

# **Dynamical Aerospace Systems**

## **Theory and Practice**

2021 Edition

Jordan D. Larson

©Jordan D. Larson 2021

# Contents

<b>Contents</b>	<b>1</b>
<b>I Introductory Flight Dynamics and Control</b>	<b>5</b>
<b>1 Introductory Dynamical Systems Theory</b>	<b>6</b>
1.1 Introduction to Dynamical Systems Theory . . . . .	6
1.2 Introduction to Transfer Functions and State-Space . . . . .	13
1.3 Free Response of SISO LTI Systems and Stability . . . . .	20
1.4 Forced and Step Response of SISO LTI Systems . . . . .	28
1.5 Impulse and Sinusoidal Response of SISO LTI Systems . . . . .	35
1.6 Frequency Response of SISO LTI Systems . . . . .	39
<b>2 Introductory Flight Vehicle Dynamics</b>	<b>52</b>
2.1 Introduction to Flight Vehicles . . . . .	52
2.2 Flight Vehicle Reference Frame Rotations . . . . .	59
2.3 Rigid Flight Vehicle Dynamics . . . . .	67
<b>3 Introductory Airplane Dynamics</b>	<b>74</b>
3.1 Rigid Airplane Dynamics . . . . .	74
3.2 Airplane Trimmed Steady Flight . . . . .	85
3.3 Airplane Static Stability . . . . .	93
3.4 Linearized Rigid Airplane Dynamics . . . . .	103
3.5 Longitudinal Stability and Control Derivatives . . . . .	113
3.6 Lateral-Directional Stability and Control Derivatives . . . . .	119
3.7 Airplane Dynamic Stability and Flying Qualities . . . . .	126
<b>4 Introduction to Control Systems</b>	<b>135</b>
4.1 Open- and Closed-Loop Control Systems . . . . .	135
4.2 MIMO Feedback Control and Stability Augmentation Systems . . . . .	143
4.3 Proportional-Integral Control . . . . .	148
4.4 Proportional-Integral-Derivative Control . . . . .	156
<b>5 SISO LTI Control System Robustness</b>	<b>165</b>

5.1	SISO LTI Control System Analysis . . . . .	165
5.2	Open-Loop Transfer Function . . . . .	172
5.3	Nyquist Plots and Stability . . . . .	177
5.4	SISO LTI System Stability Margins and Robustness . . . . .	184
<b>6</b>	<b>SISO Loop-Shaping Robust Control</b>	<b>193</b>
6.1	SISO Loop-Shaping Control Stages . . . . .	193
6.2	SISO Loop-Shaping Control Design . . . . .	200
<b>7</b>	<b>Airplane Guidance and Control Systems</b>	<b>209</b>
7.1	Introduction to Guidance and Control Systems . . . . .	209
7.2	Airplane Longitudinal Guidance and Control . . . . .	217
7.3	Airplane Lateral-Directional Guidance and Control . . . . .	224
<b>II</b>	<b>Optimal Control and Estimation</b>	<b>230</b>
<b>8</b>	<b>Advanced Dynamical Systems Theory</b>	<b>231</b>
8.1	Advanced Dynamical Systems Theory . . . . .	231
8.2	Advanced Linear State-Space Systems . . . . .	236
8.3	Lyapunov Stability and Methods . . . . .	243
8.4	Linear System Controllability and Observability . . . . .	249
8.5	Uncertain Dynamical Systems and Random Variables . . . . .	255
8.6	Random Vectors . . . . .	260
8.7	Random Processes and Sequences . . . . .	266
<b>9</b>	<b>Introductory Optimal Parameter Estimation</b>	<b>272</b>
9.1	Introduction to Optimal Parameter Estimation . . . . .	272
9.2	Ordinary Least-Squares Estimation . . . . .	276
9.3	Generalized and Bayesian Least-Squares Estimation . . . . .	282
9.4	Nonlinear and Constrained Least-Squares Estimation . . . . .	287
<b>10</b>	<b>Introductory Optimal Control</b>	<b>291</b>
10.1	Introduction to Optimal Control . . . . .	291
10.2	Unconstrained Linear-Quadratic Regulator . . . . .	296
10.3	Unconstrained Linear-Quadratic Regulator Continued . . . . .	302
10.4	Robust Servomechanism Linear-Quadratic Regulator . . . . .	308
10.5	Extended and Iterative Linear-Quadratic Regulators . . . . .	310
10.6	Constrained Linear-Quadratic Regulator . . . . .	310
<b>11</b>	<b>Introductory Optimal State Estimation</b>	<b>313</b>
11.1	Optimal Control of Stochastic State-Space Systems . . . . .	313
11.2	Introduction to Optimal State Estimation . . . . .	318
11.3	Kalman Filter Continued . . . . .	322

<b>CONTENTS</b>	<b>3</b>
11.4 Extended and Iterative Kalman Filters . . . . .	326
<b>12 Advanced Optimal State Estimation</b>	<b>329</b>
12.1 Sigma-Point Kalman Filters . . . . .	329
12.2 Sigma-Point Kalman Filters Continued . . . . .	331
12.3 Particle Filter . . . . .	331
12.4 Particle Filter Continued . . . . .	332
<b>13 Advanced Optimal Control</b>	<b>333</b>
13.1 Advanced Methods of Optimization . . . . .	333
13.2 Convex Optimization in Control . . . . .	333
13.3 Optimal Control for LPV Systems . . . . .	334
13.4 Receding Horizon Control . . . . .	334
<b>III Advanced Flight Dynamics and Control</b>	<b>339</b>
<b>14 Advanced Rigid Flight Vehicle Dynamics</b>	<b>340</b>
14.1 Mass Effects on Flight Dynamics . . . . .	340
14.2 Atmospheric and Gravity Effects on Flight Dynamics . . . . .	346
14.3 Rotating Spherical Earth Effects on Flight Dynamics . . . . .	353
14.4 Advanced Rigid Airplane Dynamics Simulation . . . . .	359
<b>15 Elastic Flight Vehicle Dynamics</b>	<b>364</b>
15.1 Lumped-Mass Vibrations . . . . .	364
15.2 Elastic Body Dynamics . . . . .	372
15.3 Elastic Vehicle Mean Axes . . . . .	379
15.4 Introduction to Elastic Flight Vehicle Dynamics . . . . .	385
15.5 Dynamic-Elastic Effects on Flight Vehicles . . . . .	390
15.6 Advanced Elastic Flight Vehicle Dynamics . . . . .	397
<b>16 MIMO LTI Control System Robustness</b>	<b>403</b>
16.1 MIMO LTI Control System Analysis . . . . .	403
16.2 Multivariate Frequency Response . . . . .	410
16.3 MIMO LTI System Uncertainty Modeling . . . . .	412
16.4 Multivariate Nyquist Stability Criterion . . . . .	415
16.5 Structured Singular Value Analysis . . . . .	420
<b>17 MIMO Loop-Shaping Robust Control</b>	<b>426</b>
17.1 Introduction to Robust Optimal Control . . . . .	426
17.2 $H_\infty$ Optimal Control . . . . .	430
17.3 Mixed-Sensitivity $H_\infty$ Loop-Shaping . . . . .	433
17.4 $\mu$ -Synthesis Robust Control . . . . .	436
<b>18 Introductory Adaptive Control</b>	<b>440</b>

18.1 Introduction to Adaptive Control . . . . .	440
18.2 Direct SISO Model Reference Adaptive Control . . . . .	445
18.3 Direct MIMO Model Reference Adaptive Control . . . . .	452
<b>IV Appendices</b>	<b>460</b>
<b>A Dynamical Systems Programming</b>	<b>461</b>
A.1 Dynamical Systems in MATLAB . . . . .	461
A.2 Dynamical Systems in Python . . . . .	477
A.3 MATLAB and Python Comparison . . . . .	484
A.4 Optimal Control and Estimation in Python . . . . .	486
<b>B Miscellaneous Topics</b>	<b>490</b>
B.1 Discrete-Time Feedback Control Systems . . . . .	490
<b>Bibliography</b>	<b>496</b>
<b>Index</b>	<b>497</b>

## **Part I**

# **Introductory Flight Dynamics and Control**

# Chapter 1

## Introductory Dynamical Systems Theory

### 1.1 Introduction to Dynamical Systems Theory

The study of flight dynamics and control (FDC) can be formalized in terms of signals and systems. A **signal** is a mathematical description of how a parameter varies with time. Signals can be continuous or discrete in time, as well as continuous or discrete in the values the parameter may take. **Analog signals** are continuous in time and continuous in value. **Digital signals** are discrete in time and discrete in value. **Discrete-time signals** are discrete in time and continuous in value. Signals that are continuous in time and discrete in value rarely occur and do not have a particular name.

A **system** is any process that produces output signals in response to input signals. Thus, the output signal is also known as the **system response**. The connection between signals and systems is often depicted visually using **block diagrams**, for which each “block” represents a system and each arrow represents a “signal.” An example of a basic block diagram can be drawn as

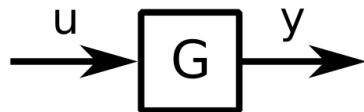


Figure 1.1: Basic Block Diagram

which consists of a system  $G$  with a single input signal  $u$  and single output signal  $y$ .

A system may be characterized by its different properties. Of particular interest to FDC is the theory of **dynamical systems** whose output signal is a time-dependent quantity that evolves according to a fixed mathematical rule, also referred to as the **dynamics equation**. This is opposed to a **static system** which is not time-dependent. It should be noted that for many physical dynamical systems, one often refers to the dynamics equation as its **equation of motion** which is true for FDC. The theory of dynamical systems encompasses three broad topics: simulation, control, and **system identification (SID)**. Each of these concepts will be introduced in this part for flight vehicles.

A system may be characterized by its number of input and output signals, namely as a **single input, single output (SISO) system**; a **single input, multiple output (SIMO) system**; a **multiple input, single**

**output (MISO) system ; or a multiple input, multiple output (MIMO) system .**

A systems may be characterized by its signal types. **Analog systems** have only analog signals while **digital systems** have only digital signals. However, the vast majority of realistic dynamical systems contain some level of both analog and digital signals and are analyzed separately along with the analog-to-digital and/or digital-to-analog converters.

A system may be characterized by its mathematical rule. A **linear system** satisfies the principle of superposition which states that one may add these systems together as well as multiplied them by scalars and the overall system output signal will be equivalent to the added and scaled output signals of the individual systems. A **nonlinear system** does not satisfy the principle of superposition. A **time-invariant system** has a mathematical rule that does not depend *explicitly* on time. A **time-varying system** has a mathematical rule that does depend on time. Lastly, a **deterministic system** will always produce the same output signal for a given input signal. If this is not true, they are **stochastic systems**.

This chapter of the textbook introduces the mathematical theory for continuous-time, linear time-invariant (LTI), deterministic dynamical systems. Unfortunately, no dynamical system is perfectly linear, time-invariant, or deterministic. First, linearity implies that the operation of a system can be scaled to arbitrarily large magnitudes which is not physically possible. Second, time-invariance is violated by aging effects that can change the outputs of analog systems over time. Lastly, thermal noise and other random phenomena ensure that the operation of any analog system will have some degree of random or stochastic behavior. However, despite this limitation, these assumptions greatly simplify the mathematical theory of dynamical systems while still providing valuable insight and intuition into the later practice of FDC. In particular, the concepts of Jacobian linearization, equilibrium, and Lyapunov stability methods allow for sufficient design and analysis of nonlinear dynamical systems with linear dynamical systems theory which will be introduced in this section.

## Ordinary Differential Equations

One standard mathematical representation of continuous-time dynamical systems is differential equations which explicitly use time derivatives to describe the mathematical rule of the dynamical system. While differential equations and can be univariate or multivariate and dynamical systems in FDC are inherently MIMO, one can make simplifying assumptions which allow the use of SISO dynamical systems to model and analyze some aspects of FDC. The remainder of this section will discuss important concepts from dynamical systems theory for continuous-time SISO systems, namely the use of **ordinary differential equations (ODEs)** . ODEs generally have the following form

$$\frac{d^n y}{dt^n} = f \left( t, y, \frac{dy}{dt}, \dots, \frac{d^{n-1} y}{dt^{n-1}}, u, \frac{du}{dt}, \dots, \frac{d^{p-1} u}{dt^{p-1}} \right) \quad (1.1)$$

where  $y(t)$  is the single **output signal** ,  $u(t)$  is the single **input signal** , and  $n$  is the **order of the ODE** with  $p \leq n$  denoting a **proper** ordinary differential equation!proper . To solve for the system response, i.e. the solution of the ODE, one requires the use of **boundary conditions** for the input-output pair at some time  $t$ . Boundary conditions given at  $t = 0$  are also called **initial conditions** . Furthermore, the ODE is said to be **autonomous** if  $f()$  is not a function of an input  $u$ , i.e.

$$\frac{d^n y}{dt^n} = f \left( t, y, \frac{dy}{dt}, \dots, \frac{d^{n-1} y}{dt^{n-1}} \right) \quad (1.2)$$

Lastly, it should be noted that from here on, this textbook will use Newton's dot notation to represent time derivatives up to the third order and a bracketed exponent for higher orders.

The analysis of general ODEs is an advanced topic and full consideration of these types of systems are discussed in subsequent parts of this textbook. Of importance to this course on dynamics and control is the analysis of **linear ordinary differential equations** whose standard form is

$$y^{[n]}(t) + a_{n-1}y^{[n-1]}(t) + \cdots + a_1\dot{y}(t) + a_0y(t) = b_p u^{[p]}(t) + \cdots + b_1\dot{u}(t) + b_0u(t) \quad (1.3)$$

Associated with the standard form of a linear ODE is its **characteristic equation** which has the form

$$\lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0 = 0 \quad (1.4)$$

The characteristic equation will play a significant role in SISO linear dynamical system analysis, in particular through the **roots** of the characteristic equation, i.e. the values of  $\lambda$  which solve the characteristic equation.

As stated previously, linear systems satisfy the **principle of superposition** described by two properties, scaling and additivity. First, let the output  $y_1(t)$  be a solution to a linear ODE with input  $u_1(t)$  and zero initial conditions. The scaling property states that for any real number  $c$ , the solution of the linear ODE with input  $u_s(t) = cu_1(t)$  and zero initial conditions is given by  $y_s(t) = cy_1(t)$ . Next, let  $y_2(t)$  be the solution with  $u_2(t)$  and zero initial conditions. The additivity property states that the solution of the linear ODE with input  $u_a(t) = u_1(t) + u_2(t)$  and zero initial conditions is  $y_a(t) = y_1(t) + y_2(t)$ . Nonlinear systems do not, in general, satisfy these properties.

## Jacobian Linearization of Univariate Functions

The Jacobian linearization for univariate functions derives from the **Taylor series** of a function  $f(y)$  about  $\bar{y}$  which is defined as

$$f(y) = f(\bar{y}) + \left[ \frac{df}{dy} \right]_{\bar{y}} (y - \bar{y}) + \frac{1}{2} \left[ \frac{d^2f}{dy^2} \right]_{\bar{y}} (y - \bar{y})^2 + \cdots \quad (1.5)$$

which is an infinite series of increasing order.

The **Jacobian linearization of  $f(y)$  about  $\bar{y}$**  approximates this series by the inclusion of the zeroth and first order terms, i.e. the constant and proportional terms, which produces a linear function as

$$f(y) \approx f(\bar{y}) + \left[ \frac{df}{dy} \right]_{\bar{y}} (y - \bar{y}) \quad (1.6)$$

which can be said to approximate the true function. Thus, the Jacobian linearization is also known as a **first order approximation**. Here the difference between the exact solution and the linearization is known as the **linearization error** and is a result of the removal of **higher order terms (HOT)** from the Taylor series. In addition, it should also be noted that often one uses the substitution

$$y = \bar{y} + \Delta y \quad (1.7)$$

where  $\Delta y$  is a **perturbation** about the point  $\bar{y}$ . This type of substitution is known as **perturbation form**. Then, the linearization can be rewritten as

$$f(\bar{y} + \Delta y) \approx f(\bar{y}) + \left[ \frac{df}{dy} \right]_{\bar{y}} \Delta y \quad (1.8)$$

which is a fundamental result of **perturbation theory**.

An important linearization is for the trigonometric sine and cosine functions about any angle  $\bar{\theta}$ . It can be shown that the Taylor series for sine and cosine are

$$\sin(\bar{\theta}) = \sin(\bar{\theta}) + \cos(\bar{\theta})(\theta - \bar{\theta}) - \frac{1}{2!} \sin(\bar{\theta})(\theta - \bar{\theta})^2 - \frac{1}{3!} \cos(\bar{\theta})(\theta - \bar{\theta})^3 + \dots \quad (1.9)$$

and

$$\cos(\bar{\theta}) = \cos(\bar{\theta}) - \sin(\bar{\theta})(\theta - \bar{\theta}) - \frac{1}{2!} \cos(\bar{\theta})(\theta - \bar{\theta})^2 + \frac{1}{3!} \sin(\bar{\theta})(\theta - \bar{\theta})^3 + \dots \quad (1.10)$$

where  $\theta$  must be in radians. Thus, the linearizations for sine and cosine about  $\bar{\theta}$  are given by

$$\sin(\bar{\theta}) \approx \sin \bar{\theta} + \cos(\bar{\theta})(\theta - \bar{\theta}) \quad (1.11)$$

and

$$\cos(\bar{\theta}) \approx \cos \bar{\theta} - \sin(\bar{\theta})(\theta - \bar{\theta}) \quad (1.12)$$

Of particular note, for  $\bar{\theta} = 0$  rad, the linearization simplifies to

$$\sin \theta \approx \theta \quad (1.13)$$

and

$$\cos \theta \approx 1 \quad (1.14)$$

In this case, one can also approximate

$$\tan \theta \approx \theta \quad (1.15)$$

These three approximations are commonly called the **small angle approximations** and will be used throughout this textbook. These are decent approximations when  $\theta < 15^\circ = 0.2618$  rad, producing a linearization error of 0.0028, 0.034, 0.0061 and for sine, cosine, and tangent, respectively.

## Linearization of ODEs

Recall the general form for an ODE

$$y^{[n]} = f(y, \dot{y}, \dots, y^{[n-1]}, u, \dot{u}, \dots, u^{[p-1]}) \quad (1.16)$$

Then, modeling the output signal in perturbation form as

$$y(t) = \bar{y} + \Delta y(t) \quad (1.17)$$

and the input signal in perturbation form as

$$u(t) = \bar{u} + \Delta u(t) \quad (1.18)$$

the linearization of a general ODE can be written as

$$\Delta y^{[n]} + a_{n-1} \Delta y^{[n-1]} + \dots + a_1 \Delta \dot{y} + a_0 \Delta y = b_p \Delta u^{[p]} + \dots + b_1 \Delta \dot{u} + b_0 \Delta u \quad (1.19)$$

where the coefficients of this linear ODE are

$$a_0 = -\frac{\partial f}{\partial y} (\bar{y}, 0, \dots, 0, \bar{u}, 0, \dots, 0) \quad (1.20)$$

$$a_i = -\frac{\partial f}{\partial y^{[i]}} (\bar{y}, 0, \dots, 0, \bar{u}, 0, \dots, 0) \quad (1.21)$$

for  $i = 1, \dots, n-1$ , and

$$b_0 = \frac{\partial f}{\partial u} (\bar{y}, 0, \dots, 0, \bar{u}, 0, \dots, 0) \quad (1.22)$$

$$b_j = \frac{\partial f}{\partial u^{[j]}} (\bar{y}, 0, \dots, 0, \bar{u}, 0, \dots, 0) \quad (1.23)$$

for  $j = 1, \dots, p$ . Note that the negative signs appear for the  $a$  coefficients due to the standard form for linear ODEs where the left side of the equation contains all  $y$  terms and the right side contains all  $u$  terms. Lastly, note that the derivatives of  $y(t)/u(t)$  are equivalent to those of  $\Delta y(t)/u(t)$  since  $\bar{y}/\bar{u}$  are constants.

## Equilibrium Points of ODEs

For autonomous ODEs,  $\bar{y}$  is an **equilibrium point** if all derivatives of  $y$  are zero for all  $t > 0$ , i.e. if

$$f(t, \bar{y}, 0, \dots, 0) = 0 \quad (1.24)$$

is a valid solution for all  $t \geq 0$ . Thus, if one initializes  $y(t) = \bar{y}$  at  $t = 0$ , then  $y(t) = \bar{y}$  for all  $t \geq 0$ . An important property of equilibrium points to analyze is their stability, a subject which was studied by Lyapunov for autonomous dynamical systems in his dissertation *The General Problem of Stability of Motion*. Here, an equilibrium point  $\bar{y}$  of an ODE is **stable in the sense of Lyapunov (SISL)** if for some  $|\epsilon| > 0$  with  $y(0) = \bar{y} + \epsilon$ , there exists some  $\delta$  for which as  $t \rightarrow \infty$ ,  $|y| < |\bar{y} + \delta|$ , i.e. if one initializes the dynamical system “near” enough to equilibrium, the system will remain “close” to the equilibrium point. If a  $\delta$  exists for all  $\epsilon \in \mathbb{R}$ , then the equilibrium point is **globally stable**. If  $y \rightarrow 0$ , then the equilibrium point is **asymptotically stable**. Lastly, if  $y \rightarrow 0$  for all  $\epsilon \in \mathbb{R}$ , then the equilibrium point is **globally asymptotically stable (GAS)**.

For non-autonomous ODEs, an output-input pair  $(\bar{y}, \bar{u})$  is an **equilibrium point** if all derivatives of  $y$  and  $u$  are zero for all  $t > 0$ , i.e. if

$$f(t, \bar{y}, 0, \dots, 0, \bar{u}, 0, \dots, 0) = 0 \quad (1.25)$$

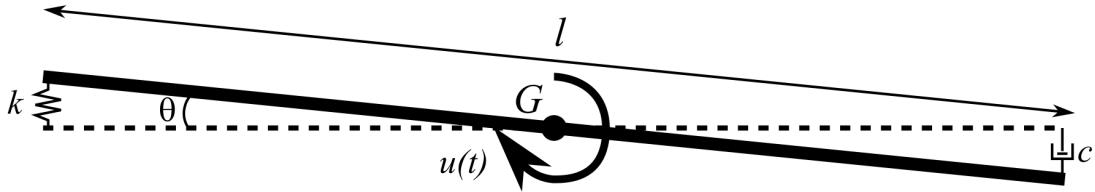
is a valid solution for all  $t \geq 0$ . Thus, if one initializes  $y(t) = \bar{y}$  at  $t = 0$  and sets  $u(t) = \bar{u}$  for  $t \geq 0$ , then  $y(t) = \bar{y}$  for all  $t \geq 0$ . Note that one may choose two variables  $\bar{y}$  and  $\bar{u}$  to solve for equilibrium which may lead to infinitely many solutions. Thus, an equilibrium point for non-autonomous ODEs is also known as a **trim point**, as one is “trimming” the system to accomplish equilibrium using the additional independent input variable to the dynamical system. With conditions on  $u(t) = \bar{u}$ , Lyapunov stability definitions can be extended to definitions of stability for non-autonomous dynamical systems. In addition, one can also define other types of stability, primarily, **bounded input, bounded output (BIBO) stability** which states that

To prove the stability of equilibrium points, Lyapunov developed two methods. The first method of Lyapunov states that if the linearized dynamical system about the equilibrium point is strictly stable, then the original dynamical system is also stable for some “neighborhood” about the trim point. This is also

known as the **linearization theorem** and is a fundamental theorem for dynamical systems theory which justifies the linearized system analysis about trim points in FDC. The size of this “stability neighborhood” is directly related to the effects of the neglected HOT in the linearization. Thus, for highly nonlinear ODEs one typically analyzes the stability of equilibrium points using Lyapunov’s second method. However, this part on introductory FDC focuses on the analysis of linearized flight dynamics and will only consider Lyapunov’s first method. A later part of this textbook will discuss Lyapunov’s second method.

## Example Problem

Given: A rod is allowed to pitch about its center with the following spring and damper forces and forcing moment  $u(t)$  applied



It is also known that

- $c = 1.1$
- $\ell = 2$
- $I_y = 0.1$
- $k = 1$

Determine: the linearized ODE for small angles in standard linear ODE form

Solution:

$$\Sigma M_G = I_y \ddot{\theta} \quad (1.26)$$

$$-c \left( \frac{\ell}{2} \frac{d \sin \theta}{dt} \right) \frac{\ell}{2} - k \left( \frac{\ell}{2} \sin \theta \right) \frac{\ell}{2} + u(t) = I_y \ddot{\theta} \quad (1.27)$$

$$-\frac{c\ell^2}{4} \cos \theta \dot{\theta} - \frac{k\ell^2}{4} \sin \theta + u(t) = I_y \ddot{\theta} \quad (1.28)$$

$$\ddot{\theta} + \frac{c\ell^2}{4I_y} \cos \theta \dot{\theta} + \frac{k\ell^2}{4I_y} \sin \theta = \frac{1}{I_y} u(t) \quad (1.29)$$

Using the small angle approximation for the linearization

$$\ddot{\theta} + \frac{c\ell^2}{4I_y} \dot{\theta} + \frac{k\ell^2}{4I_y} \theta = \frac{1}{I_y} u(t) \quad (1.30)$$

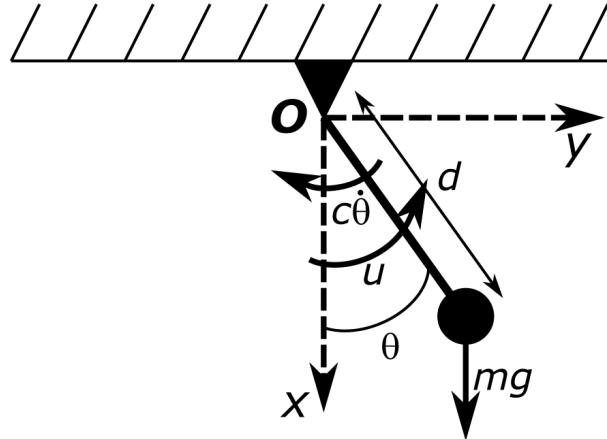
Substituting values

$$\underline{\ddot{\theta} + 11\dot{\theta} + 10\theta = 10u(t)} \quad (1.31)$$

## Example Problem 2

**Given:**

The free body diagram of a pendulum



*Note:  $c\dot{\theta}$  can be thought of as a resistance term*

**Determine:**

- the nonlinear ODE representing the dynamical system
- $\bar{\theta}$  for all trim point(s) with  $\bar{u} = 0$
- the linearized ODEs about trim point(s) with  $\bar{u} = 0$  in standard form
- comment on the stability of the trim point(s)

**Solution:**

- a) The equation of motion for the pendulum is given by

$$\sum M_O = I_P \ddot{\theta} \quad (1.32)$$

where the moments are generated by two applied torques,  $c\dot{\theta}$ ,  $u$ , and the moment arm due to the weight,  $mgd \sin \theta$ . Recalling the moment of inertia of the mass is  $md^2$ , one has the EOM

$$-c\dot{\theta} + u - mgd \sin \theta = (md^2)\ddot{\theta} \quad (1.33)$$

or in standard form

$$\ddot{\theta} = -\frac{c}{md^2}\dot{\theta} - \frac{g}{d} \sin \theta + \frac{u}{md^2} \quad (1.34)$$

where  $0 \leq \theta < 2\pi$

- b) For trim at  $\bar{u} = 0$ ,  $\ddot{\theta} = 0$ , and  $\dot{\theta} = 0$  by definition, one has

$$0 = -0 - \frac{g}{d} \sin \bar{\theta} + 0 \quad (1.35)$$

$$\bar{\theta} = \sin^{-1} 0 \quad (1.36)$$

$$\underline{\bar{\theta} = 0 \text{ & } \pi} \quad (1.37)$$

c) To linearize the second order ODE into the standard form

$$\Delta\ddot{\theta} + a_1\Delta\dot{\theta} + a_0\Delta\theta = b_0\Delta u \quad (1.38)$$

take the partial derivative with respect to each derivative term and evaluate at trim, i.e.

$$a_0 = -\frac{\partial f}{\partial \theta}(\bar{\theta}, 0, 0, \bar{u}) = \frac{g}{d} \cos \theta_0 \quad (1.39)$$

which equals  $\frac{g}{d}$  for  $\theta_0 = 0$  and  $-\frac{g}{d}$  for  $\theta_0 = \pi$ ,

$$a_1 = -\frac{\partial f}{\partial \dot{\theta}}(\bar{\theta}, 0, 0, \bar{u}) = \frac{c}{md^2} \quad (1.40)$$

for both cases and

$$b_0 = \frac{\partial f}{\partial u}(\bar{\theta}, 0, 0, \bar{u}) = \frac{1}{md^2} \quad (1.41)$$

for both cases.

Thus, for  $\bar{\theta} = 0$ , the linearized ODE is

$$\underline{\Delta\ddot{\theta} + \frac{c}{md^2}\Delta\dot{\theta} + \frac{g}{d}\Delta\theta = \frac{1}{md^2}\Delta u} \quad (1.42)$$

and for  $\bar{\theta} = \pi$ , the linearized ODE is

$$\underline{\Delta\ddot{\theta} + \frac{c}{md^2}\Delta\dot{\theta} - \frac{g}{d}\Delta\theta = \frac{1}{md^2}\Delta u} \quad (1.43)$$

d) If one considers perturbing the pendulum about  $\bar{\theta} = 0$ , i.e. small angles about 0, the pendulum will tend back to the  $\theta = 0$  as gravity works to pull it back the mass as it swings. Thus, this is a stable trim point.

Conversely, if one considers perturbing the pendulum about  $\bar{\theta} = \pi$ , i.e. small angles about  $\pi$ , the pendulum will tend to fall back down towards  $\theta = 0$  as gravity works in the opposite direction of the mass. Thus, this is a unstable trim point.

## 1.2 Introduction to Transfer Functions and State-Space

In addition to ODEs, there are two common alternative representations for continuous-time SISO LTI dynamical systems, namely the transfer function representation and the LTI state-space representation. Furthermore, state-space representation also provides a general form for general MIMO dynamical systems as well as the notion of transfer function matrices. These two representations will be introduced in this section and related to the standard form of LTI ODEs. A more comprehensive treatment of state-space representation and transfer function matrices is given in the later parts of this textbook.

## Transfer Function Representation

The **transfer function representation** of SISO LTI systems uses a change of variables from the real variable  $t$  to a complex variable  $s$  under zero initial conditions by the **Laplace transform**

$$F(s) = \mathcal{L}\{f(t)\} = \int_0^\infty f(t)e^{-st} dt \quad (1.44)$$

and vice versa by the **inverse Laplace transform**

$$f(t) = \mathcal{L}^{-1}\{F(s)\} = \frac{1}{2\pi j} \lim_{T \rightarrow \infty} \int_{\gamma-jT}^{\gamma+jT} F(s)e^{st} ds \quad (1.45)$$

where  $j$  here represents  $\sqrt{-1}$ , a dynamical systems notation resulting from electrical systems already using lowercase  $i$  for current. From this transform, one can define the conversion of derivatives as

$$x^{[i]}(t) = s^i x(s) - \sum_{k=1}^i s^{i-k} x^{[k-1]}(0) \quad (1.46)$$

and linear expressions as

$$ax_1(t) + bx_2 = aX_1(s) + bX_2(s) \quad (1.47)$$

Next, for the standard linear ODE

$$y^{[m]}(t) + a_{m-1}y^{[m-1]}(t) + \cdots + a_1\dot{y}(t) + a_0y(t) = b_p u^{[p]}(t) + \cdots + b_1\dot{u}(t) + b_0u(t) \quad (1.48)$$

with zero initial conditions, one can transform a linear ODE to the  $s$  domain as

$$s^m y(s) + a_{m-1}s^{m-1}y(s) + \cdots + a_1sy(s) + a_0y(s) = b_ps^p u(s) + \cdots + b_1su(s) + b_0u(s) \quad (1.49)$$

and rearranging, one has

$$(s^m + s^{m-1} + \cdots + a_1s + a_0)y(s) = (b_ps^p + \cdots + b_1s + b_0)u(s) \quad (1.50)$$

or

$$y(s) = \frac{b_ps^p + \cdots + b_1s + b_0}{s^m + a_{m-1}s^{m-1} + \cdots + a_1s + a_0}u(s) \quad (1.51)$$

Then, defining the transfer function of an LTI system  $G(s)$  as

$$y(s) = G(s)u(s) \quad (1.52)$$

which “transfers” the transformed input  $u(s)$  to the transformed output  $y(s)$  through simple multiplication, the **standard transform function form** can be defined as

$$G(s) = \frac{y(s)}{u(s)} = \frac{b_ps^p + \cdots + b_1s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1s + a_0} \quad (1.53)$$

As the numerator and denominator of the transfer function are polynomials, they have real and/or complex roots. A **zero** of a transfer function is a root of the numerator while a **pole** of a transfer function is a root

of the denominator. These terms comes from the fact that the transfer function's magnitude,  $|G(s)|$ , in the complex plane will go to 0 for  $s$  equal to a zero and infinity for  $s$  equal to a pole. Plotting  $|G(s)|$  over the complex plane, this infinite value will look like a tent pole in a three dimensional plot. Note: that if there are no **pole-zero cancellations**, i.e. no poles and zeros with the same value, then the denominator of the transfer function is equivalent to the characteristic equation of the ODE and the transfer function is said to be a **minimal realization** and the denominator of the transfer function is equivalent to the characteristic equation of the ODE. Thus, one can interchange the terms roots and poles of an LTI system for minimal realizations.

For reference, some additional conversions for functions in the  $t$  and  $s$  domains are provided in the following table which are often used for converting between  $u(t)$  and  $u(s)$  and  $y(t)$  and  $y(s)$ , respectively.

Function	$t$ Domain ( $\forall t \geq 0$ )	$s$ Domain
Unit Step	1	$\frac{1}{s}$
Exponential	$e^{-at}$	$\frac{1}{s+a}$
Sine	$\sin \omega t$	$\frac{\omega}{s^2+\omega^2}$
Cosine	$\cos \omega t$	$\frac{s}{s^2+\omega^2}$
Exponentially Decaying Sine	$e^{-at} \sin \omega t$	$\frac{\omega}{(s+a)^2+\omega^2}$
Exponentially Decaying Cosine	$e^{-at} \cos \omega t$	$\frac{s+a}{(s+a)^2+\omega^2}$

## Continuous-Time State-Space Representation

A more general method for modeling a continuous-time dynamical system is the **continuous-time state-space representation** which can be defined as the following two equations

$$\begin{aligned}\dot{\vec{x}}(t) &= f(t, \vec{x}, \vec{u}) \\ \vec{y}(t) &= h(t, \vec{x}, \vec{u})\end{aligned}\tag{1.54}$$

where  $\vec{u}(t) \in \mathbb{R}^{n_u}$  is the **input vector** of  $n_u$  input signals,  $\vec{y}(t) \in \mathbb{R}^{n_y}$  is the **output vector** of  $n_y$  output signals, and  $\vec{x} \in \mathbb{R}^{n_x}$  is the **state vector** of the  $n_x^{\text{th}}$  order dynamical system. The first vector-valued differential equation is called the **state equation** and also known as the **dynamics equation** for state-space systems. The second vector-valued algebraic equation is the **output equation** and relates the state and input vectors to the output vector. In this formulation, the state of the system is dynamically controlled by the input  $\vec{u}$  and is observed through the output  $\vec{y}$ . Furthermore, the dimension of the state vector,  $n_x$ , is equal to the system order.

It should be noted that this vector-valued state-space system can be rewritten as  $n_x$  first order ODEs for

the state equation and  $n_y$  algebraic equations for the output in the general form

$$\begin{aligned}\dot{x}_1 &= f_1(t, x_1, \dots, x_{n_x}, u_1, \dots, u_{n_u}) \\ &\vdots = \vdots \\ \dot{x}_{n_x} &= f_{n_x}(t, x_1, \dots, x_{n_x}, u_1, \dots, u_{n_u}) \\ y_1 &= h_1(t, x_1, \dots, x_{n_x}, u_1, \dots, u_{n_u}) \\ &\vdots = \vdots \\ y_{n_y} &= h_{n_y}(t, x_1, \dots, x_{n_x}, u_1, \dots, u_{n_u})\end{aligned}\tag{1.55}$$

where  $x_i$  denotes the  $i^{\text{th}}$  element of  $\vec{x}$ ,  $u_j$  denotes the  $j^{\text{th}}$  element of  $\vec{u}$ , and  $y_k$  denotes the  $k^{\text{th}}$  element of  $\vec{y}$ .

When the functions  $f()$  and  $h()$  are LTI, then one has the **continuous-time LTI state-space representation** defined as

$$\begin{aligned}\dot{\vec{x}}(t) &= A \vec{x}(t) + B \vec{u}(t) \\ \vec{y}(t) &= C \vec{x}(t) + D \vec{u}(t)\end{aligned}\tag{1.56}$$

where  $A \in \mathbb{R}^{n_x \times n_x}$  matrix is the **state matrix**. The  $B$  matrix is the **input matrix**. The  $C$  matrix is the **output matrix** and  $D$  is the **feedthrough matrix**. With this standard form, a particular LTI state-space model can be denoted by the quadruple  $(A, B, C, D)$ . However, it should be noted that there are different choices for  $(A, B, C, D)$  that can represent the same dynamical system in terms of the input-to-output relationship, although the internal state will be different for each representation. It should also be noted that if one wishes to set  $\vec{y} = \vec{x}$ , then one may set the output matrix to the **identity matrix**, i.e.

$$C = I = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \vdots & 1 & 0 \\ 0 & 0 & \vdots & 0 & 1 \end{bmatrix}\tag{1.57}$$

and the feedthrough matrix to the **zero matrix**, i.e.

$$D = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}\tag{1.58}$$

The mathematics and analysis of LTI systems using the state-space representation borrow heavily from linear algebra theory which will be presented when necessary in this textbook, though this textbook should not be considered a thorough treatment of linear algebra.

## Linearization of Time-Invariant State-Space Systems

Trim points for time-invariant state-space systems occur when the state does not change, i.e.

$$\begin{aligned}\dot{\vec{x}} &= f(\bar{x}, \bar{u}) = 0 \\ \vec{y} &= h(\bar{x}, \bar{u})\end{aligned}\tag{1.59}$$

where there may be multiple solutions (even infinite) for trim states since there will be fewer equations than free variables for the vector-valued state equation (i.e.  $n_x$  elements and  $n_x + n_u$  free variables).

For the linearization of the general time-invariant state-space model, first, recall the Taylor Series for multivariate functions about the vector pair  $(\bar{x}, \bar{u})$  is

$$\vec{x}(t) = f(x, u) = f(\bar{x}, \bar{u}) + \left[ \frac{\partial f}{\partial \vec{x}}(\bar{x}, \bar{u}) \right] (\vec{x} - \bar{x}) + \left[ \frac{\partial f}{\partial \vec{u}}(\bar{x}, \bar{u}) \right] (\vec{u} - \bar{u}) + \text{HOT} \quad (1.60)$$

or

$$\vec{y}(t) = h(x, u) = h(\bar{x}, \bar{u}) + \left[ \frac{\partial h}{\partial \vec{x}}(\bar{x}, \bar{u}) \right] (\vec{x} - \bar{x}) + \left[ \frac{\partial h}{\partial \vec{u}}(\bar{x}, \bar{u}) \right] (\vec{u} - \bar{u}) + \text{HOT} \quad (1.61)$$

where  $\left[ \frac{\partial f}{\partial \vec{x}}(\bar{x}, \bar{u}) \right]$  is the Jacobian of  $f()$ . Thus, multivariate linearization is sometimes referred to as **Jacobian linearization**. Defining the state, input, and output perturbation vectors about constants  $\bar{x}$ ,  $\bar{u}$ , and  $\bar{y}$  as

$$\Delta \vec{x}(t) = \vec{x}(t) - \bar{x} \quad (1.62)$$

$$\Delta \vec{u}(t) = \vec{u}(t) - \bar{u} \quad (1.63)$$

$$\Delta \vec{y}(t) = \vec{y}(t) - \bar{y} \quad (1.64)$$

and recognizing for trim,  $f(\bar{x}, \bar{u}) = 0$  and  $h(\bar{x}, \bar{u}) = \bar{y}$ , one has

$$\Delta \vec{x}(t) = f(x, u) = \left[ \frac{\partial f}{\partial \vec{x}}(\bar{x}, \bar{u}) \right] \Delta \vec{x}(t) + \left[ \frac{\partial f}{\partial \vec{u}}(\bar{x}, \bar{u}) \right] \Delta \vec{u}(t) + \text{HOT} \quad (1.65)$$

$$\Delta \vec{y}(t) = h(x, u) - \bar{y} = \left[ \frac{\partial h}{\partial \vec{x}}(\bar{x}, \bar{u}) \right] \Delta \vec{x}(t) + \left[ \frac{\partial h}{\partial \vec{u}}(\bar{x}, \bar{u}) \right] \Delta \vec{u}(t) + \text{HOT} \quad (1.66)$$

Thus, setting

$$A = \left[ \frac{\partial f}{\partial \vec{x}}(\bar{x}, \bar{u}) \right] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\bar{x}, \bar{u}) & \cdots & \frac{\partial f_1}{\partial x_n}(\bar{x}, \bar{u}) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1}(\bar{x}, \bar{u}) & \cdots & \frac{\partial f_n}{\partial x_n}(\bar{x}, \bar{u}) \end{bmatrix} \quad (1.67)$$

$$B = \left[ \frac{\partial f}{\partial \vec{u}}(\bar{x}, \bar{u}) \right] = \begin{bmatrix} \frac{\partial f_1}{\partial u_1}(\bar{x}, \bar{u}) & \cdots & \frac{\partial f_1}{\partial u_p}(\bar{x}, \bar{u}) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1}(\bar{x}, \bar{u}) & \cdots & \frac{\partial f_n}{\partial u_p}(\bar{x}, \bar{u}) \end{bmatrix} \quad (1.68)$$

$$C = \left[ \frac{\partial h}{\partial \vec{x}}(\bar{x}, \bar{u}) \right] = \begin{bmatrix} \frac{\partial h_1}{\partial x_1}(\bar{x}, \bar{u}) & \cdots & \frac{\partial h_1}{\partial x_n}(\bar{x}, \bar{u}) \\ \vdots & \ddots & \vdots \\ \frac{\partial h_m}{\partial x_1}(\bar{x}, \bar{u}) & \cdots & \frac{\partial h_m}{\partial x_n}(\bar{x}, \bar{u}) \end{bmatrix} \quad (1.69)$$

$$D = \left[ \frac{\partial h}{\partial \vec{u}}(\bar{x}, \bar{u}) \right] = \begin{bmatrix} \frac{\partial h_1}{\partial u_1}(\bar{x}, \bar{u}) & \cdots & \frac{\partial h_1}{\partial u_p}(\bar{x}, \bar{u}) \\ \vdots & \ddots & \vdots \\ \frac{\partial h_m}{\partial u_1}(\bar{x}, \bar{u}) & \cdots & \frac{\partial h_m}{\partial u_p}(\bar{x}, \bar{u}) \end{bmatrix} \quad (1.70)$$

yields an approximate LTI state-space model about  $\bar{x}$  and  $\bar{u}$  as

$$\begin{aligned} \Delta \vec{x}(t) &\approx A \Delta \vec{x}(t) + B \Delta \vec{u}(t) \\ \Delta \vec{y}(t) &\approx C \Delta \vec{x}(t) + D \Delta \vec{u}(t) \end{aligned} \quad (1.71)$$

### LTI ODE Conversion to Continuous-Time State-Space

A general proper  $n^{\text{th}}$  order LTI ODE, i.e.

$$y^{[n]}(t) + a_{n-1}y^{[n-1]}(t) + \cdots + a_1\dot{y}(t) + a_0y(t) = b_nu^{[n]}(t) + \cdots + b_1\dot{u}(t) + b_0u(t) \quad (1.72)$$

can be converted to any number of continuous-time SISO LTI state-space representations. However, one common method is the **Controllable Canonical Form (CCF)** which is performed as follows. Let

$$\begin{aligned} x_1 &= y \\ x_2 &= \dot{y} \\ &\vdots = \vdots \\ x_{n_x-1} &= y^{[n-1]} \\ x_{n_x} &= y^{[n]} \end{aligned} \quad (1.73)$$

Then, the matrices of the CCF state-space system can be defined as

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{bmatrix} \quad (1.74)$$

$$B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad (1.75)$$

$$C = [(b_0 - a_0b_n) \quad (b_1 - a_1b_n) \quad \cdots \quad (b_{n-2} - a_{n-2}b_n) \quad (b_{n-1} - a_{n-1}b_n)] \quad (1.76)$$

$$D = b_n \quad (1.77)$$

Note that if  $b_n = 0$  then the formula is much simpler.

### Continuous-Time LTI State-Space Conversion to Transfer Function

For continuous-time LTI state-space systems, i.e.

$$\begin{aligned} \dot{\vec{x}}(t) &= A\vec{x}(t) + B\vec{u}(t) \\ \vec{y}(t) &= C\vec{x}(t) + D\vec{u}(t) \end{aligned} \quad (1.78)$$

one can perform the Laplace transform for matrices to obtain

$$\begin{aligned} s\vec{x}(s) &= A\vec{x}(s) + B\vec{u}(s) \\ \vec{y}(s) &= C\vec{x}(s) + D\vec{u}(s) \end{aligned} \quad (1.79)$$

To find the transfer function  $G(s) = \frac{y(s)}{u(s)}$  one needs to find  $x(s)$  in terms of  $u(s)$  and substitute into the output equation, i.e.

$$s\vec{x}(s) - A\vec{x}(s) = B\vec{u}(s) \quad (1.80)$$

$$(sI - A)\vec{x}(s) = B\vec{u}(s) \quad (1.81)$$

$$\vec{x}(s) = (sI - A)^{-1}B\vec{u}(s) \quad (1.82)$$

Then, by substitution

$$\vec{y}(s) = C(sI - A)^{-1}B\vec{u}(s) + D\vec{u}(s) \quad (1.83)$$

$$\vec{y}(s) = \left(C(sI - A)^{-1}B + D\right)\vec{u}(s) \quad (1.84)$$

and by the definition of the transfer function

$$G(s) = \frac{y(s)}{u(s)} \quad (1.85)$$

one has

$$G(s) = C(sI - A)^{-1}B + D \quad (1.86)$$

### Example Problem

Given: the second order linear ODE

$$\ddot{y}(t) + 2\dot{y}(t) - 4y(t) = 3u(t) \quad (1.87)$$

Determine:

- a) the equivalent transfer function
- b) the CCF LTI state-space representation

Solution:

- a) Taking the Laplace transform of

$$\ddot{y}(t) + 2\dot{y}(t) - 4y(t) = 3u(t) \quad (1.88)$$

with zero initial conditions yields

$$s^2y(s) + 2sy(s) - 4y(s) = 3u(s) \quad (1.89)$$

$$\left(s^2 + 2s - 4\right)y(s) = 3u(s) \quad (1.90)$$

$$y(s) = \frac{3}{s^2 + 2s - 4}u(s) \quad (1.91)$$

Thus, by definition

$$\underline{G(s) = \frac{3}{s^2 + 2s - 4}} \quad (1.92)$$

b) For the CCF, let  $x_1 = y$  and  $x_2 = \dot{y}$  and the formula for a 2<sup>nd</sup> order linear ODE is

$$A = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & 4 \end{bmatrix} \quad (1.93)$$

$$B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad (1.94)$$

$$C = [(b_0 - a_0 b_2) \quad (b_1 - a_1 b_2)] = [3 \quad 0] \quad (1.95)$$

$$D = b_2 = 0 \quad (1.96)$$

which provides the state-space model

$$\begin{aligned} \dot{\vec{x}}(t) &= \begin{bmatrix} 0 & 1 \\ -2 & 4 \end{bmatrix} \vec{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \\ y(t) &= \underline{\begin{bmatrix} 3 & 0 \end{bmatrix} \vec{x}(t)} \end{aligned} \quad (1.97)$$

### 1.3 Free Response of SISO LTI Systems and Stability

Recall the standard linear ODE representation for a SISO LTI system

$$y^{[n]}(t) + a_{n-1}y^{[n-1]}(t) + \cdots + a_1\dot{y}(t) + a_0y(t) = b_p u^{[p]}(t) + \cdots + b_1\dot{u}(t) + b_0u(t) \quad (1.98)$$

This lecture will look at the **free response** of a continuous-time LTI ODE which occurs when

$$u(t) = 0 \quad \forall t \geq 0 \quad (1.99)$$

which also infers that all derivatives of  $u$  are also zero. Thus, one must find the solution to the equation

$$\begin{aligned} y^{[n]}(t) + a_{n-1}y^{[n-1]}(t) + \cdots + a_1\dot{y}(t) + a_0y(t) &= 0 \\ \text{with initial conditions: } y^{[n-1]}(0), \dots, \dot{y}(0), y(0) \end{aligned} \quad (1.100)$$

which is also known as a **homogeneous solution** of the ODE. Note that as the solution only depends on the initial conditions, this is also known as an **initial value problem (IVP)** with the solution called the **initial value response**.

#### Free Response of SISO LTI Systems

As an initial guess, assume  $y(t) = e^{\lambda t}$ . Then, by substitution, one has

$$\lambda^n e^{\lambda t} + a_{n-1}\lambda^{n-1}e^{\lambda t} + \cdots + a_1\lambda e^{\lambda t} + a_0e^{\lambda t} = 0 \quad (1.101)$$

$$(\lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0) e^{\lambda t} = 0 \quad (1.102)$$

or

$$\lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0 = 0 \quad (1.103)$$

which is the characteristic equation of the linear ODE which has  $n$  roots which may be distinct or repeated, and real or complex conjugate pairs.

From calculus, it can be shown that the homogeneous solution for linear ODEs will be the sum of  $n$  terms. For a distinct real root, the single term is of the form

$$ce^{\lambda t} \quad (1.104)$$

for  $k$  repeated real roots, the  $k$  terms are

$$e^{\lambda t} \left( c_1 + c_2 t + \cdots + c_k t^{k-1} \right) \quad (1.105)$$

for distinct complex conjugate roots  $\lambda = \alpha \pm j\omega$ , the two terms are

$$e^{\alpha t} [c_1 \cos(\omega t) + c_2 \sin(\omega t)] \quad (1.106)$$

and for  $k$  repeated complex conjugate roots  $\lambda = \alpha \pm j\omega$ , the  $2k$  terms are

$$e^{\alpha t} \left[ (c_1 \cos(\omega t) + c_2 \sin(\omega t)) + (c_3 \cos(\omega t) + c_4 \sin(\omega t)) t + \cdots + (c_{2k-1} \cos(\omega t) + c_{2k} \sin(\omega t)) t^{k-1} \right] \quad (1.107)$$

where the  $c$  coefficients are real constants. These  $n$  unknowns can be determined from the  $n$  initial conditions. Thus, in general, the free response can be explicitly computed by

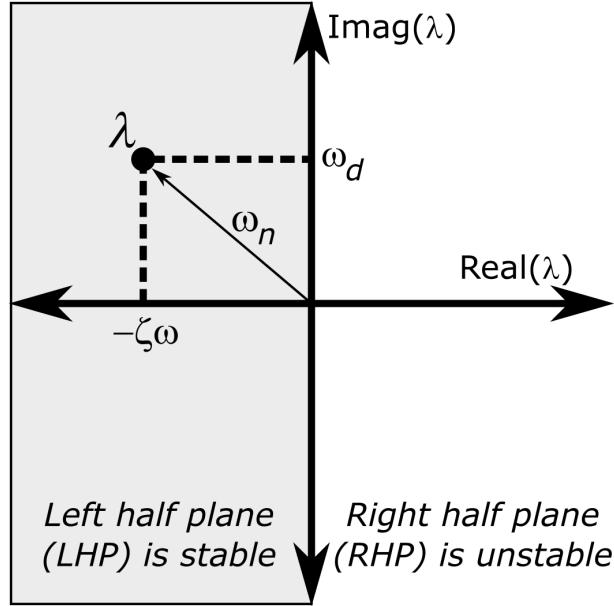
1. Solve for the  $n$  roots of the characteristic equation
2. Form the general solution from the sum of  $n$  terms
3. Use the  $n$  initial conditions to solve for the  $n$  unknown coefficients

It should also be noted that the products  $\lambda t$  have units of radians. Thus, if time is in units of seconds then the roots have units of radians/second.

## Modes and Stability

From the previous solution, note that for each root with a unique real part, there will be a unique exponential term in the free response. Each of these terms is said to refer to a unique **mode** of the system. Then, recalling that the stability of a dynamical system is described by the behavior of the system response,  $y(t)$ , as  $t \rightarrow \infty$ , and realizing the free response of LTI systems is dominated by the exponential terms, the stability of SISO LTI systems can be characterized by an analysis of the individual modes. Here one can define a mode as **strictly stable** if the real part of  $\lambda < 0$ , **marginally stable** if the real part of  $\lambda = 0$ , and **unstable** if the real part of  $\lambda > 0$ . Then, by the properties of the summation of modes on the response by extension, if *all* the modes of a SISO LTI system are stable, the system is defined as stable. Otherwise, if *any* mode is unstable, the LTI system is unstable.

A useful method for quickly visualizing the stability of SISO LTI systems is to plot the roots of characteristic equation,  $\lambda$ , (or the poles of the transfer function) in the **complex plane** where roots located in the **left half plane (LHP)** correspond to stable modes, i.e. the real part of  $\lambda$  is negative.



Thus, if all the roots/poles of the SISO LTI system are in the LHP, then the system is stable. Lastly, it should be noted the **time constant** of the  $i^{\text{th}}$  mode is defined as the inverse of the exponential decay/growth factor, i.e.

$$\tau_i = \frac{1}{\text{Real}(\lambda)} \quad (1.108)$$

where  $\text{Real}(\lambda)$  represents the real part of  $\lambda$ .

The remaining sections of this

### Free Response for First Order LTI ODE

Consider a first order LTI ODE with no input, i.e.

$$\begin{aligned} \dot{y}(t) + a_0 y(t) &= 0 \\ \text{with initial condition: } y(0) & \end{aligned} \quad (1.109)$$

The characteristic equation is

$$\lambda + a_0 = 0 \quad (1.110)$$

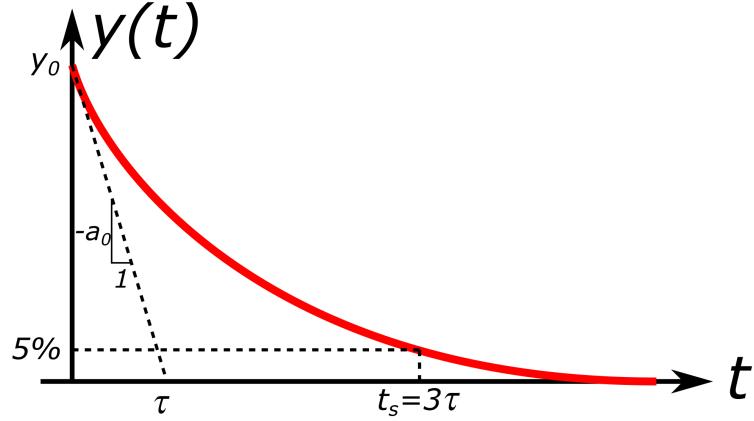
which is true for  $\lambda = -a_0$ . Thus, the stability of first order LTI system is solely based on the value of  $a_0$ , i.e. **stable**:  $a_0 > 0$  and **unstable**:  $a_0 \leq 0$ . In addition, using the initial condition at  $t = 0$

$$y(0) = c e^{-a_0(0)} = c \quad (1.111)$$

Thus, the free response for first order LTI ODEs is

$$y(t) = y(0) e^{-a_0 t} \quad (1.112)$$

For stable first order LTI systems, the output signal is an exponential decay, i.e.



where  $\tau$  is equal to  $\frac{1}{a_0}$  and the **settling time**,  $t_s$ , is the time to decay to roughly 5% of  $y(0)$ , i.e.

$$0.05y(0) = e^{-a_0 t_s} y(0) \quad (1.113)$$

$$-a_0 t_s = \ln 0.05 \quad (1.114)$$

$$t_s = \frac{2.996}{a_0} \quad (1.115)$$

$$t_s \approx 3\tau \quad (1.116)$$

### Free Response for Second Order LTI ODE

Next, consider a second order LTI ODE with no input, i.e.

$$\ddot{y}(t) + a_1 \dot{y}(t) + a_0 y(t) = 0 \quad (1.117)$$

with initial conditions:  $y(0), \dot{y}(0)$

The characteristic equation is

$$\lambda^2 + a_1 \lambda + a_0 = 0 \quad (1.118)$$

and has two roots, i.e.

$$\lambda_1 = \frac{-a_1}{2} + \frac{\sqrt{a_1^2 - 4a_0}}{2} \quad (1.119)$$

$$\lambda_2 = \frac{-a_1}{2} - \frac{\sqrt{a_1^2 - 4a_0}}{2} \quad (1.120)$$

There are three different possibilities for the values of these expressions.

1. If  $a_1^2 - 4a_0 > 0$  then the roots are real and distinct and the free response is

$$y(t) = c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t} \quad (1.121)$$

2. If  $a_1^2 - 4a_0 = 0$ , then the roots are repeated, i.e.  $\lambda_1 = \lambda_2 = -a_1/2$ , and the free response is

$$y(t) = c_1 e^{-a_1 t/2} + c_2 t e^{-a_1 t/2} \quad (1.122)$$

3. If  $a_1^2 - 4a_0 < 0$ , then the roots are a complex conjugate pair, i.e.  $\lambda_1 = \alpha + j\omega$  and  $\lambda_2 = \alpha - j\omega$ , and the free response is

$$y(t) = e^{\alpha t} [c_1 \cos(\omega t) + c_2 \sin(\omega t)] \quad (1.123)$$

Note that the constants  $c_1$  and  $c_2$  will depend on  $y(0)$  and  $\dot{y}(0)$ . Also note that a second order LTI system can be described as having two first order modes if  $\lambda_1$  and  $\lambda_2$  are both distinct real roots otherwise it has only one mode.

For stable second order LTI systems, one can also define the following two parameters, the **undamped natural frequency** as

$$\omega_n = \sqrt{a_0} > 0 \quad (1.124)$$

and the **damping ratio** as

$$\zeta = \frac{a_1}{2\omega_n} > 0 \quad (1.125)$$

Then, the characteristic equation can be rewritten as

$$\lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2 = 0 \quad (1.126)$$

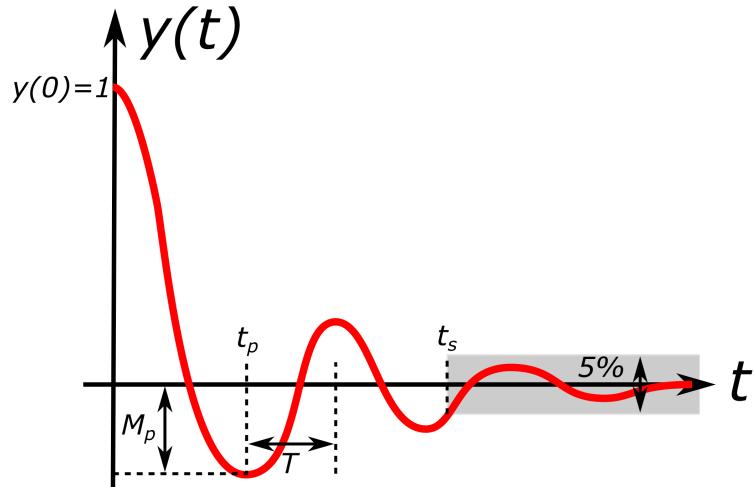
and the roots are

$$\lambda_1 = -\zeta\omega_n + \omega_n\sqrt{\zeta^2 - 1} \quad \lambda_2 = -\zeta\omega_n - \omega_n\sqrt{\zeta^2 - 1} \quad (1.127)$$

Three cases can be defined for the free response behavior

1. **underdamped**:  $0 < \zeta < 1$ , i.e. one oscillatory decaying mode
2. **overdamped**:  $\zeta > 1$ , i.e. two exponentially decaying modes
3. **critically damped**:  $\zeta = 1$ , i.e. one exponentially decaying mode

The **free response for underdamped second order LTI ODEs** behaves as



where the **damped natural frequency** is defined as

$$\omega_d = \omega_n \sqrt{1 - \zeta^2} \quad (1.128)$$

and the corresponding **period of oscillation** is defined as

$$T = \frac{2\pi}{\omega_d} \quad (1.129)$$

the **rise time** is defined as the time for the response to first reach 0 and can be shown to be

$$t_r = \quad (1.130)$$

the **peak time** is defined as the time for the response to reach peak value and can be shown to be

$$t_p = \frac{\pi}{\omega_d} \quad (1.131)$$

the **maximum overshoot** at the peak is defined as the  $|y(t_p)|$  and can be shown to be

$$M_p = e^{-\left(\zeta/\sqrt{1-\zeta^2}\right)\pi} \quad (1.132)$$

and the **settling time** is the time for the response to reach and stay within 5% of 0 and for  $\zeta < 0.8$  is approximately

$$t_s \approx \frac{3}{\zeta \omega_n} \quad (1.133)$$

The free response of overdamped second order LTI systems can be approximated by a first order LTI ODE response with the “slower” root of the characteristic equation, i.e. the root closest to zero. This approximation improves as  $\zeta \rightarrow \infty$ . The **free response of critically damped second order LTI ODEs** behave similarly to first order LTI ODEs except the settling time is no longer  $3\tau$ , but is approximately

$$t_s \approx \frac{4.744}{\omega_n} \quad (1.134)$$

## Free Response for LTI State-Space Systems

Finally, consider an LTI state-space system with no input

$$\begin{aligned} \dot{\vec{x}}(t) &= A \vec{x}(t) \\ \vec{y}(t) &= C \vec{x}(t) \end{aligned} \quad (1.135)$$

with initial conditions  $\vec{x}(0)$ . The free response can be defined as

$$\vec{y}(t) = C \vec{x}(t) \quad (1.136)$$

where, similar to the first order LTI ODE,

$$\vec{x}(t) = e^{At} \vec{x}(0) \quad (1.137)$$

and  $e^{At}$  is a **matrix exponential** since  $A$  is a matrix. The matrix exponential can be shown to be

$$e^{At} = \sum_{k=0}^{\infty} \frac{1}{k!} A^k t^k \quad (1.138)$$

where  $A^0$  is the identity matrix  $I$  and has the property

$$\frac{d}{dt} e^{At} = A e^{At} \quad (1.139)$$

Alternatively, one can use also model the free response as

$$\vec{x}(t) = e^{-\lambda t} \vec{v}(t) \quad (1.140)$$

where  $\lambda$  is an **eigenvalue** of  $A$  and  $\vec{v}$  is the corresponding **eigenvector**. To solve for the suitable eigenvalues and eigenvectors, substitute this solution into the state equation

$$(\lambda I - A) \vec{v} = 0 \quad (1.141)$$

$I$  is the identity matrix, i.e.

$$I = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \quad (1.142)$$

For nontrivial solutions, i.e.  $\vec{v} \neq 0$ , one must solve the equation

$$\det(\lambda I - A) = 0 \quad (1.143)$$

which is also known as the **eigenvalue problem**. For SISO LTI state-space systems, this equation is equivalent to the characteristic equation of the ODE. Thus, the state matrix eigenvalues of a SISO LTI system are equivalent to the roots of the characteristic equation and the poles of the transfer function.

## Second Order LTI State-Space System

For a second order LTI state system, i.e.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (1.144)$$

this equation becomes

$$\det \begin{pmatrix} \lambda - A_{11} & -A_{12} \\ -A_{21} & \lambda - A_{22} \end{pmatrix} = 0 \quad (1.145)$$

or

$$\lambda^2 + (-A_{11} - A_{22})\lambda + (A_{11}A_{22} - A_{12}A_{21}) = 0 \quad (1.146)$$

which is a polynomial equation with solutions

$$\lambda_1 = \frac{A_{11} + A_{22}}{2} + \sqrt{\frac{(A_{11} + A_{22})^2}{4} + A_{12}A_{21} - A_{11}A_{22}} \quad (1.147)$$

$$\lambda_2 = \frac{A_{11} + A_{22}}{2} - \sqrt{\frac{(A_{11} + A_{22})^2}{4} + A_{12}A_{21} - A_{11}A_{22}} \quad (1.148)$$

Notably,  $\lambda_1$  and  $\lambda_2$  can be two unique real numbers, the same real number, or a complex conjugate pair. Then, to compute corresponding eigenvectors,  $\vec{v}_1$  and  $\vec{v}_2$ , substitute  $\lambda_1$  and  $\lambda_2$  back into

$$[\lambda I - A] \vec{v} = 0 \quad (1.149)$$

It should be noted that  $\vec{v}$  can be scaled arbitrarily. Three formulas for the eigenvectors can be constructed as follows.

If  $A_{12} \neq 0$ , then

$$\vec{v}_1 = \begin{bmatrix} A_{12} \\ \lambda_1 - A_{11} \end{bmatrix} \quad \vec{v}_2 = \begin{bmatrix} A_{12} \\ \lambda_2 - A_{11} \end{bmatrix} \quad (1.150)$$

If  $A_{21} \neq 0$ , then

$$\vec{v}_1 = \begin{bmatrix} \lambda_1 - A_{22} \\ A_{21} \end{bmatrix} \quad \vec{v}_2 = \begin{bmatrix} \lambda_2 - A_{22} \\ A_{21} \end{bmatrix} \quad (1.151)$$

If  $A_{12} = A_{21} = 0$ , then

$$\vec{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \vec{v}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (1.152)$$

Finally, the free response can be written as

$$\vec{x}(t) = c_1 \vec{v}_1 e^{\lambda_1 t} + c_2 \vec{v}_2 e^{\lambda_2 t} \quad (1.153)$$

where scalar constants  $c_1$  and  $c_2$  are determined from initial conditions.

This result is similar to second order LTI ODEs, except the amplitudes of the two exponential terms are scaled by the eigenvectors as well as constants  $c_1$  and  $c_2$ . Thus, certain exponential terms affect one of the two states more than the other due to this relative eigenvector scaling. For example, if  $A_{12} = A_{21} = 0$ , then each exponential only applies to one of the states. This is known as **decoupled** states. If the relative scaling between different eigenvectors is small, then one may refer to this as **weakly coupled** states. These concepts are vital for FDC analysis and control design.

## Example Problem

Given: coupled nonlinear ODEs

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} x_2(t)^2 - 4x_1(t) \\ -9x_1(t) + u(t)^2 \end{bmatrix} \quad (1.154)$$

Determine: the approximate LTI state equation about the trim point

$$(\bar{x}, \bar{u}) = \left( \begin{bmatrix} 1 \\ 2 \end{bmatrix}, 3 \right) \quad (1.155)$$

Solution:

Let  $\vec{x}(t) = \bar{x} + \Delta\vec{x}(t)$ ,  $u(t) = \bar{u} + \Delta u(t)$

Recalling the state equation from the state-space form (without an explicit  $t$ )

$$\dot{\vec{x}}(t) = f(x, u) = \begin{bmatrix} x_2(t)^2 - 4x_1(t) \\ -9x_1(t) + u(t)^2 \end{bmatrix} \quad (1.156)$$

one can compute the Jacobian linearization as

$$A = \frac{\partial f}{\partial \vec{x}}(\bar{x}, \bar{u}) = \begin{bmatrix} -4 & 2\bar{x}_2 \\ -9 & 0 \end{bmatrix} \quad (1.157)$$

$$B = \frac{\partial f}{\partial u}(\bar{x}, \bar{u}) = \begin{bmatrix} 0 \\ 2\bar{u} \end{bmatrix} \quad (1.158)$$

Substituting  $\bar{x} = 2$  and  $\bar{u} = 3$

$$A = \begin{bmatrix} -4 & 4 \\ -9 & 0 \end{bmatrix} \quad (1.159)$$

$$B = \begin{bmatrix} 0 \\ 6 \end{bmatrix} \quad (1.160)$$

one has the LTI state-space

---


$$\Delta \dot{\vec{x}}(t) = \begin{bmatrix} -4 & 4 \\ -9 & 0 \end{bmatrix} \Delta \vec{x}(t) + \begin{bmatrix} 0 \\ 6 \end{bmatrix} \Delta u(t) \quad (1.161)$$

## 1.4 Forced and Step Response of SISO LTI Systems

The standard ODE representation for a SISO LTI system is

$$y^{[n]}(t) + a_{n-1}y^{[n-1]}(t) + \cdots + a_1\dot{y}(t) + a_0y(t) = b_p u^{[p]}(t) + \cdots + b_1\dot{u}(t) + b_0u(t) \quad (1.162)$$

The **forced response** of an LTI system occurs when

$$u(t) \neq 0 \quad \forall t \geq 0 \quad (1.163)$$

and the general solution for this ODE to can be written as

$$y(t) = y_H(t) + y_P(t) \quad (1.164)$$

where  $y_H(t)$  is the **homogeneous solution**, i.e. the solution for  $u(t) = 0$ , and  $y_P(t)$  is the **particular solution** due to  $u(t)$ . The problem of computing an analytical expression for the particular solution of arbitrary input signals can be intractable, thus, in many cases these equations must be solved numerically. For an example, the solutions for a first order LTI system without input derivatives will be computed, i.e.

$$\begin{aligned} \dot{y}(t) + a_0y(t) &= b_0u(t) \\ \text{with initial condition: } y(0) &= 0 \end{aligned} \quad (1.165)$$

From the previous lecture, the homogeneous solution is

$$y_H(t) = e^{-a_0 t} y(0) \quad (1.166)$$

and the particular solution for any  $u(t)$  for  $t \geq 0$  can be computed using a integrating factor  $e^{at}$

$$e^{at} \dot{y}_P(t) + e^{a_0 t} a_0 y_P(t) = e^{a_0 t} b_0 u(t) \quad (1.167)$$

and by recognizing the product rule of the left hand side, one has

$$\frac{d}{dt} (e^{a_0 t} y_P(t)) = e^{a_0 t} b_0 u(t) \quad (1.168)$$

Next, integrating over the input signal one has

$$e^{at} y_P(t) = \int_0^t e^{a_0 \tau} b_0 u(\tau) d\tau \quad (1.169)$$

finally, multiplying both sides by  $e^{-a_0 t}$  yields

$$y_P(t) = b_0 \int_0^t e^{-a_0(t-\tau)} u(\tau) d\tau \quad (1.170)$$

Thus,

$$y(t) = e^{-a_0 t} y(0) + b_0 \int_0^t e^{-a_0(t-\tau)} u(\tau) d\tau \quad (1.171)$$

Similarly, in the state-space formulation, it can be shown that the same form can be computed for vectors and matrices as

$$\begin{aligned} \vec{x}(t) &= e^{At} \vec{x}(0) + \int_0^t e^{A(t-\tau)} B \vec{u}(\tau) d\tau \\ \vec{y}(t) &= C \vec{x}(t) + D \vec{u}(t) \end{aligned} \quad (1.172)$$

which will be discussed later in this textbook.

## Step Response for SISO LTI Systems

An important input signal generally used to characterize SISO LTI systems is the **step input**, i.e.

$$u(t) = \begin{cases} 0 & \forall t < 0 \\ c & \forall t \geq 0 \end{cases} \quad (1.173)$$

with zero initial conditions and  $c$  is some constant. The output signal due to a step input is known as the **step response** of a system.

For a first order LTI system without an input derivative, the step response can be explicitly computed from

$$y(t) = e^{-a_0 t} y(0) + b_0 \int_0^t e^{-a_0(t-\tau)} u(\tau) d\tau \quad (1.174)$$

with  $u(\tau) = c$  and  $y(0) = 0$ . Thus,

$$y(t) = e^{-a_0 t}(0) + cb_0 \int_0^t e^{-a_0(t-\tau)} d\tau \quad (1.175)$$

Integrating with respect to  $\tau$  yields

$$y(t) = cb_0 \left[ \frac{1}{a_0} e^{-a_0(t-\tau)} \right]_0^t \quad (1.176)$$

$$y(t) = c \frac{b_0}{a_0} [1 - e^{-a_0 t}] \quad (1.177)$$

$$y(t) = c \frac{b_0}{a_0} - \frac{b_0}{a_0} e^{-a_0 t} \quad (1.178)$$

which is an exponential decay/growth dependent on the value of  $a_0$  (same as the first order free response). It should be noted that for LTI systems, the step size will simply result in a direct scaling of the output. Thus, one often analyzes the **unit step response**, i.e.  $c = 1$ .

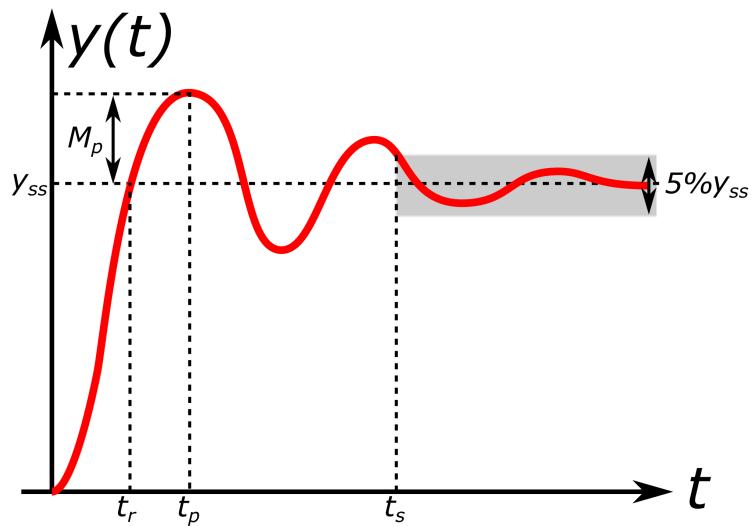
It should also be noted that for FDC another common input signal is the **doublet input**, i.e.

$$u(t) = \begin{cases} 0 & \forall t < 0 \\ c & \forall 0 \leq t < \frac{\Delta t}{2} \\ -c & \forall \frac{\Delta t}{2} \leq t < \Delta t \\ 0 & \forall t \geq \Delta t \end{cases} \quad (1.179)$$

where  $\Delta t$  is the time length of the doublet. This is noticeably similar to the step where the primary benefit is that stable systems will return to their original output as well as the integrated output state, particularly useful when the system output is an angle as the angle will return to its initial value.

### Step Response Characteristics

In general, the step response for stable SISO LTI systems has the form



which is very similar to the free response except that the system reaches some non-zero constant value as  $t \rightarrow \infty$  and is dependent on the step input magnitude. This value is the **steady-state output**,  $y_{ss}$ , also known as the **final value** for a step input. This steady-state condition occurs when all derivatives of the input and output are zero, i.e. the output or state is steady in value. Thus, by inspection of the ODE standard form for LTI systems, one has

$$y_{ss} = \frac{b_0}{a_0} u_{ss} \quad (1.180)$$

where  $u_{ss}$  is the **steady-state input** and the ratio  $\frac{b_0}{a_0}$  is called the **steady-state gain**. Accordingly, for the step response the **settling time**,  $t_s$ , is time for response to reach and stay within 5% of  $y_{ss}$  while the **rise time**,  $t_r$ , is the first time that the step response reaches  $y_{ss}$ . Furthermore, the **peak time**,  $t_p$ , is the time for the response to reach peak value and the corresponding **maximum overshoot** is

$$M_p = \frac{y(t_p) - y_{ss}}{y_{ss}} \quad (1.181)$$

Note that the formulas for these step response characteristics for first and second order ODEs *without* input derivative terms are the same as the free response and can be found in the previous section. For example, if the LTI system has no complex conjugate poles, then there will be no underdamped modes and no peak in the step response, thus one could only compute a settling time.

For higher order systems and input derivatives, one can qualitatively describe the effects from these base cases. In general, an additional pole increases the peak time and decreases the overshoot. However, this effect is only significant if the extra poles are within a factor of 10 in magnitude of the real part of the other poles. One can explain the effects of a zero as follows. Let the nominal system be

$$Y_0(s) = G(s)U(s) \quad (1.182)$$

and let the new system,  $X_1$ , have an extra zero,  $s = z$ , i.e.

$$Y_1(s) = G(s)U(s) \left( \frac{-1}{z} s + 1 \right) \quad (1.183)$$

Then,

$$Y_1(s) = \left( \frac{-1}{z} s + 1 \right) Y_0(s) \quad (1.184)$$

and in the time domain

$$y_1(t) = \frac{-1}{z} \dot{y}_0(t) + y_0(t) \quad (1.185)$$

Thus, the step response consists of the nominal response plus the derivative (or slope) of the response and since the derivative is initially positive for the nominal response, one can infer that if  $z < 0$ , then there will be more overshoot and if  $z > 0$ , then there will be an initial undershoot and more overshoot. However, this effect is only significant if the extra zeros are within a factor of 10 in magnitude of the real part of the poles. If the real part of the zero is positive, then it is a **right half plane (RHP) zero**, otherwise its a **left half plane (LHP) zero**, and a RHP zero causes initial undershoot and more overshoot while a LHP zero simply causes more overshoot.

As an example of these effects, consider the following second order LTI system as an example

$$\ddot{y} + 2\dot{y} + 4y = 4u \quad (1.186)$$

or as a transfer function,  $G_1(s)$ , is

$$G_1(s) = \frac{4}{s^2 + 2s + 4} \quad (1.187)$$

$G_1$  is underdamped, has poles at  $-1 \pm j1.73$  with  $\omega_n = 2$  and  $\zeta = 0.5$  which corresponds to an overshoot,  $M_p \approx 0.16$ , and has a steady-state output of  $y_{ss} = \frac{4}{4} = 1$ .

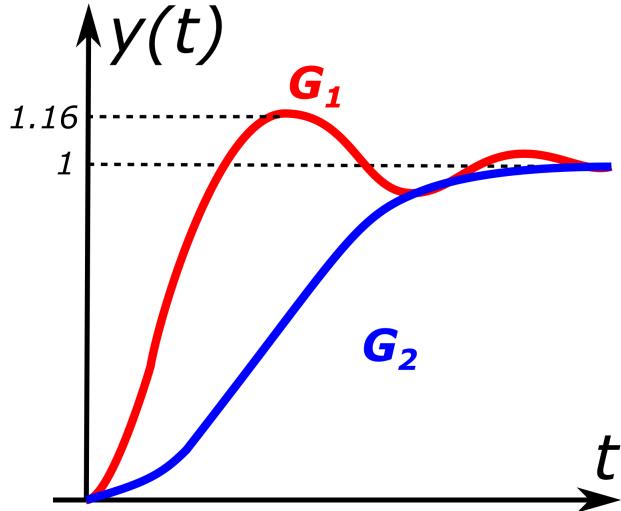
First, consider an additional pole at  $s = -1$ , then one has the third order LTI ODE

$$\ddot{y} + 3\dot{y} + 6y + 4y = 4u \quad (1.188)$$

and transfer function

$$G_2(s) = \frac{4}{s^2 + 2s + 4} \frac{1}{s + 1} \quad (1.189)$$

Plotting the unit step response for both  $G_1(s)$  and  $G_2(s)$ , one has



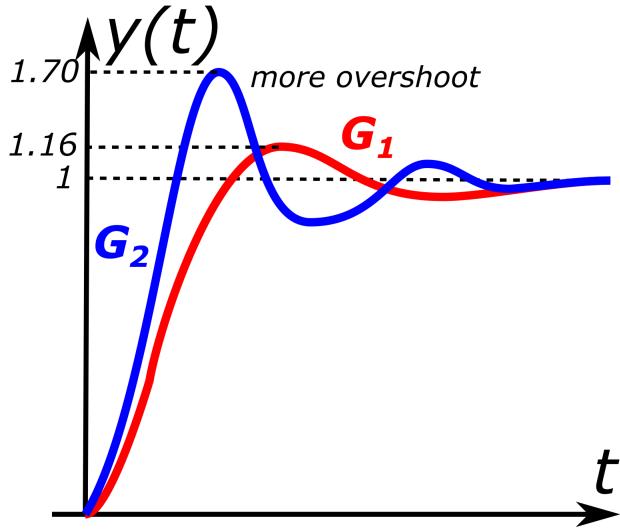
Next, consider a LHP zero at  $s = -1$ , then one has the second order LTI ODE

$$\ddot{y} + 2\dot{y} + 4y = 4\dot{u} + 4u \quad (1.190)$$

and transfer function

$$G_2(s) = \frac{4(s+1)}{s^2 + 2s + 4} \quad (1.191)$$

Plotting the unit step response for both  $G_1(s)$  and  $G_2(s)$ , one has



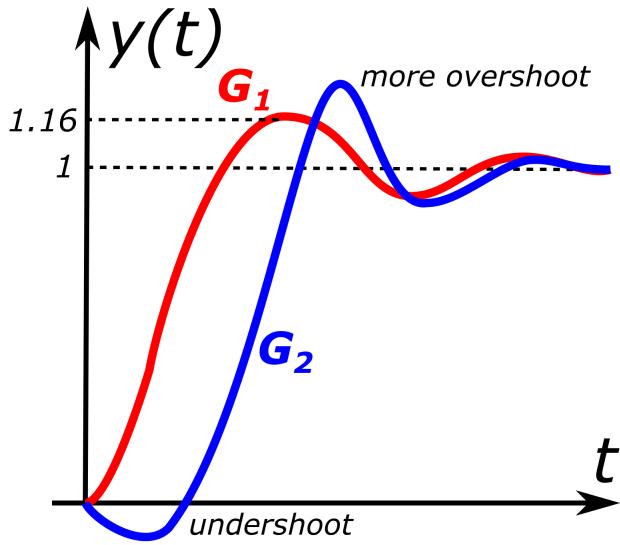
Lastly, consider a RHP zero at  $s = 1$ , then one has the second order LTI ODE

$$\ddot{y} + 2\dot{y} + 4y = -4\dot{u} + 4u \quad (1.192)$$

and transfer function

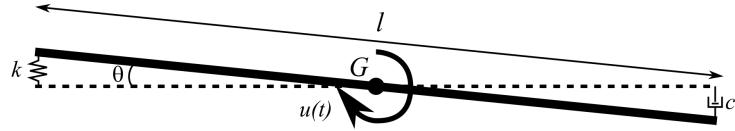
$$G_2(s) = \frac{4(-s+1)}{s^2 + 2s + 4} \quad (1.193)$$

Plotting the unit step response for both  $G_1(s)$  and  $G_2(s)$ , one has



### Example Problem

Given: A rod is allowed to pitch about  $G$  with the following forces and moments applied as shown in the free body diagram



For small angles

$\theta_0 = 0$	$\dot{\theta}_0 = 0$	$c = 1.1$
$\ell = 2$	$I_y = 0.1$	$k = 1$

Determine:

- a)  $\omega_n$
- b)  $\zeta$
- c)  $\theta(t)$  for a unit step input
- d) the dominant term in the response

Solution:

$$\Sigma M_G = I_y \ddot{\theta} \quad (1.194)$$

$$-c \left( \frac{\ell}{2} \frac{d \sin \theta}{dt} \right) \frac{\ell}{2} - k \left( \frac{\ell}{2} \sin \theta \right) \frac{\ell}{2} + u(t) = I_y \ddot{\theta} \quad (1.195)$$

$$-\frac{c\ell^2}{4} \cos \theta \dot{\theta} - \frac{k\ell^2}{4} \sin \theta + u(t) = I_y \ddot{\theta} \quad (1.196)$$

$$\ddot{\theta} + \frac{c\ell^2}{4I_y} \cos \theta \dot{\theta} + \frac{k\ell^2}{4I_y} \sin \theta = \frac{1}{I_y} u(t) \quad (1.197)$$

Using the small angle approximation,

$$\ddot{\theta} + \frac{c\ell^2}{4I_y} \dot{\theta} + \frac{k\ell^2}{4I_y} \theta = \frac{1}{I_y} u(t) \quad (1.198)$$

Substituting values

$$\ddot{\theta} + 11\dot{\theta} + 10\theta = 10u(t) \quad (1.199)$$

which reflects the LTI ODE standard form

$$\ddot{y} + a_1 \dot{y} + a_0 y = b_0 u \quad (1.200)$$

a)

$$\omega_n = \sqrt{a_0} \quad (1.201)$$

$$\omega_n = \sqrt{10} \quad (1.202)$$

$$\underline{\omega_n = 3.16} \quad (1.203)$$

b)

$$\zeta = \frac{a_1}{2\omega_n} \quad (1.204)$$

$$\zeta = \frac{11}{2\sqrt{10}} \quad (1.205)$$

$$\underline{\zeta = 1.74} \quad (> 1, \text{overdamped}) \quad (1.206)$$

c) Solving for the roots of the characteristic equation

$$\lambda^2 + 11\lambda + 10 = 0 \quad (1.207)$$

provides

$$\lambda_1 = -1 \quad \lambda_2 = -10 \quad (1.208)$$

which is stable. The general solution for a second order unit step

$$\theta(t) = \frac{b_0}{\omega_n^2} + c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t} \quad (1.209)$$

$$\theta(t) = \frac{10}{\sqrt{10}^2} + c_1 e^{-t} + c_2 e^{-10t} \quad (1.210)$$

$$\theta(t) = 1 + c_1 e^{-t} + c_2 e^{-10t} \quad (1.211)$$

and taking the derivative

$$\dot{\theta}(t) = -c_1 e^{-t} + -10c_2 e^{-10t} \quad (1.212)$$

Solving for  $c_1$  and  $c_2$ 

$$\theta(0) = 0 = 1 + c_1 + c_2 \quad \dot{\theta}(0) = 0 = -c_1 - 10c_2 \quad (1.213)$$

gives

$$c_1 = -\frac{10}{9} \quad c_2 = \frac{1}{9} \quad (1.214)$$

Thus,

$$\underline{\theta(t) = 1 - \frac{10}{9}e^{-t} + \frac{1}{9}e^{-10t}} \quad (1.215)$$

d) The slower component,  $e^{-t}$ , dominates response (e.g.  $t_s$  of 3 s vs. 0.33 s).

## 1.5 Impulse and Sinusoidal Response of SISO LTI Systems

Dynamical systems use signals as the time-dependent input and output variables. An important type of signal is a **periodic signal** which is one that repeats its sequence of values exactly after a fixed length of time, known as the period. Similar to the Taylor series for arbitrary functions, the **Fourier series** of a periodic function is its representation as the summation of harmonically related sinusoids. Furthermore, as the period of the function is allowed to approach infinity, the Fourier series becomes the **Fourier transform**

$$F(\omega) = \mathcal{F}\{f(t)\} = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \quad (1.216)$$

where  $\omega$  is the **angular frequency**, typically given in radians/second. Sometimes the **periodic frequency**,  $\xi$ , is used where  $\omega = 2\pi\xi$  and typically given in Hertz (Hz) or cycles/second. Note that this uses **Euler's formula** for the harmonically related sinusoids, i.e.

$$e^{-j\omega t} = \cos(\omega t) - j\sin(\omega t) \quad (1.217)$$

The **Fourier inversion theorem** states for "well-behaved" functions (e.g. linear ODEs) it is possible to recover a function from its Fourier transform and one can define the **inverse Fourier transform** as

$$f(t) = \mathcal{F}^{-1}\{F(\omega)\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{j\omega t} d\omega \quad (1.218)$$

Thus, for dynamical systems analysis if one knows all the frequency information about a signal then one may reconstruct the original signal precisely in time. Thus, using the frequency domain for signals instead of the time domain allows one to completely analyze "well-behaved" system responses to *any* input. Furthermore, as the Fourier and inverse Fourier transforms are exactly equivalent to the Laplace transforms with the substitution  $s = j\omega$ , frequency domain analysis directly uses the transfer function representation for SISO LTI systems. For MIMO systems, one must employ multivariate frequency domain analysis which is discussed later in this textbook.

## Impulse Response

The **Dirac delta** can be loosely thought of as a function on the real line which is zero everywhere except at the origin, where it is infinite, i.e.

$$\delta(x) = \begin{cases} +\infty, & x = 0 \\ 0, & x \neq 0 \end{cases} \quad (1.219)$$

and which is also constrained to satisfy the identity

$$\int_{-\infty}^{\infty} \delta(x) dx = 1 \quad (1.220)$$

However, this is merely a heuristic characterization as the Dirac delta is not a function in the traditional sense as no function defined on the real numbers has these properties. The Dirac delta function can be rigorously defined either as a distribution or as a measure, but an intuitive understanding is as an **impulse function**. For example, suppose that a force is uniformly distributed over a small time interval  $\Delta t$  and imparts some momentum  $P$  to another object, i.e.

$$F_{\Delta t}(t) = \begin{cases} \frac{P}{\Delta t}, & 0 < t < \Delta t \\ 0, & \text{otherwise.} \end{cases} \quad (1.221)$$

Then, the momentum at any time  $t$  is found by integration

$$p(t) = \int_0^t F_{\Delta t}(\tau) d\tau = \begin{cases} P, & t > \Delta t \\ \frac{Pt}{\Delta t}, & 0 < t < \Delta t \\ 0, & t \leq 0. \end{cases} \quad (1.222)$$

Next, by considering a model situation of an instantaneous transfer of momentum, i.e. an impulse, which requires taking the limit as  $\Delta t \rightarrow 0$ , providing

$$p(t) = \begin{cases} P, & t > 0 \\ 0, & t \leq 0. \end{cases} \quad (1.223)$$

Here  $F_{\Delta t}$  can be thought of as a useful approximation to the idea of instantaneous transfer of momentum. The Dirac delta function allows one to construct the idealized limit of these approximations where  $\lim_{\Delta t \rightarrow 0} F_{\Delta t}$  is zero everywhere and infinite at a single point, but is also limited by

$$\int_{-\infty}^{\infty} F_{\Delta t}(t) dt = P \quad \forall \Delta t > 0 \quad (1.224)$$

or

$$F(t) = \lim_{\Delta t \rightarrow 0} F_{\Delta t}(t) = P\delta(t) \quad (1.225)$$

With this example in mind, consider the SISO LTI system response for an **impulse input**,

$$u(t) = \delta(t) \quad (1.226)$$

Then, by the Laplace transform one has

$$U(s) = \int_0^{\infty} \delta(t) e^{-st} dt \quad (1.227)$$

or by the definition of  $\delta$  as zero everywhere except at  $t = 0$  and its integral constraint as well as  $e^{-s(0)} = 1$ , one has

$$U(s) = 1 \quad (1.228)$$

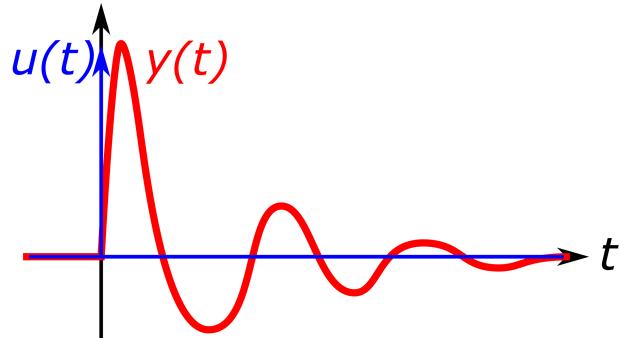
Thus,

$$Y(s) = G(s) \quad (1.229)$$

or in the time domain by the inverse Laplace transform

$$y(t) = g(t) \quad (1.230)$$

where  $g(t)$  is called the **impulse response**. Thus, by simulating an LTI in the time domain using an impulse input, one can completely determine the transfer function transformed to the time domain. This is a useful technique for system identification (ID) of an *unknown* LTI system. An example of an impulse response plot is



which is somewhat similar to the free response, except the excitation of the system occurs via the input and not the initial conditions (assumed to be zero).

Furthermore, using the impulse response, one can define the transfer function output response

$$Y(s) = G(s)U(s) \quad (1.231)$$

in the time domain as a **convolution integral** between the impulse response and any input as

$$y(t) = \int_0^t g(\tau)u(t-\tau)d\tau \quad (1.232)$$

Thus, another system ID method is known as **deconvolution** which uses measured output and input signals in the time domain to estimate the impulse response. Lastly, it should be noted that the discrete system version to the Dirac delta is called the **Kronecker delta** which allows for the definition of the discrete-time impulse response.

## Sinusoidal and Frequency Response

Now consider a sinusoidal input at frequency  $\omega$  in the time domain as

$$u(t) = \cos(\omega t) \quad (1.233)$$

Then, using the convolution integral for the output, one has

$$y(t) = \int_0^t g(\tau) (\cos(\omega(t-\tau))) d\tau \quad (1.234)$$

or

$$y(t) = \int_0^\infty g(\tau) (\cos(\omega(t-\tau))) d\tau - \int_t^\infty g(\tau) (\cos(\omega(t-\tau))) d\tau \quad (1.235)$$

where, if the system is stable, the first term is the steady-state sinusoidal response and the second term is the **transient response** which decays with  $t$ .

Thus, any stable system output will reach the steady-state sinusoidal response given by

$$y_{sss}(t) = \int_0^\infty g(\tau) (\cos(\omega(t-\tau))) d\tau \quad (1.236)$$

which using Euler's formula for the input as

$$u(t) = \frac{1}{2} (e^{j\omega t} + e^{-j\omega t}) \quad (1.237)$$

one has

$$y_{sss}(t) = \frac{1}{2} \int_0^\infty g(\tau) e^{j\omega(t-\tau)} d\tau + \frac{1}{2} \int_0^t g(\tau) e^{-j\omega(t-\tau)} d\tau \quad (1.238)$$

or

$$y_{sss}(t) = \frac{1}{2} e^{j\omega t} \int_0^\infty g(\tau) e^{-j\omega\tau} d\tau + \frac{1}{2} e^{-j\omega t} \int_0^\infty g(\tau) e^{j\omega\tau} d\tau \quad (1.239)$$

and using the definition of the transfer function using the Laplace transform

$$y_{sss}(t) = \frac{1}{2}e^{j\omega t}G(j\omega) + \frac{1}{2}e^{-j\omega t}G(-j\omega) \quad (1.240)$$

or finally, the **steady-state sinusoidal response**

$$y_{sss}(t) = \operatorname{Re}\{G(j\omega)\} \cos(\omega t) - \operatorname{Im}\{G(j\omega)\} \sin(\omega t) \quad (1.241)$$

which can alternatively be written in magnitude and phase as

$$y_{sss}(t) = |G(j\omega)| \cos(\omega t + \angle G(j\omega)) \quad (1.242)$$

which demonstrates that the steady-state sinusoidal response is a sinusoid at the same frequency, but with a different magnitude and phase dependent on the transfer function evaluated at  $s = j\omega$ , i.e.  $G(j\omega)$ .

Lastly, note that if one uses the Fourier transform for a sinusoidal input as

$$f(t) = e^{j\omega_0 t} \quad (1.243)$$

the Fourier transform provides

$$F(\omega) = \int_{-\infty}^{\infty} e^{j\omega_0 t} e^{-j\omega t} dt \quad (1.244)$$

or

$$F(\omega) = \int_{-\infty}^{\infty} e^{j(\omega_0 - \omega)t} dt \quad (1.245)$$

which is an alternative definition to the Dirac delta function,  $\delta(\omega - \omega_0)$ , as the frequency response will be singular at  $\omega = \omega_0$  and zero everywhere else.

## 1.6 Frequency Response of SISO LTI Systems

When  $G(j\omega)$  is considered as a continuous function of all  $\omega \in [0, \infty)$  one is said to have the **frequency response** of the SISO LTI system. The frequency response can be regarded as the magnitude and phase values for all steady-state sinusoidal responses. Alternatively, it can also be regarded as the Fourier transform of the impulse response, i.e.  $g(t) \rightarrow G(j\omega)$  for all  $\omega \in [0, \infty)$ . Furthermore, with the Fourier inversion theorem in mind, the frequency response provides the SISO LTI system response to *all* harmonic sinusoids within any “well-behaved” input signal from which one can exactly replicate the output signal.

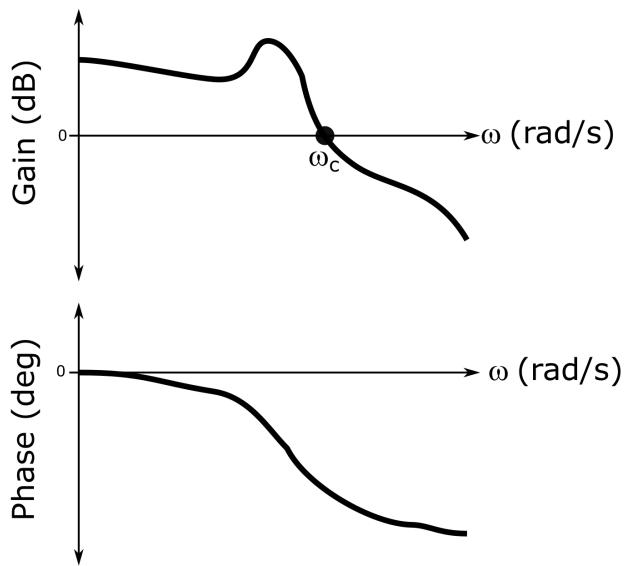
It should be noted that this also includes the step input which suggest that the step response and frequency response characteristics can be directly related. Furthermore, the steady-state output is equivalent to the frequency response at  $\omega = 0$ . The magnitude of the steady-state step response is called the **DC gain** due to the importance of direct current (DC) and alternating current (AC) for performing circuit analysis.

Two common plots used to analyze the frequency response of an LTI system are the Bode and Nyquist plots.

The **Bode plot** consists of two subplots:

1. the magnitude/gain,  $|G(j\omega)|$  (vertical) versus  $\omega$  (horizontal)
2. the phase  $\angle G(j\omega)$  (vertical) versus  $\omega$  (horizontal)

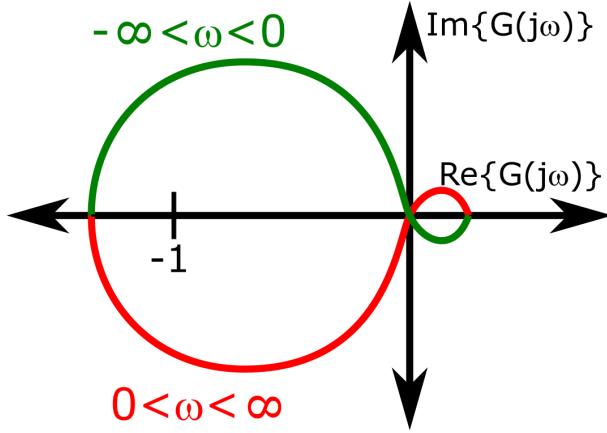
For the Bode plot, it is common to plot the frequency,  $\omega$ , on a  $\log_{10}$  scale in (radians/sec), the phase,  $\angle G(j\omega)$ , in degrees ( $^\circ$ ), and the magnitude/gain,  $|G(j\omega)|$ , in units of decibels (dB), i.e.  $20 \log_{10} |G(j\omega)|$ . An example of a Bode plot is the following:



Some useful conversions for decibels to gain are as follows.

Decibels (dB)	Gain
40 dB	100
20 dB	10
6 dB	2
3 dB	$\sqrt{2}$
0 dB	1
-3 dB	$\frac{1}{\sqrt{2}}$
-6 dB	$\frac{1}{2}$
-20 dB	$\frac{1}{10}$
-40 dB	$\frac{1}{100}$

A companion to the Bode plot, the **Nyquist plot** consists of the real and imaginary parts of  $G(j\omega)$  plotted as a single curve in the complex plane from  $\omega = -\infty$  to  $\omega = \infty$ . An example of a Nyquist plot is the following:



This part of the textbook will make extensive use of the Bode plots and use the Nyquist plot in a later chapter on SISO LTI system robustness. This section will discuss simple Bode plots for first order LTI systems with a single pole or zero as well as underdamped second order LTI systems, then extend these characteristics to understanding Bode plots for higher order LTI systems.

### Bode Plot of First Order SISO LTI System

Consider the first order LTI system with linear ODE

$$\dot{y} + a_0 y = b_0 u \quad (1.246)$$

and transfer function

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_0}{s + a_0} \quad (1.247)$$

which has a root/pole at  $s = -a_0$ .

Recall the following characteristics for first order LTI systems.

1. The LTI system is stable if  $a_0 > 0$ .
2. If  $u(t) = \bar{u} \ \forall t \geq 0$  where  $\bar{u}$  is some constant, then  $y(t) \rightarrow y_{ss} = \frac{b_0}{a_0} \bar{u}$  as  $t \rightarrow \infty$ .
3. The settling time is  $t_s = \frac{3}{a_0}$ , thus as  $a_0$  increases  $t_s$  decreases, i.e. the speed of response increases.

To sketch the Bode plot of the frequency response, i.e.  $|G(s)|$  and  $\angle G(s)$  at  $s = j\omega$ , requires computing  $G(j\omega)$  explicitly

$$G(j\omega) = \frac{b_0}{j\omega + a_0} \quad (1.248)$$

$$G(j\omega) = \frac{b_0}{j\omega + a_0} \frac{a_0 - j\omega}{a_0 - j\omega} \quad (1.249)$$

$$G(j\omega) = \frac{a_0 b_0}{a_0^2 + \omega^2} - j \frac{\omega b_0}{a_0^2 + \omega^2} \quad (1.250)$$

Now,

$$|G(j\omega)| = \sqrt{\left(\frac{a_0 b_0}{a_0^2 + \omega^2}\right)^2 + \left(\frac{\omega b_0}{a_0^2 + \omega^2}\right)^2} \quad (1.251)$$

$$|G(j\omega)| = \sqrt{\frac{b_0^2(a_0^2 + \omega^2)}{(a_0^2 + \omega^2)^2}} \quad (1.252)$$

$$|G(j\omega)| = \frac{b_0}{\sqrt{a_0^2 + \omega^2}} \quad (1.253)$$

$$20 \log_{10} |G(j\omega)| = 20 \log_{10} \frac{b_0}{\sqrt{a_0^2 + \omega^2}} \quad (1.254)$$

and

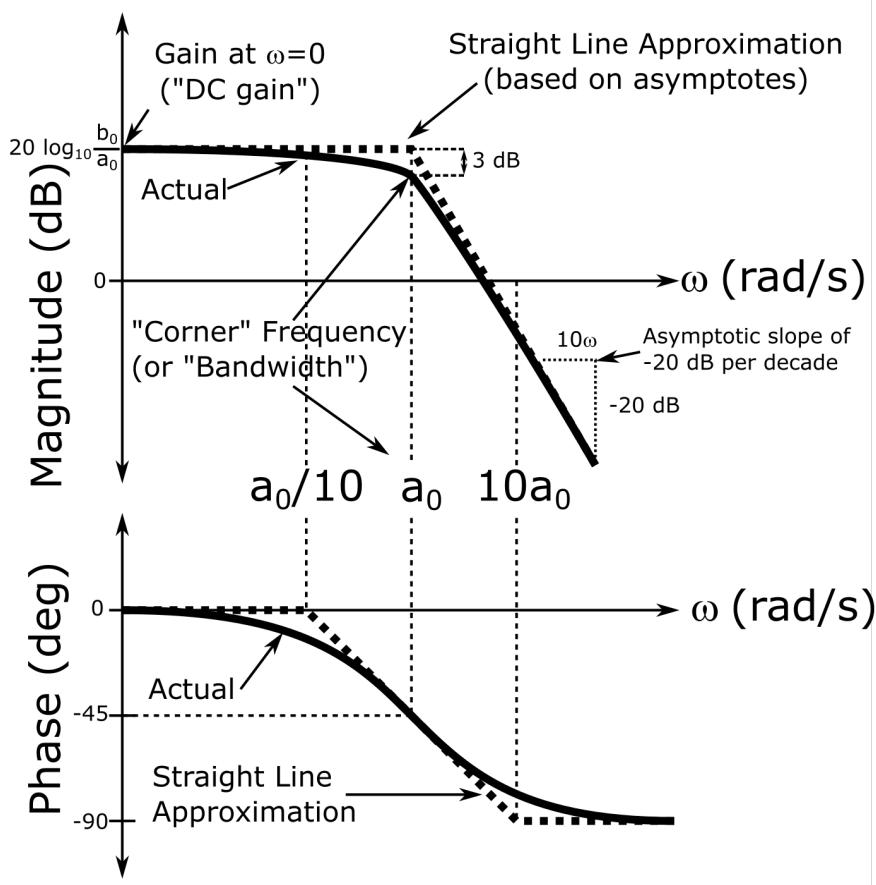
$$\angle G(j\omega) = \tan^{-1} \begin{bmatrix} -\omega b_0 \\ \frac{a_0^2 + \omega^2}{a_0 b_0} \end{bmatrix} \quad (1.255)$$

$$\angle G(j\omega) = -\tan^{-1} \left[ \frac{\omega}{a_0} \right] \quad (1.256)$$

To sketch the frequency response, one should consider the asymptotic behavior and the frequency response at the pole. If  $b_0 > 0$  and recalling  $G(j\omega) = \frac{b_0}{j\omega + a_0}$ , one can deduce the following

	$G(j\omega)$	$20 \log_{10}  G(j\omega) $	$\angle G(j\omega)$
$\omega \ll a_0:$	$\approx \frac{b_0}{a_0}$	$\approx 20 \log_{10} \frac{b_0}{a_0}$	$\approx 0$
$\omega \gg a_0:$	$\approx \frac{b_0}{j\omega} = -j \frac{b_0}{\omega}$	$\approx 20 \log_{10} \frac{b_0}{\omega}$	$\approx -90^\circ$
$\omega = a_0:$	$= \left(\frac{1}{1+j}\right) \frac{b_0}{a_0} = \left(\frac{1-j}{2}\right) \frac{b_0}{a_0}$	$= 20 \log_{10} \left(\frac{1}{\sqrt{2}}\right) \frac{b_0}{a_0}$	$= -45^\circ$

Plotting for a stable system, one has



For an unstable system the phase plot will go from  $-180^\circ$  to  $-90^\circ$ . The step response settling time for a first order LTI system is given by  $t_s = \frac{3}{a_0}$ , thus increasing  $a_0$  will decrease  $t_s$ . In the frequency domain, this corresponds to increasing the bandwidth. Secondly, note the steady-state step response,  $y_{ss} = \frac{b_0}{a_0}$  whose absolute value  $|G(0)| = \left| \frac{b_0}{a_0} \right|$  is the DC gain. Lastly, it should be noted that changing  $a_0 > 0$  and/or  $b_0 < 0$  will only change the phase of the Bode plot, i.e.

	$b_0 > 0$	$b_0 < 0$
$a_0 > 0$ (i.e. stable)	$\angle G(j\omega)$ from $0^\circ \rightarrow -90^\circ$	$\angle G(j\omega)$ from $180^\circ \rightarrow 90^\circ$
$a_0 < 0$ (i.e. unstable)	$\angle G(j\omega)$ from $-180^\circ \rightarrow -90^\circ$	$\angle G(j\omega)$ from $0^\circ \rightarrow 90^\circ$

where it should be noted that for  $a_0 < 0$  or  $b_0 < 0$ , the step response steady-state is still  $y_{ss} = \frac{b_0}{a_0}$  which corresponds to the DC gain  $|G(0)| = \left| \frac{b_0}{a_0} \right|$  at  $\angle G(j\omega) = 0^\circ$  or  $\pm 180^\circ$ .

## Bode Plots for Real Zero LTI System

Consider the SISO LTI system with linear ODE

$$u(t) = ae + b\dot{e} \quad (1.257)$$

and transfer function

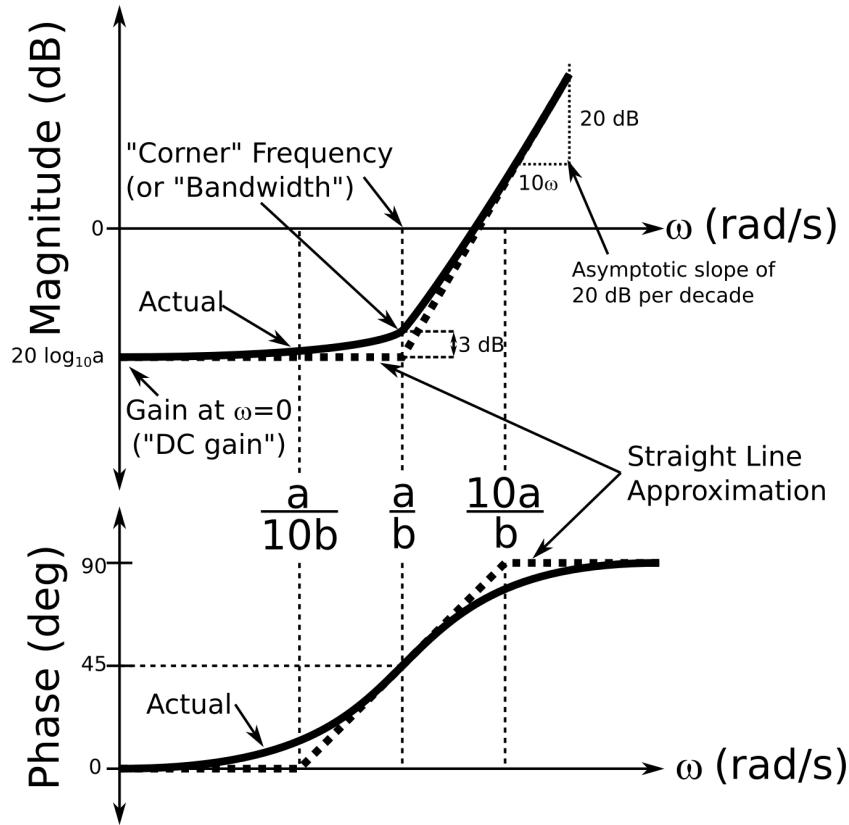
$$G(s) = \frac{U(s)}{E(s)} = a + bs \quad (1.258)$$

which has a zero at  $s = -\frac{a}{b}$ .

To sketch the frequency response, one should consider the asymptotic behavior of  $G(j\omega) = a + jb\omega$  (assuming  $a > 0$  and  $b > 0$ ) and the frequency response at the zero.

	$G(j\omega)$	$20 \log_{10}  G(j\omega) $	$\angle G(j\omega)$
$\omega \ll \frac{a}{b}$ :	$\approx a$	$\approx 20 \log_{10} a$	$\approx 0^\circ$
$\omega \gg \frac{a}{b}$ :	$\approx jb\omega$	$\approx 20 \log_{10} b\omega$	$\approx 90^\circ$
$\omega = \frac{a}{b}$ :	$= a + ja$	$= 20 \log_{10} a\sqrt{2}$	$= 45^\circ$

where it should be noted that  $\angle G(j\omega)$  will change if  $a < 0$  and/or  $b < 0$ . Plotting, one has



## Bode Plots for Underdamped Second Order LTI Systems

Consider an underdamped second order LTI system with linear ODE

$$\ddot{y} + 2\zeta\omega_n\dot{y} + \omega_n^2 y = b_0 u \quad (1.259)$$

and transfer function

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_0}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (1.260)$$

which has poles at  $s = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}$ .

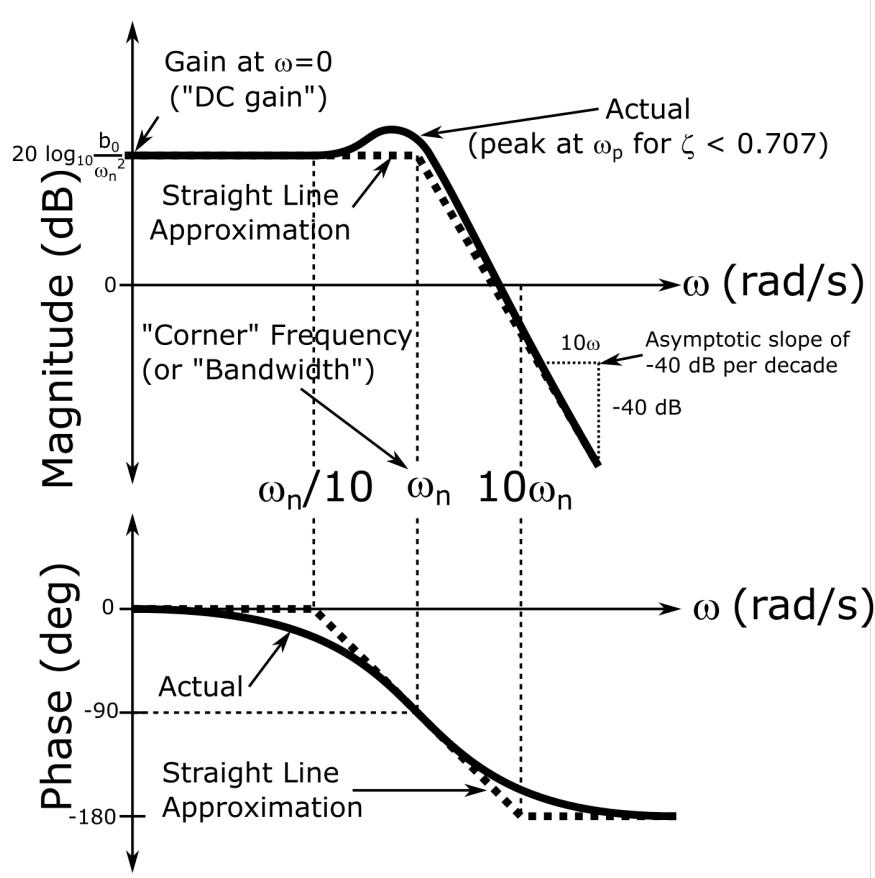
Recall the following characteristics for underdamped second order LTI systems.

1. The system is stable if  $\zeta\omega_n > 0$  and  $\omega_n^2 > 0$ .
2. If  $u(t) = \bar{u} \ \forall t \geq 0$  where  $\bar{u}$  is some constant, then  $y(t) \rightarrow y_{ss} = \frac{b_0}{\omega_n^2}\bar{u}$  as  $t \rightarrow \infty$ .
3. The settling time,  $t_s \approx \frac{3}{\zeta\omega_n}$ , thus increasing  $\zeta\omega_n$  decreases  $t_s$ , i.e. increases the speed of response.
4. The overshoot is  $M_p = e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}}$ .

If  $b_0 > 0$ , one can sketch the Bode plot by considering the asymptotes.

	$G(j\omega)$	$20 \log_{10}  G(j\omega) $	$\angle G(j\omega)$
$\omega \ll \omega_n:$	$\approx \frac{b_0}{\omega_n^2}$	$\approx 20 \log_{10} \frac{b_0}{\omega_n^2}$	$\approx 0$
$\omega \gg \omega_n:$	$\approx \frac{b_0}{(j\omega)^2} = -\frac{b_0}{\omega^2}$	$\approx 20 \log_{10} \frac{b_0}{\omega^2}$	$\approx \pm 180^\circ$
$\omega = \omega_n:$	$= \frac{b_0}{j2\zeta\omega_n^2} = -j\frac{b_0}{2\zeta\omega_n^2}$	$= 20 \log_{10} \frac{b_0}{2\zeta\omega_n^2}$	$= -90^\circ$

where it should be noted that  $\angle G(j\omega)$  will change if  $b < 0$ . and plotting these results, one has



where it should be noted that if  $\zeta < \frac{1}{\sqrt{2}} = 0.707$ , then at  $\omega_p = \omega_n\sqrt{1 - 2\zeta^2}$  there will be a **resonant peak** of magnitude/gain

$$|G(j\omega_p)| = \frac{b_0}{2\zeta\omega_n^2\sqrt{1 - \zeta^2}} \quad (1.261)$$

which can be proven using calculus, but is left for the reader. This peak becomes more pronounced for small damping ratios, e.g. if  $\zeta \leq 0.3$ , the  $|G(j\omega_p)| \approx \frac{b_0}{2\zeta\omega_n^2}$ .

## Bode Plots for Higher Order LTI Systems

For higher order SISO LTI systems one can use the fact that for complex numbers, e.g. multiplying  $n_1 = A_1 e^{j\phi_1}$  and  $n_2 = A_2 e^{j\phi_2}$  using polar form results in the expression

$$n_1 n_2 = A_1 A_2 e^{j(\phi_1 + \phi_2)} \quad (1.262)$$

which is a complex number with magnitude/gain  $|A_1 A_2|$  and phase  $\phi_1 + \phi_2$ . Furthermore, if the magnitude/gain is in dB, then one has

$$20 \log_{10} |A_1 A_2| = 20 \log_{10} |A_1| + 20 \log_{10} |A_2| \quad (1.263)$$

$$|A_1 A_2|_{\text{dB}} = |A_1|_{\text{dB}} + |A_2|_{\text{dB}} \quad (1.264)$$

Thus, the Bode plot for higher order SISO LTI systems can be constructed by factoring the numerator and denominator of its transfer function into its real and complex conjugate pairs of poles and zeros, then, adding together the Bode plot contributions from each pole and zero. It should be noted that the inverse is also true. From the Bode plot, one can estimate the poles and zeros of the SISO LTI system, i.e. the modes, which also acts as the basis for some system ID methods.

Lastly, it should also be noted that the **asymptotic slope** of the Bode plot can be calculated by assuming that the highest order term (order  $n$ ) in the numerator and denominator will dominate the frequency response as  $\omega \rightarrow \infty$ . Thus, either the approximate gain response will be 1, i.e. 0 dB, if the highest order terms are the same, otherwise,

$$|G(j\omega)|_{\text{dB}} \approx 20 \log_{10} \omega^n \quad (1.265)$$

where  $n$  will be positive if the numerator order is higher than the denominator and negative if the denominator order is higher than the numerator. Then, considering what happens when one increases  $\omega$  by a factor of 10, one has

$$|G(j\omega)|_{\text{dB}} \approx 20 \log_{10}(10\omega)^n \quad (1.266)$$

$$|G(j\omega)|_{\text{dB}} \approx 20 \log_{10} \omega^n + 20 \log_{10} 10^n \quad (1.267)$$

or

$$|G(j\omega)|_{\text{dB}} \approx 20 \log_{10} \omega^n + 20n \quad (1.268)$$

Thus, as one moves up a decade in frequency  $\omega$ , one will increase by  $20n$  where  $n$  is the highest order term of the transfer function (may be negative).

## Example Problem 1

Given: the first order SISO LTI system,  $K$ , with linear ODE

$$0.1\dot{u}(t) + 10u(t) = \dot{e}(t) + 10e(t) \quad (1.269)$$

Determine:

- a) the transfer function,  $K(s)$
- b) a sketch of the Bode plot using the straight line approximation

Solution:

- a) The transfer function is

$$K(s) = \frac{s + 10}{0.1s + 10} \quad (1.270)$$

which has a real zero at  $s = -10$  and a real pole at  $s = -100$ .

- b) One can write  $K(s)$  as the product of

$$K(s) = K_1(s)K_2(s) = \left( \frac{1}{0.1s + 10} \right) (s + 10) \quad (1.271)$$

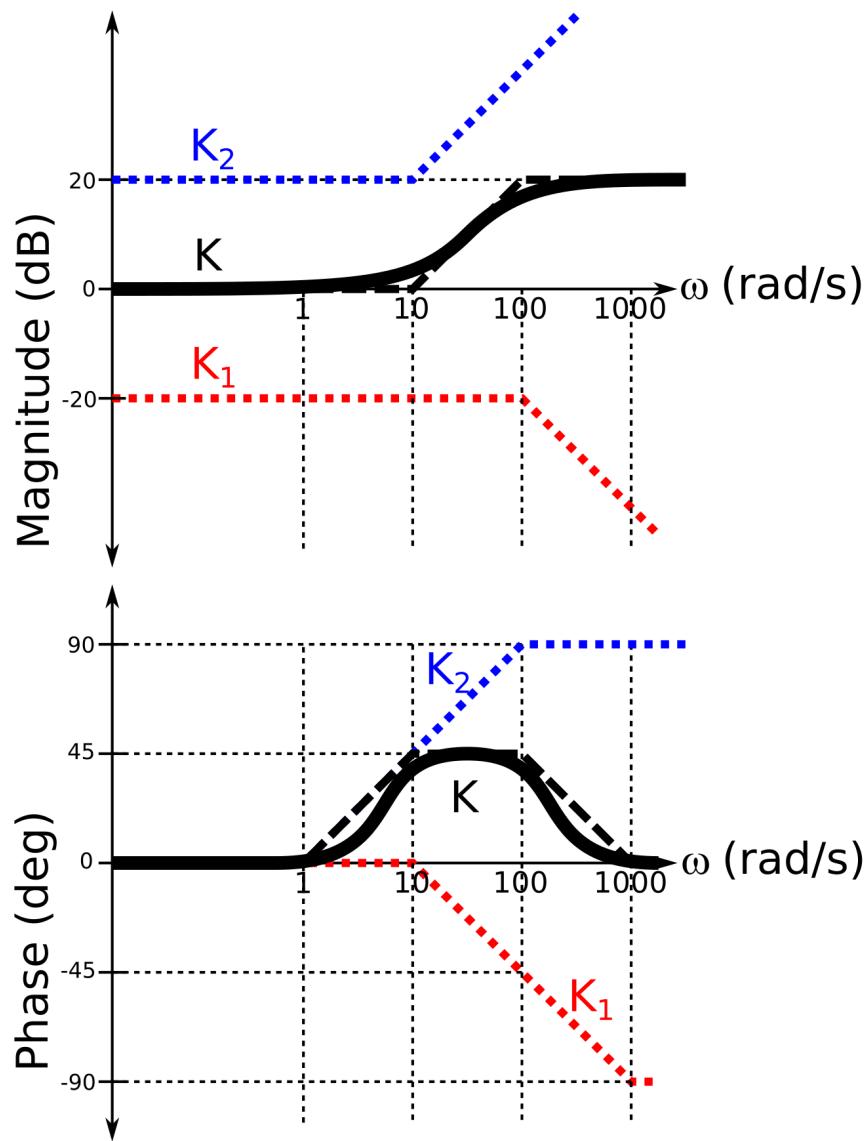
$K_1(j\omega)$  is a first order LTI system, thus the primary characteristics of the Bode plot are:

1. DC gain ( $\omega = 0$ ) of  $20 \log_{10} \frac{1}{10} = -20$  dB
2. corner frequency/bandwidth at  $\omega = 100$
3. high frequency slope of  $-20$  dB per  $10\omega$
4.  $\angle K_1(j\omega)$  will go from  $0^\circ \rightarrow -90^\circ$

$K_2(j\omega)$  is a real zero LTI system, thus the primary characteristics of the Bode plot are:

1. DC gain ( $\omega = 0$ ) of  $20 \log_{10} 10 = 20$  dB
2. corner frequency/bandwidth at  $\omega = 10$
3. high frequency slope of  $+20$  dB per  $10\omega$
4.  $\angle K_2(j\omega)$  will go from  $0^\circ \rightarrow 90^\circ$

Plotting these two together provides



Note that since  $\angle K(s) > 0 \quad \forall \omega$ , the steady-state output sinusoid will “lead” the input sinusoid in its oscillation. Hence, the name **lead controller** is used for this type of control system which will be discussed later in this part of the textbook.

### Example Problem 2

Given: the second order SISO LTI system,  $G$ , with linear ODE

$$\ddot{y}(t) + 1.2\dot{y}(t) + 0.2y(t) = 0.5u(t) \quad (1.272)$$

Determine:

- a) the transfer function,  $G(s)$
- b) a sketch of the Bode plot using the straight line approximation

Solution:

- a) The transfer function is

$$G(s) = \frac{0.5}{s^2 + 1.2s + 0.2} \quad (1.273)$$

which has two poles at  $s = -0.2$  and  $s = -1$  and no zeros.

- b) One can write  $G(s)$  as the product of

$$G(s) = G_1(s)G_2(s) = \left(\frac{1}{s+0.2}\right)\left(\frac{0.5}{s+1}\right) \quad (1.274)$$

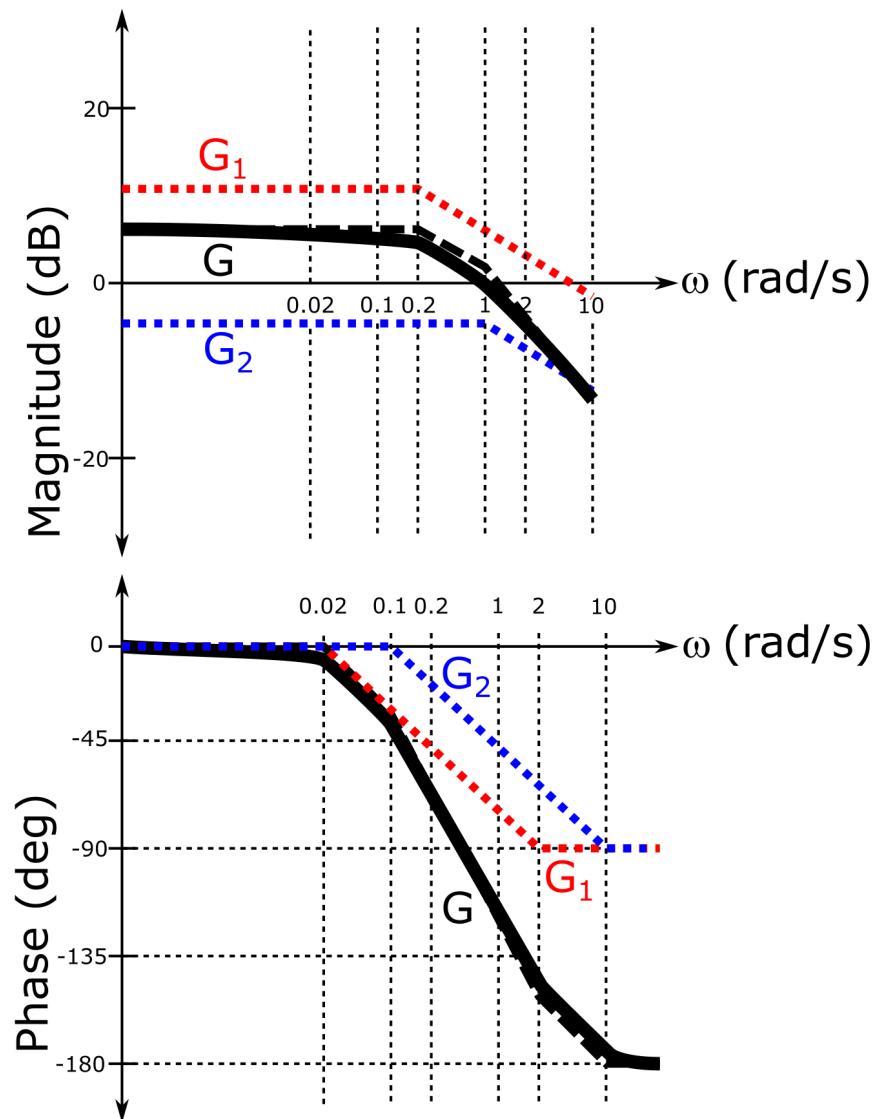
$G_1(j\omega)$  is a first order LTI system, thus the primary characteristics of the Bode plot are:

1. DC gain ( $\omega = 0$ ) of  $20 \log_{10} \frac{1}{0.2} = 20 \log_{10} 5 = 14$  dB
2. corner frequency/bandwidth at  $\omega = 0.2$
3. high frequency slope of  $-20$  dB per  $10\omega$
4.  $\angle G_1(j\omega)$  will go from  $0^\circ \rightarrow -90^\circ$

$G_2(j\omega)$  is a first order LTI system, thus the primary characteristics of the Bode plot are:

1. DC gain ( $\omega = 0$ ) of  $20 \log_{10} 0.5 = -6$  dB
2. corner frequency/bandwidth at  $\omega = 1$
3. high frequency slope of  $-20$  dB per  $10\omega$
4.  $\angle G_2(j\omega)$  will go from  $0^\circ \rightarrow -90^\circ$

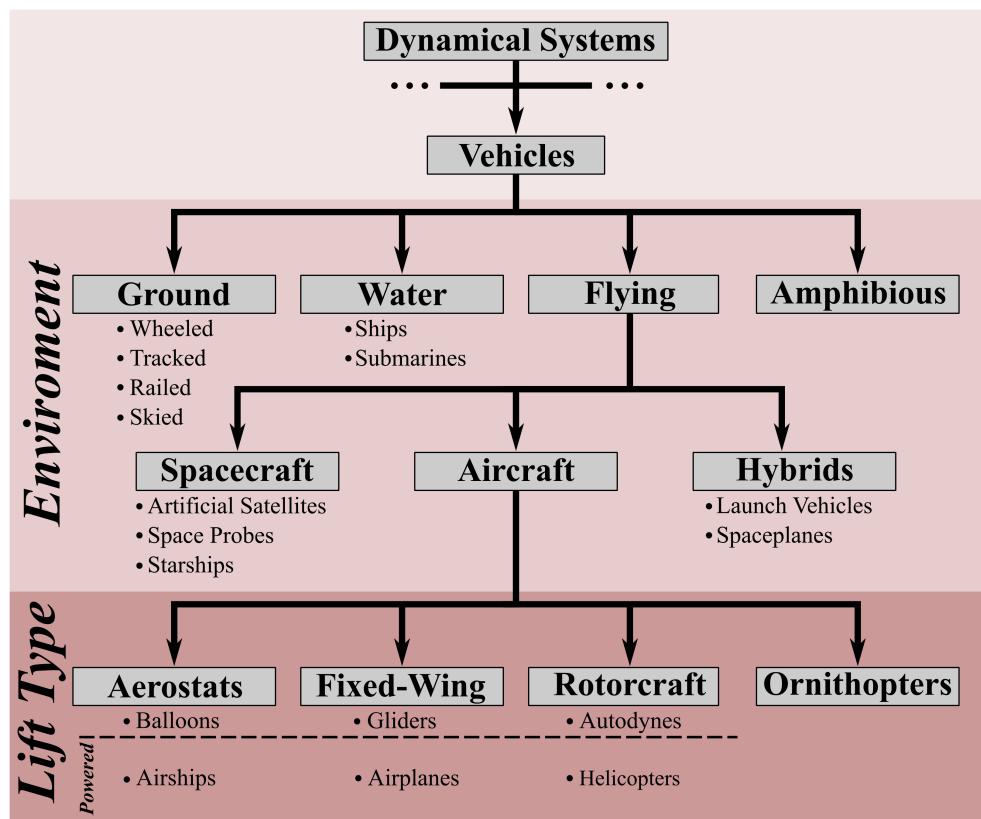
Plotting these two together provides



## Chapter 2

# Introductory Flight Vehicle Dynamics

### 2.1 Introduction to Flight Vehicles



A **vehicle** is defined as a dynamical system that transports a payload, e.g. cargo, people, munitions, sensors. Vehicles are typically classified by the environment in which they operate, i.e. ground, water, flying, or am-

phibious. **Flying vehicles** are a category of vehicle encompassing aircraft, spacecraft, and air/space hybrids. These are also known as **aerospace vehicles**. There are many different types of flying vehicles. These include fixed-wing (gliders, airplanes), rotorcraft (autodynes, helicopters), aerostats (balloons, airships), spacecraft (artificial satellites, space probes, starships), and hybrid flying vehicles that operate in air and space (launch vehicles, spaceplanes). The primary focus of this part of the textbook is *airplanes*, i.e. powered fixed-wing aircraft. However, it is important to note that the dynamics and control concepts in this part of the textbook can be applied to *any* dynamical system to varying degrees as one diverges further away from airplanes.

**Vehicle dynamics and control** is the study of the changes in a vehicle's motion due to the forces and moments applied to and by that vehicle in its environment. The motion of a vehicle is typically described by its position, orientation/attitude, and velocity. When applied in particular to flying vehicles, this study is called **flight dynamics and control (FDC)** and is the subject of this part of the textbook.

The important differentiating factor in FDC from other vehicles is that flying vehicles are affected by only three external forces: propulsion, gravity, and aerodynamics, whereas ground, water, and amphibious vehicles experience ground forces, hydrodynamics, or both, respectively. The **propulsive forces and moments**, also known as the **thrust forces and moments**, are produced by an engine, e.g. propeller, jet, rocket, ion. High fidelity design and modeling of spacecraft and aircraft propulsion systems is beyond the scope of this textbook and will be given in simple functional form using basic dynamical system models. The **gravitational forces and moments**, also known as the **weight**, are typically modeled as according to Newton's law of gravitation. It should be noted that though Earth's gravity field contains variation, this textbook will assume that the gravitational field gradient does not produce any significant moments on a flying vehicle which is often not neglected for spacecraft. The **aerodynamic forces and moments** are due to the pressure distribution around the flying vehicle and can typically be resolved as the aircraft's lift, side, and drag forces as well as three aerodynamic moments. The **lift force** causes an aircraft to overcome its weight and fly. The **side force** is used to steer the aircraft from side-to-side through the air mass. The **drag force** is a motion-hindering force that resists motion through the air mass. The **aerodynamic moments** cause the aircraft to rotate due to the varying pressure distribution over the aircraft body. The aerodynamic forces are the primary factor in aircraft FDC and are often negligible for spacecraft FDC, except at some low planetary orbits.

Flying vehicles are designed so that the **operator** of the flying vehicle, also known as the **pilot**, can affect the aerodynamic and propulsive forces and moments imparted on the flying vehicle in order to control or affect the vehicle's motion, i.e. its dynamics. The control of a flying vehicle can be performed completely by a human, also known as **control!manual**), completely by a computer, also known as **automatic control**, which constitutes an **autopilot**, or partially from both, also known as **semi-automatic control**. In this part of the textbook, linear control theory will be introduced and applied to the control of flying vehicles.

## Reference Frames

Flight vehicles operate either in or outside of the atmospheres of celestial bodies which have their own gravitational motion. Thus, in order to derive the dynamical system models, one uses various coordinate systems centered at different origins, known as **reference frames**. A reference frame  $F$  is uniquely described by its three axes  $x_F - y_F - z_F$  and origin point  $O_F$ . This section discusses relevant reference frames that are used in FDC.

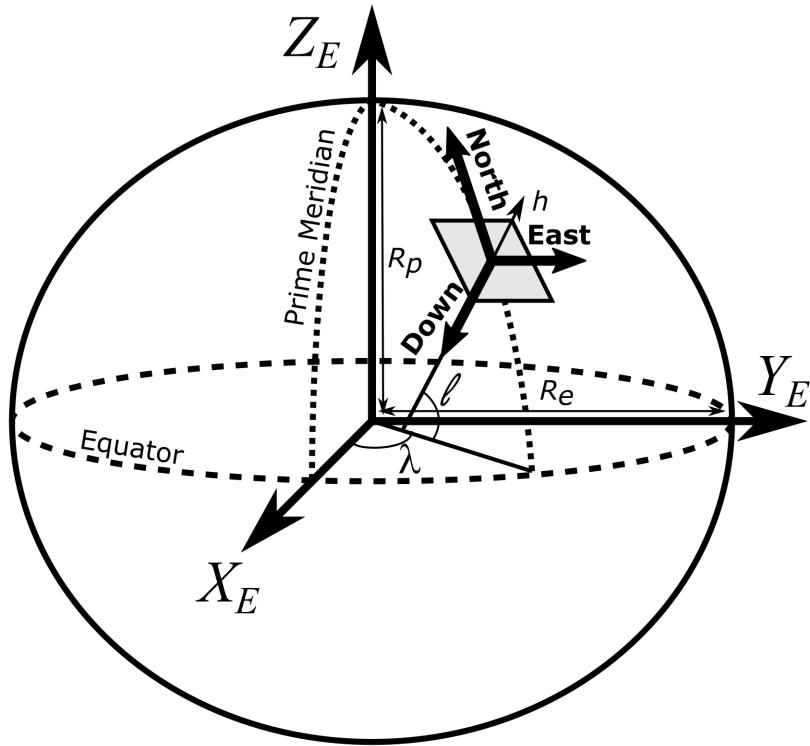
**Earth frames** in FDC can be of two different types. The first is an **Earth-Centered, Earth-Fixed**

**(ECEF)** frame (subscript  $E$ ) which has Cartesian coordinate axes represented as  $x_E - y_E - z_E$  with its origin as the Earth's center of mass. For this frame, the  $x_E$  axis passes through intersection of the prime meridian and the equator which corresponds to  $0^\circ$  longitude and  $0^\circ$  latitude. This is located just south of west Africa. The  $z^E$  axis passes through true north which does not coincide with Earth's instantaneous rotational axis because of the Earth's wobble, but its average. The  $y^E$  axis is orthogonal to both according to right-hand-rule located just south of India along the equator. Because the ECEF frame rotates with the Earth, the coordinates of a point on the surface of the Earth do not change. The second is an **Earth-Centered Inertial (ECI)** frame (subscript  $I$ ) whose origin is still Earth's center of mass but whose axes are defined with respect to the stars. Thus, the Earth's surface rotates over frame, but it's inertial properties make it useful for spacecraft orbiting around Earth and high fidelity dynamic models of aircraft traveling long distances. It should be noted that the ECI orbits the sun and is not typically used for spacecraft traveling beyond the Earth's vicinity in the Solar System. These spacecraft typically use a sun-centered reference frame such as the **International Celestial Reference Frame (ICRF)**.

An important detail when discussing Earth frames is that the surface of the Earth is not perfectly spherical. It is actually not a fully geometrically realizable shape at all, thus the discipline of **geodesy** has developed which is the study of Earth's shape. This science typically uses two different approximations for the Earth's shape, the geoid and the reference ellipsoid. The **geoid** which is an idealized equilibrium surface of the Earth's gravitational potential which varies according to its crust formation and is also known as the **International Terrestrial Reference Frame (ITRF)**.

The **reference ellipsoid** approximates the geoid and has an equatorial radius,  $R_e$ , longer than its polar radius,  $R_p$ , a shape also known as an **oblate spheroid**. A common model for the reference ellipsoid is the **World Geodetic System (WGS)**. The traditional coordinates used for the reference ellipsoid are **latitude, longitude  $\lambda$ , and altitude,  $h$**  which are also called **geodetic coordinates**. Zero altitude is referred to as **mean sea level (MSL)** and is defined as the ideal continuous surface of the ocean in the absence of currents and air pressure variations and whose surface continues under the continental masses. Geodetic coordinates are ellipsoidal coordinates. Thus, because these are ellipsoidal as opposed to spherical, the latitude angle does not define the angle between the Earth's center and a point on the surface and altitude does not always align with the Earth's center, but both have an analytical offset.

Another frame in FDC is the frame defining an aircraft's immediate motion relative to the Earth's surface, i.e. the **local tangent plane (LTP)** frame defined as tangent to the reference ellipsoid. Thus, the origin of the frame is the tangent point with the ECEF at a specified latitude,  $\ell$ , longitude,  $\lambda$ , and altitude,  $h$ , and the three Cartesian axes are in the east-west direction, i.e. tangent to the latitude parallels, the north-south direction, i.e. tangent to the longitudinal meridians, and the up-down direction, i.e. normal to the reference ellipsoid). Two common right-handed LTPs for the LTP axes are the East-North-Up (ENU) and the **North-East-Down (NED)** frames. The relationship between the ECEF frame, LLA, and the NED LTP frame can be visualized through the following figure.



Closely related to the LTP frame, the **navigation frame** (subscript  $N$ ) is defined with an origin as the flying vehicle's center of gravity  $G$  and the coordinate axes aligned with instantaneous LTP. This frame is ideal for aircraft navigation as it uses though aircraft dynamics typically use the NED LTP frame since most objects of interest are below aircraft. It is important to note that the LTP frame is not an inertial frame and thus for large accelerations or large deviations from the origin, an aircraft's equations of motion must be adjusted to handle this.

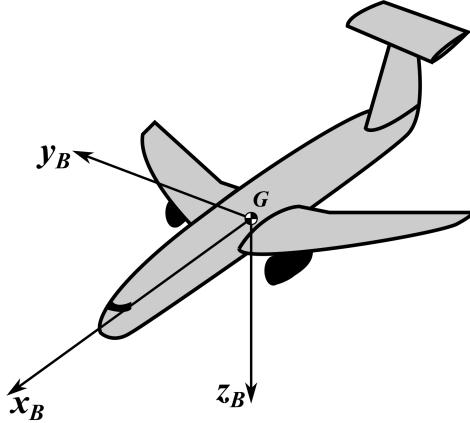
However, this part of the textbook will use the **flat Earth approximation** which assumes that the entire Earth's surface can be approximated by a single coordinate frame which can be considered as the “average” LTP in the area of operation. Furthermore, in this case, the navigation frame coordinate axes remain static, i.e. “inertial,” which allows one to more easily compute the equations of motion for aircraft. This approximation is typically “good enough” for aircraft dynamics and control analysis and design. Although with hypersonic aircraft velocities and/or long distance flight analysis, the flat Earth approximation may not be suitable even for this case. The additional effects of non-flat Earth modeling on aircraft equations of motion is presented in another part of this textbook.

The **body frame** (subscript  $B$ ) is attached to the flying vehicle's body structure, thus it is ideal for geometric configuration and structural modeling. This frame is defined as follows

1. The origin is the flying vehicle's center of gravity  $G$ .
2. The  $x_B$  axis points out the front of the flying vehicle, typically “along” the nominal path of travel, also known as the **longitudinal axis**.
3. The  $y_B$  axis points out the right side of the flying vehicle, also known as the **lateral axis**.

4. The  $z_B$  axis points straight beneath the flying vehicle, also known as the **vertical axis**.

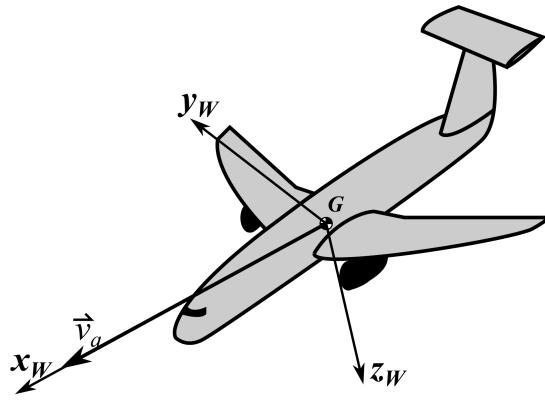
This frame is depicted for an airplane in the following figure.



The **wind frame** (subscript  $W$ ) relates the **free-stream airflow** that an aircraft encounters as it flies, thus it is ideal for aerodynamics modeling and is particular to aircraft dynamics and control. The frame is defined as

1. The origin is the aircraft's center of gravity  $G$ .
2. The  $x_W$  axis is colinear with free-stream airflow  $\vec{v}_\infty$ .
3. The  $z_W$  axis is in the plane of symmetry of the aircraft, positive below the aircraft.
4. The  $y_W$  axis is perpendicular to both.

This frame is depicted for an airplane in the following figure.

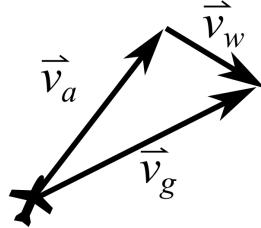


Here the aircraft velocity  $\vec{v}_\infty$  is the velocity relative to the air mass that the aircraft is traveling through, i.e. the free-stream **airspeed** vector. This is different than the velocity of the aircraft with respect to the

planet's surface, i.e. the **ground speed** vector  $\vec{v}_g$ . These velocity vectors are related through the **wind speed** vector as

$$\vec{v}_g = \vec{v}_\infty + \vec{v}_w \quad (2.1)$$

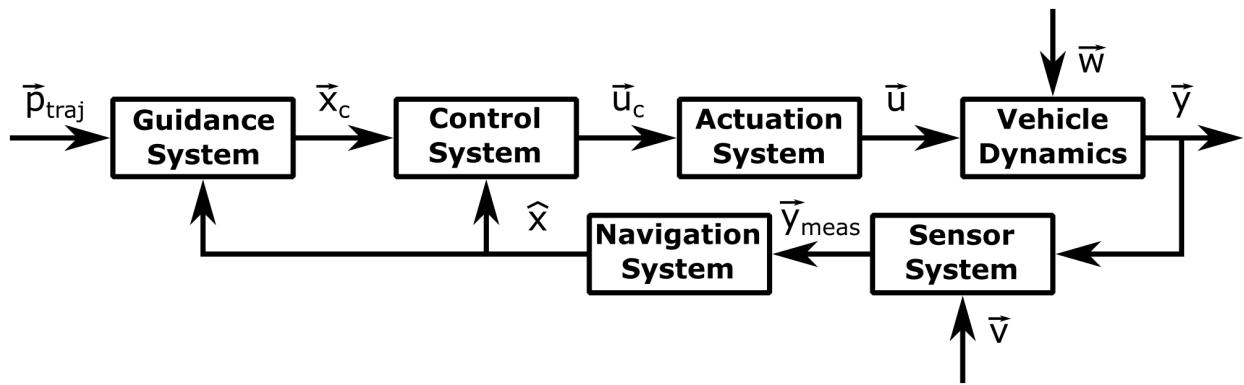
an equation also known as the **wind triangle** as this vector relationship forms a triangle, i.e.



Similar to the flat Earth approximation, this part of the textbook will use the **no wind approximation** to introduce aircraft dynamics. Although with wind speeds significantly close to the nominal aircraft airspeed and/or long distance flight analysis, the no wind approximation may not be suitable. The additional effects of both steady and unsteady wind on the aircraft dynamics is presented in another part of this textbook.

### Guidance, Navigation, and Control

The study of vehicle control systems is also known as **guidance, navigation and control (GNC)** which refers to the design and analysis of the three traditional sub-systems used to control the movement of vehicles. This part of the textbook will touch on aspects of GNC system design with an introduction to guidance and control for flight vehicles. Later parts will discuss flight vehicle navigation and advanced guidance and control. However, this section will introduce the high level concepts associated with GNC systems. The block diagram of a general GNC system is shown as follows.



where the vehicle state vector,  $\vec{x}$ , traditionally includes the position, velocity, and attitude of the vehicle. It should be noted that for manual control of flight vehicles, the guidance and control systems can be considered as the human operator and not a computer system.

Given the current vehicle state vector, the **guidance system** determines the commanded vehicle state vector,  $\vec{x}_c$ , for following some reference trajectory,  $\vec{p}_{traj}$ , e.g. determines a commanded velocity, attitude, and acceleration. Notably, this system is often simply controlled by a human operator and does not use automatic control algorithms. However, as the input to the operator is the available human sensory information and/or an information display from sensors, as it would be for electronic information from the navigation system, the operator is still using the basic logic of GNC reasoning.

The **control system** achieves the commanded state,  $\vec{x}_c$ , while also maintaining vehicle dynamic stability and typically uses either automatic or semi-automatic control. In this way, the vehicle's state,  $\vec{x}$ , evolves in time according to its natural dynamics, the control inputs,  $\vec{u}$ , and any disturbances,  $\vec{w}$ , from the environment.

Furthermore, as the control system include the direct manipulation of the forces applied to the vehicle through some sort of physical phenomena, one should also consider the effects of any **actuation system** on the GNC system. Using dynamical systems theory, one can typically model actuation systems for flight vehicles,  $A(s)$ , using first or second order transfer functions, e.g.

$$A(s) = \frac{\omega_a}{s + \omega_a} \quad (2.2)$$

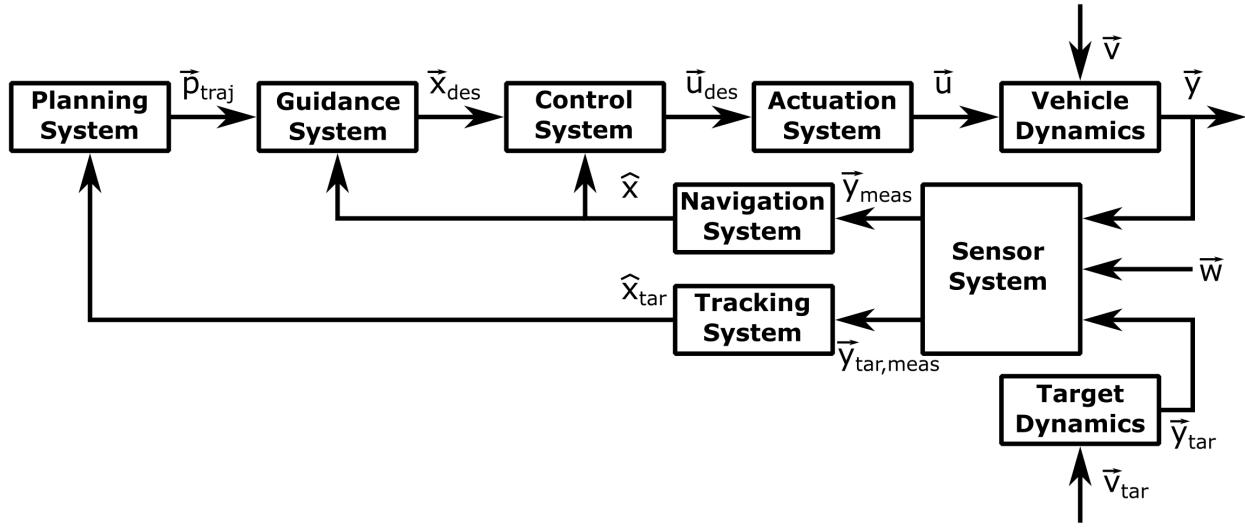
or

$$A(s) = \frac{\omega_a^2}{s^2 + 2\zeta_a\omega_a s + \omega_a^2} \quad (2.3)$$

where  $\omega_a$  is the bandwidth of the actuator and  $\zeta_a$  is the damping. Actuators also typically have hard limits on the minimum and maximum output as well as hard rate limits. Coupled with the vehicle dynamics, these actuator dynamics play a vital role in the control system design for vehicles.

At the base of guidance, navigation, and control systems is the necessity of the pilot or autopilot to use a **sensor!system** to sense the vehicle's state vector. In addition, a corresponding **navigation!system** is typically necessary to process, i.e. “filter,” the raw signals obtained by the sensor system. Furthermore, when only certain aspects of the vehicle dynamics, i.e. the outputs  $\vec{y}$ , are measurable by the sensor system, the navigation system estimates the state vector,  $\hat{x}$ , that is required for the guidance and control systems to operate. Thus, navigation can be defined as **vehicle state estimation**.

Using the most general definition of a “plan” or task for a vehicle, a vehicle must travel from a starting location to one (or more) designated target(s). As such, one can also consider the operation of flight vehicles at a level higher than GNC, namely that of **path planning**, also known as **trajectory planning**, which is performed by a **planning system** that determines the reference path or trajectory,  $\vec{p}_{traj}$ , that the vehicle should follow from its starting location to the current designated target. This stage is often completely manually derived and pre-programmed without real-time signal inputs. Furthermore, if one must reach a dynamic target, than an additional **target tracking system** is necessary to accomplish the plan. If more than one target is specified, then the planning system must have some feedback from the navigation or target tracking system to assess when to switch from one target to another. The block diagram of a general planning, tracking and GNC system is shown as follows.



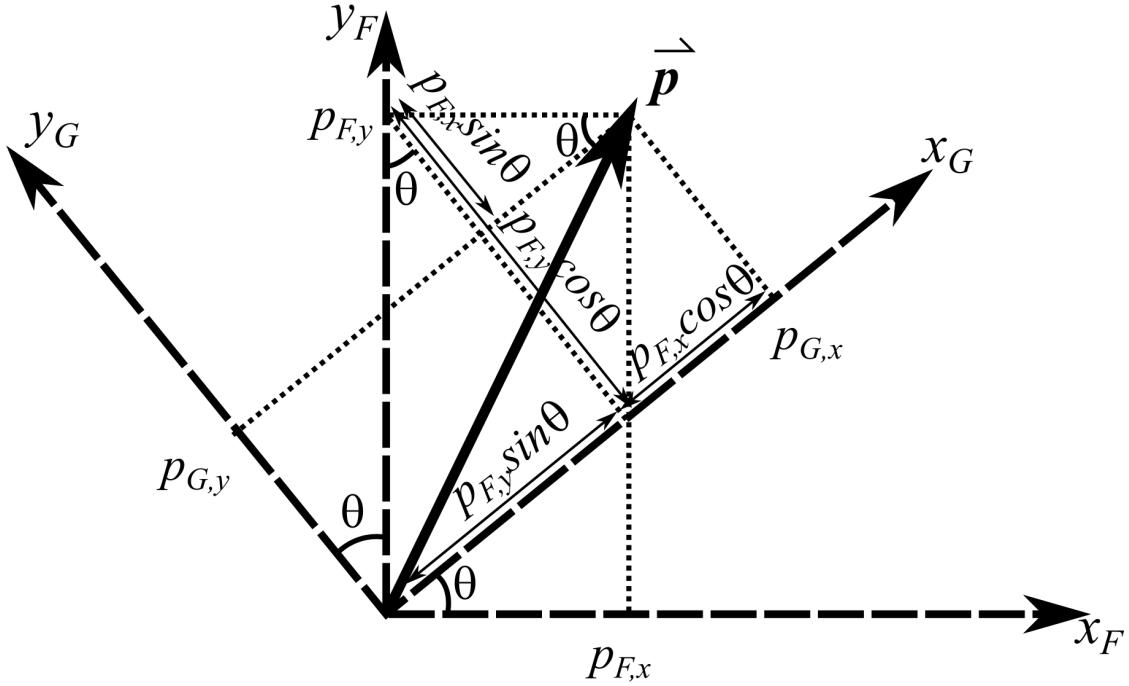
While this part of the textbook will introduce guidance and control systems for flight vehicles, later parts of this textbook discuss the design and analysis of planning, navigation, and target tracking systems for flight vehicles.

## 2.2 Flight Vehicle Reference Frame Rotations

When discussing flight vehicle dynamics, one must be able to transform the vehicle state between different reference frames. This section will discuss vector transformations for two and three dimensional vectors referenced to the same origin point, also known as **reference frame rotations**.

### Two-Dimensional Rotations

Consider the vector  $\vec{p}$  in frames  $F$  and  $G$ , expressed as  $\vec{p}_F = [p_{F,x} \ p_{F,y}]^T$  and  $\vec{p}_G = [p_{G,x} \ p_{G,y}]^T$  with the same origin.



By trigonometry,

$$\vec{p}_G = \begin{bmatrix} p_{G,x} \\ p_{G,y} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} p_{F,x} \\ p_{F,y} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \vec{p}_F \quad (2.4)$$

or

$$\vec{p}_G = C_{G \leftarrow F}(\theta) \vec{p}_F \quad (2.5)$$

where  $C_{G \leftarrow F}(\theta)$  is the **rotation matrix** from frame  $F$  to  $G$  by the angle  $\theta$ .

To transform back, one can form the inverse rotation matrix by using the negative angle, i.e.

$$\vec{p}_F = \begin{bmatrix} \cos(-\theta) & \sin(-\theta) \\ -\sin(-\theta) & \cos(-\theta) \end{bmatrix} \vec{p}_G = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \vec{p}_G \quad (2.6)$$

or

$$\vec{p}_F = C_{F \leftarrow G}(\theta) \vec{p}_G \quad (2.7)$$

which by inspection can also be seen to be

$$\vec{p}_F = C_{G \leftarrow F}^T(\theta) \vec{p}_G \quad (2.8)$$

which implies the property

$$C_{G \leftarrow F}^{-1}(\theta) = C_{G \leftarrow F}^T(\theta) \quad (2.9)$$

for rotation matrices.

Next, consider if the vector and the rotation angle are changing in time, then one can represent the time derivative of the vector coordinates in the rotating frame  $G$ ,  $\frac{d}{dt} \vec{p}_G(t)$ , also known as the velocity, as represented in the rotating frame, then one must calculate the following

$$\frac{d}{dt} \vec{p}_G(t) = \frac{d}{dt} \left( \begin{bmatrix} \cos \theta(t) & \sin \theta(t) \\ -\sin \theta(t) & \cos \theta(t) \end{bmatrix} \vec{p}_F(t) \right) \quad (2.10)$$

which by the product rule provides

$$\frac{d}{dt} \vec{p}_G(t) = \begin{bmatrix} \cos \theta(t) & \sin \theta(t) \\ -\sin \theta(t) & \cos \theta(t) \end{bmatrix} \dot{\vec{p}}_F(t) + \begin{bmatrix} -\sin \theta(t) & \cos \theta(t) \\ -\cos \theta(t) & -\sin \theta(t) \end{bmatrix} \dot{\theta}(t) \vec{p}_F(t) \quad (2.11)$$

where  $\dot{\vec{p}}$  is simply the partial time derivative of the vector itself, i.e.

$$\dot{\vec{p}}_F(t) = \frac{\partial}{\partial t} \vec{p}_F(t) \quad (2.12)$$

and not the rotating frame.

By rearranging the second term, one can write

$$\frac{d}{dt} \vec{p}_G(t) = \begin{bmatrix} \cos \theta(t) & \sin \theta(t) \\ -\sin \theta(t) & \cos \theta(t) \end{bmatrix} \dot{\vec{p}}_F(t) + \dot{\theta}(t) \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \cos \theta(t) & \sin \theta(t) \\ -\sin \theta(t) & \cos \theta(t) \end{bmatrix} \vec{p}_F(t) \quad (2.13)$$

Finally, by defining  $\omega(t) = \dot{\theta}(t)$  as the **angular velocity** and the definition of the rotation matrix, one has

$$\frac{d}{dt} \vec{p}_G(t) = \dot{\vec{p}}_G(t) + \begin{bmatrix} 0 & -\omega(t) \\ \omega(t) & 0 \end{bmatrix} \vec{p}_G(t) \quad (2.14)$$

Similarly, if one also wishes to compute the second time derivative of the vector coordinates in the rotating frame  $G$ ,  $\frac{d^2}{dt^2} \vec{p}_G(t)$ , also known as the **acceleration**, as represented in the rotating frame, then another instance of the product rule for both terms can be employed to obtain

$$\frac{d^2}{dt^2} \vec{p}_G(t) = \frac{d}{dt} \left( \dot{\vec{p}}_G(t) \right) + \frac{d}{dt} \left( \begin{bmatrix} 0 & -\omega(t) \\ \omega(t) & 0 \end{bmatrix} \vec{p}_G(t) \right) \quad (2.15)$$

$$\frac{d^2}{dt^2} \vec{p}_G(t) = \ddot{\vec{p}}_G(t) + \begin{bmatrix} 0 & -\omega(t) \\ \omega(t) & 0 \end{bmatrix} \dot{\vec{p}}_G(t) + \begin{bmatrix} 0 & -\dot{\omega}(t) \\ \dot{\omega}(t) & 0 \end{bmatrix} \vec{p}_G(t) + \begin{bmatrix} 0 & -\omega(t) \\ \omega(t) & 0 \end{bmatrix} \frac{d}{dt} \vec{p}_G(t) \quad (2.16)$$

and using the previous formula for  $\frac{d}{dt} \vec{p}_G(t)$  and the substitution  $\alpha = \dot{\omega}$  as the **angular acceleration**, one has

$$\begin{aligned} \frac{d^2}{dt^2} \vec{p}_G(t) = & \ddot{\vec{p}}_G(t) + 2 \begin{bmatrix} 0 & -\omega(t) \\ \omega(t) & 0 \end{bmatrix} \dot{\vec{p}}_G(t) + \begin{bmatrix} 0 & -\alpha(t) \\ \alpha(t) & 0 \end{bmatrix} \vec{p}_G(t) \\ & + \begin{bmatrix} 0 & -\omega(t) \\ \omega(t) & 0 \end{bmatrix} \begin{bmatrix} 0 & -\omega(t) \\ \omega(t) & 0 \end{bmatrix} \vec{p}_G(t) \end{aligned} \quad (2.17)$$

where

$$2 \begin{bmatrix} 0 & -\omega(t) \\ \omega(t) & 0 \end{bmatrix} \dot{\vec{p}}_G(t) \quad (2.18)$$

is the **Coriolis acceleration**,

$$\begin{bmatrix} 0 & -\alpha(t) \\ \alpha(t) & 0 \end{bmatrix} \vec{p}_G(t) \quad (2.19)$$

is the **Euler acceleration**, and

$$\begin{bmatrix} 0 & -\omega(t) \\ \omega(t) & 0 \end{bmatrix} \begin{bmatrix} 0 & -\omega(t) \\ \omega(t) & 0 \end{bmatrix} \vec{p}_G(t) \quad (2.20)$$

is the **centrifugal acceleration**. Collectively, these are known as **fictitious accelerations** and produce **fictitious forces**.

### Three-Dimensional Rotations

In two dimensional rotations, the rotation axis can only be perpendicular to the plane. However, for three dimensional rotations, one must not only specify the rotation angle, but also the axis about which the angular rotation is performed. However, of special note are the **basic rotation matrices** defined as

$$C_1(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & \sin \theta_x \\ 0 & -\sin \theta_x & \cos \theta_x \end{bmatrix} \quad (2.21)$$

$$C_2(\theta_y) = \begin{bmatrix} \cos \theta_y & 0 & -\sin \theta_y \\ 0 & 1 & 0 \\ \sin \theta_y & 0 & \cos \theta_y \end{bmatrix} \quad (2.22)$$

$$C_3(\theta_z) = \begin{bmatrix} \cos \theta_z & \sin \theta_z & 0 \\ -\sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.23)$$

which make use of rotations about each primary coordinate axis. It can be shown that every possible rotation in three dimensions, i.e. the rotation angle and axis, can be represented by, at most, three rotations. Thus, all possible rotation matrices can be obtained from, at most, three sequential combinations of the basic rotation matrices. The three sequential angles used to form these rotation matrices are known as **Euler angles**. In FDC, one typically defines the Euler angles for a 3 – 2 – 1 rotation sequence which corresponds to the equation

$$\vec{v}_G = C_1(\theta_x)C_2(\theta_y)C_3(\theta_z)\vec{v}_F \quad (2.24)$$

where  $G$  is the frame rotating with respect to  $F$ . Here  $\theta_x$ ,  $\theta_y$ , and  $\theta_z$  are the 3 – 2 – 1 Euler angles.

As an alternative to describing a rotation matrix using the sequential Euler angles, one can also specify a single rotation matrix known as the **direction cosine matrix (DCM)**,  $C_{G \leftarrow F}$ , i.e.

$$\vec{v}_G = C_{G \leftarrow F}\vec{v}_F \quad (2.25)$$

which is often denoted component-wise as

$$C_{G \leftarrow F} = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \quad (2.26)$$

For the  $3 - 2 - 1$  Euler angle designation, one can show that the DCM and Euler angle rotations are related explicitly by the formulas

$$C_{G \leftarrow F} = C_1(\theta_x)C_2(\theta_y)C_3(\theta_z) \quad (2.27)$$

$$C_{G \leftarrow F} = \begin{bmatrix} \cos \theta_y \cos \theta_z & \cos \theta_y \sin \theta_z & -\sin \theta_y \\ \sin \theta_x \sin \theta_y \cos \theta_z - \cos \theta_x \sin \theta_z & \sin \theta_x \sin \theta_y \sin \theta_z + \cos \theta_x \cos \theta_z & \sin \theta_x \cos \theta_y \\ \cos \theta_x \sin \theta_y \cos \theta_z + \sin \theta_x \sin \theta_z & \cos \theta_x \sin \theta_y \sin \theta_z - \sin \theta_x \cos \theta_z & \cos \theta_x \cos \theta_y \end{bmatrix} \quad (2.28)$$

and

$$\theta_x = \arctan \frac{C_{23}}{C_{33}} \quad (2.29)$$

$$\theta_y = -\arcsin C_{13} \quad (2.30)$$

$$\theta_z = \arctan \frac{C_{12}}{C_{11}} \quad (2.31)$$

It should be noted that if  $\theta_y = \pm 90^\circ$ , then the Euler angle transformation above is undefined, also known as the **Euler angle ambiguity** that must be dealt with in some flight vehicle applications. Thus, using the DCM directly can be safer than using Euler angles, but requires six separate quantities to describe. However, **quaternions** are another alternative for representing rotations that is defined by four quantities and overcomes the Euler angle ambiguity with the least amount of quantities necessary. Quaternions are considered in a later part of this textbook.

### Navigation-to-Wind Euler Angles

The Euler angles for describing the wind frame relative to the navigation frame are the **bank angle**,  $\mu$ , **flight path angle**,  $\gamma$ , and **heading angle**,  $\sigma$ . Thus, a vector expressed in navigation frame coordinates,  $\vec{v}_N$ , can be expressed as a vector in wind frame coordinates,  $\vec{v}_W$ , through the sequence

$$\vec{v}_W = C_1(\mu)C_2(\gamma)C_3(\sigma)\vec{v}_N \quad (2.32)$$

$$\vec{v}_W = C_{W \leftarrow N}\vec{v}_N \quad (2.33)$$

where

$$C_{W \leftarrow N} = \begin{bmatrix} \cos \gamma \cos \sigma & \cos \gamma \sin \sigma & -\sin \gamma \\ \sin \mu \sin \gamma \cos \sigma - \cos \mu \sin \sigma & \sin \mu \sin \gamma \sin \sigma + \cos \mu \cos \sigma & \sin \mu \cos \gamma \\ \cos \mu \sin \gamma \cos \sigma + \sin \mu \sin \sigma & \cos \mu \sin \gamma \sin \sigma - \sin \mu \cos \sigma & \cos \mu \cos \gamma \end{bmatrix} \quad (2.34)$$

It should be noted that the heading angle can be arbitrarily set as subsequent rotations in heading can be performed before the other navigation-to-body frame Euler angles, but is typically referenced to north.

### Navigation-to-Body Euler Angles

The Euler angles for describing the body frame relative to the navigation frame are the **roll angle**,  $\phi$ , **pitch angle**,  $\theta$ , and **yaw angle**,  $\psi$ . Thus, a vector expressed in navigation frame coordinates,  $\vec{v}_N$ , can be expressed as a vector in body frame coordinates,  $\vec{v}_B$ , through the sequence

$$\vec{v}_B = C_1(\phi)C_2(\theta)C_3(\psi)\vec{v}_N \quad (2.35)$$

$$\vec{v}_B = C_{B \leftarrow N} \vec{v}_N \quad (2.36)$$

where

$$C_{B \leftarrow N} = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix} \quad (2.37)$$

These Euler angles are used to represent the orientation or **attitude** of the aircraft as they provide a description of how the rigid body of the aircraft is currently oriented in 3D space. It should be noted that the yaw angle can be arbitrarily set as subsequent rotations in yaw can be performed before the other angles.

### Wind-to-Body Euler Angles

The Euler angles for describing the body frame relative to the wind frame are the **angle of attack**,  $\alpha$ , and the **sideslip angle**,  $\beta$ . Thus, a vector expressed in wind frame coordinates,  $\vec{v}_W$ , can be expressed as a vector in body frame coordinates,  $\vec{v}_B$ , through the sequence

$$\vec{v}_B = C_2(\alpha)C_3(-\beta)\vec{v}_W \quad (2.38)$$

$$\vec{v}_B = C_{B \leftarrow W} \vec{v}_W \quad (2.39)$$

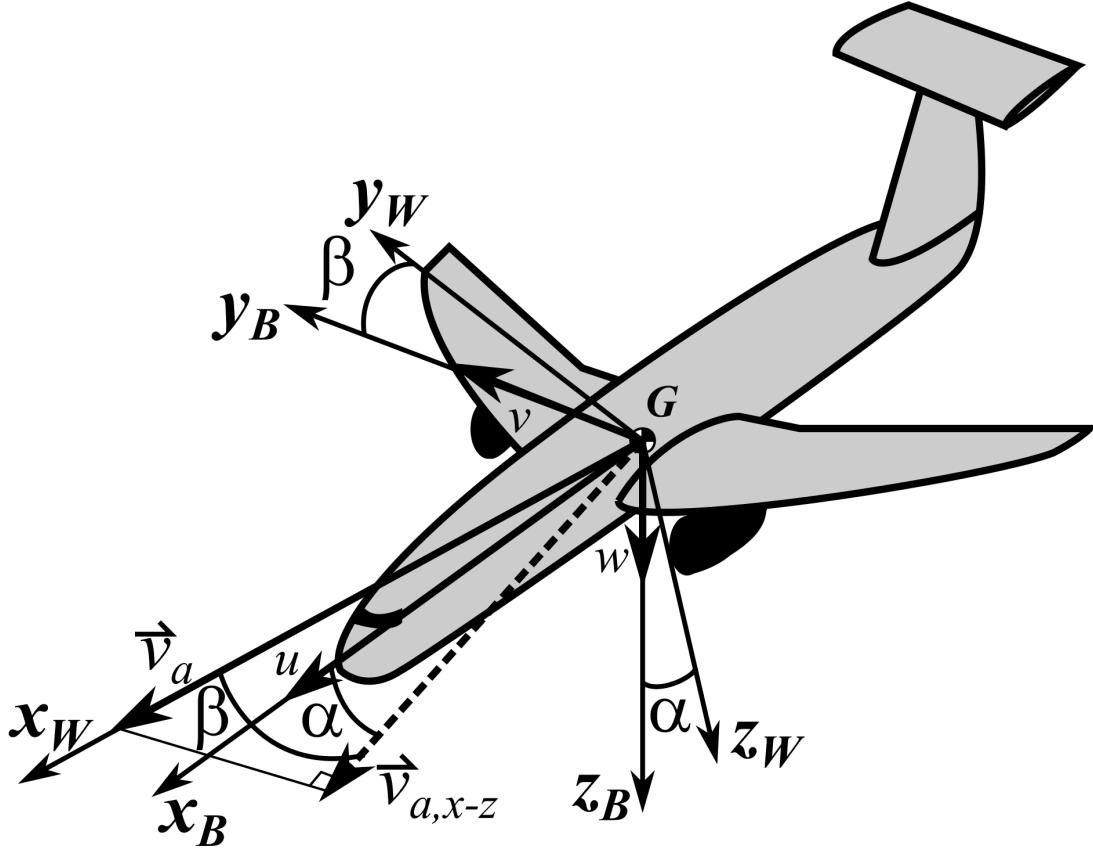
where

$$\vec{v}_B = \begin{bmatrix} \cos \alpha \cos \beta & -\cos \alpha \sin \beta & -\sin \alpha \\ \sin \beta & \cos \beta & 0 \\ \sin \alpha \cos \beta & -\sin \alpha \sin \beta & \cos \alpha \end{bmatrix} \vec{v}_W \quad (2.40)$$

Of particular importance for this rotation is the representation of the airspeed vector  $\vec{v}_a$  with magnitude  $v_a$  expressed in the body frame as the components

$$\vec{v}_{a,B} = [u \ v \ w]^T \quad (2.41)$$

which can be visualized in the following diagram



From the previous equations, one has

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \cos \alpha \cos \beta & -\cos \alpha \sin \beta & -\sin \alpha \\ \sin \beta & \cos \beta & 0 \\ \sin \alpha \cos \beta & -\sin \alpha \sin \beta & \cos \alpha \end{bmatrix} \begin{bmatrix} v_a \\ 0 \\ 0 \end{bmatrix} \quad (2.42)$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} v_a \cos \alpha \cos \beta \\ v_a \sin \beta \\ v_a \sin \alpha \cos \beta \end{bmatrix} \quad (2.43)$$

Dividing the first row by the third row, one has

$$\frac{u}{w} = \frac{v_a \cos \alpha \cos \beta}{v_a \sin \alpha \cos \beta} \quad (2.44)$$

$$\frac{u}{w} = \frac{\cos \alpha}{\sin \alpha} \quad (2.45)$$

$$\frac{u}{w} = \frac{1}{\tan \alpha} \quad (2.46)$$

which provides the relationship between the angle of attack and the components of the airspeed vector as

$$\alpha = \tan^{-1} \frac{w}{u} \quad (2.47)$$

While isolating the second row, one has

$$v = v_a \sin \beta \quad (2.48)$$

which provides the relationship between the sideslip angle and the components of the airspeed vector as

$$\beta = \sin^{-1} \frac{v}{v_a} \quad (2.49)$$

### Approximate Relationship between Euler Angles

Moreover, it should be pointed out that there is an implicit relationship between these sets of Euler angles, e.g.

$$C_{W \leftarrow N} = C_{W \leftarrow B} C_{B \leftarrow N} \quad (2.50)$$

which, in general, results in complicated trigonometric equations.

However, for linearized flight dynamics, it is useful to form the following approximate relationships between the Euler angles using the small angle approximation. For Equation 2.50, this becomes

$$\begin{bmatrix} 1 & \sigma & -\gamma \\ -\sigma & 1 & \mu \\ 0 & 0 & 1 \end{bmatrix} \approx \begin{bmatrix} 1 & \beta & \alpha \\ -\beta & 1 & 0 \\ -\alpha & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \psi & -\theta \\ -\psi & 1 & \phi \\ 0 & 0 & 1 \end{bmatrix} \quad (2.51)$$

By computing the element in the second row and third column, one has

$$\mu \approx -\theta \beta + \phi \quad (2.52)$$

and discarding higher order terms

$$\mu \approx \phi \quad (2.53)$$

By computing the element in the first row and third column

$$-\gamma \approx -\theta + \beta \phi + \alpha \quad (2.54)$$

and discarding higher order terms

$$\gamma \approx \theta - \alpha \quad (2.55)$$

By computing the element in the first row and second column, one has

$$\sigma \approx \psi + \beta \quad (2.56)$$

It should be noted that small  $\sigma$  and  $\psi$  here correspond to small *changes* in the *nominal* heading and yaw as these both can be set to an arbitrary reference direction in the rotation sequence.

## 2.3 Rigid Flight Vehicle Dynamics

A **rigid body** is a solid object which does not deform, i.e. the distance between any two points remain constant regardless of the external forces exerted on it. Thus a rigid body can be considered a continuous distribution of mass in three-dimensional space. The rigid body approximation for flight vehicle dynamics is “good enough” for the majority of flight vehicles, as their structures are designed so that the actual deformations can often be neglected in the dynamics and control analysis and design. However, modern aircraft often incorporate elastic, i.e. flexible, structures which complicate the flight vehicle dynamics modeling. This elastic body modeling is considered in later part of this textbook.

General rigid body dynamics are represented using the **Newton-Euler equations of motion** which relate the forces and moments acting on the rigid body to the linear and angular momentum, respectively, i.e.

$$\begin{aligned}\vec{F} &= \frac{d}{dt} \vec{p} \\ \vec{M} &= \frac{d}{dt} \vec{H}\end{aligned}\tag{2.57}$$

where  $\vec{F}$  is the total force on the rigid body,  $\vec{p}$  is the linear momentum,  $\vec{M}$  is the total moment on the rigid body, and  $\vec{H}$  is the angular momentum. Since a rigid body can be represented as a continuous mass distribution, these quantities can be computed as integrals over the volume  $V$  by the following equations

$$\begin{aligned}\int_V \vec{f} \rho dV &= \frac{d}{dt} \int_V \vec{v} \rho dV \\ \int_V \vec{r} \times \vec{f} \rho dV &= \frac{d}{dt} \int_V \vec{r} \times \vec{v} \rho dV\end{aligned}\tag{2.58}$$

where  $\vec{f}$  is the forces acting on the rigid body per unit mass,  $dm = \rho dV$  is an infinitesimal mass element of the body with density  $\rho$ ,  $\vec{v}$  is the velocity of that mass element, and  $\vec{r}$  is the position vector of the mass element with respect to the origin of an inertial reference frame. It should be noted that the first equation of the Newton-Euler EOMs is called the **translation equation** while the second is called the **rotation equation**.

### Rigid Body Dynamics

The following rigid body dynamics derivation will assume that the total mass  $m$  is constant over time for the rigid body. This aspect will be addressed in a later part of this textbook. However, for the majority of flight vehicles, engine fuel consumption will cause mass to change. This change may be slow in the case of aircraft engines and is often neglected, but for rocket engines the mass rate of change is crucial in deriving appropriate EOMs.

For rigid bodies, if one defines the coordinate frame origin at the center of mass  $G$  and the coordinates axes as attached to the rigid body, i.e. the body frame, then the Newton-Euler EOMs become

$$\begin{aligned}\vec{F}_B &= \frac{d}{dt} (m \vec{v}) = m (\dot{\vec{v}}_B + \vec{\omega}_B \times \vec{v}_B) \\ \vec{M}_B &= \frac{d}{dt} (I_G \vec{\omega}) = I_G \dot{\vec{\omega}}_B + \vec{\omega}_B \times I_G \vec{\omega}_B\end{aligned}\tag{2.59}$$

where  $\vec{F}_B$  is the total force acting on  $G$  in the body frame,  $\vec{v}_B$  is the velocity of  $G$ ,  $\vec{M}_B$  is the total moment acting about  $G$ ,  $I_G$  is the inertia matrix, and  $\vec{\omega}_B$  is the **angular velocity of the body frame**. Note that the additional cross product terms must be included due to the body frame being non-inertial, i.e. its coordinate axes are rotating and origin is accelerating. These cross-products can also be rewritten using the **skew-symmetric matrix operation** defined as

$$\vec{a} \times \vec{b} = [\vec{a}]^{\times} \vec{b} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}^{\times} \vec{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \vec{b} \quad (2.60)$$

where the upper left  $2 \times 2$  matrix was used explicitly in the two-dimensional rotation derivative derived previously which implicitly used the  $z$  axis for its angular velocity. The three-dimensional rotation derivative is simply the extension of this concept for an arbitrary vector in three-dimensional space.

The **inertia matrix**, also known as the **moment of inertia tensor**, is composed of the **moments of inertia** and the **products of inertia** about the body frame coordinates axes  $x_B - y_B - z_B$ , i.e.

$$I_G = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix}, \quad (2.61)$$

where

$$I_{xx} = \int_V (y_B^2 + z_B^2) \rho dV \quad (2.62)$$

$$I_{yy} = \int_V (x_B^2 + z_B^2) \rho dV \quad (2.63)$$

$$I_{zz} = \int_V (x_B^2 + y_B^2) \rho dV \quad (2.64)$$

$$I_{xy} = I_{yx} = \int_V x_B y_B \rho dV \quad (2.65)$$

$$I_{xz} = I_{zx} = \int_V x_B z_B \rho dV \quad (2.66)$$

$$I_{yz} = I_{zy} = \int_V y_B z_B \rho dV \quad (2.67)$$

For the body frame linear and angular velocity components, one typically uses the following representations

$$\vec{v}_B = [u \ v \ w]^T \quad (2.68)$$

and

$$\vec{\omega}_B = [p \ q \ r]^T \quad (2.69)$$

which allows one to write out the force equation as

$$\vec{F}_B = m \left( \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \right) \quad (2.70)$$

$$\vec{F}_B = m \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \quad (2.71)$$

and the moment equation as

$$\vec{M}_B = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.72)$$

$$\vec{M}_B = \begin{bmatrix} I_{xx}\dot{p} - I_{xy}\dot{q} - I_{xz}\dot{r} \\ -I_{xy}\dot{p} + I_{yy}\dot{q} - I_{yz}\dot{r} \\ -I_{xz}\dot{p} - I_{yz}\dot{q} + I_{zz}\dot{r} \end{bmatrix} + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} I_{xx}p - I_{xy}q - I_{xz}r \\ -I_{xy}p + I_{yy}q - I_{yz}r \\ -I_{xz}p - I_{yz}q + I_{zz}r \end{bmatrix} \quad (2.73)$$

$$\vec{M}_B = \begin{bmatrix} I_{xx}\dot{p} + (I_{zz} - I_{yy})qr - I_{xy}(\dot{q} - pr) - I_{xz}(\dot{r} + pq) - I_{yz}(q^2 - r^2) \\ I_{yy}\dot{q} + (I_{xx} - I_{zz})pr - I_{xy}(\dot{p} + qr) - I_{xz}(r^2 - p^2) - I_{yz}(\dot{r} - pq) \\ I_{zz}\dot{r} + (I_{yy} - I_{xx})pq + I_{xy}(q^2 - p^2) - I_{xz}(\dot{p} - qr) - I_{yz}(\dot{q} + pr) \end{bmatrix} \quad (2.74)$$

Thus, for any rigid body in a rotating reference frame, the Newton-Euler EOMs are

$$\vec{F}_B = m \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix}$$

$$\vec{M}_B = \begin{bmatrix} I_{xx}\dot{p} + (I_{zz} - I_{yy})qr - I_{xy}(\dot{q} - pr) - I_{xz}(\dot{r} + pq) + I_{yz}(r^2 - q^2) \\ I_{yy}\dot{q} + (I_{xx} - I_{zz})pr - I_{xy}(\dot{p} + qr) + I_{xz}(p^2 - r^2) - I_{yz}(\dot{r} - pq) \\ I_{zz}\dot{r} + (I_{yy} - I_{xx})pq + I_{xy}(q^2 - p^2) - I_{xz}(\dot{p} - qr) - I_{yz}(\dot{q} + pr) \end{bmatrix} \quad (2.75)$$

These represent six coupled nonlinear ODEs with 6 free variables,  $u$ ,  $v$ ,  $w$ ,  $p$ ,  $q$ ,  $r$ , thus forming a six degree-of-freedom (6-DOF) equations of motion (EOM).

## Supplemental Equations

The Newton-Euler EOMs provide the dynamics for the linear and angular velocities of the rigid body in body frame coordinates. However, one is typically also interested in the position, velocity, and attitude of the rigid body in the navigation frame coordinates. Letting  $[\dot{x}_N \dot{y}_N \dot{z}_N]^T$  represent the velocity in the NED navigation frame, these quantities can be related to the previous body frame by

$$\begin{bmatrix} \dot{x}_N \\ \dot{y}_N \\ \dot{z}_N \end{bmatrix} = C_{N \leftarrow B} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.76)$$

or

$$\begin{bmatrix} \dot{x}_N \\ \dot{y}_N \\ \dot{z}_N \end{bmatrix} = \begin{bmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.77)$$

Using this representation in the navigation frame, it is then possible to integrate the linear velocity to obtain the position of the rigid body over time,  $\vec{x}(t)$  with the flat Earth assumption. For a more accurate inertial

velocity, one would also need to include the rotation of the navigation frame relative to the Earth and the rotation of the Earth about its axis before integrating. Regardless, integration of the velocity is still difficult to compute analytically because  $\phi$ ,  $\theta$ , and  $\psi$  are functions of time and is typically done numerically, usually Runge-Kutta methods. A **first-order Runge-Kutta method**, also known as **Euler integration**, is

$$\vec{x}_{k+1} = \vec{x}_k + (t_{k+1} - t_k) \vec{v}_{g,k} \quad (2.78)$$

To complete the rigid body dynamics, one must also relate the **Euler angle rates**,  $(\dot{\phi}, \dot{\theta}, \dot{\psi})$ , to the angular velocity of the body frame,  $\vec{\omega} = [p \ q \ r]^T$ . These quantities are generally different as the Euler angle rates define the instantaneous change in the *sequential* rotations describing the attitude of the rigid body while the angular velocity describes the rotational speed and axis about which the rigid body is rotating. However, these quantities can be shown to be related by the following equations.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.79)$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & -\sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.80)$$

It should be noted that these equations do not use rotation transformations as  $[\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$  is not a vector in three-dimensional space.

For linearized FDC, these relationships can be simplified small angle approximation for  $\phi$  and  $\theta$ , then

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & -\theta \\ 0 & 1 & -\phi \\ 0 & -\phi & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.81)$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} \approx \begin{bmatrix} \dot{\phi} - \theta \dot{\psi} \\ \dot{\theta} - \phi \dot{\psi} \\ \dot{\psi} - \phi \dot{\theta} \end{bmatrix} \quad (2.82)$$

and assuming that the Euler angle derivatives are also small, one can discard the higher order terms and obtain the simpler

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} \approx \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.83)$$

which will be used in the linearized flight vehicle dynamics later in this textbook.

## Rigid Flight Vehicle Dynamics

For flight vehicle dynamics, the forces and moments are a result of gravity, propulsion, and aerodynamics. The modeling of these three forces and moments are the differentiating factors between rigid flight vehicle dynamics and other other vehicles.

**Newton's law of gravitation** states that the **gravitational force**,  $\vec{F}_g$ , also known as the **weight**,  $W$ , is a function of the distance  $r$  between the centers of mass for the vehicle  $m$  and a celestial body  $M$ , i.e.

$$\vec{F}_g = W = \frac{GMm}{r^2} = \frac{\mu m}{r^2} \quad (2.84)$$

where  $G$  is the gravitational constant, i.e.  $6.674 \times 10^{-11} \text{ m}^3 \text{kg}^{-1} \text{s}^{-2}$  and  $\mu = GM$  is the **standard gravitational parameter** of the celestial body. The gravity of Earth and other planets are the primary forces in spacecraft FDC. For flight vehicles “near” Earth and assuming a spherical Earth, one can model the gravitational force vector,  $\vec{F}_g$ , in the NED navigation frame as

$$\vec{F}_{g,N} = \vec{W}_N = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (2.85)$$

where  $g$  is Earth’s **gravitational acceleration**. This force can be rotated to the body frame by the equation

$$\vec{F}_{g,B} = \vec{W}_N = C_{B \leftarrow N} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (2.86)$$

or

$$\vec{F}_{g,B} = \vec{W}_N = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (2.87)$$

$$\vec{F}_{g,B} = \vec{W}_N = \begin{bmatrix} -mg \sin \theta \\ mg \sin \phi \cos \theta \\ mg \cos \phi \cos \theta \end{bmatrix} \quad (2.88)$$

It should be noted again that Earth’s gravity field gradient has been neglected in this formulation and that  $g$  varies as a function of latitude, altitude, and local topography and geology. This part of the textbook will assume a simple model for the gravitational acceleration as a function of altitude,  $h$ , by the equation

$$g(h) = g_0 \left( \frac{R_e}{R_e + h} \right)^2 \quad (2.89)$$

where  $R_e$  is the **Earth’s mean radius**, i.e.  $6.3710088 \times 10^6 \text{ m}$ , and  $g_0$  is the **standard gravitational acceleration**, i.e.  $9.80665 \text{ m/s}^2$ . Higher fidelity modeling of the gravitational force is discussed in later parts of this textbook.

The propulsion forces and moments result from engines fixed in the body frame and the geometry of where the engines are placed. As the propulsion system for flight vehicles provides the vehicle the ability to travel along its path, one typically defines the **propulsive force** vector,  $\vec{F}_p$ , also known as the **thrust** vector,  $\vec{T}$ , in the body frame along the longitudinal axis, i.e.

$$\vec{F}_{p,B} = \vec{T}_B = \begin{bmatrix} T \\ 0 \\ 0 \end{bmatrix} \quad (2.90)$$

where  $T$  is the magnitude of the thrust from the engines. Note that as the propulsion system is meant to translate the entire vehicle, the moment from the nominal propulsion system,  $\vec{M}_{p,B}$ , is often designed to be zero. However, some propulsion engines can be used to steer flight vehicles, thus allowing one to produce a moment by rotating the thrust vector which serves as a control input of the flight vehicle, a design known as **thrust vectoring**. This textbook will not consider engine modeling in detail, but will assume that one can define a dynamical system model for the engine(s) that may be dependent on the flight conditions and can then be approximated by first or second order dynamics, similar to other actuators.

While typically neglected for spacecraft, the aerodynamic forces and moments result from the complex behavior between the air and an aircraft which will be discussed in later in this textbook for fixed-wing and rotary-wing aircraft. The next chapter of this textbook will introduce simple traditional models for modeling the aerodynamic forces and moments for airplanes using the mass and geometric properties as well as the flight conditions. It should be noted that the determination of these models for general aircraft is typically studied as **aircraft system identification** which typically employs **optimal parameter estimation** and is discussed in later parts of this textbook.

Thus, one can write the 6-DOF **rigid flight vehicle equations of motion** as

$$\begin{aligned} \vec{F}_{a,B} + \vec{F}_{p,B} + mg \begin{bmatrix} -\sin \theta \\ \sin \phi \cos \theta \\ \cos \phi \cos \theta \end{bmatrix} &= m \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \\ \vec{M}_{a,B} + \vec{M}_{p,B} &= \begin{bmatrix} I_{xx}\dot{p} + (I_{zz} - I_{yy})qr - I_{xy}(\dot{q} - pr) - I_{xz}(\dot{r} + pq) + I_{yz}(r^2 - q^2) \\ I_{yy}\dot{q} + (I_{xx} - I_{zz})pr - I_{xy}(\dot{p} + qr) + I_{xz}(p^2 - r^2) - I_{yz}(\dot{r} - pq) \\ I_{zz}\dot{r} + (I_{yy} - I_{xx})pq + I_{xy}(q^2 - p^2) - I_{xz}(\dot{p} - qr) - I_{yz}(\dot{q} + pr) \end{bmatrix} \end{aligned} \quad (2.91)$$

### Example Problem

Given: an aircraft is initially flying due north with wings level, then it starts maneuvering. At some instant in this maneuver, the aircraft's attitude is given by the following Euler angles:  $\phi = 90^\circ$ ,  $\theta = 30^\circ$ , and  $\psi = -120^\circ$ . Assume all Euler angles are  $0^\circ$  initially.

Determine:

- Describe the aircraft's attitude in words (e.g., nose down or up; left wing down or right wing down; heading east, west, northwest, etc.).
- If the aircraft weighs 35,000 lbs, what are the components of the weight in the body frame, i.e.  $W_{B,x}$ ,  $W_{B,y}$  and  $W_{B,z}$ ?
- If the aircraft's center of mass is moving North at 175 knots, what are  $u$ ,  $v$  and  $w$ ?

Solution:

a) For  $\psi = -120^\circ$ , the plane is heading southwest.

For  $\theta = 30^\circ$ , the nose is pointing up.

For  $\phi = 90^\circ$  the right wing has turned straight downward.

b) From the previous lecture, recall

$$C_{B \leftarrow N} = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix} \quad (2.92)$$

and for Euler angles  $(90^\circ, 30^\circ, -120^\circ)$

$$C_{B \leftarrow N} = \begin{bmatrix} -0.4330 & -0.75 & -0.5 \\ -0.25 & -0.4330 & 0.8660 \\ -0.8660 & 0.5 & 0 \end{bmatrix} \quad (2.93)$$

Then,

$$\begin{bmatrix} W_{B,x} \\ W_{B,y} \\ W_{B,z} \end{bmatrix} = C_{B \leftarrow N} \begin{bmatrix} 0 \\ 0 \\ 35000 \end{bmatrix} \quad (2.94)$$

$$\begin{bmatrix} W_{B,x} \\ W_{B,y} \\ W_{B,z} \end{bmatrix} = \begin{bmatrix} -0.4330 & -0.75 & -0.5 \\ -0.25 & -0.4330 & 0.8660 \\ -0.8660 & 0.5 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 35000 \end{bmatrix} \quad (2.95)$$

$$\begin{bmatrix} W_{B,x} \\ W_{B,y} \\ W_{B,z} \end{bmatrix} = \begin{bmatrix} -17500 \\ 30300 \\ 0 \end{bmatrix} \text{ lb} \quad (2.96)$$

c) Also from the previous lecture

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = C_{B \leftarrow N} \begin{bmatrix} 175 \\ 0 \\ 0 \end{bmatrix} \quad (2.97)$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} -0.4330 & -0.75 & -0.5 \\ -0.25 & -0.4330 & 0.8660 \\ -0.8660 & 0.5 & 0 \end{bmatrix} \begin{bmatrix} 175 \\ 0 \\ 0 \end{bmatrix} \quad (2.98)$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} -75 \\ -43.75 \\ -151.55 \end{bmatrix} \text{ knots} \quad (2.99)$$

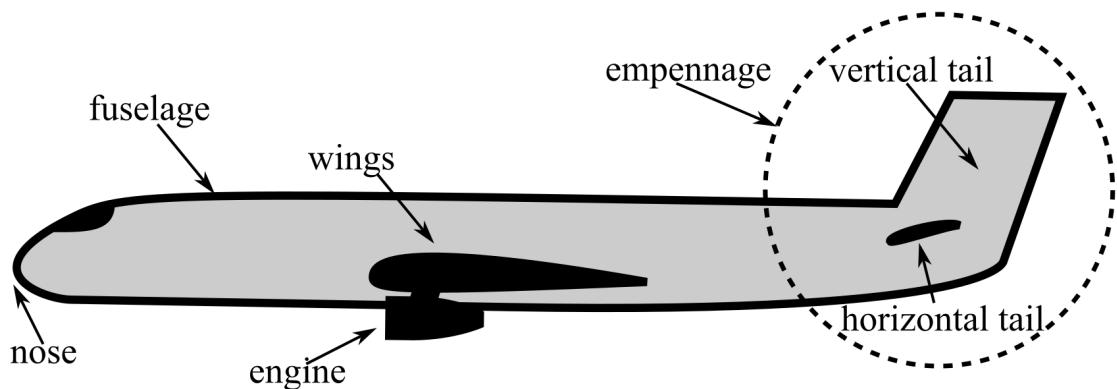
## Chapter 3

# Introductory Airplane Dynamics

### 3.1 Rigid Airplane Dynamics

#### Airplane Anatomy

The basic geometric components of a traditional airplane design are shown in the following diagram



The **nose** is the front of the airplane and houses the **cockpit** where the pilot(s) and/or other operators are located. The **fuselage** is the tubular structure of the airplane that houses the payload, fuel, and/or avionics. The **wings** extend horizontally out of the fuselage and generate the majority of the lift force for the aircraft to fly. The **empennage** is the rear section of an airplane and creates dynamic effects to steer the airplane in the direction of flight. The **horizontal tail** produces a vertical up or down lift force to steer the airplane and maintain horizontal stability while the **vertical tail** produces a horizontal left or right force to steer the airplane and maintain vertical stability. Lastly, the **engines** produce the thrust forces which are controlled by the pilot. These can be mounted on the wings, empennage, or nose. Some airplanes are designed with different aerodynamic features than these. One example is a **canard**, i.e. a forward horizontal stabilizer, either in addition or instead of a rear horizontal stabilizer, i.e. a horizontal tail. Another is a **v-tail** which configures the entire tail in a “v” shape instead of a “t” shape.

Fixed-wing aircraft generally have three major control surfaces that allow the pilot or autopilot to affect the aerodynamic forces on the aircraft thereby altering its motion in a controlled manner. The **elevator** is mounted on trailing edge of horizontal tail and is used as the primary pitch angle,  $\theta$ , control input as shown in the following graphic.

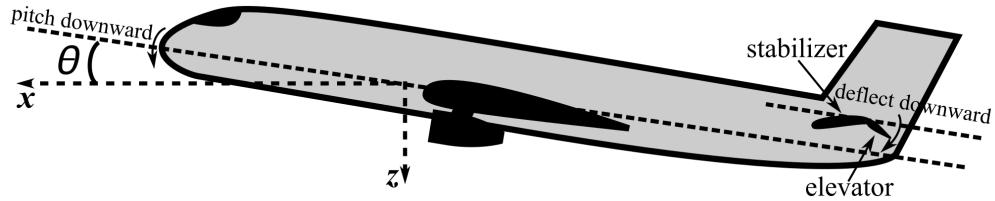


Figure 3.1: A deflection upwards pushes the airplane's nose up.

The **rudder** is mounted on trailing edge of the vertical tail and is used as the primary yaw angle,  $\psi$ , control input as shown in the following graphic.

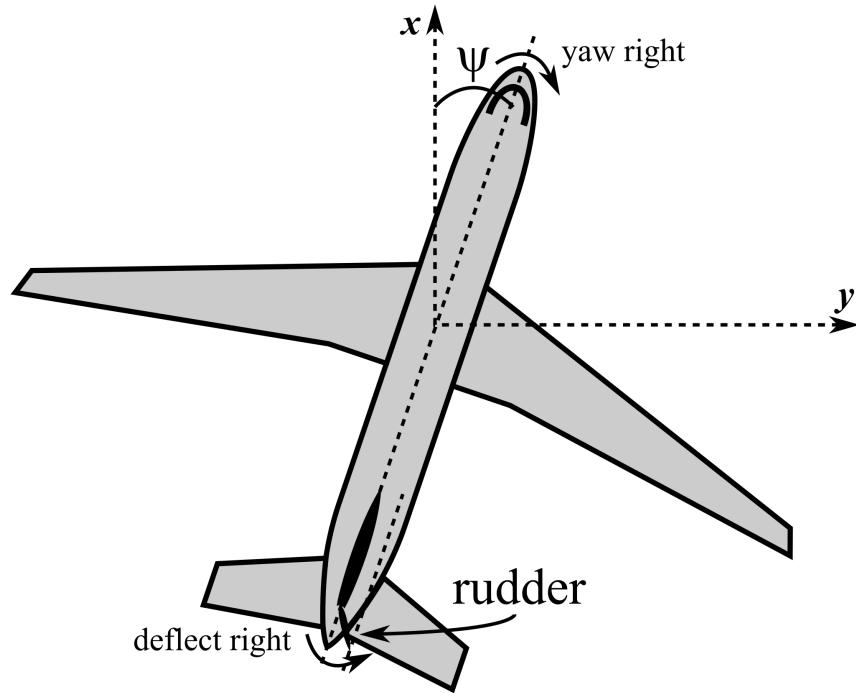


Figure 3.2: A deflection right pushes airplane's nose right.

The **ailerons** are a differential pair of surfaces, i.e. if one deflects up, the other deflects down, mounted on trailing edge of wings near the tips and are used as the primary roll angle,  $\phi$ , control input as shown in the following graphic.

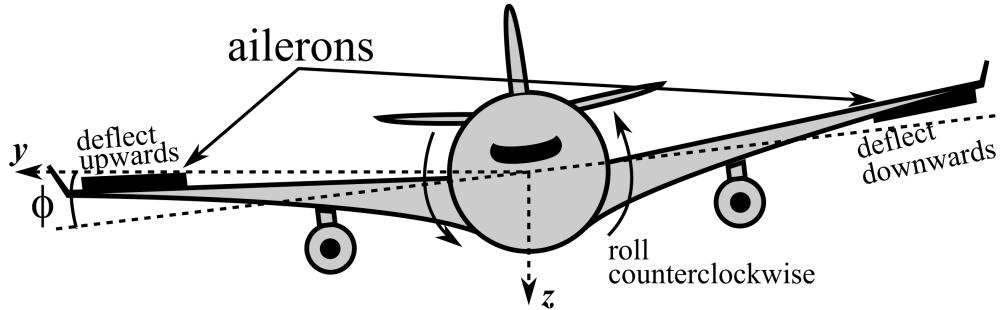


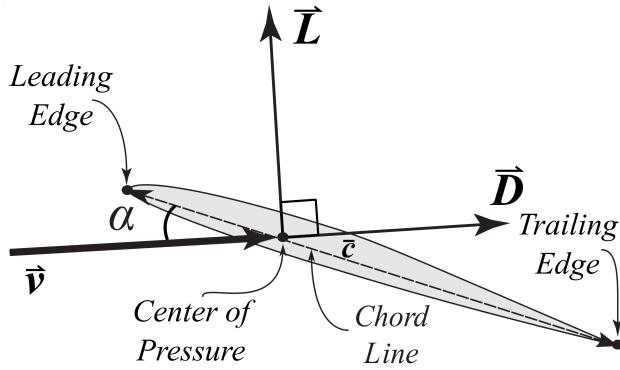
Figure 3.3: The airplane rolls to the side that deflects up.

Some aircraft may also compound primary control surfaces which include a compound rudder and elevator called a **ruddervator**, a compound pair of ailerons and elevator called **elevons** or **tailerons**, and a compound horizontal stabilizer and elevator called a **stabilator**.

Many airplanes also have secondary control surfaces that can be controlled to change the airplane aerodynamics and thus alter its motion. These are typically grouped into two different categories. **Spoilers** are hinged flat plates attached to wing facing oncoming airflow. These “spoil” the lift on the wing when rotated up from the surface of the wing and thus are typically used as speed brakes, but sometimes can be used for roll control with or without ailerons. **Slats** and **flaps** are mechanized leading and trailing edges of wings, respectively, that can be extended/retracted from the nominal wing shape, thereby changing the airfoil cross-sectional shape and effectively increasing/decreasing the lift potential of the wing with higher/lower induced drag. As such, these are primarily extended during takeoff and landing and are not dynamically active control surfaces, i.e. they are either extended or retracted for different flight conditions and tasks.

### Finite Wing Theory

To develop the basic analytical models for modeling the aerodynamic forces and moments for airplanes in this chapter, one can use **finite wing theory** for each unique structure on a airplane while also accounting for some interactive aerodynamics between these structures. Finite wing theory uses the simpler analysis of the flow over any two-dimensional cross-section of the wing, also known as an **airfoil** and resolves the two-dimensional pressure distribution of the moving air over the wing into two perpendicular force contributions as shown in the following free body diagram.



where  $c$  is the **aerodynamic chord** from the leading edge to trailing edge,  $v_\infty$  is the **free-stream velocity**,  $\alpha$  is the **angle of attack**,  $L$  is the airfoil's **lift force** defined as perpendicular to the free-stream, and  $D$  is the airfoil's **drag force** defined as parallel to the free-stream velocity. It should be noted that the **center of pressure** defines the location about which the pitching moment due to the pressure distribution is currently zero, thus producing no pitching moment. However, as this location will generally change as a function of angle of attack, one often uses the **aerodynamic center** of the airfoil about which the pitching moment will be constant regardless of the angle of attack. Thus, for any airfoil at any angle of attack, one can resolve the pressure distribution as the lift, drag, and pitching moment at some location along the aerodynamic chord.

Considering a finite wing as a distribution of airfoils at a single angle of attack, one can model the lift, drag, and pitching moment on the entire wing as

$$L_w = \frac{1}{2} \rho v_\infty^2 S_w C_{L,w} = Q_\infty S_w C_{L,w} \quad (3.1)$$

$$D_w = \frac{1}{2} \rho v_\infty^2 S_w C_{D,w} = Q_\infty S_w C_{D,w} \quad (3.2)$$

and

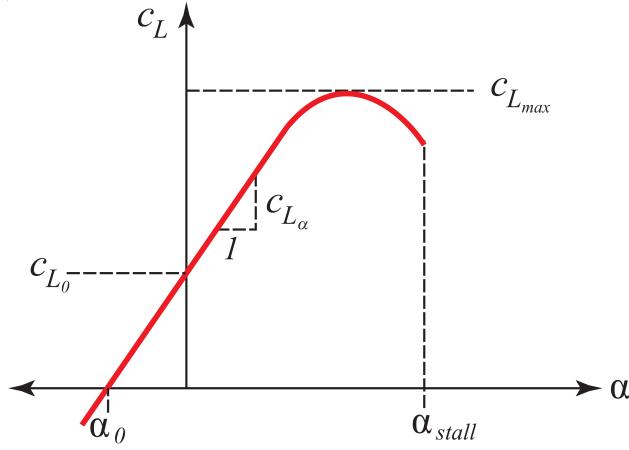
$$M_w = \frac{1}{2} \rho v_\infty^2 S_w \bar{c}_w C_{M,w} = Q_\infty S_w \bar{c}_w C_{M,w} \quad (3.3)$$

where  $\rho$  is the **air density**,  $S_w$  is the surface area of the wing,  $Q_\infty$  is **dynamic pressure** for the free-stream velocity,  $\bar{c}_w$  is the **mean aerodynamic chord** of the wing,  $C_{L,w}$  is the **lift coefficient** of the wing,  $C_{D,w}$  is the **drag coefficient** of the wing,  $M_w$  is the **pitching moment**, and  $C_{M,w}$  is the **pitching moment coefficient** of the wing. Furthermore, due to their finite nature, wings may also have a **rolling moment**,  $L_{roll,w}$ , defined as

$$L_{roll,w} = \frac{1}{2} \rho v_\infty^2 S_w b_w C_{L_{roll},w} = Q_\infty S_w b_w C_{L_{roll},w} \quad (3.4)$$

where  $b_w$  is the **span** of the wing and  $C_{L_{roll},w}$  is the **rolling moment coefficient** of the wing. It should be noted that often the “*roll*” subscript is dropped which noticeably overloads the use of the letter,  $L$ , and must be inferred by context whether one is referring to the lift or the rolling moment.

The lift coefficient for a finite wing generally depends on the angle of attack as depicted in the following plot.



At low angles of attack, the relationship is linear with **lift coefficient slope**,  $C_{L_{\alpha,w}} = \frac{dC_{L,w}}{d\alpha}$ , or

$$C_{L,w} = C_{L_{\alpha,w}}(\alpha_w - \alpha_{0,w}) \quad (3.5)$$

where  $\alpha_{0,w} \leq 0$  is the **zero lift angle of attack** for the wing. This can also be written as

$$C_{L,w} = C_{L_{\alpha,w}}\alpha_w + C_{L_0,w} \quad (3.6)$$

where  $C_{L_0,w}$  is the lift coefficient at  $\alpha = 0$ . If  $C_{L_0,w} = \alpha_{0,w} = 0$ , then the airfoil is **symmetric**, otherwise it is **cambered**. Note that at high angles of attack, a maximum lift coefficient is reached,  $C_{L_{max}}$ , after which there is a slight decrease in the lift coefficient and eventually a large decrease. With this large decrease in lift there is also a large increase in drag, a phenomenon known as **stall** which occurs at some  $\alpha_{stall}$ . As the purpose of airfoils are generally to generate lift with a small amount of drag, aircraft typically operate at angles of attack well below stall and will be assumed in this textbook.

The drag coefficient for a finite wing can generally be separated into two terms as

$$C_{D,w} = C_{D_0,w} + C_{D_i,w} \quad (3.7)$$

where  $C_{D_0,w}$  is the **parasitic drag coefficient** of the wing due to air pressure and skin friction and  $C_{D_i,w}$  is the **induced drag coefficient** of the wing due to the production of lift. The parasitic drag coefficient is a constant term with respect to the flight conditions and thus is also known as the **zero-lift drag coefficient** while the induced drag coefficient is typically modeled as a parabolic function

$$C_{D_i,w} = kC_{L,w}^2 \quad (3.8)$$

where  $k$  is some constant.

For a wing with constant airfoil shape, one can use **lifting-line theory** which predicts that the lift coefficient slope for a wing can be approximately related to the airfoil's lift coefficient slope,  $c_{L_{alpha}}$ , by

$$C_{L_{\alpha,w}} = c_{L_\alpha} \left( \frac{AR_w}{AR_w + 2} \right) \quad (3.9)$$

where  $AR_w$  is the **aspect ratio** of the wing defined as

$$AR_w = \frac{b_w^2}{S_w} \quad (3.10)$$

For drag, lifting-line theory predicts that

$$C_{D_i,w} = \frac{C_{L,w}^2}{\pi AR_w e_w} \quad (3.11)$$

where  $e_w$  is the wing efficiency factor where if  $e_w = 1$ , the wing is defined as **elliptical**. However, it should be noted that most aircraft use multiple types of airfoils at different points of a lifting surface, a technique called **aerodynamic twisting** which makes an analytical solution more difficult to obtain than lifting-line theory. This simple theory is a good approximation for aircraft at low speeds with a high aspect ratio and little sweep to their wings.

### Rigid Airplane Dynamics

Recall the 6-DOF rigid flight vehicle equations of motion as

$$\begin{aligned} \vec{F}_{a,B} + \vec{F}_{p,B} + mg \begin{bmatrix} -\sin \theta \\ \sin \phi \cos \theta \\ \cos \phi \cos \theta \end{bmatrix} &= m \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \\ \vec{M}_{a,B} + \vec{M}_{p,B} &= \begin{bmatrix} I_{xx}\dot{p} + (I_{zz} - I_{yy})qr - I_{xy}(\dot{q} - pr) - I_{xz}(\dot{r} + pq) + I_{yz}(r^2 - q^2) \\ I_{yy}\dot{q} + (I_{xx} - I_{zz})pr - I_{xy}(\dot{p} + qr) + I_{xz}(p^2 - r^2) - I_{yz}(\dot{r} - pq) \\ I_{zz}\dot{r} + (I_{yy} - I_{xx})pq + I_{xy}(q^2 - p^2) - I_{xz}(\dot{p} - qr) - I_{yz}(\dot{q} + pr) \end{bmatrix} \end{aligned} \quad (3.12)$$

As can be seen in the previous airplane anatomy subsection, one can see that the vast majority of airplanes are designed as symmetric in the  $x_B - y_B$  and  $y_B - z_B$  planes, thus the inertia matrix can be simplified to

$$I_G = \begin{bmatrix} I_{xx} & 0 & -I_{xz} \\ 0 & I_{yy} & 0 \\ -I_{xz} & 0 & I_{zz} \end{bmatrix} \quad (3.13)$$

where often  $I_{xz}$  is also neglected due to its relatively small magnitude.

For the rigid flight vehicle dynamics, airplanes generate aerodynamic forces and moments from several different static lifting surfaces. For each of these surfaces, one can define an aerodynamic center about which the lift, drag, and aerodynamic moment are resolved for that surface. For traditional airplanes, there are four primary lifting surfaces to consider in aerodynamic coefficient modeling, namely the wing aerodynamic coefficients denoted with subscript  $w$ , the horizontal tail aerodynamic coefficients denoted with subscript  $h$ , the vertical tail aerodynamic coefficients denoted with subscript  $v$ , and the fuselage aerodynamic coefficients denoted with subscript  $f$ . For inclusion into the rigid flight vehicle dynamics, one must define the aerodynamic forces and moments for the airplane in the body frame. Traditionally, the aerodynamic and propulsive forces along the  $x_B$ -,  $y_B$ -, and  $z_B$ -axes are normalized by the mass of the airplane,  $m$ , and denoted by  $X$ ,  $Y$ , and  $Z$ , respectively. This infers the following force equation for airplanes as

$$m \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \vec{F}_{p,B} + \vec{F}_{a,B} \quad (3.14)$$

Furthermore, the aerodynamic and propulsive moments about the  $x_B$ -,  $y_B$ -, and  $z_B$ -axes are traditionally normalized by the moment of inertia about that axis, i.e.  $x_B$ -,  $y_B$ -, and  $z_B$ , respectively, and denoted by  $L$ ,  $M$ , and  $N$ , respectively. This infers the following moment equation for airplanes as

$$\begin{bmatrix} I_{xx}L \\ I_{yy}M \\ I_{zz}N \end{bmatrix} = \vec{M}_{p,B} + \vec{M}_{a,B} \quad (3.15)$$

It should be noted that the gravitational force is kept separate from these terms. Finally, with these substitutions, one obtains the **rigid airplane equations of motion**

$$\begin{aligned} \begin{bmatrix} X - g \sin \theta \\ Y + g \sin \phi \cos \theta \\ Z - \cos \phi \cos \theta \end{bmatrix} &= \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \\ \begin{bmatrix} L \\ M \\ N \end{bmatrix} &= \begin{bmatrix} \dot{p} + \frac{I_{zz}-I_{yy}}{I_{xx}} qr - \frac{I_{xz}}{I_{xx}} (\dot{r} + pq) \\ \dot{q} + \frac{I_{xx}-I_{zz}}{I_{yy}} pr + \frac{I_{xz}}{I_{yy}} (p^2 - r^2) \\ \dot{r} + \frac{I_{yy}-I_{xx}}{I_{zz}} pq - \frac{I_{xz}}{I_{zz}} (\dot{p} - qr) \end{bmatrix} \end{aligned} \quad (3.16)$$

with supplemental equations

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (3.17)$$

which notably only requires that eight states be known to calculate the derivatives at any instant, namely  $u$ ,  $v$ ,  $w$ ,  $p$ ,  $q$ ,  $r$ ,  $\phi$ , and  $\theta$ , while  $\psi$  is simply a derived parameter from  $p$ ,  $q$ , and  $r$ . Furthermore, though these equations use eight states, there are only six degrees-of-freedom as  $\dot{\phi}$ ,  $\dot{\theta}$  are completely dictated by  $p$ ,  $q$ , and  $r$ .

As an alternative to this form of the EOMs, one may also substitute for the lateral and vertical velocity terms

$$v = u \tan \beta \quad (3.18)$$

$$w = u \sin \alpha \quad (3.19)$$

as well as their derivatives

$$\dot{v} = \dot{u} \tan \beta + \dot{\beta} u \sec^2 \beta \quad (3.20)$$

$$\dot{w} = \dot{u} \sin \alpha + \dot{\alpha} u \cos \alpha \quad (3.21)$$

in order to obtain the alternative EOMs

$$\begin{bmatrix} X - g \sin \theta \\ Y + g \cos \theta \sin \phi \\ Z + g \cos \theta \cos \phi \\ L \\ M \\ N \end{bmatrix} = \begin{bmatrix} \dot{u} + qu \sin \alpha - ru \tan \beta \\ \dot{u} \tan \beta + \dot{\beta} u \sec^2 \beta + ru - pu \sin \alpha \\ \dot{u} \sin \alpha + \dot{\alpha} u \cos \alpha + pu \tan \beta - qu \\ \dot{p} + \frac{I_{zz}-I_{yy}}{I_{xx}} qr - \frac{I_{xz}}{I_{xx}} (\dot{r} + pq) \\ \dot{q} + \frac{I_{xx}-I_{zz}}{I_{yy}} pr - \frac{I_{xz}}{I_{yy}} (r^2 - p^2) \\ \dot{r} + \frac{I_{yy}-I_{xx}}{I_{zz}} pq - \frac{I_{xz}}{I_{zz}} (\dot{p} - qr) \end{bmatrix} \quad (3.22)$$

Similar to finite wing theory, one can also model an airplane's total aerodynamic force vector,  $\vec{F}_a$ , in the wind frame as

$$\vec{F}_{a,W} = \begin{bmatrix} -D \\ S \\ -L \end{bmatrix} \quad (3.23)$$

where  $D$  is the **drag force**,  $S$  is the **side force**, and  $L$  is the **lift force** for the entire airplane. These wind frame aerodynamic forces can be rotated to the body frame by

$$\vec{F}_{a,B} = C_{B \leftarrow W} \begin{bmatrix} -D \\ S \\ -L \end{bmatrix} \quad (3.24)$$

$$\vec{F}_{a,B} = \begin{bmatrix} \cos \alpha \cos \beta & -\cos \alpha \sin \beta & -\sin \alpha \\ \sin \beta & \cos \beta & 0 \\ \sin \alpha \cos \beta & -\sin \alpha \sin \beta & \cos \alpha \end{bmatrix} \begin{bmatrix} -D \\ S \\ -L \end{bmatrix} \quad (3.25)$$

$$\vec{F}_{a,B} = \begin{bmatrix} -D \cos \alpha \cos \beta - S \cos \alpha \sin \beta + L \sin \alpha \\ S \cos \beta - D \sin \beta \\ -D \sin \alpha \cos \beta - S \sin \alpha \sin \beta - L \sin \alpha \end{bmatrix} \quad (3.26)$$

where one notably has the relationship

$$m \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \vec{F}_{p,B} + \begin{bmatrix} -D \cos \alpha \cos \beta - S \cos \alpha \sin \beta + L \sin \alpha \\ S \cos \beta - D \sin \beta \\ -D \sin \alpha \cos \beta - S \sin \alpha \sin \beta - L \sin \alpha \end{bmatrix} \quad (3.27)$$

which can be used as an alternative to  $X$ ,  $Y$ , and  $Z$  in the translation equation for rigid airplane dynamics. It should be noted that this notation overloads the use of  $L$  as both the overall airplane lift and the aerodynamic moment about the  $x_B$ -axis where the ambiguity must be resolved using the context of the variable in this textbook. In addition,  $S$  here without a subscript should not be confused with the surface area of a lifting surface which always has a specifying subscript with it in this textbook. In this model, the propulsive force is separated from the aerodynamic force. For rigid airplanes, one typically models the propulsive force as

$$\vec{F}_{p,B} = \begin{bmatrix} T \cos \theta_T \\ 0 \\ T \sin \theta_T \end{bmatrix} \quad (3.28)$$

where  $\theta_T$  is a potential offset angle with respect to the  $x_B$ -axis of the body frame. Moreover, a propulsive moment may also be present nominally about the  $y_B$ -axis as

$$\vec{M}_{p,B} = [T(z_T \cos \phi_T - x_T \sin \phi_T)] \quad (3.29)$$

where  $(x_T, z_T)$  denotes the location of the thrust force in the  $x_B - z_B$  plane. It should be noted that often  $\theta_T = 0^\circ$  or can be neglected and  $T$  is generally a function of the airspeed, altitude, and throttle setting.

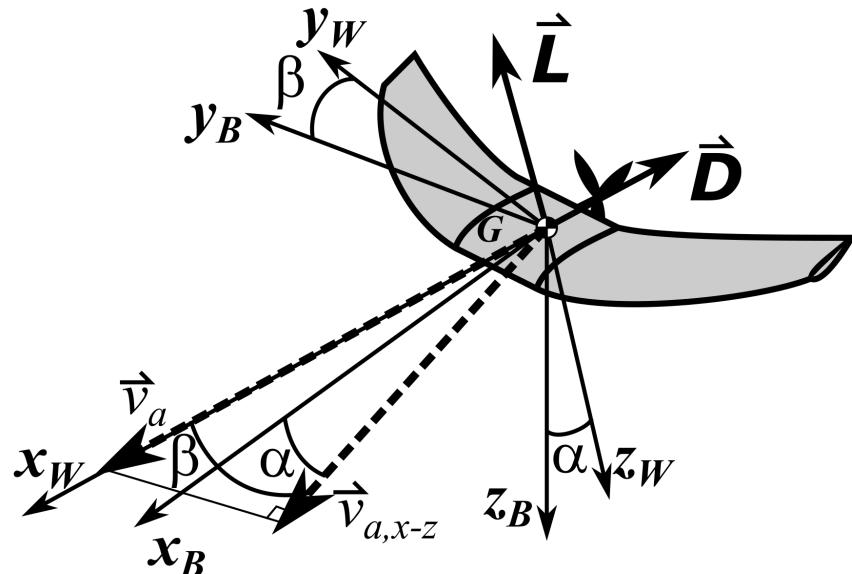
Regardless of which model is used, the overall airplane aerodynamic forces and moments are not simple to compute for arbitrary flight conditions as the lift and drag are functions of the local airspeed, angle of attack, and sideslip angle at each lifting surface, their geometric layout, as well as the control inputs to the elevator, rudder, and ailerons, and the rotational motion of the rigid body. Thus, one typically linearizes the rigid airplane dynamics about equilibrium flight conditions in order to analyze its response characteristics and design suitable control systems.

**Example Problem**Given:

The airspeed vector for a flying wing at sea level ( $\rho = 0.002378 \text{ slugs/ft}^3$ ) is measured as

$$\vec{v}_a = \begin{bmatrix} 100 \\ 5 \\ 10 \end{bmatrix} \text{ ft/s}$$

This flying wing has the following characteristics:  $S_w = 300 \text{ ft}^2$ ,  $b_w = 30$ ,  $e_w = 0.9$ ,  $C_{L\alpha,w} = 5.5$ ,  $C_{L0,w} = 0$  and  $C_{D0,w} = 0.2$ .

Determine:

- the angle of attack,  $\alpha$
- the sideslip angle,  $\beta$
- the lift vector,  $\vec{L}$ , in the body frame
- the drag vector,  $\vec{D}$ , in the body frame

Assume:

- Lifting-line theory

Solution:

- The angle of attack is given by

$$\alpha = \arctan \frac{w}{u} \quad (3.30)$$

$$\alpha = \arctan \frac{10}{100} \quad (3.31)$$

$$\underline{\alpha = 5.71^\circ} \quad (3.32)$$

(b) The sideslip angle is given by

$$\beta = \arcsin \frac{v}{|\vec{v}|} \quad (3.33)$$

$$\beta = \arcsin \frac{v}{\sqrt{u^2 + v^2 + w^2}} \quad (3.34)$$

$$\beta = \arcsin \frac{5}{\sqrt{100^2 + 5^2 + 10^2}} \quad (3.35)$$

$$\underline{\beta = 2.85^\circ} \quad (3.36)$$

(c) The magnitude of the lift force is

$$L = 0.5\rho |\vec{v}_a|^2 S_w C_{L,w} = 0.5\rho |\vec{v}_a|^2 S_w C_{L_{\alpha,w}} \alpha \quad (3.37)$$

The transformation matrix from the wind frame to the body frame is

$$\vec{L}_B = \begin{bmatrix} \cos \alpha \cos \beta & -\cos \alpha \sin \beta & -\sin \alpha \\ \sin \beta & \cos \beta & 0 \\ \sin \alpha \cos \beta & -\sin \alpha \sin \beta & \cos \alpha \end{bmatrix} \vec{L}_W \quad (3.38)$$

Recalling the lift force is aligned with the  $-z_W$  axis, one has

$$\vec{L}_B = \begin{bmatrix} \cos \alpha \cos \beta & -\cos \alpha \sin \beta & -\sin \alpha \\ \sin \beta & \cos \beta & 0 \\ \sin \alpha \cos \beta & -\sin \alpha \sin \beta & \cos \alpha \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -0.5\rho |\vec{v}_a|^2 S_w C_{L_{\alpha,w}} \alpha \end{bmatrix} \quad (3.39)$$

$$\vec{L}_B = \begin{bmatrix} 0.5\rho |\vec{v}_a|^2 S_w C_{L_{\alpha,w}} \alpha \sin \alpha \\ 0 \\ -0.5\rho |\vec{v}_a|^2 S_w C_{L_{\alpha,w}} \alpha \cos \alpha \end{bmatrix} \quad (3.40)$$

Substituting values,

$$\vec{L}_B = \begin{bmatrix} 0.5(0.002378)(100^2 + 5^2 + 10^2)(300)(5.5) \left(5.71^\circ \frac{\pi}{180^\circ}\right) \sin(5.71^\circ) \\ 0 \\ -0.5(0.002378)(100^2 + 5^2 + 10^2)(300)(5.5) \left(5.71^\circ \frac{\pi}{180^\circ}\right) \cos(5.71^\circ) \end{bmatrix} \quad (3.41)$$

$$\underline{\vec{L}_B = \begin{bmatrix} 197 \\ 0 \\ -1970 \end{bmatrix} \text{lb.}} \quad (3.42)$$

(d) The magnitude of the drag force is

$$D = 0.5\rho |\vec{v}_a|^2 S_w C_{D,w} \quad (3.43)$$

and the drag coefficient,  $C_D$ , as a function of the lift coefficient is

$$C_D = C_{D_{0,w}} + \frac{C_L^2}{\pi e_w A R_w} \quad (3.44)$$

Recalling  $AR = \frac{b^2}{S_w}$  and substituting for  $C_L$

$$C_D = C_{D_{0,w}} + \frac{(C_{L_{\alpha,w}} \alpha)^2}{\pi e_w \frac{b_w^2}{S_w}} \quad (3.45)$$

Then,

$$D = 0.5\rho |\vec{v}_a|^2 S_w \left( C_{D_{0,w}} + \frac{(C_{L_{\alpha,w}} \alpha)^2}{\pi e_w \frac{b_w^2}{S_w}} \right) \quad (3.46)$$

The transformation matrix from the wind frame to the body frame is

$$\vec{D}_B = \begin{bmatrix} \cos \alpha \cos \beta & -\cos \alpha \sin \beta & -\sin \alpha \\ \sin \beta & \cos \beta & 0 \\ \sin \alpha \cos \beta & -\sin \alpha \sin \beta & \cos \alpha \end{bmatrix} \vec{D}_W \quad (3.47)$$

Recalling the drag force is aligned with the  $-x_W$  axis,

$$\vec{D}_B = \begin{bmatrix} \cos \alpha \cos \beta & -\cos \alpha \sin \beta & -\sin \alpha \\ \sin \beta & \cos \beta & 0 \\ \sin \alpha \cos \beta & -\sin \alpha \sin \beta & \cos \alpha \end{bmatrix} \begin{bmatrix} -0.5\rho |\vec{v}_a|^2 S_w \left( C_{D_0} + \frac{(C_{L_\alpha} \alpha)^2}{\pi e \frac{b^2}{S_w}} \right) \\ 0 \\ 0 \end{bmatrix} \quad (3.48)$$

$$\vec{D}_B = \begin{bmatrix} -0.5\rho |\vec{v}_a|^2 S_w \left( C_{D_0} + \frac{(C_{L_\alpha} \alpha)^2}{\pi e \frac{b^2}{S_w}} \right) \cos \alpha \cos \beta \\ -0.5\rho |\vec{v}_a|^2 S_w \left( C_{D_0} + \frac{(C_{L_\alpha} \alpha)^2}{\pi e \frac{b^2}{S_w}} \right) \sin \beta \\ -0.5\rho |\vec{v}_a|^2 S_w \left( C_{D_0} + \frac{(C_{L_\alpha} \alpha)^2}{\pi e \frac{b^2}{S_w}} \right) \sin \alpha \cos \beta \end{bmatrix} \quad (3.49)$$

Substituting values,

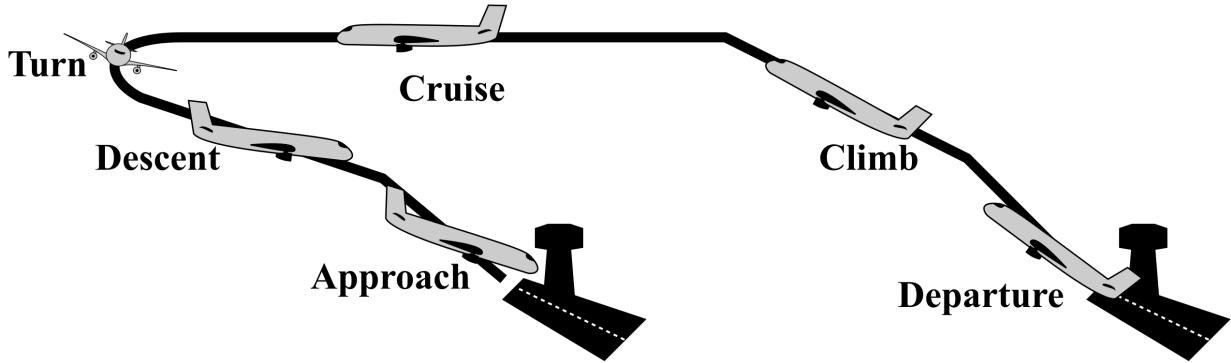
$$\vec{D}_B = \begin{bmatrix} -0.5(0.002378)(100^2 + 5^2 + 10^2)(300) \left( 0.2 + \frac{((5.5)(5.71^\circ \frac{\pi}{180^\circ}))^2}{\pi(0.9)\frac{30^2}{300}} \right) \cos(5.71^\circ) \cos(2.85^\circ) \\ -0.5(0.002378)(100^2 + 5^2 + 10^2)(300) \left( 0.2 + \frac{((5.5)(5.71^\circ \frac{\pi}{180^\circ}))^2}{\pi(0.9)\frac{30^2}{300}} \right) \sin(2.85^\circ) \\ -0.5(0.002378)(100^2 + 5^2 + 10^2)(300) \left( 0.2 + \frac{((5.5)(5.71^\circ \frac{\pi}{180^\circ}))^2}{\pi(0.9)\frac{30^2}{300}} \right) \sin(5.71^\circ) \cos(2.85^\circ) \end{bmatrix} \quad (3.50)$$

$$\underline{\vec{D}_B = \begin{bmatrix} -845 \\ -42.3 \\ -84.5 \end{bmatrix} \text{ lb.}} \quad (3.51)$$

## 3.2 Airplane Trimmed Steady Flight

### Steady Flight Conditions

The six phases of flight can generally be described as departure, climb, cruise, descent, and approach as shown in the following graphic.



where each phase would have different flight conditions including potential turns during climb, cruise, and/or descent. The airplane may also be configured differently at each phase, e.g. for takeoff/departure and approach/landing, flaps may be deployed and the landing gear will be extended down, which will directly impact the aerodynamic forces and moments. To analyze the response of an airplane during each of these flight phases, one can solve for the **equilibrium flight conditions**, also known as the **steady flight conditions**, for each phase which, by definition, occur when the state variables in the rigid airplane EOMs are constant, i.e.

$$\dot{u} = \dot{v} = \dot{w} = \dot{p} = \dot{q} = \dot{r} = \dot{\phi} = \dot{\theta} = 0 \quad (3.52)$$

which imply that the steady flight conditions solve the **steady flight equations** in the body frame as

$$\begin{bmatrix} \bar{X} - g \sin \bar{\theta} \\ \bar{Y} + g \sin \bar{\phi} \cos \bar{\theta} \\ \bar{Z} - \cos \bar{\phi} \cos \bar{\theta} \\ \bar{L} \\ \bar{M} \\ \bar{N} \end{bmatrix} = \begin{bmatrix} \bar{q}\bar{u} \sin \bar{\alpha} - \bar{r}\bar{u} \tan \bar{\beta} \\ \bar{r}\bar{u} - \bar{p}\bar{u} \sin \bar{\alpha} \\ \bar{p}\bar{u} \tan \bar{\beta} - \bar{q}\bar{u} \\ \frac{I_{zz}-I_{yy}}{I_{xx}}\bar{q}\bar{r} - \frac{I_{xz}}{I_{xx}}\bar{p}\bar{q} \\ \frac{I_{xx}-I_{zz}}{I_{yy}}\bar{p}\bar{r} + \frac{I_{xz}}{I_{yy}}(\bar{p}^2 - \bar{r}^2) \\ \frac{I_{yy}-I_{xx}}{I_{zz}}\bar{p}\bar{q} + \frac{I_{xz}}{I_{zz}}\bar{q}\bar{r} \end{bmatrix} \quad (3.53)$$

and

$$\begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \bar{p} & \bar{q} \sin \bar{\phi} \tan \bar{\theta} & \bar{r} \cos \bar{\phi} \tan \bar{\theta} \\ 0 & \bar{q} \cos \bar{\phi} & -\bar{r} \sin \bar{\phi} \\ 0 & \bar{q} \sin \bar{\phi} \sec \bar{\theta} & \bar{r} \cos \bar{\phi} \sec \bar{\theta} \end{bmatrix} \quad (3.54)$$

where the  $\bar{u}$ ,  $\bar{\beta}$ ,  $\bar{\alpha}$ ,  $\bar{p}$ ,  $\bar{q}$ ,  $\bar{r}$ ,  $\bar{\phi}$ , and  $\bar{\theta}$  are the steady flight conditions while  $\bar{X}$ ,  $\bar{Y}$ ,  $\bar{Z}$ ,  $\bar{L}$ ,  $\bar{M}$ , and  $\bar{N}$  are the aerodynamic and propulsive forces and moments at steady flight and are functions of the steady flight conditions. It should also be noted that the gravitational acceleration,  $g$  and air density,  $\rho$ , will also vary

as a function of altitude which further requires that a strictly steady flight condition would be at a constant altitude. However, as these variations occur slowly, one typically assumes these are constant for analyzing different steady flight maneuvers. It should also be noted that one may alternatively use the thrust, lift, side, and drag forces at steady flight,  $\bar{T}$ ,  $\bar{L}$ ,  $\bar{S}$ , and  $\bar{D}$ , respectively, instead of  $\bar{X}$ ,  $\bar{Y}$ , and  $\bar{Z}$ .

While each of these steady flight equations allow for a wide variety of conditions, an important condition is when  $\bar{S} = 0$ , i.e. **coordinated flight**, for which one often assumes  $\beta = 0^\circ$ . Furthermore, one is often interested in determining the steady flight conditions for an approximate point-mass airplane, e.g. in a performance analysis of the airplane. In this case, one may use directly rewrite the translation equations of motion for coordinated steady flight in the wind frame as

$$\bar{F}_{a,W} + \bar{F}_{p,W} + \bar{F}_{g,W} = m \begin{bmatrix} \bar{p}_W \\ \bar{q}_W \\ \bar{r}_W \end{bmatrix} \times \begin{bmatrix} \bar{v}_\infty \\ 0 \\ 0 \end{bmatrix} = m\bar{v}_\infty \begin{bmatrix} 0 \\ \bar{r}_W \\ -\bar{q}_W \end{bmatrix} \quad (3.55)$$

where  $[\bar{p}_W \bar{q}_W \bar{r}_W]^T$  is the angular velocity of the wind frame which can be related to the wind frame Euler angles by

$$\begin{bmatrix} p_W \\ q_W \\ r_W \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \gamma \\ 0 & \cos \mu & -\sin \mu \cos \gamma \\ 0 & -\sin \mu & \cos \mu \cos \gamma \end{bmatrix} \begin{bmatrix} \dot{\mu} \\ \dot{\gamma} \\ \dot{\sigma} \end{bmatrix} \quad (3.56)$$

and for a prescribed  $m\bar{u}$  and  $\dot{\gamma}$  at steady flight, one has

$$\begin{bmatrix} 0 \\ \bar{r}_W \\ -\bar{q}_W \end{bmatrix} = \begin{bmatrix} 0 \\ \dot{\sigma} \cos \bar{\mu} \cos \bar{\gamma} \\ \dot{\sigma} \sin \bar{\mu} \cos \bar{\gamma} \end{bmatrix} \quad (3.57)$$

For the aerodynamic force, one can define

$$\bar{F}_{a,W} = \begin{bmatrix} -\bar{D} \\ 0 \\ -\bar{L} \end{bmatrix} \quad (3.58)$$

For the propulsive force assuming  $\theta_T = 0^\circ$  and  $\beta = 0^\circ$ , one can define

$$\bar{F}_{p,W} = C_{W \leftarrow B} \begin{bmatrix} \bar{T} \\ 0 \\ 0 \end{bmatrix}_B = \begin{bmatrix} \bar{T} \cos \bar{\alpha} \\ 0 \\ -\bar{T} \sin \bar{\alpha} \end{bmatrix} \quad (3.59)$$

For the force of gravity, one can define

$$\bar{F}_{g,W} = C_{W \leftarrow N} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}_N = \begin{bmatrix} -mg \sin \bar{\gamma} \\ mg \sin \bar{\mu} \cos \bar{\gamma} \\ mg \cos \bar{\mu} \cos \bar{\gamma} \end{bmatrix} \quad (3.60)$$

Then, by substitution, one has

$$\begin{bmatrix} \bar{T} \cos \bar{\alpha} - \bar{D} - mg \sin \bar{\gamma} \\ mg \sin \bar{\mu} \cos \bar{\gamma} \\ -\bar{T} \sin \bar{\alpha} - \bar{L} + mg \cos \bar{\mu} \cos \bar{\gamma} \end{bmatrix} = m\bar{v}_\infty \begin{bmatrix} 0 \\ \dot{\sigma} \cos \bar{\mu} \cos \bar{\gamma} \\ \dot{\sigma} \sin \bar{\mu} \cos \bar{\gamma} \end{bmatrix} \quad (3.61)$$

Finally, defining  $R$  as the instantaneous radius of curvature in the navigation frame, i.e.

$$R = \frac{\bar{v}_\infty \cos \bar{\gamma}}{\dot{\sigma}} \quad (3.62)$$

one has

$$\begin{bmatrix} \bar{T} \cos \bar{\alpha} - \bar{D} - mg \sin \bar{\gamma} \\ mg \sin \bar{\mu} \cos \bar{\gamma} \\ -\bar{T} \sin \bar{\alpha} - \bar{L} + mg \cos \bar{\mu} \cos \bar{\gamma} \end{bmatrix} = \begin{bmatrix} 0 \\ m \frac{(\bar{v}_\infty \cos \bar{\gamma})^2}{R} \cos \bar{\mu} \\ m \frac{(\bar{v}_\infty \cos \bar{\gamma})^2}{R} \sin \bar{\mu} \end{bmatrix} \quad (3.63)$$

which are known as the **performance steady flight equations**. It should be noted that from the second equation, i.e.

$$mg \sin \bar{\mu} \cos \bar{\gamma} = m \frac{(\bar{v}_\infty \cos \bar{\gamma})^2}{R} \cos \bar{\mu} \quad (3.64)$$

one can see that

$$R = \frac{\bar{v}_\infty^2 \cos \bar{\gamma}}{g \tan \bar{\mu}} \quad (3.65)$$

and

$$\dot{\sigma} = \frac{g \tan \bar{\mu}}{\bar{v}_\infty} \quad (3.66)$$

which is defined for any non-zero bank angle.

The most general maneuver described by these simplified steady flight equations is a steady climbing or descending coordinated turn and are primarily controlled by altering the lift, thrust, and bank angle of the airplane. The trajectory the airplane flies during this maneuver is a helix about the  $z_E$ -axis and a circular projection on the  $x_E - y_E$  plane. Three special cases of this steady flight maneuver are straight climbs/descents, level turns, and straight-and-level where **straight flight** occurs when  $\dot{\sigma} = \bar{\mu} = 0^\circ$  and **level flight** occurs when  $\bar{\gamma} = 0^\circ$ . As the lift and drag are primarily a function of the steady-state air density, airspeed, and angle of attack, the performance steady flight equations can be considered as a balance of six conditions, the altitude (affects air density), bank angle, flight path angle, angle of attack, airspeed, and thrust for a given airplane's aerodynamic and mass properties. However, once one considers the airplane as a rigid body, the additional moment equations must also be balanced to ensure that the airplane remains at the prescribed steady flight conditions. As these additional moment equations can be altered by the control inputs to the ailerons, rudder, and elevator, one typically refers to this moment balance as **trimming the airplane**.

### Airplane Trim Points for Steady Flight

Recall the moment equations for steady flight are

$$\begin{bmatrix} \bar{L} \\ \bar{M} \\ \bar{N} \end{bmatrix} = \begin{bmatrix} \frac{I_{zz} - I_{yy}}{I_{xx}} \bar{q} \bar{r} - \frac{I_{xz}}{I_{xx}} \bar{p} \bar{q} \\ \frac{I_{xx} - I_{zz}}{I_{yy}} \bar{p} \bar{r} + \frac{I_{xz}}{I_{yy}} (\bar{p}^2 - \bar{r}^2) \\ \frac{I_{yy} - I_{xx}}{I_{zz}} \bar{p} \bar{q} + \frac{I_{xz}}{I_{zz}} \bar{q} \bar{r} \end{bmatrix} \quad (3.67)$$

and by the relationship

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & -\sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (3.68)$$

for a prescribed  $\bar{\phi}$  and  $\bar{\theta}$  at steady flight, one has

$$\begin{bmatrix} \bar{p} \\ \bar{q} \\ \bar{r} \end{bmatrix} = \begin{bmatrix} -\dot{\psi} \sin \bar{\theta} \\ -\dot{\psi} \sin \bar{\phi} \cos \bar{\theta} \\ \dot{\psi} \cos \bar{\phi} \cos \bar{\theta} \end{bmatrix} \quad (3.69)$$

which, in general results in a complicated moment equation. However, if  $\dot{\psi} = 0^\circ$ , i.e. straight flight, then  $\bar{L} = \bar{M} = \bar{N} = 0$  at trim. This steady flight requirement for straight flight results in the **static stability equations** and are typically written in terms of the moment coefficients,  $C_l$ ,  $C_m$ , and  $C_n$ , which remove the inherent variation due to air density and airspeed. Note that here lowercase letters are used so that  $C_L$  is not overloaded for both lift and the rolling moment. This formulation allows for the direct calculation of the aileron, elevator, and rudder trim values through **static control coefficients**:  $C_{m_{\delta_e}}$ ,  $C_{l_{\delta_a}}$ ,  $C_{l_{\delta_r}}$ ,  $C_{n_{\delta_a}}$ , and  $C_{n_{\delta_r}}$ . It should be noted that for establishing steady flight, the thrust control input is adjusted for the prescribed translation balance.

For the  $M$  moment coefficient, one can define the following equation

$$C_m = C_{m_0} + C_{m_\alpha} \alpha + C_{m_{\delta_e}} \delta_e \quad (3.70)$$

where  $C_{m_{\delta_e}}$  is the **elevator control power**. For trimmed straight flight,  $C_m = 0$  or

$$0 = C_{m_0} + C_{m_\alpha} \bar{\alpha} + C_{m_{\delta_e}} \bar{\delta}_e \quad (3.71)$$

or

$$\bar{\delta}_e = -\frac{C_{m_0} + C_{m_\alpha} \bar{\alpha}}{C_{m_{\delta_e}}} \quad (3.72)$$

However, the elevator will also affect the lift coefficient of the airplane due to its effect on the lift vector of the tail. This can be modeled as

$$C_L = C_{L_0} + C_{L_\alpha} \alpha + C_{L_{\delta_e}} \delta_e \quad (3.73)$$

Thus, at some prescribed  $\bar{C}_L$  for straight flight

$$\bar{\alpha} = \frac{\bar{C}_L - C_{L_{\delta_e}} \delta_e}{C_{L_\alpha}} \quad (3.74)$$

and by substitution

$$\bar{\delta}_e = -\frac{C_{m_0} C_{L_\alpha} + C_{m_\alpha} \bar{C}_L}{C_{m_{\delta_e}} C_{L_\alpha} - C_{m_\alpha} C_{L_{\delta_e}}} \quad (3.75)$$

For the  $L$  and  $N$  moment coefficients, one can define the following equations

$$C_l = C_{l_\beta} \beta + C_{l_{\delta_a}} \delta_a + C_{l_{\delta_r}} \delta_r \quad (3.76)$$

and

$$C_n = C_{n_\beta} \beta + C_{n_{\delta_a}} \delta_a + C_{n_{\delta_r}} \delta_r \quad (3.77)$$

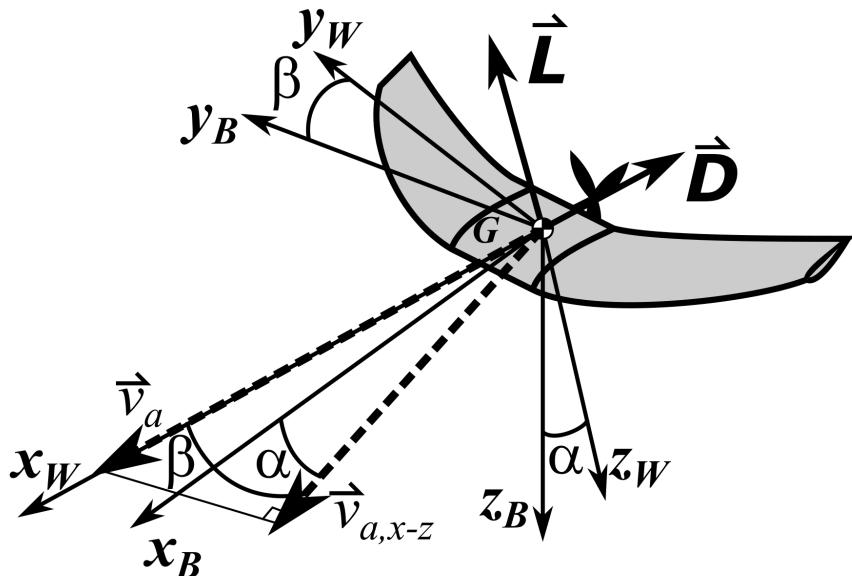
where  $C_{l_{\delta_a}}$  is the **aileron control power** and  $C_{n_{\delta_r}}$  is the **rudder control power** as these primarily affect the roll and yaw moments, respectively, but are still coupled effects. Though steady flight primarily occurs for  $\bar{\beta} = \bar{\delta}_a = \bar{\delta}_r = 0^\circ$ , there are two primary situations where aileron and rudder commands will be needed for trim: crosswind landings and engine outage. To maintaining alignment with the runway during crosswind, landing requires trimmed flight at a sideslip angle. An engine outage creates asymmetric thrust adding a moment term to  $C_n$ , thus enough rudder power is needed to overcome the resulting moment for low flight speeds (i.e. high  $\bar{C}_L$ ).

**Example Problem 1**Given:

The airspeed vector for a flying wing flying at sea level ( $\rho = 0.002378 \text{ slugs/ft}^3$ ) is measured as

$$\vec{v}_a = \begin{bmatrix} 100 \\ 5 \\ 10 \end{bmatrix} \text{ ft/s}$$

This flying wing has the following characteristics:  $S_w = 300 \text{ ft}^2$ ,  $b_w = 30$ ,  $e_w = 0.9$ ,  $C_{L\alpha} = 5.5$ ,  $C_{L,0} = 0$  and  $C_{D,0} = 0.2$ .

Determine:

- the angle of attack,  $\alpha$
- the sideslip angle,  $\beta$
- the lift vector,  $\vec{L}$ , in the body frame
- the drag vector,  $\vec{D}$ , in the body frame

Assume:

- $\vec{L}$  and  $\vec{D}$  are aligned with the  $-z_w$  and  $-x_w$  axes, respectively.
- Lifting-line theory

Solution:

- The angle of attack is given by

$$\alpha = \arctan \frac{w}{u} \quad (3.78)$$

$$\alpha = \arctan \frac{10}{100} \quad (3.79)$$

$$\underline{\alpha = 5.71^\circ} \quad (3.80)$$

(b) The sideslip angle is given by

$$\beta = \arcsin \frac{v}{|\vec{v}|} \quad (3.81)$$

$$\beta = \arcsin \frac{v}{\sqrt{u^2 + v^2 + w^2}} \quad (3.82)$$

$$\beta = \arcsin \frac{5}{\sqrt{100^2 + 5^2 + 10^2}} \quad (3.83)$$

$$\underline{\beta = 2.85^\circ} \quad (3.84)$$

(c) The magnitude of the lift force is

$$L = 0.5\rho |\vec{v}_a|^2 S_w C_L = 0.5\rho |\vec{v}_a|^2 S_w C_{L_\alpha} \alpha \quad (3.85)$$

The transformation matrix from the wind frame to the body frame is

$$\vec{L}_B = \begin{bmatrix} \cos \alpha \cos \beta & -\cos \alpha \sin \beta & -\sin \alpha \\ \sin \beta & \cos \beta & 0 \\ \sin \alpha \cos \beta & -\sin \alpha \sin \beta & \cos \alpha \end{bmatrix} \vec{L}_W \quad (3.86)$$

Recalling the lift force is aligned with the  $-z_W$  axis, one has

$$\vec{L}_B = \begin{bmatrix} \cos \alpha \cos \beta & -\cos \alpha \sin \beta & -\sin \alpha \\ \sin \beta & \cos \beta & 0 \\ \sin \alpha \cos \beta & -\sin \alpha \sin \beta & \cos \alpha \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -0.5\rho |\vec{v}_a|^2 S_w C_{L_\alpha} \alpha \end{bmatrix} \quad (3.87)$$

$$\vec{L}_B = \begin{bmatrix} 0.5\rho |\vec{v}_a|^2 S_w C_{L_\alpha} \alpha \sin \alpha \\ 0 \\ -0.5\rho |\vec{v}_a|^2 S_w C_{L_\alpha} \alpha \cos \alpha \end{bmatrix} \quad (3.88)$$

Substituting values,

$$\vec{L}_B = \begin{bmatrix} 0.5(0.002378)(100^2 + 5^2 + 10^2)(300)(5.5) \left(5.71^\circ \frac{\pi}{180^\circ}\right) \sin(5.71^\circ) \\ 0 \\ -0.5(0.002378)(100^2 + 5^2 + 10^2)(300)(5.5) \left(5.71^\circ \frac{\pi}{180^\circ}\right) \cos(5.71^\circ) \end{bmatrix} \quad (3.89)$$

$$\underline{\vec{L}_B = \begin{bmatrix} 197 \\ 0 \\ -1970 \end{bmatrix} \text{lb.}} \quad (3.90)$$

(d) The magnitude of the drag force is

$$D = 0.5\rho |\vec{v}_a|^2 S_w C_D \quad (3.91)$$

and the drag coefficient,  $C_D$ , as a function of the lift coefficient is

$$C_D = C_{D,0} + \frac{C_L^2}{\pi e_w A R_w} \quad (3.92)$$

Recalling  $AR = \frac{b^2}{S_w}$  and substituting for  $C_L$

$$C_D = C_{D,0} + \frac{(C_{L_a} \alpha)^2}{\pi e_w \frac{b_w^2}{S_w}} \quad (3.93)$$

Then,

$$D = 0.5\rho |\vec{v}_a|^2 S_w \left( C_{D,0} + \frac{(C_{L_a} \alpha)^2}{\pi e_w \frac{b_w^2}{S_w}} \right) \quad (3.94)$$

The transformation matrix from the wind frame to the body frame is

$$\vec{D}_B = \begin{bmatrix} \cos \alpha \cos \beta & -\cos \alpha \sin \beta & -\sin \alpha \\ \sin \beta & \cos \beta & 0 \\ \sin \alpha \cos \beta & -\sin \alpha \sin \beta & \cos \alpha \end{bmatrix} \vec{D}_W \quad (3.95)$$

Recalling the drag force is aligned with the  $-x_W$  axis,

$$\vec{D}_B = \begin{bmatrix} \cos \alpha \cos \beta & -\cos \alpha \sin \beta & -\sin \alpha \\ \sin \beta & \cos \beta & 0 \\ \sin \alpha \cos \beta & -\sin \alpha \sin \beta & \cos \alpha \end{bmatrix} \begin{bmatrix} -0.5\rho |\vec{v}_a|^2 S_w \left( C_{D,0} + \frac{(C_{L_a} \alpha)^2}{\pi e_w \frac{b_w^2}{S_w}} \right) \\ 0 \\ 0 \end{bmatrix} \quad (3.96)$$

$$\vec{D}_B = \begin{bmatrix} -0.5\rho |\vec{v}_a|^2 S_w \left( C_{D,0} + \frac{(C_{L_a} \alpha)^2}{\pi e_w \frac{b_w^2}{S_w}} \right) \cos \alpha \cos \beta \\ -0.5\rho |\vec{v}_a|^2 S_w \left( C_{D,0} + \frac{(C_{L_a} \alpha)^2}{\pi e_w \frac{b_w^2}{S_w}} \right) \sin \beta \\ -0.5\rho |\vec{v}_a|^2 S_w \left( C_{D,0} + \frac{(C_{L_a} \alpha)^2}{\pi e_w \frac{b_w^2}{S_w}} \right) \sin \alpha \cos \beta \end{bmatrix} \quad (3.97)$$

Substituting values,

$$\vec{D}_B = \begin{bmatrix} -0.5(0.002378)(100^2 + 5^2 + 10^2)(300) \left( 0.2 + \frac{((5.5)(5.71^\circ \frac{\pi}{180^\circ}))^2}{\pi(0.9)\frac{30^2}{300}} \right) \cos(5.71^\circ) \cos(2.85^\circ) \\ -0.5(0.002378)(100^2 + 5^2 + 10^2)(300) \left( 0.2 + \frac{((5.5)(5.71^\circ \frac{\pi}{180^\circ}))^2}{\pi(0.9)\frac{30^2}{300}} \right) \sin(2.85^\circ) \\ -0.5(0.002378)(100^2 + 5^2 + 10^2)(300) \left( 0.2 + \frac{((5.5)(5.71^\circ \frac{\pi}{180^\circ}))^2}{\pi(0.9)\frac{30^2}{300}} \right) \sin(5.71^\circ) \cos(2.85^\circ) \end{bmatrix} \quad (3.98)$$

$$\underline{\vec{D}_B = \begin{bmatrix} -845 \\ -42.3 \\ -84.5 \end{bmatrix} \text{ lb.}} \quad (3.99)$$

**Example Problem 2**

Given: the following information for a modified Cessna 172

$\rho = 0.002378 \text{ slug/ft}^3$	$W = 2300 \text{ lb.}$	$C_{L_0} = 0.27$	$C_{L_{max}} = 1.24$
$C_{D,0} = 0.035$	$S_w = 174 \text{ ft}^2$	$b = 36 \text{ ft}$	$e = 0.97$

Determine:

- (a)  $v_{\infty,min}$  for straight-and-level steady flight (in knots)
- (b)  $T$  at  $v_{\infty,min}$  for straight-and-level steady flight

Assume:

1.  $\alpha = 0$  in steady flight equations
2. Lifting-line theory

Solution:

a) For straight-and-level flight with  $\alpha = 0$ ,  $\gamma = \mu = 0$  and the steady flight equations dictate  $L = mg = W$ , or

$$W = \frac{1}{2}\rho v_{\infty}^2 S_w C_L \quad (3.100)$$

Rearranging,

$$v_{\infty} = \sqrt{\frac{2W}{\rho S_w C_L}} \quad (3.101)$$

Minimum airspeed occurs at  $C_{L,max}$

$$v_{\infty,min} = \sqrt{\frac{2W}{\rho S_w C_{L,max}}} = \sqrt{\frac{2 \times 2300 \text{ lb}}{0.002378 \text{ slug/ft}^3 \times 174 \text{ ft}^2 \times 1.24}} = 94.7 \text{ ft/s} \quad (3.102)$$

$$\underline{v_{\infty,min} = 55.6 \text{ knots}} \quad (3.103)$$

b) Also for straight-and-level flight with  $\alpha = 0$ , the steady flight equations dictate  $T = D$ , or

$$T = \frac{1}{2}\rho v_{\infty}^2 S_w C_D \quad (3.104)$$

$$T = \frac{1}{2}\rho v_{\infty}^2 S_w (C_{D,0} + C_{D,i}) \quad (3.105)$$

$$T = \frac{1}{2}\rho v_{\infty}^2 S_w (C_{D,0} + \frac{C_{L,max}^2}{\pi A R_w e_w}) \quad (3.106)$$

Then,

$$T \text{ at } v_{\infty,min} = \frac{1}{2}\rho v_{\infty,min}^2 S_w (C_{D,0} + \frac{C_{L,max}^2}{\pi A R_w e_w}) \quad (3.107)$$

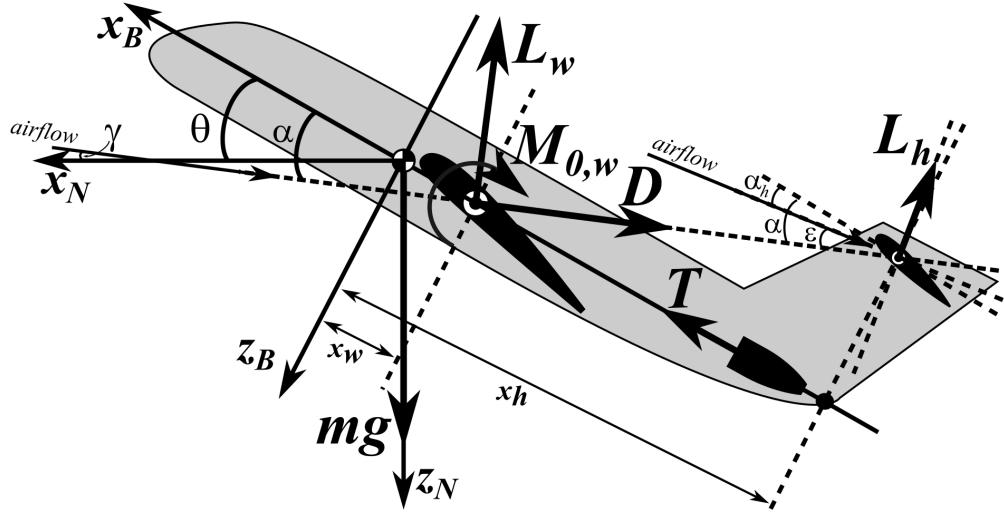
$$T \text{ at } v_{\infty,min} = \frac{1}{2}(0.002378 \text{ slug/ft}^3)(94.7 \text{ ft/s})^2(174 \text{ ft}^2) \left( 0.035 + \frac{1.24^2}{\pi \times \frac{36^2}{174} \times 0.97} \right) \quad (3.108)$$

$$\underline{T \text{ at } v_{\infty,min} = 190.6 \text{ lb}} \quad (3.109)$$

### 3.3 Airplane Static Stability

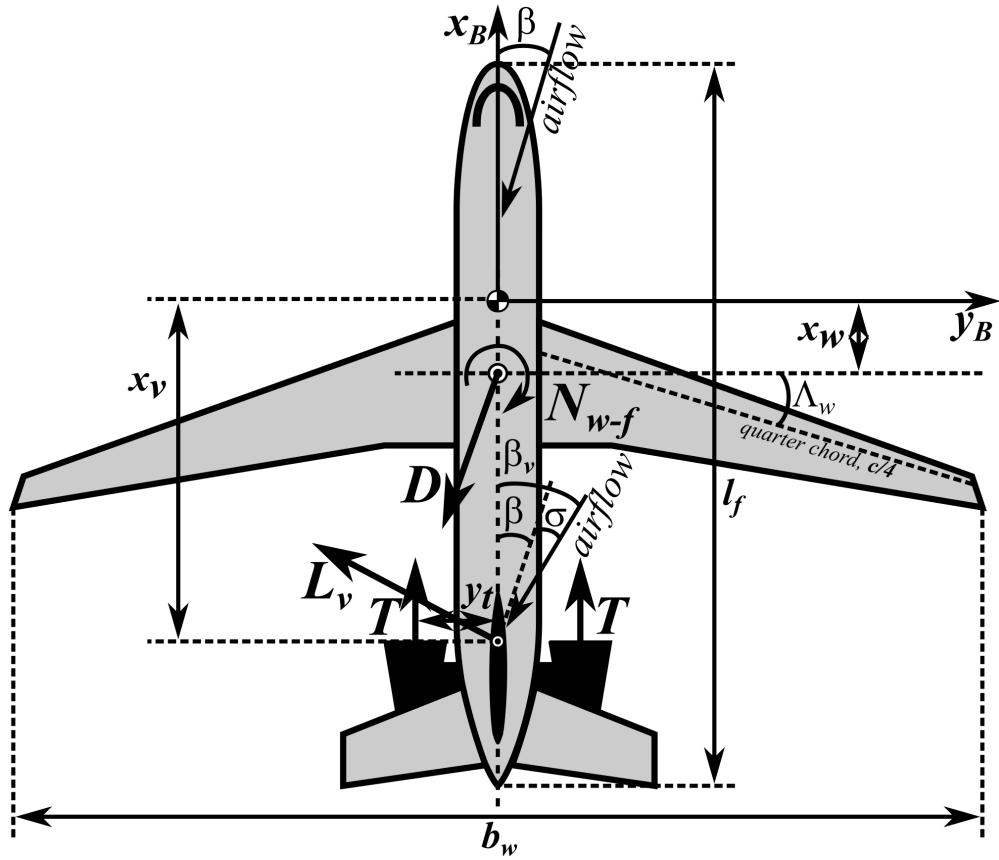
To analyze the aerodynamics moments and static stability, consider the following simplified free-body diagram (FBD) in three dimensions.

From the view of the  $x_B - z_B$ -plane, also known as the **longitudinal plane**, the simplified FBD with the  $y_B$ -axis drawn into the page can be drawn as



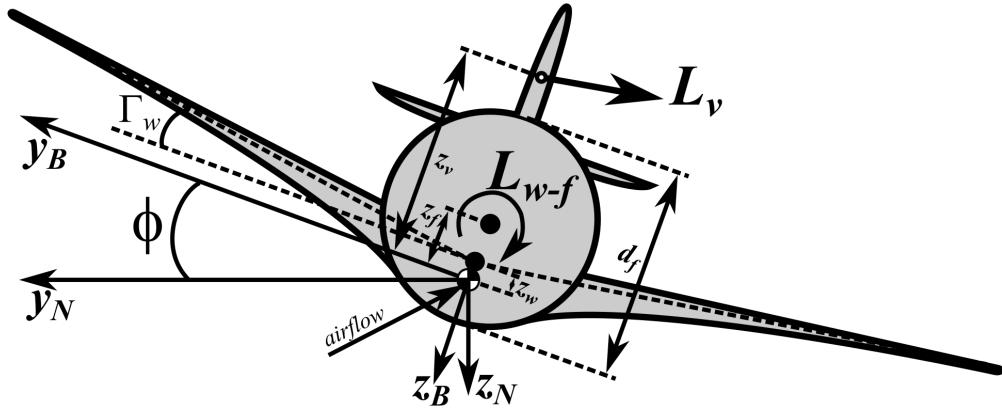
where the airplane's induced drag is dominated by the wing and the parasitic drag has been resolved at the wing (i.e. this replaces  $D_f$ ,  $D_w$ ,  $D_h$ , and  $D_v$  by  $D$  at wing aerodynamic center), the drag and thrust contribution to the pitching moment on the airplane has been neglected, the horizontal tail airfoil is symmetric (i.e.  $M_{0,h} = 0$ ), and  $\epsilon$  is the **downwash angle** which results from the airflow interacting with the wings and engines upstream of the horizontal tail. Note that the aerodynamic centers of each lifting surface in this plane are labeled by their  $x$  and  $z$  coordinates in the body frame. It should also be noted that the lift from the fuselage is often quite small in this plane.

From the view of the  $x_B - y_B$ -plane, also known as the **directional plane**, the simplified FBD with the  $z_B$ -axis drawn into the page can be drawn as



where the wing and fuselage effects can be combined into two terms,  $L_{w-f}$  and  $N_{w-f}$ , the vertical tail airfoil is symmetric (i.e.  $N_{0,v} = 0$ ), and  $\sigma$  is the **sidewash angle** which results from the airflow interacting with the wings and engines upstream of the vertical tail. It should be noted that each thrust vector has a  $y_B$  component which cancel each other with respect to the yawing moment.

From the view of the  $y_B - z_B$ -plane, also known as the **lateral plane**, the simplified FBD with the  $x_B$ -axis drawn out of the page can be drawn as



It should be noted that the aerodynamic centers of the fuselage and wing are additionally represented by  $z_B$  coordinates in the body frame which was neglected in the longitudinal frame.

Furthermore, if one assumes the small angle approximation for  $\alpha$  and  $\beta$ ,  $\alpha - \epsilon$ , and  $\beta + \sigma$ , then, one obtains the simplified wind-to-body aerodynamic and propulsive force and moment equations as

$$\begin{aligned}
 mX &= (L_w + L_h)\alpha - D + T \\
 mY &= -D\beta - L_v \\
 mZ &= (L_w + L_h) - D\alpha \\
 I_{xx}L &= L_{w-f} - z_v L_v \\
 I_{yy}M &= M_{0,w} + x_w L_w + x_h L_h \\
 I_{zz}N &= N_{w-f} - x_v L_v
 \end{aligned} \tag{3.110}$$

where  $\alpha$  and  $\beta$  are in radians and  $x_w$ ,  $x_h$ , and  $x_v$  are the body frame coordinates of the aerodynamic centers, i.e. negative values in the FBD. Note that  $L$  and  $L_{w-f}$  are moment terms and not lift terms here. By substituting for  $(L_w + L_h)$ ,  $D$ ,  $T$ ,  $L_w$ ,  $L_h$ ,  $L_v$ , and  $N_v$  by their aerodynamic coefficients, one has

$$\begin{aligned}
 mX &= Q_\infty S_w C_L \alpha - Q_\infty S_w C_D + Q_\infty S_w C_T \\
 mY &= -Q_\infty S_w C_D \beta - Q_v S_v C_{L,v} \\
 mZ &= -Q_\infty S_w C_L - Q_\infty S_w C_D \alpha \\
 I_{xx}L &= Q_\infty S_w b_w C_{l,w-f} - z_v Q_v S_v C_{L,v} \\
 I_{yy}M &= Q_\infty S_w \bar{c}_w C_{m_0,w} + x_w Q_\infty S_w C_{L,w} + x_h Q_h S_h C_{L,h} \\
 I_{zz}N &= Q_\infty S_w b_w C_{n,w-f} - x_v Q_v S_v C_{L,v}
 \end{aligned} \tag{3.111}$$

where the lowercase  $l$ ,  $m$ , and  $n$  are used for the body frame moment coefficient subscripts as  $C_L$  is used for the overall airplane lift coefficient, in this case  $L_w + L_h$ . These equations can be further simplified into

nondimensional coefficient equations by the definitions

$$\begin{aligned} X &= \frac{Q_\infty S_w}{m} C_X \\ Y &= \frac{Q_\infty S_w}{m} C_Y \\ Z &= \frac{Q_\infty S_w}{m} C_Z \\ L &= \frac{Q_\infty S_w b_w}{I_{xx}} C_l \\ M &= \frac{Q_\infty S_w \bar{c}_w}{I_{yy}} C_m \\ N &= \frac{Q_\infty S_w b_w}{I_{zz}} C_n \end{aligned} \quad (3.112)$$

which provides the following approximation for the force and moment coefficients as

$$\begin{aligned} C_X &= C_L \alpha - C_D + C_T \\ C_Y &= -C_D \beta - \frac{Q_v S_v}{Q_\infty S_w} C_{L,v} \\ C_Z &= -C_L - C_D \alpha \\ C_l &= C_{l,w-f} - \frac{Q_v S_v z_v}{Q_\infty S_w} C_{L,v} \\ C_m &= C_{m_0,w} + \frac{x_w}{\bar{c}_w} C_{L,w} - \frac{Q_h S_h x_h}{Q_\infty S_w \bar{c}_w} C_{L,h} \\ C_n &= C_{n,w-f} - \frac{Q_v S_v x_v}{Q_\infty S_w b_w} C_{L,v} \end{aligned} \quad (3.113)$$

Note that these simplified equations will also be used in this course to derive analytical models for studying airplane dynamics about steady flight conditions. The remainder of this section will consider only the moment coefficients and the relative angles of attack of the wing, horizontal tail, and vertical tail for computing the static stability equations.

### Moment Coefficient Models and Trim

For the  $M$  moment, the coefficient is

$$C_m = C_{m_0,w} + \frac{x_w}{\bar{c}_w} C_{L,w} - \frac{Q_h S_h x_h}{Q_\infty S_w \bar{c}_w} C_{L,h} \quad (3.114)$$

Next, modeling the lift coefficients for small angles of attack as

$$C_{L,w} = C_{L_0,w} + C_{L_{\alpha},w} \alpha_w \quad (3.115)$$

and

$$C_{L,h} = C_{L_{\alpha},h} \alpha_h \quad (3.116)$$

where it has been assumed that the vertical tail has a symmetric airfoil, thus providing no net force at  $\alpha_h = 0^\circ$ . Then, one has

$$C_m = C_{m_0,w} + \frac{x_w}{\bar{c}_w} (C_{L_0,w} + C_{L_\alpha,w}\alpha) - \frac{Q_h S_h x_h}{Q_\infty S_w \bar{c}_w} C_{L_\alpha,h} \alpha_h \quad (3.117)$$

Then, by the previous FBD, one also has

$$\alpha_h = \alpha - \epsilon \quad (3.118)$$

where  $\epsilon$  is typically approximated by a first order Taylor series expansion

$$\epsilon = \epsilon_0 + \frac{d\epsilon}{d\alpha} \alpha \quad (3.119)$$

where for an elliptical lift distribution

$$\epsilon_0 = \frac{2C_{L,w}}{\pi AR_w} \quad (3.120)$$

and

$$\frac{d\epsilon}{d\alpha} = \frac{2C_{L_\alpha,w}}{\pi AR_w} \quad (3.121)$$

though these are typically estimated using test data and/or CFD. Next, substituting for  $\alpha_h$  and  $\epsilon$ , one has

$$\begin{aligned} C_m = & C_{m_0,w} + \frac{x_w}{\bar{c}_w} (C_{L_0,w} + C_{L_\alpha,w}\alpha) \\ & + \frac{Q_h S_h x_h}{Q_\infty S_w \bar{c}_w} C_{L_\alpha,h} \left( \alpha - \left( \epsilon_0 + \frac{d\epsilon}{d\alpha} \alpha \right) \right) \end{aligned} \quad (3.122)$$

and finally grouping constant and  $\alpha$ -varying terms provides

$$C_m = \left[ C_{m_0,w} + \frac{x_w}{\bar{c}_w} (C_{L_0,w} - \frac{Q_h S_h x_h}{Q_\infty S_w \bar{c}_w} C_{L_\alpha,h} \epsilon_0) \right] + \left[ \frac{x_w}{\bar{c}_w} C_{L_\alpha,w} + \frac{Q_h S_h x_h}{Q_\infty S_w \bar{c}_w} C_{L_\alpha,h} \left( 1 - \frac{d\epsilon}{d\alpha} \right) \right] \alpha \quad (3.123)$$

and defining

$$\eta_h = \frac{Q_h}{Q_\infty} \quad (3.124)$$

as the **horizontal tail efficiency** and

$$V_h = \frac{-x_h S_h}{\bar{c}_w S_w} \quad (3.125)$$

as the **horizontal tail volume ratio**, one has

$$C_m = C_{m_0} + C_{m_\alpha} \alpha \quad (3.126)$$

where

$$C_{m_0} = C_{m_0,w} + \frac{x_w}{\bar{c}_w} (C_{L_0,w}) + \eta_h V_h C_{L_\alpha,h} \epsilon_0 \quad (3.127)$$

and

$$C_{m_\alpha} = \frac{x_w}{\bar{c}_w} C_{L_\alpha,w} - \eta_h V_h C_{L_\alpha,h} \left( 1 - \frac{d\epsilon}{d\alpha} \right) \quad (3.128)$$

For  $L$  and  $N$  moments, the coefficients are

$$C_l = C_{l,w-f} - \frac{Q_v S_v z_v}{Q_\infty S_w} C_{L,v} \quad (3.129)$$

and

$$C_n = C_{n,w-f} - \frac{Q_v S_v x_v}{Q_\infty S_w b_w} C_{L,v} \quad (3.130)$$

Next, modeling the moment and lift coefficients for small angles of attack as

$$C_{l,w-f} = C_{l_\beta,w-f} \beta \quad (3.131)$$

$$C_{n,w-f} = C_{n_\beta,w-f} \beta \quad (3.132)$$

and

$$C_{L,v} = C_{L_\alpha,v} \alpha_v \quad (3.133)$$

where these surfaces are symmetric with respect to  $\beta$  and  $\alpha_v$ , respectively, thus providing no net moment or force at  $0^\circ$ . Then, one has

$$C_l = C_{l_\beta,w-f} \beta - \frac{Q_v S_v z_v}{Q_\infty S_w b_w} C_{L_\alpha,v} \alpha_v \quad (3.134)$$

and

$$C_n = C_{n_\beta,w-f} \beta - \frac{Q_v S_v x_v}{Q_\infty S_w b_w} C_{L_\alpha,v} \alpha_v \quad (3.135)$$

Then, inspecting the FBD above, one can see that

$$\alpha_v = \beta + \sigma \quad (3.136)$$

where  $\sigma$  can be approximated by a linear function

$$\sigma = \frac{d\sigma}{d\beta} \beta \quad (3.137)$$

which is typically estimated using test data and/or CFD. Next, substituting for  $\alpha_v$  and  $\sigma$ , one has

$$C_l = C_{l_\beta,w-f} \beta - \frac{Q_v S_v z_v}{Q_\infty S_w b_w} C_{L_\alpha,v} \left( \beta + \frac{d\sigma}{d\beta} \beta \right) \quad (3.138)$$

and

$$C_n = C_{n_\beta,w-f} \beta - \frac{Q_v S_v x_v}{Q_\infty S_w b_w} C_{L_\alpha,v} \left( \beta + \frac{d\sigma}{d\beta} \beta \right) \quad (3.139)$$

Finally, defining

$$\eta_v = \frac{Q_v}{Q_\infty} \quad (3.140)$$

as the **vertical tail efficiency** and

$$V_v = \frac{-x_v S_v}{b_w S_w} \quad (3.141)$$

as the **vertical tail volume ratio**, one has

$$C_l = C_{l\beta}\beta \quad (3.142)$$

$$C_n = C_{n\beta}\beta \quad (3.143)$$

where

$$C_{l\beta} = C_{l\beta,w-f} - \eta_v \frac{S_v z_v}{S_w b_w} C_{L\alpha_v} \left( 1 + \frac{d\sigma}{d\beta} \right) \quad (3.144)$$

and

$$C_{n\beta} = C_{n\beta,w-f} + \eta_v V_v C_{L\alpha_v} \left( 1 + \frac{d\sigma}{d\beta} \right) \quad (3.145)$$

By the results above, the moment coefficients can be simply modeled as

$$\begin{bmatrix} C_l \\ C_m \\ C_n \end{bmatrix} = \begin{bmatrix} C_{l\beta}\beta \\ C_{m_0} + C_{m_\alpha}\alpha \\ C_{n\beta}\beta \end{bmatrix} \quad (3.146)$$

where  $C_{l\beta}$ ,  $C_{m_\alpha}$ , and  $C_{n\beta}$  are called the **static stability coefficients**. Steady flight requires that  $C_l = C_m = C_n = 0$  from which follows

$$\bar{\beta} = 0^\circ \quad (3.147)$$

and

$$\bar{\alpha} = -\frac{C_{m_0}}{C_{m_\alpha}} \quad (3.148)$$

### Static Stability

An initial analysis of an airplane's motion first assumes the airplane is nominally operating at some trimmed steady flight condition but is suddenly perturbed in the angles of attack and sideslip. To assess if the *initial* motion of the airplane is to return to the trim condition, i.e. **static stability** (as opposed to if it will eventually return to the trim condition, i.e. **dynamic stability**), one can consider the moment equations of motion

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} \dot{p} + \frac{I_z - I_y}{I_x} qr - \frac{I_{xz}}{I_x} (\dot{r} + pq) \\ \dot{q} + \frac{I_x - I_z}{I_y} pr + \frac{I_{xz}}{I_y} (p^2 - r^2) \\ \dot{r} + \frac{I_y - I_x}{I_z} pq + \frac{I_{xz}}{I_z} (qr - \dot{p}) \end{bmatrix} \quad (3.149)$$

which can be simplified by decoupling the  $\dot{p}$  and  $\dot{r}$  equations from the  $\dot{q}$ , an approximation which holds for small Euler angles. Then, one has

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} \approx \begin{bmatrix} \dot{p} - \frac{I_{xz}}{I_x} \dot{r} \\ \dot{q} \\ \dot{r} - \frac{I_{xz}}{I_z} \dot{p} \end{bmatrix} \quad (3.150)$$

Next, recall the small angle approximations for the Euler angle rates in steady flight

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} \approx \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (3.151)$$

and for the Euler angles

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \approx \begin{bmatrix} \mu \\ \gamma + \alpha \\ \sigma - \beta \end{bmatrix} \quad (3.152)$$

By assuming that  $\gamma$  and  $\sigma$  are held constant, one can approximate

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \approx \begin{bmatrix} \ddot{\mu} \\ \ddot{\alpha} \\ -\ddot{\beta} \end{bmatrix} \quad (3.153)$$

which results in

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} \approx \begin{bmatrix} \ddot{\mu} + \frac{I_{xz}}{I_x} \ddot{\beta} \\ \ddot{\alpha} \\ -\ddot{\beta} - \frac{I_{xz}}{I_z} \dot{\mu} \end{bmatrix} \quad (3.154)$$

which infers that the static stability of this trim point requires  $\dot{p}$ ,  $\dot{q}$  and  $\dot{r}$  be related to  $\alpha$  and  $\beta$  through the static stability coefficients. Substituting the static stability coefficient definitions for the moments, one has

$$\begin{bmatrix} \frac{Q_\infty S_w b_w}{I_{xx}} C_{l\beta} \beta \\ \frac{Q_\infty S_w \bar{c}_w}{I_{yy}} (C_{m_0} + C_{m_\alpha} \alpha) \\ \frac{Q_\infty S_w b_w}{I_{zz}} C_{n\beta} \beta \end{bmatrix} \approx \begin{bmatrix} \ddot{\mu} + \frac{I_{xz}}{I_x} \ddot{\beta} \\ \ddot{\alpha} \\ -\ddot{\beta} - \frac{I_{xz}}{I_z} \dot{\mu} \end{bmatrix} \quad (3.155)$$

Thus, for **airplane static stability**, if the airplane is perturbed from  $\bar{\alpha}$  by some  $\Delta\alpha$ , the resulting  $M$  moment should act in the opposite sense of  $\Delta\alpha$  to restore the airplane to  $\bar{\alpha}$ . Likewise, if the airplane is perturbed from  $\bar{\beta}$  by some  $\Delta\beta$ , the resulting  $L$  moment should act in the opposite sense as  $\Delta\beta$  to restore the airplane to  $\bar{\beta}$  while the resulting  $N$  moment should act in the same sense as  $\Delta\beta$  to restore the airplane to  $\bar{\beta}$ . This requirement can be separated into three different static stabilities, one for each axis, namely longitudinal, lateral, and directional.

An airplane has **longitudinal static stability** if  $C_{m_\alpha} < 0$  which depends on two design parameters: the wings' horizontal position and the size and position of the horizontal tail. To understand this requirement, consider the  $M$  stability coefficient

$$C_{m_\alpha} = \frac{x_w}{\bar{c}_w} C_{L_\alpha, w} - \eta_h V_h C_{L_\alpha, h} \left( 1 - \frac{d\epsilon}{d\alpha} \right) \quad (3.156)$$

which contains two terms which may be positive or negative and by inspection depends on the sign of  $x_w$  and  $x_h$ , the second of which appears in  $V_h$ . In particular, each term will be negative for negative values of  $x_w$  and  $x_h$ , i.e. the wing and tail aerodynamic centers are behind the center of gravity. Thus, most airplanes use a tail for their horizontal control surface to achieve longitudinal static stability. However, for high maneuverability airplanes may be designed as longitudinal statically unstable and thus, some use a canard. An alternative representation for this requirement for longitudinal static stability uses the **static margin**,  $SM$ , which is the normalized  $x_B$  coordinate for the aerodynamic center for the entire airplane, i.e. where one can resolve the total resultant lift of the airplane without a moment. Thus,  $SM$  is related to the static stability coefficient by

$$C_{m_\alpha} = -SM C_{L_\alpha} \quad (3.157)$$

and as  $C_{L\alpha} > 0$ , longitudinal static stability requires  $SM > 0$ , i.e. the total lift must resolve aft of the center of gravity to result in a negative moment. It should be noted that the length from the nose of the airplane to the location of the airplane aerodynamic center is also known as the **neutral point**,  $l_{np}$ , because if the center of gravity is located at this point on the airplane, the airplane will have a zero or “neutral” moment. To solve for the neutral point and static margin, consider rewriting  $x_w$  relative to the nose of the aircraft, i.e.

$$x_w = l_{cg} - l_w \quad (3.158)$$

where  $l_{cg}$  is the length from the nose to the center of gravity relative to the nose and  $l_w$  is the length from the nose to the aerodynamic center of the wing. Then,

$$C_{m\alpha} = \frac{l_{cg} - l_w}{\bar{c}_w} C_{L\alpha,w} - \eta_h V_h C_{L\alpha,h} \left( 1 - \frac{d\epsilon}{d\alpha} \right) \quad (3.159)$$

and  $l_{np}$  occurs for the value of  $l_{cg}$  where  $C_{m\alpha} = 0$ , i.e.

$$\frac{l_{np} - l_w}{\bar{c}_w} C_{L\alpha,w} - \eta_h V_h C_{L\alpha,h} \left( 1 - \frac{d\epsilon}{d\alpha} \right) = 0 \quad (3.160)$$

and solving, one has

$$l_{np} = l_w + \bar{c}_w \eta_h V_h \frac{C_{L\alpha,h}}{C_{L\alpha,w}} \left( 1 - \frac{d\epsilon}{d\alpha} \right) \quad (3.161)$$

Finally, by definition one can write the static margin as

$$SM = \frac{l_{np} - l_{cg}}{\bar{c}_w} \quad (3.162)$$

which is positive (i.e. statically stable) when the neutral point is aft of the center of gravity. Note that by rewriting this as

$$l_{np} = l_{cg} + \bar{c}_w SM \quad (3.163)$$

one has, by substitution

$$l_{cg} + \bar{c}_w SM = l_w + \bar{c}_w \eta_h V_h \frac{C_{L\alpha,h}}{C_{L\alpha,w}} \left( 1 - \frac{d\epsilon}{d\alpha} \right) \quad (3.164)$$

or rewriting in terms of  $SM$

$$SM = \frac{l_w - l_{cg}}{\bar{c}_w} + \eta_h V_h \frac{C_{L\alpha,h}}{C_{L\alpha,w}} \left( 1 - \frac{d\epsilon}{d\alpha} \right) \quad (3.165)$$

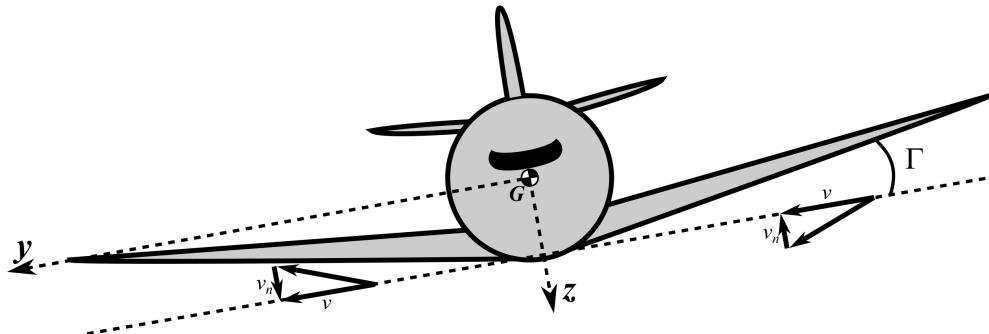
An airplane has **lateral static stability** if

1.  $C_l = 0$  at  $\beta = 0$
2.  $C_{l\beta} < 0$

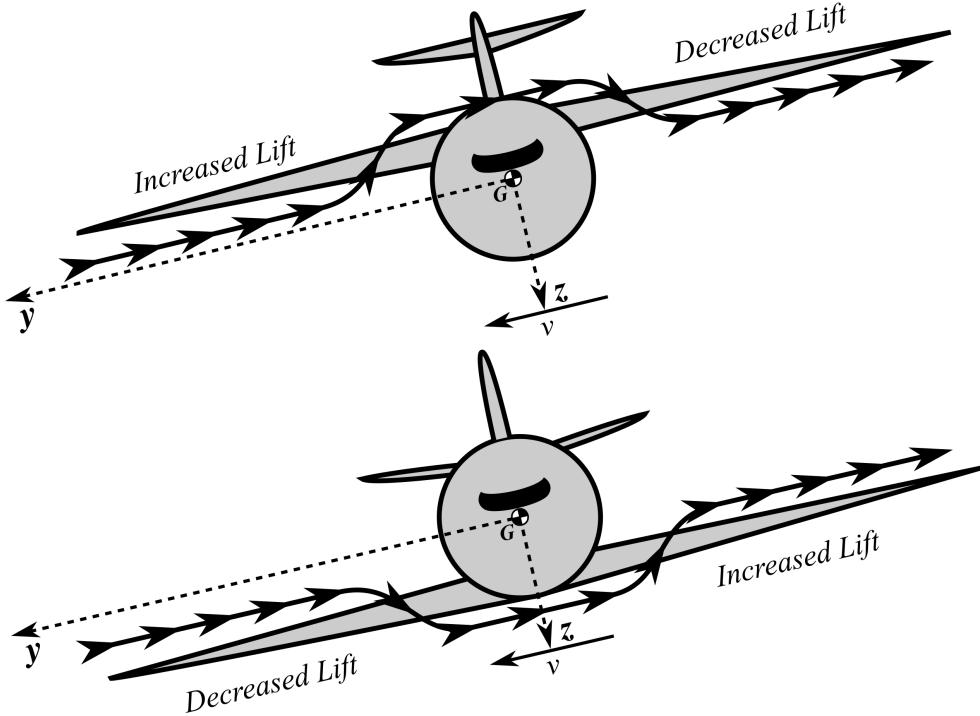
and **directional static stability** if

1.  $C_n = 0$  at  $\beta = 0$
2.  $C_{n\beta} > 0$

As structural symmetry will satisfy the conditions for  $\beta = 0$ , the lateral-directional stability depends on the two stability derivatives  $C_{l\beta}$  and  $C_{n\beta}$  also known as the **dihedral effect** and the **weathercock stability derivative**, respectively. These are influenced by four design parameters: the wing dihedral angle, the wing sweep angle, the wings' vertical position, and the size and position of the vertical tail. In particular, if the center of pressure of the vertical tail is above the center of gravity, the rolling moment due to the tail will be restorative and its effect will be increased as the size of the vertical tail is increased. It should be noted that these design parameters appear in the stability and control derivatives in the analytical models in the previous lecture. It should also be noted that the  $C_{l\beta}$  stability derivative is affected primarily by the **dihedral angle** of the wing,  $\Gamma_w$ . However, it is also influenced to a lesser extent by the wing sweep and vertical wing position. To understand these effects, consider a non-zero  $\beta$  which changes the local airflow angles at the wings as a function of its dihedral and sweep angles as well as forces the airflow to travel around the fuselage. It can be shown that these effects are stabilizing for a positive dihedral angle, swept-back wings, and by positioning the wings above the fuselage centerline. Conversely, the effect is destabilizing for a negative dihedral angle (a.k.a. **anhedral**), swept-forward wings, and positioning the wings below the fuselage centerline. These concepts are demonstrated in the following figures which use the fact that  $\beta v$  and that a sideslip change induces a roll to the airplane which should be restored to level.



where  $v_n$  is the normal component of the side velocity relative to the airfoil of the angled wings which increases and decreases the relative angle of attack and, thereby, lift for each wing depending on the dihedral angle.



Here the airplane movement creates a perturbed airflow around the fuselage which increases/decreases relative angle of attack and, thereby, lift for each wing depending on position.

### 3.4 Linearized Rigid Airplane Dynamics

Previous sections developed methods for calculating trimmed steady flight and static stability for airplanes. For analyzing the dynamic response of airplanes, in particular its dynamic stability, one employs Lyapunov stability theory to assess the linearized rigid airplane dynamics about a trimmed steady flight condition. To perform the Jacobian linearization of the dynamics, the airplane LTI state-space system representation will use states  $u$ ,  $\alpha$ ,  $\beta$ ,  $p$ ,  $q$ ,  $r$ ,  $\phi$ , and  $\theta$ , inputs  $\delta_a$ ,  $\delta_e$ ,  $\delta_r$ ,  $\delta_t$ , and aerodynamic and propulsive forces and moments  $X$ ,  $Y$ ,  $Z$ ,  $L$ ,  $M$ ,  $N$  expressed in trim and perturbation notation, i.e.

$$a = \bar{a} + \Delta a \quad (3.166)$$

where  $a$  is a state, input, force, or moment.

As a general airplane modeling principle, the perturbed aerodynamic and propulsive forces and moments are modeled as two sets

$$\begin{bmatrix} \Delta X \\ \Delta Z \\ \Delta M \end{bmatrix} = \begin{bmatrix} X_{\dot{u}} & X_{\dot{\alpha}} & X_{\dot{q}} \\ Z_{\dot{u}} & Z_{\dot{\alpha}} & Z_{\dot{q}} \\ M_{\dot{u}} & M_{\dot{\alpha}} & M_{\dot{q}} \end{bmatrix} \begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{\alpha} \\ \Delta \dot{q} \end{bmatrix} + \begin{bmatrix} X_u & X_\alpha & X_q \\ Z_u & Z_\alpha & Z_q \\ M_u & M_\alpha & M_q \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta q \end{bmatrix} + \begin{bmatrix} X_{\delta_e} & X_{\delta_t} \\ Z_{\delta_e} & Z_{\delta_t} \\ M_{\delta_e} & M_{\delta_t} \end{bmatrix} \begin{bmatrix} \Delta \delta_e \\ \Delta \delta_t \end{bmatrix} \quad (3.167)$$

and

$$\begin{bmatrix} \Delta Y \\ \Delta L \\ \Delta N \end{bmatrix} = \begin{bmatrix} Y_{\dot{\beta}} & Y_{\dot{p}} & Y_{\dot{r}} \\ L_{\dot{\beta}} & L_{\dot{p}} & L_{\dot{r}} \\ N_{\dot{\beta}} & N_{\dot{p}} & N_{\dot{r}} \end{bmatrix} \begin{bmatrix} \Delta \dot{\beta} \\ \Delta \dot{p} \\ \Delta \dot{r} \end{bmatrix} + \begin{bmatrix} Y_\beta & Y_p & Y_r \\ L_\beta & L_p & L_r \\ N_\beta & N_p & N_r \end{bmatrix} \begin{bmatrix} \Delta \beta \\ \Delta p \\ \Delta r \end{bmatrix} + \begin{bmatrix} Y_{\delta_a} & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{bmatrix} \begin{bmatrix} \Delta \delta_a \\ \Delta \delta_r \end{bmatrix} \quad (3.168)$$

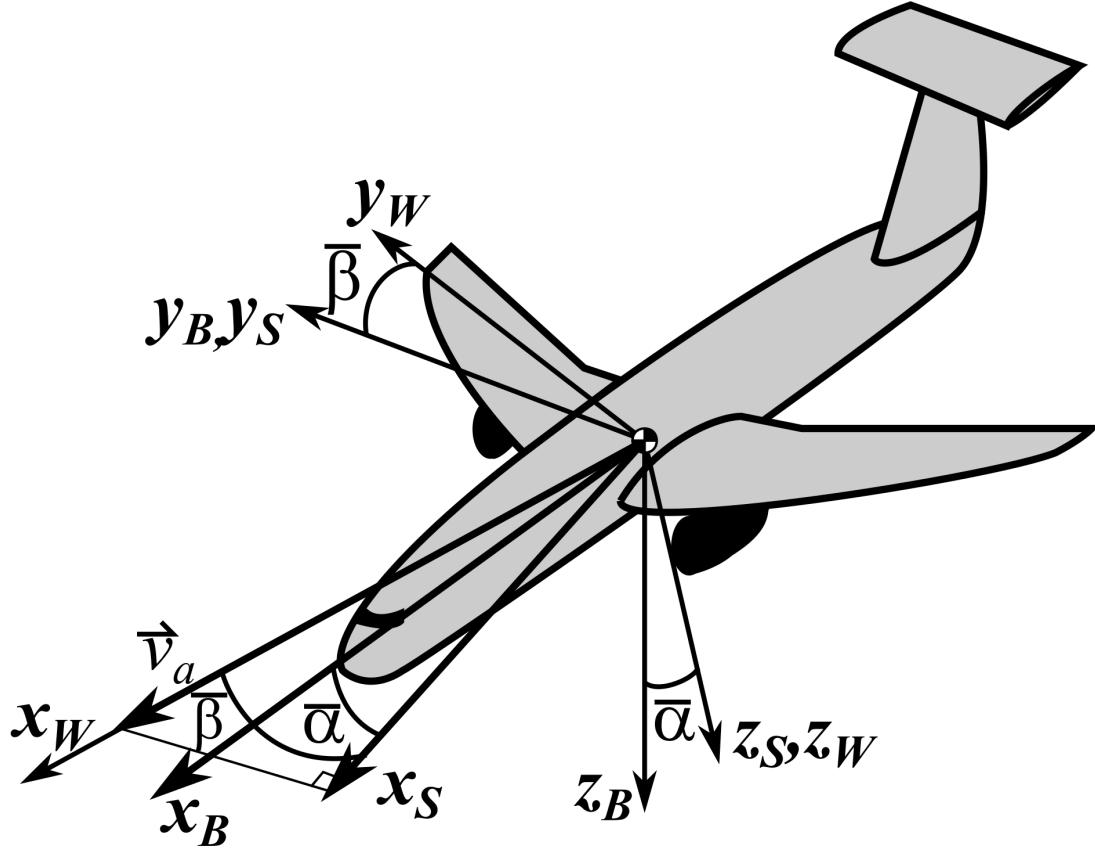
where the coefficients of the perturbed states and inputs inside these matrices are called the **stability and control derivatives** and correspond to the Jacobian partial derivative terms about trimmed steady flight. As these derivatives generally change with an airplane's trim conditions, one typically calculates tables of these derivatives at many steady flight conditions using wind tunnels tests, flight tests, and/or computational fluid dynamics (CFD). Methods for determining these derivatives from such data fall under the discipline of **aircraft system identification**, in particular, optimal parameter estimation, a topic addressed later in this textbook. In this part of the textbook, the following two sections will derive basic analytical models for the primary derivatives at subsonic, coordinated flight conditions using additive-component models based on traditional airplane design. As such, the following linearized dynamics derivation for airplanes will assume the following stability and control derivatives dominate the perturbed forces and moments

$$\begin{bmatrix} \Delta X \\ \Delta Z \\ \Delta M \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & M_{\dot{\alpha}} & 0 \end{bmatrix} \begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{\alpha} \\ \Delta \dot{q} \end{bmatrix} + \begin{bmatrix} X_u & X_\alpha & 0 \\ Z_u & Z_\alpha & 0 \\ M_u & M_\alpha & M_q \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta q \end{bmatrix} + \begin{bmatrix} 0 & X_{\delta_t} \\ Z_{\delta_e} & Z_{\delta_t} \\ M_{\delta_e} & M_{\delta_t} \end{bmatrix} \begin{bmatrix} \Delta \delta_e \\ \Delta \delta_t \end{bmatrix} \quad (3.169)$$

and

$$\begin{bmatrix} \Delta Y \\ \Delta L \\ \Delta N \end{bmatrix} = \begin{bmatrix} Y_\beta & Y_p & Y_r \\ L_\beta & L_p & L_r \\ N_\beta & N_p & N_r \end{bmatrix} \begin{bmatrix} \Delta \beta \\ \Delta p \\ \Delta r \end{bmatrix} + \begin{bmatrix} 0 & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{bmatrix} \begin{bmatrix} \Delta \delta_a \\ \Delta \delta_r \end{bmatrix} \quad (3.170)$$

In addition, recall that previous analysis defined the aerodynamic and propulsive forces in a body frame whose  $x_B$ -axis was aligned with a fuselage reference line, e.g. the centerline, also known as a **fuselage frame**. However, for the linearized rigid airplane dynamics, it is common to use an alternative body-fixed frame known as the **stability frame** (subscript  $S$ ). In particular, the stability frame is related to the fuselage frame through a rotation of  $\bar{\alpha}$  about the  $y_B$ -axis as shown in the following figure.



Note that by being defined by  $\bar{\alpha}$ , *different* stability frames are defined for *different* steady flight conditions. However, the stability frame rotates with the airplane body frame as the perturbed states vary, not remaining fixed to the free-stream velocity vector as for the wind frame. It is important to note that here the force and moment vector elements,  $X$ ,  $Y$ ,  $Z$ ,  $L$ ,  $M$ , and  $N$ , are now defined in the stability frame, but can be related to the fuselage-fixed body frame using a frame rotation matrix, i.e.

$$\vec{v}_B = C_2(\bar{\alpha}) \vec{v}_S \quad (3.171)$$

where  $\vec{v}$  is some vector. In addition, the **inertia matrix in the stability frame**,  $I_S$ , would be related to the inertia matrix in fuselage frame,  $I_B$ , can be transformed using the formula

$$I_B = C_2(\bar{\alpha}) I_S C_2^T(\bar{\alpha}) \quad (3.172)$$

### Airplane Linearized Dynamics about Straight Flight

If one assumes straight flight in the stability frame,  $\bar{\beta} = \bar{\alpha} = 0$  and  $\bar{u} = \vec{v}_a$  which greatly simplifies the linearized rigid airplane dynamics that follow. In particular, one can decouple the dynamics into longitudinal and lateral-directional dynamics which will be assumed here.

First, recall the following form for the 6-DOF airplane EOMs

$$\begin{bmatrix} X - g \sin \theta \\ Y + g \cos \theta \sin \phi \\ Z + g \cos \theta \cos \phi \\ L \\ M \\ N \end{bmatrix} = \begin{bmatrix} \dot{u} + qu \sin \alpha - ru \tan \beta \\ \dot{u} \tan \beta + \dot{\beta} u \sec^2 \beta + ru - pu \sin \alpha \\ \dot{u} \sin \alpha + \dot{\alpha} u \cos \alpha + pu \tan \beta - qu \\ \dot{p} + \frac{I_{zz} - I_{yy}}{I_{xx}} qr - \frac{I_{xz}}{I_{xx}} (\dot{r} + pq) \\ \dot{q} + \frac{I_{xx} - I_{zz}}{I_{yy}} pr - \frac{I_{xz}}{I_{yy}} (r^2 - p^2) \\ \dot{r} + \frac{I_{yy} - I_{xx}}{I_{zz}} pq - \frac{I_{xz}}{I_{zz}} (\dot{p} - qr) \end{bmatrix} \quad (3.173)$$

Next, for coordinated flight, one can decouple these into the linearized longitudinal and lateral-directional EOMs using the trim and perturbed states, forces, and moments as

$$\begin{bmatrix} \bar{X} + \Delta X - g \sin (\bar{\theta} + \Delta \theta) \\ \bar{Z} + \Delta Z + g \sin \bar{\phi} \cos (\bar{\theta} + \Delta \theta) \\ \bar{M} + \Delta M \end{bmatrix} = \begin{bmatrix} \Delta \dot{u} + \Delta q (\bar{u} + \Delta u) \sin \Delta \alpha \\ \Delta \dot{u} \sin \Delta \alpha + \Delta \dot{\alpha} (\bar{u} + \Delta u) \cos \Delta \alpha - \Delta q (\bar{u} + \Delta u) \\ \Delta \dot{q} \end{bmatrix} \quad (3.174)$$

and

$$\begin{bmatrix} \bar{Y} + \Delta Y + g \cos \bar{\theta} \sin (\bar{\phi} + \Delta \phi) \\ \bar{L} + \Delta L \\ \bar{N} + \Delta N \end{bmatrix} = \begin{bmatrix} \Delta \dot{\beta} \bar{u} \sec^2 \Delta \beta + \bar{u} (\bar{r} + \Delta r) \\ \Delta \dot{p} - \frac{I_{xz}}{I_{xx}} \Delta \dot{r} \\ \Delta \dot{r} - \frac{I_{xz}}{I_{zz}} \Delta \dot{p} \end{bmatrix} \quad (3.175)$$

Then, using the trigonometric addition formulas and the small angle approximation, i.e.

$$\sin(\bar{a} + \Delta a) = \sin \bar{a} \cos \Delta a + \cos \bar{a} \sin \Delta a = \sin \bar{a} + \cos \bar{a} \Delta a \quad (3.176)$$

and

$$\cos(\bar{a} + \Delta a) = \cos \bar{a} \cos \Delta a - \sin \bar{a} \sin \Delta a = \cos \bar{a} - \sin \bar{a} \Delta a \quad (3.177)$$

for  $\theta$ ,  $\phi$ ,  $\alpha$ , and  $\beta$  where  $\bar{\alpha} = \bar{\beta} = 0$  has already been assumed, one has

$$\begin{bmatrix} \bar{X} + \Delta X - g \sin \bar{\theta} - g \cos \bar{\theta} \Delta \theta \\ \bar{Z} + \Delta Z + g \sin \bar{\phi} \cos \bar{\theta} - g \sin \bar{\phi} \sin \bar{\theta} \Delta \theta \\ \bar{M} + \Delta M \end{bmatrix} = \begin{bmatrix} \Delta \dot{u} + \Delta q (\bar{u} + \Delta u) \Delta \alpha \\ \Delta \dot{u} \Delta \alpha + \Delta \dot{\alpha} (\bar{u} + \Delta u) - \Delta q (\bar{u} + \Delta u) \\ \Delta \dot{q} \end{bmatrix} \quad (3.178)$$

and

$$\begin{bmatrix} \bar{Y} + \Delta Y + g \cos \bar{\theta} (\sin \bar{\phi} - \cos \bar{\phi} \Delta \phi) \\ \bar{L} + \Delta L \\ \bar{N} + \Delta N \end{bmatrix} = \begin{bmatrix} \Delta \dot{\beta} \bar{u} + \Delta r \bar{u} \\ \Delta \dot{p} - \frac{I_{xz}}{I_{xx}} \Delta \dot{r} \\ \Delta \dot{r} - \frac{I_{xz}}{I_{zz}} \Delta \dot{p} \end{bmatrix} \quad (3.179)$$

which can be further linearized by eliminating higher order terms of the perturbations and separating out the perturbation terms, i.e.

$$\begin{bmatrix} \bar{X} - g \sin \bar{\theta} \\ \bar{Z} + g \sin \bar{\phi} \cos \bar{\theta} \\ \bar{M} \end{bmatrix} + \begin{bmatrix} \Delta X \\ \Delta Z \\ \Delta M \end{bmatrix} + \begin{bmatrix} -g \cos \bar{\theta} \\ -g \sin \bar{\phi} \sin \bar{\theta} \\ 0 \end{bmatrix} \Delta \theta = \begin{bmatrix} \Delta \dot{u} \\ \bar{u} \Delta \dot{\alpha} - \bar{u} \Delta q \\ \Delta \dot{q} \end{bmatrix} \quad (3.180)$$

and

$$\begin{bmatrix} \bar{Y} + g \cos \bar{\theta} \sin \bar{\phi} \\ \bar{L} \\ \bar{N} \end{bmatrix} + \begin{bmatrix} \Delta Y \\ \Delta L \\ \Delta N \end{bmatrix} + \begin{bmatrix} -g \cos \bar{\theta} \cos \bar{\phi} \\ 0 \\ 0 \end{bmatrix} \Delta \phi = \begin{bmatrix} \Delta \dot{\beta} \bar{u} + \Delta r \bar{u} \\ \Delta \dot{p} - \frac{I_{xz}}{I_{xx}} \Delta \dot{r} \\ \Delta \dot{r} - \frac{I_{xz}}{I_{zz}} \Delta \dot{p} \end{bmatrix} \quad (3.181)$$

Next, by definition of steady flight as the conditions for which

$$\begin{aligned} \bar{X} - g \sin \bar{\theta} &= 0 \\ \bar{Y} + g \sin \bar{\phi} \cos \bar{\theta} &= 0 \\ \bar{Z} + g \sin \bar{\phi} \cos \bar{\theta} &= 0 \\ \bar{L} &= 0 \\ \bar{M} &= 0 \\ \bar{N} &= 0 \end{aligned} \quad (3.182)$$

due to the zero angular velocities, the linearized models can be rewritten using matrices and the states as

$$\begin{bmatrix} \Delta X \\ \Delta Z \\ \Delta M \end{bmatrix} + \begin{bmatrix} -g \cos \bar{\theta} \\ -g \sin \bar{\phi} \sin \bar{\theta} \\ 0 \end{bmatrix} \Delta \theta + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \bar{u} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta q \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \bar{u} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{\alpha} \\ \Delta \dot{q} \end{bmatrix} \quad (3.183)$$

and

$$\begin{bmatrix} \Delta Y \\ \Delta L \\ \Delta N \end{bmatrix} + \begin{bmatrix} -g \cos \bar{\theta} \cos \bar{\phi} \\ 0 \\ 0 \end{bmatrix} \Delta \phi - \begin{bmatrix} 0 & 0 & \bar{u} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \beta \\ \Delta p \\ \Delta r \end{bmatrix} = \begin{bmatrix} \bar{u} & 0 & 0 \\ 0 & 1 & -\frac{I_{xz}}{I_{xx}} \\ 0 & -\frac{I_{xz}}{I_{xx}} & 1 \end{bmatrix} \begin{bmatrix} \Delta \dot{\beta} \\ \Delta \dot{p} \\ \Delta \dot{r} \end{bmatrix} \quad (3.184)$$

and substituting for the perturbed aerodynamic and propulsive forces and moments as assumed in this course, one has

$$\begin{aligned} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & M_{\dot{\alpha}} & 0 \end{bmatrix} \begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{\alpha} \\ \Delta \dot{q} \end{bmatrix} + \begin{bmatrix} X_u & X_{\alpha} & 0 \\ Z_u & Z_{\alpha} & 0 \\ M_u & M_{\alpha} & M_q \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta q \end{bmatrix} + \begin{bmatrix} 0 & X_{\delta_t} \\ Z_{\delta_e} & Z_{\delta_t} \\ M_{\delta_e} & M_{\delta_t} \end{bmatrix} \begin{bmatrix} \Delta \delta_e \\ \Delta \delta_t \\ \Delta \delta_r \end{bmatrix} \\ + \begin{bmatrix} -g \cos \bar{\theta} \\ -g \sin \bar{\phi} \sin \bar{\theta} \\ 0 \end{bmatrix} \Delta \theta + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \bar{u} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta q \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \bar{u} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{\alpha} \\ \Delta \dot{q} \end{bmatrix} \end{aligned} \quad (3.185)$$

and

$$\begin{aligned} \begin{bmatrix} Y_{\beta} & Y_p & Y_r \\ L_{\beta} & L_p & L_r \\ N_{\beta} & N_p & N_r \end{bmatrix} \begin{bmatrix} \Delta \beta \\ \Delta p \\ \Delta r \end{bmatrix} + \begin{bmatrix} 0 & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{bmatrix} \begin{bmatrix} \Delta \delta_a \\ \Delta \delta_r \\ \Delta \delta_t \end{bmatrix} \\ + \begin{bmatrix} -g \cos \bar{\theta} \cos \bar{\phi} \\ 0 \\ 0 \end{bmatrix} \Delta \phi - \begin{bmatrix} 0 & 0 & \bar{u} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \beta \\ \Delta p \\ \Delta r \end{bmatrix} = \begin{bmatrix} \bar{u} & 0 & 0 \\ 0 & 1 & -\frac{I_{xz}}{I_{xx}} \\ 0 & -\frac{I_{xz}}{I_{xx}} & 1 \end{bmatrix} \begin{bmatrix} \Delta \dot{\beta} \\ \Delta \dot{p} \\ \Delta \dot{r} \end{bmatrix} \end{aligned} \quad (3.186)$$

Next, combining matrices of similar terms and reversing the equations, one has

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \bar{u} & 0 \\ 0 & -M_{\dot{\alpha}} & 1 \end{bmatrix} \begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{\alpha} \\ \Delta \dot{q} \end{bmatrix} = \begin{bmatrix} X_u & X_\alpha & 0 & -g \cos \bar{\theta} \\ Z_u & Z_\alpha & \bar{u} & -g \sin \bar{\phi} \sin \bar{\theta} \\ M_u & M_\alpha & M_q & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta q \\ \Delta \theta \end{bmatrix} + \begin{bmatrix} 0 & X_{\delta_t} \\ Z_{\delta_e} & Z_{\delta_t} \\ M_{\delta_e} & M_{\delta_t} \end{bmatrix} \begin{bmatrix} \Delta \delta_e \\ \Delta \delta_t \end{bmatrix} \quad (3.187)$$

and

$$\begin{bmatrix} \bar{u} & 0 & 0 \\ 0 & 1 & -\frac{I_{xz}}{I_{xx}} \\ 0 & -\frac{I_{xz}}{I_{xx}} & 1 \end{bmatrix} \begin{bmatrix} \Delta \dot{\beta} \\ \Delta \dot{p} \\ \Delta \dot{r} \end{bmatrix} = \begin{bmatrix} Y_\beta & Y_p & Y_r - \bar{u} & -g \cos \bar{\theta} \cos \bar{\phi} \\ L_\beta & L_p & L_r & 0 \\ N_\beta & N_p & N_r & 0 \end{bmatrix} \begin{bmatrix} \Delta \beta \\ \Delta p \\ \Delta r \\ \Delta \phi \end{bmatrix} + \begin{bmatrix} 0 & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{bmatrix} \begin{bmatrix} \Delta \delta_a \\ \Delta \delta_r \end{bmatrix} \quad (3.188)$$

Then, recalling for small angles

$$\Delta \dot{\theta} = \Delta q \quad (3.189)$$

and

$$\Delta \dot{\phi} = \Delta p \quad (3.190)$$

one has

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \bar{u} & 0 & 0 \\ 0 & -M_{\dot{\alpha}} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{\alpha} \\ \Delta \dot{q} \\ \Delta \dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_\alpha & 0 & -g \cos \bar{\theta} \\ Z_u & Z_\alpha & \bar{u} & -g \sin \bar{\phi} \sin \bar{\theta} \\ M_u & M_\alpha & M_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta q \\ \Delta \theta \end{bmatrix} + \begin{bmatrix} 0 & X_{\delta_t} \\ Z_{\delta_e} & Z_{\delta_t} \\ M_{\delta_e} & M_{\delta_t} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \delta_e \\ \Delta \delta_t \end{bmatrix} \quad (3.191)$$

and

$$\begin{bmatrix} \bar{u} & 0 & 0 & 0 \\ 0 & 1 & -\frac{I_{xz}}{I_{xx}} & 0 \\ 0 & -\frac{I_{xz}}{I_{xx}} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta \dot{\beta} \\ \Delta \dot{p} \\ \Delta \dot{r} \\ \Delta \dot{\phi} \end{bmatrix} = \begin{bmatrix} Y_\beta & Y_p & Y_r - \bar{u} & -g \cos \bar{\theta} \cos \bar{\phi} \\ L_\beta & L_p & L_r & 0 \\ N_\beta & N_p & N_r & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta \beta \\ \Delta p \\ \Delta r \\ \Delta \phi \end{bmatrix} + \begin{bmatrix} 0 & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \delta_a \\ \Delta \delta_r \end{bmatrix} \quad (3.192)$$

Finally, by normalizing by the coefficients on the time derivative terms, one has the **linearized longitudinal dynamics** or EOMs

$$\begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{\alpha} \\ \Delta \dot{q} \\ \Delta \dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_\alpha & 0 & -g \cos \bar{\theta} \\ \frac{Z_u}{\bar{u}} & \frac{Z_\alpha}{\bar{u}} & 1 & -\frac{g}{\bar{u}} \sin \bar{\theta} \cos \bar{\phi} \\ M_u + M_{\dot{\alpha}} \frac{Z_u}{\bar{u}} & M_\alpha + M_{\dot{\alpha}} \frac{Z_\alpha}{\bar{u}} & M_q + M_{\dot{\alpha}} & -M_{\dot{\alpha}} \frac{g}{\bar{u}} \sin \bar{\theta} \cos \bar{\phi} \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta q \\ \Delta \theta \end{bmatrix} + \begin{bmatrix} 0 & X_{\delta_t} \\ \frac{Z_{\delta_e}}{\bar{u}} & \frac{Z_{\delta_t}}{\bar{u}} \\ M_{\delta_e} + M_{\dot{\alpha}} \frac{Z_{\delta_e}}{\bar{u}} & M_{\delta_t} + M_{\dot{\alpha}} \frac{Z_{\delta_t}}{\bar{u}} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \delta_e \\ \Delta \delta_t \end{bmatrix} \quad (3.193)$$

where the  $M_\alpha$  term in the  $\Delta\dot{q}$  equation occurs through back-substitution of  $\Delta\dot{\alpha}$  and one has the **linearized lateral-directional dynamics** or EOMs

$$\begin{bmatrix} \Delta\dot{\beta} \\ \Delta\dot{p} \\ \Delta\dot{r} \\ \Delta\dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{Y_\beta}{\bar{u}} & \frac{Y_p}{\bar{u}} & \frac{Y_r}{\bar{u}} - 1 & \frac{g}{\bar{u}} \cos \bar{\theta} \cos \bar{\phi} \\ L_\beta^* + \frac{I_{xz}}{I_{xx}} N_\beta^* & L_p^* + \frac{I_{xz}}{I_{xx}} N_p^* & L_r^* + \frac{I_{xz}}{I_{xx}} N_r^* & 0 \\ N_\beta^* + \frac{I_{xz}}{I_{zz}} L_\beta^* & N_p^* + \frac{I_{xz}}{I_{zz}} L_p^* & N_r^* + \frac{I_{xz}}{I_{zz}} L_r^* & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\beta \\ \Delta p \\ \Delta r \\ \Delta\phi \end{bmatrix} + \begin{bmatrix} 0 & \frac{Y_{\delta_r}}{\bar{u}} \\ L_{\delta_a}^* + \frac{I_{xz}}{I_{xx}} N_{\delta_a}^* & L_{\delta_r}^* + \frac{I_{xz}}{I_{xx}} N_{\delta_r}^* \\ N_{\delta_a}^* + \frac{I_{xz}}{I_{zz}} L_{\delta_a}^* & N_{\delta_r}^* + \frac{I_{xz}}{I_{zz}} L_{\delta_r}^* \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\delta_a \\ \Delta\delta_r \end{bmatrix} \quad (3.194)$$

where

$$L_a^* = \frac{L_a}{1 - \frac{I_{xz}^2}{I_{xx} I_{zz}}} \quad (3.195)$$

and

$$N_a^* = \frac{N_a}{1 - \frac{I_{xz}^2}{I_{xx} I_{zz}}} \quad (3.196)$$

for  $a = \beta, p, r, \delta_a$ , and  $\delta_r$  due to the coupling of  $L$  and  $N$  for non-zero  $I_{xz}$ .

First, note that this derivation results in two sets of fourth order LTI state-space systems. Secondly, it should be noted that one may also form these EOMs using the  $v$  and  $w$  states instead of  $\beta$  and  $\alpha$ , respectively. Third, it should also be noted that often these equations are extended to fifth order systems by including the perturbed altitude,  $\Delta h$ , as a perturbed longitudinal state and the perturbed yaw,  $\Delta\psi$ , as a perturbed lateral-directional state which are simply related to the other states for coordinated steady flight by

$$\Delta\dot{h} = \bar{u} (\Delta\theta - \Delta\alpha) \quad (3.197)$$

and

$$\Delta\dot{\psi} = \Delta r \quad (3.198)$$

and do not have any direct control derivatives. Lastly, note that these state-space systems do not have an explicit output equation whose form will depend on which states should be used as “outputs,” e.g. in a control system.

### Example Problem 1

Given: Analysis of an aerial fire fighter has provided the following preliminary information.

$C_{m_0} = 0.05$	$C_{m_{\delta_e}} = -0.01/^\circ$	$l_{cg} = 6/2$ ft when full/empty	$l_{np} = 8$ ft
$C_{L_\alpha} = 0.1/^\circ$	$C_{L_0} = 0.2$	sea level $\rho = 0.002378$ slug/ft <sup>3</sup>	$ (\delta_e)_{max}  = 20$
$\bar{c}_w = 20$ ft	$b_w = 80$ ft	$W = 120,000/100,000$ lb when full/empty	

Determine:

- (a) the static margin at full and empty water tank
- (b)  $\bar{\delta}_e$  at  $\bar{\alpha} = 10^\circ$  with full tank
- (c)  $v_{\infty,min}$  for trimmed straight-and-level flight with empty tank

Assume:

1. Fuselage and thrust contributions to  $C_m$  are negligible
2. Rectangular wing

Solution:

- a) The equation for the static margin is

$$SM = \frac{l_{np} - l_{cg}}{\bar{c}_w} \quad (3.199)$$

$$SM_{full} = \frac{8 - 6}{20} = \underline{0.1} \quad (3.200)$$

$$SM_{empty} = \frac{8 - 2}{20} = \underline{0.3} \quad (3.201)$$

- b) For trim,  $C_m = 0$

$$C_{m_0} + C_{m_\alpha} \bar{\alpha} + C_{m_{\delta_e}} \bar{\delta}_e = 0 \quad (3.202)$$

$$C_{m_0} - SM_{full} C_{L_{\alpha,w}} \bar{\alpha} + C_{m_{\delta_e}} \bar{\delta}_e = 0 \quad (3.203)$$

$$\bar{\delta}_e = \frac{-C_{m_0} + SM_{full} C_{L_{\alpha,w}} \bar{\alpha}}{C_{m_{\delta_e}}} \quad (3.204)$$

$$\bar{\delta}_e = \frac{-0.05 + 0.1(0.1)(10)}{-0.01} \quad (3.205)$$

$$\underline{\bar{\delta}_e = -5^\circ} \quad (3.206)$$

- c) For trim at straight-and-level flight,

$$W = L \quad (3.207)$$

$$W = \frac{1}{2} \rho v_\infty^2 S_w C_L \quad (3.208)$$

and setting  $S_w = \bar{c} b_w$ ,

$$v_\infty = \sqrt{\frac{2W}{\rho \bar{c} b} \frac{1}{C_L}} \quad (3.209)$$

$$v_{\infty,min} = \sqrt{\frac{2W}{\rho \bar{c} b} \frac{1}{C_{L,max}}} \quad (3.210)$$

$C_{L,max}$  for a trimmed airplane occurs when  $\delta_e = -|(\delta_e)_{max}|$

$$0 = C_{m_0} - SM_{empty} C_{L_\alpha} \bar{\alpha} + C_{m_{\delta_e}} (-|(\delta_e)_{max}|) \quad (3.211)$$

and since  $C_{L,max} = C_{L_0} + C_{L_\alpha} \bar{\alpha}$

$$0 = C_{m_0} - SM_{empty}(C_{L,max} - C_{L_0}) + C_{m_{\delta_e}}(-|(\delta_e)_{max}|) = 0 \quad (3.212)$$

or

$$C_{L,max} = \frac{C_{m_0} + C_{m_{\delta_e}}(-|(\delta_e)_{max}|)}{-SM_{empty}} + C_{L_0} \quad (3.213)$$

So,

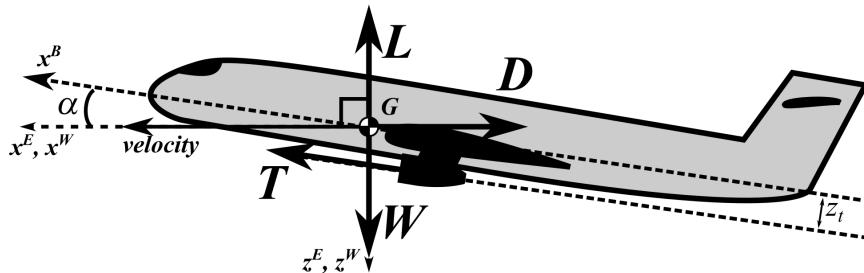
$$v_{\infty,min} = \sqrt{\frac{2W}{\rho \bar{c} b} \left( \frac{1}{\frac{C_{m_0} + C_{m_{\delta_e}}(-|(\delta_e)_{max}|)}{-SM_{empty}} + C_{L_0}} \right)} \quad (3.214)$$

$$v_{\infty,min} = \sqrt{\frac{2(100,000)}{0.002378(20)(80)} \left( \frac{1}{\frac{0.05+0.01(-20)}{0.3} + 0.2} \right)} \quad (3.215)$$

$$\underline{v_{\infty,min} = 225 \text{ ft/s} = 153 \text{ mph}} \quad (3.216)$$

### Example Problem 2

Given: an airplane weighing 500,000 kg with off-center engines is flying at sea level ( $\rho = 1.347 \text{ kg/m}^3$ ) at a velocity of 200 m/s.



The following data has been identified.

$C_{m_\alpha} = -0.004/\text{°}$	$C_{m_{\delta_e}} = -0.007/\text{°}$	$C_{m_{\delta_t}} = 10^{-8}/\text{N}$	$C_{m_0} = 0.001$
$C_{L_0} = 0.02$	$S_w = 500 \text{ m}^2$	$\bar{c}_w = 10 \text{ m}$	

Determine:

- (a)  $\bar{\delta}_e$  for trim at  $\bar{\alpha} = 5^\circ$  and  $\bar{T} = \bar{\delta}_t = 500,000 \text{ N}$
- (b)  $\bar{L}$  (lift) for trim at  $\bar{\alpha} = 5^\circ$  and  $\bar{T} = \bar{\delta}_t = 500,000 \text{ N}$
- (c)  $\bar{C}_{L_\alpha}$
- (d) the static margin,  $SM$

Assume:

1.  $T = \delta_t$  already accounts for both engines
2.  $C_{L_{\alpha,w}} = C_{L_\alpha}$

**Solution:**

(a) The equation for the pitching moment can be written as

$$C_m = C_{m_0} + C_{m_\alpha} \alpha + C_{m_{\delta_e}} \bar{\delta}_e + C_{m_{\delta_t}} \bar{\delta}_t \quad (3.217)$$

At trim,  $C_m = 0$ , thus

$$0 = C_{m_0} + C_{m_\alpha} \bar{\alpha} + C_{m_{\delta_e}} \bar{\delta}_e + C_{m_{\delta_t}} \bar{\delta}_t \quad (3.218)$$

Rearranging for  $\bar{\delta}_e$

$$\bar{\delta}_e = -\frac{C_{m_0} + C_{m_\alpha} \bar{\alpha} + C_{m_{\delta_t}} \bar{\delta}_t}{C_{m_{\delta_e}}} \quad (3.219)$$

Substituting values provides

$$\bar{\delta}_e = -\frac{0.001 - 0.004(5) + (10^{-8})(500,000)}{-0.007} \quad (3.220)$$

$$\underline{\bar{\delta}_e = -2^\circ} \quad (3.221)$$

(b) For trim,

$$\bar{L} - W + \bar{T} \sin \bar{\alpha} = 0 \quad (3.222)$$

Rearranging and substituting for the weight

$$\bar{L} = mg - \bar{T} \sin \bar{\alpha} \quad (3.223)$$

Substituting values,

$$\bar{L} = 500,000(9.81) - 500,000 \sin 10 \quad (3.224)$$

$$\underline{\bar{L} = 1,818,176 \text{ N}} \quad (3.225)$$

(c) The lift coefficient is given by

$$\bar{C}_L = \frac{\bar{L}}{QS_w} \quad (3.226)$$

and

$$\bar{C}_L = C_{L_\alpha} \bar{\alpha} + C_{L_0} \quad (3.227)$$

Combining,

$$\frac{\bar{L}}{QS_w} = C_{L_\alpha} \bar{\alpha} + C_{L_0} \quad (3.228)$$

Rearranging,

$$\bar{C}_{L_\alpha} = \frac{\frac{\bar{L}}{QS_w} - C_{L_0}}{\bar{\alpha}} \quad (3.229)$$

Substituting values

$$\bar{C}_{L_\alpha} = \frac{\frac{1818176}{5(1.347)(200)^2(500)} - 0.02}{5} \quad (3.230)$$

$$\bar{C}_{L_\alpha} = 0.0675 / {}^\circ \quad (3.231)$$

(d) The static margin,  $SM$  is given by

$$C_{m_\alpha} = -SM C_{L_\alpha} \quad (3.232)$$

Rearranging,

$$SM = \frac{-C_{m_\alpha}}{C_{L_\alpha}} \quad (3.233)$$

$$SM = \frac{0.004}{0.0675} \quad (3.234)$$

$$\underline{SM = 5.92\%} \quad (3.235)$$

## 3.5 Longitudinal Stability and Control Derivatives

This section will derive models for the primary longitudinal stability and control derivatives used in the Jacobian linearization of the aerodynamic and propulsive normalized forces and moments about trimmed steady flight, i.e.

$$\begin{aligned} X &= \bar{X} + X_u \Delta u + X_\alpha \Delta \alpha + X_{\delta_t} \Delta \delta_t \\ Z &= \bar{Z} + Z_u \Delta u + Z_\alpha \Delta \alpha + Z_q \Delta q + Z_{\dot{\alpha}} \Delta \dot{\alpha} + Z_{\delta_e} \Delta \delta_e + Z_{\delta_t} \Delta \delta_t \\ M &= \bar{M} + M_u \Delta u + M_\alpha \Delta \alpha + M_q \Delta q + M_{\dot{\alpha}} \Delta \dot{\alpha} + M_{\delta_e} \Delta \delta_e + M_{\delta_t} \Delta \delta_t \end{aligned} \quad (3.236)$$

where the coefficients of the linear terms are called the **longitudinal stability and control derivatives**. Alternatively, these equations can be written in terms of nondimensional coefficients as

$$\begin{aligned} C_X &= \bar{C}_X + C_{X_u} \Delta u + C_{X_\alpha} \Delta \alpha + C_{X_{\delta_t}} \Delta \delta_t \\ C_Z &= \bar{C}_Z + C_{Z_u} \Delta u + C_{Z_\alpha} \Delta \alpha + C_{Z_q} \Delta q + C_{Z_{\dot{\alpha}}} \Delta \dot{\alpha} + C_{Z_{\delta_e}} \Delta \delta_e + C_{Z_{\delta_t}} \Delta \delta_t \\ C_m &= \bar{C}_m + C_{m_u} \Delta u + C_{m_\alpha} \Delta \alpha + C_{m_q} \Delta q + C_{m_{\dot{\alpha}}} \Delta \dot{\alpha} + C_{m_{\delta_e}} \Delta \delta_e + C_{m_{\delta_t}} \Delta \delta_t \end{aligned} \quad (3.237)$$

where the coefficients of the linear terms are called the **longitudinal stability and control coefficients**.

It should be noted that a more comprehensive collection of aerodynamic stability and control prediction techniques can be found in the *USAF Stability and Control DATCOM (Data Compendium)* which was compiled between 1960 and 1978 by the McDonnell Douglas Corporation in conjunction with the Flight Dynamics Laboratory at Wright-Patterson Air Force Base.

### *u* Stability Derivatives and Coefficients

For  $X$ , consider the dominant drag and thrust forces for steady flight, i.e.

$$mX = -D + T \quad (3.238)$$

or

$$mX = -Q_w S_w C_D + T \quad (3.239)$$

and taking the derivative with respect to  $u$

$$mX_u = -Q_w S_w \left( \frac{2\bar{C}_D}{u} + C_{D_u} \right) + T_u \quad (3.240)$$

and defining the coefficient

$$X_u = \frac{Q_w S_w}{m\bar{u}} C_{X_u} \quad (3.241)$$

one has

$$C_{X_u} = - (2\bar{C}_D + C_{D_u}) + C_{T_u} \quad (3.242)$$

where  $C_{T_u} \approx 0$  for jet engines,  $C_{T_u} \approx -\bar{C}_D$  for piston engines, and  $C_{D_u}$  typically depends on the trim Mach number  $\bar{\mathcal{M}}$ , i.e.

$$C_{D_u} = C_{D_M} \bar{\mathcal{M}} \quad (3.243)$$

where the Mach number is defined as

$$\bar{\mathcal{M}} = \frac{\bar{v}_a}{\bar{v}_s} \quad (3.244)$$

where  $\bar{v}_s$  is the speed of sound at trim (and varies with altitude). For low speed flight

$$C_{D_u} \approx 0 \quad (3.245)$$

For  $Z$ , consider the dominant force

$$mZ = -L \quad (3.246)$$

or by lifting-line theory

$$mZ = -Q_w S_w C_L \quad (3.247)$$

and taking the derivative with respect to  $u$ , one has

$$mZ_u = -Q_w S_w \left( \frac{2\bar{C}_L}{u} + C_{L_u} \right) \quad (3.248)$$

Defining the coefficient as

$$Z_u = \frac{Q_w S_w}{m\bar{u}} C_{Z_u} \quad (3.249)$$

one has

$$C_{Z_u} = - (2\bar{C}_L + \bar{u} C_{L_u}) \quad (3.250)$$

The Prantl-Glauent formula for the lift coefficient of compressible flow given the incompressible lift coefficient states

$$C_L = \frac{1}{\sqrt{1 - \bar{\mathcal{M}}^2}} C_{L,M=0} \quad (3.251)$$

one can show

$$\bar{u} C_{L_u} = \left( \frac{\bar{\mathcal{M}}^2}{1 - \bar{\mathcal{M}}^2} \right) \bar{C}_L \quad (3.252)$$

Then,

$$C_{Z_u} = - \left( 2 + \frac{\bar{\mathcal{M}}^2}{1 - \bar{\mathcal{M}}^2} \right) \bar{C}_L \quad (3.253)$$

For  $M$ , the stability coefficient is defined as

$$M_u = \frac{Q_w S_w \bar{c}_w}{I_{yy} \bar{u}} C_{m_u} \quad (3.254)$$

for which the stability coefficient can also be rewritten as

$$C_{m_u} = C_{m_M} M \quad (3.255)$$

and for low speed flight

$$C_{m_u} \approx 0 \quad (3.256)$$

### $\alpha$ Stability Derivatives and Coefficients

For  $X$ , consider the coefficient form where one has

$$C_X = C_L \alpha - C_D + C_T \quad (3.257)$$

and modeling the drag according to lifting-line theory provides

$$C_X = C_L \alpha - \left[ C_{D_0} + \frac{C_L^2}{\pi A R_w e_w} \right] + C_T \quad (3.258)$$

Then, defining the stability derivative as

$$X_\alpha = \frac{Q S_w}{m} C_{X_\alpha} \quad (3.259)$$

taking the derivative provides the stability coefficient as

$$C_{X_\alpha} = \bar{C}_L - \frac{2 \bar{C}_L C_{L_\alpha}}{\pi A R_w e_w} \quad (3.260)$$

or

$$C_{X_\alpha} = \bar{C}_L \left( 1 - \frac{2 C_{L_\alpha}}{\pi A R_w e_w} \right) \quad (3.261)$$

For  $Z$ , consider

$$C_Z = -(C_L + C_D \alpha) \quad (3.262)$$

$$C_Z = -(C_{L_\alpha} \alpha + C_{L_0} + C_D \alpha) \quad (3.263)$$

$$C_Z = -(C_{L_\alpha} + C_D) \alpha - C_{L_0} \quad (3.264)$$

thus, the stability coefficient with respect to  $\alpha$  is

$$C_{Z_\alpha} = -(C_{L_\alpha} + \bar{C}_D) \quad (3.265)$$

where the stability derivative is defined as

$$Z_\alpha = \frac{Q_w S_w}{m} C_{Z_\alpha} \quad (3.266)$$

For  $M$ , the stability derivative is defined as

$$M_\alpha = \frac{Q_w S_w \bar{c}_w}{I_{yy}} C_{m_\alpha} \quad (3.267)$$

where the stability coefficient was derived previously for static stability as

$$C_{m_\alpha} = \frac{x_w}{\bar{c}_w} C_{L_{\alpha,w}} - \eta_h V_h C_{L_{\alpha,h}} \left( 1 - \frac{d\epsilon}{d\alpha} \right) \quad (3.268)$$

It should be noted that sometimes a  $C_{m_{\alpha,f}}$  term is included in this derivative to account for fuselage effects.

### *q* Stability Derivatives and Coefficients

For  $Z$ , the primary change is due to the change of the lift at the horizontal tail. Thus, consider modeling the change in tail lift as

$$\Delta L_h = Q_h S_h C_{L_{\alpha,h}} \Delta \alpha_h \quad (3.269)$$

where due to a rotation rate of  $q$  provides by small angle approximation

$$\Delta \alpha_h = -\frac{x_h q}{\bar{u}} \quad (3.270)$$

Thus,

$$\Delta L_h = -Q_h S_h C_{L_{\alpha,h}} \frac{x_h q}{\bar{u}} \quad (3.271)$$

or in terms of the stability derivative where  $L_h$  acts opposite the  $z_B$  axis

$$m \Delta Z = -Q_h S_h C_{L_{\alpha,h}} \frac{x_h q}{\bar{u}} \quad (3.272)$$

and taking the derivative with respect to  $q$

$$Z_q = -Q_h S_h C_{L_{\alpha,h}} \frac{x_h}{m \bar{u}} \quad (3.273)$$

By definition of the stability derivative as

$$Z_q = \frac{Q_w S_w \bar{c}_w}{2m \bar{u}} C_{Z_q} \quad (3.274)$$

the stability coefficient is

$$C_{Z_q} = \frac{2Q_h x_h S_h}{Q_w S_w \bar{c}_w} C_{L_{\alpha,h}} \quad (3.275)$$

Then, defining

$$V_h = \frac{-x_h S_h}{S_w \bar{c}_w} \quad (3.276)$$

as the **horizontal tail volume ratio** and

$$\eta_h = \frac{Q_h}{Q_w} \quad (3.277)$$

as the **horizontal tail efficiency**, one has

$$C_{Z_q} = -2\eta_h V_h C_{L_{\alpha,h}} \quad (3.278)$$

Note that  $x_h$  is a negative number here. It should also be noted that typical values for  $\eta_h$  are 0.8-1.2 where  $\eta_h$  can be > 1 because of slipstream or engine stream effects, i.e. the interaction of the airflow with the wing or engine before encountering the horizontal tail.

For  $M$ , consider the moment due to  $Z_q$  as modeled by the horizontal tail, i.e.

$$I_{yy} M_q = -x_h m Z_q \quad (3.279)$$

By definition of the stability derivative as

$$M_q = \frac{Q_w S_w \bar{c}_w^2}{2I_{yy} \bar{u}} C_{m_q} \quad (3.280)$$

and substituting for  $Z_q$  from the previous derivation, the stability coefficient is

$$C_{m_q} = -\frac{x_h}{\bar{c}_w} C_{Z_q} \quad (3.281)$$

Note that  $Z_q$  was neglected in the linearized EOMs in the previous lecture due to its small magnitude relative to  $\bar{u}$ , though it appears indirectly in  $M_q$ .

Lastly, note that a common practice with these  $q$  derivatives is to increase these models by 10% to approximately account for the wing and fuselage as well.

### $\dot{\alpha}$ Stability Derivatives and Coefficients

For  $Z$ , consider the change in the angle of attack primarily changing the circulation around the wing. This most directly affects the downwash at the tail which occurs at a lag time approximated by

$$\Delta t = \frac{-x_t}{\bar{u}} \quad (3.282)$$

Furthermore, the change in downwash can be related to the change in the horizontal tail angle of attack as

$$\Delta \alpha_h = \frac{d\epsilon}{dt} \Delta t \quad (3.283)$$

Then, by substitution

$$\Delta \alpha_h = \frac{d\epsilon}{dt} \frac{-x_h}{\bar{u}} \quad (3.284)$$

$$\Delta \alpha_h = \frac{d\epsilon}{d\alpha} \frac{d\alpha}{dt} \frac{-x_h}{\bar{u}} \quad (3.285)$$

$$\Delta \alpha_h = \frac{d\epsilon}{d\alpha} \dot{\alpha} \frac{-x_h}{\bar{u}} \quad (3.286)$$

and relating this to the change in the lift, one has

$$\Delta L_h = Q_h S_h C_{L_{\alpha,h}} \Delta \alpha_h \quad (3.287)$$

or in terms of  $Z$  where  $L_h$  acts opposite the  $z_B$  axis

$$m\Delta Z = -Q_h S_h C_{L_\alpha, h} \frac{d\epsilon}{d\alpha} \dot{\alpha} \frac{-x_h}{\bar{u}} \quad (3.288)$$

and taking the derivative with respect to  $\dot{\alpha}$

$$Z_{\dot{\alpha}} = -Q_h S_h \frac{-x_h}{m\bar{u}} C_{L_\alpha, h} \frac{d\epsilon}{d\alpha} \quad (3.289)$$

By definition of the stability derivative as

$$Z_{\dot{\alpha}} = \frac{Q_w S_w \bar{c}_w}{2m\bar{u}} C_{Z_{\dot{\alpha}}} \quad (3.290)$$

the stability coefficient is

$$C_{Z_{\dot{\alpha}}} = -2V_h \eta_h C_{L_\alpha, h} \frac{d\epsilon}{d\alpha} \quad (3.291)$$

For  $M$ , consider the moment due to  $Z_{\dot{\alpha}}$  as modeled by the horizontal tail, i.e.

$$I_{yy} M_{\dot{\alpha}} = -x_h m Z_{\dot{\alpha}} \quad (3.292)$$

By definition of the stability derivative as

$$M_{\dot{\alpha}} = \frac{Q_w S_w \bar{c}_w^2}{2I_{yy}\bar{u}} C_{m_{\dot{\alpha}}} \quad (3.293)$$

and substituting for  $Z_{\dot{\alpha}}$  from the previous derivation, the stability coefficient is

$$C_{m_{\dot{\alpha}}} = -\frac{x_h}{\bar{c}_w} C_{Z_{\dot{\alpha}}} \quad (3.294)$$

Note that  $Z_{\dot{\alpha}}$  was neglected in the linearized EOMs in the previous lecture due to its small magnitude relative to  $\bar{u}$ , although it appears indirectly in  $M_{\dot{\alpha}}$ .

### Longitudinal Control Derivatives and Coefficients

The longitudinal control inputs are the elevator angle,  $\delta_e$ , and throttle input  $\delta_t$ . The throttle coefficients and derivatives are determined solely by the engine type and can vary with the placement of the engines, e.g. under the wings. However, their modeling is beyond the scope of this course, but, in general the thrust can affect  $C_X$ ,  $C_Z$ , and  $C_m$ . This implies the following definitions for the control derivatives

$$X_{\delta_t} = \frac{Q_w S_w}{m} C_{X_{\delta_t}} \quad (3.295)$$

$$Z_{\delta_t} = \frac{Q_w S_w}{m} C_{Z_{\delta_t}} \quad (3.296)$$

$$M_{\delta_t} = \frac{Q_w S_w \bar{c}_w}{I_{yy}} C_{m_{\delta_t}} \quad (3.297)$$

where the coefficients will depend on the design of the propulsion system and will be given directly in this course.

The elevator angle input affects the aerodynamics of the horizontal tail which will affect the lift and induced drag of the tail, the second of which is relatively small compared to the total airplane drag and will be neglected in this course's models. Thus, the elevator control derivatives are defined as

$$Z_{\delta_e} = \frac{Q_w S_w}{m} C_{Z_{\delta_e}} \quad (3.298)$$

$$M_{\delta_e} = \frac{Q_w S_w \bar{c}_w}{I_{yy}} C_{m_{\delta_e}} \quad (3.299)$$

where the coefficients can be modeled using the tail lift effects as

$$C_{Z_{\delta_e}} = -\eta_h \frac{S_h}{S_w} C_{L_{\delta_e},h} \quad (3.300)$$

$$C_{m_{\delta_e}} = -\frac{x_h}{\bar{c}_w} C_{Z_{\delta_e}} \quad (3.301)$$

$C_{m_{\delta_e}}$  is also known as the **elevator control power** while the derivative  $C_{L_{\delta_e},h}$  is the **elevator effectiveness** and can be rewritten as

$$C_{L_{\delta_e},h} = \frac{\partial C_{L,h}}{\partial \delta_e} = \frac{\partial C_{L,h}}{\partial \alpha_h} \frac{\partial \alpha_h}{\partial \delta_e} = C_{L_{\alpha},h} \tau_e \quad (3.302)$$

where the  $\tau$  **empirical parameter** models the relationship between the change in the lifting surface's angle of attack and the control surface deflection angle. Intuitively,  $\tau$  depends on the ratio of the control surface area to the lifting surface area, e.g. if  $S_e$  is the surface area of the elevator, then  $\tau_e$  is related to  $S_e/S_h$ . This term will also be used for the rudder and aileron control coefficients. The following table provides discrete values for  $\tau$ .

$S_{control}/S_{lifting}$	0.0	0.05	0.10	0.15	0.20	0.25	0.30	0.35
$\tau$	0.0	0.16	0.26	0.34	0.41	0.47	0.52	0.56
$S_{control}/S_{lifting}$	0.40	0.45	0.50	0.55	0.60	0.65	0.70	
$\tau$	0.60	0.64	0.68	0.72	0.75	0.78	0.80	

For intermediate values one typically uses linear interpolation.

## 3.6 Lateral-Directional Stability and Control Derivatives

This section will derive models for the lateral stability and control derivatives used in the Jacobian linearization of the normalized aerodynamic forces and moments about trimmed steady flight, i.e.

$$\begin{aligned} Y &= \bar{Y} + Y_\beta \Delta \beta + Y_p \Delta p + Y_r \Delta r + Y_{\delta_r} \Delta \delta_r \\ L &= \bar{L} + L_\beta \Delta \beta + L_p \Delta p + L_r \Delta r + L_{\delta_a} \Delta \delta_a + L_{\delta_r} \Delta \delta_r \\ N &= \bar{N} + N_\beta \Delta \beta + N_p \Delta p + N_r \Delta r + N_{\delta_a} \Delta \delta_a + N_{\delta_r} \Delta \delta_r \end{aligned} \quad (3.303)$$

where the coefficients of the linear terms are called the **lateral-directional stability and control derivatives**. Alternatively, these equations can be written in terms of nondimensional coefficients as

$$\begin{aligned} C_Y &= \bar{C}_Y + C_{Y_\beta} \Delta\beta + C_{Y_p} \Delta p + C_{Y_r} \Delta r + C_{Y_{\delta_r}} \Delta\delta_r \\ C_l &= \bar{C}_l + C_{l_\beta} \Delta\beta + C_{l_p} \Delta p + C_{l_r} \Delta r + C_{l_{\delta_a}} \Delta\delta_a + C_{l_{\delta_r}} \Delta\delta_r \\ C_n &= \bar{C}_n + C_{n_\beta} \Delta\beta + C_{n_p} \Delta p + C_{n_r} \Delta r + C_{n_{\delta_a}} \Delta\delta_a + C_{n_{\delta_r}} \Delta\delta_r \end{aligned} \quad (3.304)$$

where the coefficients of the linear terms are called the **lateral-directional stability and control coefficients**.

It should be noted that a more comprehensive collection of aerodynamic stability and control prediction techniques can be found in the *USAF Stability and Control DATCOM (Data Compendium)* which was compiled between 1960 and 1978 by the McDonnell Douglas Corporation in conjunction with the Flight Dynamics Laboratory at Wright-Patterson Air Force Base.

Lastly, it should also be noted that in the following models  $S_v$  is the vertical tail area and includes the submerged area to fuselage centerline.

### $\beta$ Stability Derivatives and Coefficients

For  $Y$ , consider the dominant force from the vertical tail

$$mY = -L_v \quad (3.305)$$

and in coefficient form

$$mY = -Q_v S_v C_{L,v} \quad (3.306)$$

or

$$mY = -Q_v S_v C_{L_{\alpha,v}} (\beta + \sigma) \quad (3.307)$$

and taking the derivative with respect to  $\beta$  provides

$$mY_\beta = -Q_v S_v C_{L_{\alpha,v}} \left( 1 + \frac{d\sigma}{d\beta} \right) \quad (3.308)$$

By definition of the stability derivative as

$$Y_\beta = \frac{Q_w S_w}{m} C_{Y_\beta} \quad (3.309)$$

and defining

$$\eta_v = \frac{Q_v}{Q_w} \quad (3.310)$$

as the **vertical tail efficiency**, the stability coefficient is

$$C_{Y_\beta} = -\eta_v \frac{S_v}{S_w} C_{L_{\alpha,v}} \left( 1 + \frac{d\sigma}{d\beta} \right) \quad (3.311)$$

The following empirical equation can be used in the previous equation

$$\eta_v \left( 1 + \frac{d\sigma}{d\beta} \right) = 0.724 + 3.06 \frac{S_v/S_w}{1 + \cos \Lambda_w} + 0.4 \frac{z_w - z_f}{d_f} + 0.009 AR_w \quad (3.312)$$

where  $\Lambda_w$  is the sweep angle of the wing (measured at quarter chord),  $z_w$  is the  $z_B$  coordinate of the aerodynamic center of the wing,  $z_f$  is the  $z_B$  coordinate of the fuselage centerline,  $d_f$  is the maximum fuselage depth, and  $AR_w$  is the wing aspect ratio.

For  $L$ , the primary factors are the dihedral angle and the wing taper. The stability derivative is defined as

$$L_\beta = \frac{Q_w S_w b_w}{I_{xx}} C_{l_\beta} \quad (3.313)$$

and the stability coefficient can be simply modeled by

$$C_{l_\beta} = \frac{\partial C_{l_\beta}}{\partial \Gamma_w} \Gamma_w \quad (3.314)$$

where  $\Gamma_w$  is the wing dihedral angle,  $\frac{\partial C_{l_\beta}}{\partial \Gamma_w}$  is related to the aspect ratio and taper ratio of the root chord and tip chord ( $\approx -0.4$  to  $-0.9$  /rad $^2$ ).

For  $N$ , the stability derivative is defined as

$$N_\beta = \frac{Q_w S_w b_w}{I_{zz}} C_{n_\beta} \quad (3.315)$$

where the stability coefficient was derived previously for static stability as

$$C_{n_\beta} = C_{n_\beta, w-f} + \eta_v V_v C_{L_{\alpha_v}} \left( 1 + \frac{d\sigma}{d\beta} \right) \quad (3.316)$$

where

$$V_v = \frac{-x_v S_v}{b_w S_w} \quad (3.317)$$

is the **vertical tail volume ratio**.

### *p* Stability Derivatives and Coefficients

For  $Y$ , there will be a resulting side force only if there is wing sweep as the change in the relative airflow will only then have a component in the  $y_B$  direction. The stability derivative is defined as

$$Y_p = \frac{Q_w S_w b_w}{2m\bar{u}} C_{Y_p} \quad (3.318)$$

It can be shown that the stability coefficient is

$$C_{Y_p} = \frac{AR_w + \cos \Lambda_w}{AR_w + 4 \cos \Lambda_w} \tan \Lambda_w \bar{C}_L \quad (3.319)$$

For  $L$ , the primary factor is the change in lift distribution over the wing. Thus, consider modeling the change in tail lift over a cross-section of the wing of width  $dy$  and chord length  $c_w(y)$  as

$$\Delta(\text{Lift}) = Q_w(c_w(y)dy) C_{l_{\alpha,w}} \Delta\alpha \quad (3.320)$$

where  $C_{l_{\alpha},w}$  is the cross-sectional lift coefficient slope. A rotation rate of  $p$  provides by the small angle approximation

$$\Delta\alpha = \frac{yp}{\bar{u}} \quad (3.321)$$

Thus, for the moment arm  $y$  for the contribution to the normalized moment  $L$  of the sectional mass

$$my^2\Delta L = -Q_w C_{l_{\alpha},w} \frac{p}{\bar{u}} c_w(y) y^2 dy \quad (3.322)$$

which can be integrated over half the span of the wing twice for the entire roll moment  $I_{xx}L$  due to the moment being in the opposite sense for the other side of the wing as

$$I_{xx}L = -2 \int_0^{b_w/2} Q_w C_{l_{\alpha},w} \frac{p}{\bar{u}} c_w(y) y^2 dy \quad (3.323)$$

Then, assuming that the cross-sectional lift coefficient slope is well approximated by the wing lift coefficient slope, one has

$$I_{xx}L = \frac{-2Q_w C_{L_{\alpha},w} p}{\bar{u}} \int_0^{b_w/2} c_w(y) y^2 dy \quad (3.324)$$

and taking the derivative with respect to  $p$

$$I_{xx}L_p = \frac{-2Q_w C_{L_{\alpha},w}}{\bar{u}} \int_0^{b_w/2} c_w(y) y^2 dy \quad (3.325)$$

By definition of the stability derivative as

$$L_p = \frac{Q_w S_w b_w^2}{2I_{xx}\bar{u}} C_{l_p} \quad (3.326)$$

The stability coefficient is

$$C_{l_p} = -\frac{4C_{L_{\alpha},w}}{S_w b_w^2} \int_0^{b_w/2} c_w(y) y^2 dy \quad (3.327)$$

where  $c_w(y)$  is the wing chord as a function of the lateral coordinate. For a tapered wing this chord function can be written as

$$c_w(y) = c_{r,w} \left[ 1 + \left( \frac{\frac{c_{t,w}}{c_{r,w}} - 1}{\frac{b_w}{2}} \right) y \right] \quad (3.328)$$

where  $c_{r,w}$  and  $c_{t,w}$  are the root and tip chords for the wing which results in a stability coefficient

$$C_{l_p} = -\left( \frac{1 + 3\lambda_w}{1 + \lambda_w} \right) \frac{C_{L_{\alpha},w}}{12} \quad (3.329)$$

where  $\lambda_w$  is the taper ratio of the wing, i.e.

$$\lambda_w = \frac{c_{t,w}}{c_{r,w}} \quad (3.330)$$

For  $N$ , the stability derivative is defined as

$$N_p = \frac{Q_w S_w b_w^2}{2I_{zz}\bar{u}} C_{n_p} \quad (3.331)$$

It can be shown that the stability coefficient is

$$C_{n_p} = -\frac{\bar{C}_L}{8} \quad (3.332)$$

which is due to conservation of angular momentum.

### r Stability Derivatives and Coefficients

For  $Y$ , consider the dominant force from the vertical tail side force and its moment arm due to the resulting sideslip, i.e.

$$Y_r = \frac{x_v}{\bar{u}} Y_{\beta,v} \quad (3.333)$$

$$Y_r = \frac{x_v Q_w S_w}{m \bar{u}} C_{Y_{\beta,v}} \quad (3.334)$$

By definition of the stability derivative as

$$Y_r = \frac{Q_w S_w b_w}{2m \bar{u}} C_{Y_r} \quad (3.335)$$

the stability coefficient is

$$C_{Y_r} = 2 \frac{x_v}{b_w} C_{Y_{\beta,v}} \quad (3.336)$$

where  $C_{Y_{\beta,v}} = C_{Y_{\beta}}$  for the tail-only model in this lecture.

For  $L$ , the stability derivative is defined as

$$L_r = \frac{Q_w S_w b_w^2}{2I_{xx} \bar{u}} C_{l_r} \quad (3.337)$$

It can be shown that the stability coefficient is

$$C_{l_r} = \frac{\bar{C}_L}{4} - 2 \frac{x_v z_v}{b_w^2} C_{Y_{\beta,v}} \quad (3.338)$$

where  $C_{Y_{\beta,v}} = C_{Y_{\beta}}$  for the tail-only model in this lecture.

For  $N$ , consider the dominant force from the vertical tail side force and its moment arm, i.e.

$$I_{zz} \Delta N = -Q_v S_v C_{L_{\alpha,v}} x_v \Delta \beta \quad (3.339)$$

where due to a rotation rate of  $r$  provides by small angle approximation

$$\Delta \beta = \frac{x_v r}{\bar{u}} \quad (3.340)$$

Thus,

$$I_{zz} \Delta N = -Q_v S_v C_{L_{\alpha,v}} \frac{x_v r}{\bar{u}} \quad (3.341)$$

and taking the derivative with respect to  $r$

$$I_{zz} N_r = -Q_v S_v C_{L_{\alpha,v}} \frac{x_v}{\bar{u}} \quad (3.342)$$

By definition of the stability derivative as

$$N_r = \frac{Q_w S_w b_w^2}{2I_{zz} \bar{u}} C_{n_r} \quad (3.343)$$

the stability coefficient is

$$C_{n_r} = 2\eta_v V_v \frac{x_v}{b_w} C_{L_{\alpha,v}} \quad (3.344)$$

Note that  $x_v$  is a negative number here.

### Lateral-Directional Control Derivatives and Coefficients

The lateral-directional control derivatives are defined as

$$Y_{\delta_r} = \frac{Q_w S_w}{m} C_{Y_{\delta_r}} \quad (3.345)$$

$$L_{\delta_a} = \frac{Q_w S_w b_w}{I_{xx}} C_{l_{\delta_a}} \quad (3.346)$$

$$L_{\delta_r} = \frac{Q_w S_w b_w}{I_{xx}} C_{l_{\delta_r}} \quad (3.347)$$

$$N_{\delta_a} = \frac{Q_w S_w b_w}{I_{zz}} C_{n_{\delta_a}} \quad (3.348)$$

$$N_{\delta_r} = \frac{Q_w S_w b_w}{I_{zz}} C_{n_{\delta_r}} \quad (3.349)$$

For low speed flight, the following models can be used for the control coefficients.

The  $\delta_a$  control coefficient for  $L$  can be modeled as

$$C_{l_{\delta_a}} = \frac{2C_{L_{\alpha,w}} \tau_a}{S_w b_w} \int_{y_{a,i}}^{y_{a,o}} c_w(y) y dy \quad (3.350)$$

where  $C_{l_{\delta_a}}$  is the **aileron control power**,  $C_{L_{\alpha,w}} \tau_a$  is the **aileron effectiveness**,  $y_{a,i}$  is the inner aileron  $y_B$  coordinate, and  $y_{a,o}$  is the outer aileron  $y_B$  coordinate. For a tapered wing, this becomes

$$C_{l_{\delta_a}} = \frac{2C_{L_{\alpha_w}} \tau_a}{S_w b_w} \int_{y_{a,i}}^{y_{a,o}} c_{r,w} \left[ 1 + \left( \frac{c_{t,w}/c_{r,w} - 1}{b_w/2} \right) y \right] y dy \quad (3.351)$$

$$C_{l_{\delta_a}} = \frac{2C_{L_{\alpha_w}} \tau_a c_{r,w}}{S_w b_w} \left[ \frac{y^2}{2} + \left( \frac{c_{t,w}/c_{r,w} - 1}{b_w/2} \right) \frac{y^3}{3} \right]_{y_{a,i}}^{y_{a,o}} \quad (3.352)$$

The  $\delta_a$  control coefficient for  $N$  can be modeled as

$$C_{n_{\delta_a}} = 2K \bar{C}_L C_{l_{\delta_a}} \quad (3.353)$$

where  $K$  is an aileron empirical factor ( $\approx -0.3$  to  $-0.1$ ) that decreases with aspect ratio, increases slightly with aileron placement further out on the wing, and increases with taper ratio; and

The  $\delta_r$  control coefficient for  $Y$  can be modeled as

$$C_{Y_{\delta_r}} = \frac{S_v}{S_w} \tau_r C_{L_{\alpha,v}} \quad (3.354)$$

The  $\delta_r$  control coefficient for  $L$  can be modeled as

$$C_{l_{\delta_r}} = \frac{S_v}{S_w} \tau_r \frac{z_v}{b_w} C_{L_{\alpha,v}} \quad (3.355)$$

The  $\delta_r$  control coefficient for  $N$  can be modeled as

$$C_{n_{\delta_r}} = -\eta_v V_v C_{L_{\alpha,v}} \tau_r \quad (3.356)$$

where  $C_{n_{\delta_r}}$  is the **rudder control power** and  $C_{L_{\alpha,v}} \tau_r$  is the **rudder effectiveness**.

## Example Problem

Given: an airplane with the following tapered wing geometry

$b_w = 33.4$ ft	$c_{r,w} = 7.2$ ft	$c_{t,w} = 3.9$ ft	$y_{a,i} = 11.1$ ft
$y_{a,o} = 16$ ft	$S_w = 184$ ft <sup>2</sup>	$C_{L_{\alpha,w}} = 4.3/\text{rad}$	$S_a = 30.2$ ft <sup>2</sup>

Determine:  $C_{l_{\delta_a}}$

Solution:

First, calculate the surface area of the ailerons to the surface area of the wing, i.e.

$$\frac{S_a}{S_w} = \frac{30.2}{184} = 0.164 \quad (3.357)$$

Using the table for  $\tau$ , one has by linear interpolation

$$\tau_a = \frac{0.164 - 0.15}{0.2 - 0.15} (0.41 - 0.34) + 0.34 \quad (3.358)$$

or

$$\tau_a = 0.36 \quad (3.359)$$

Then, for a tapered wing,

$$C_{l_{\delta_a}} = \frac{2C_{L_{\alpha,w}} \tau_a c_{r,w}}{S_w b_w} \left[ \frac{y^2}{2} + \left( \frac{c_{t,w}/c_{r,w} - 1}{b_w/2} \right) \frac{y^3}{3} \right]_{y_{a,i}}^{y_{a,o}} \quad (3.360)$$

and substituting

$$C_{l_{\delta_a}} = \frac{2(4.3)(0.36)(7.2)}{184(33.4)} (90.4 - 49) \quad (3.361)$$

$$\underline{\underline{C_{l_{\delta_a}} = 0.155/\text{rad}}} \quad (3.362)$$

## 3.7 Airplane Dynamic Stability and Flying Qualities

Recall that for LTI state-space systems, the Lyapunov stability is determined by the stability of the system modes. For flight vehicles, this type of stability is also known as **dynamic stability**. As such, this section will discuss the longitudinal and lateral-directional modes for the linearized rigid airplane dynamics using the state-space representation as well as approximations for the different modes. In addition, an analysis of these modes and their effects on the flying qualities of airplanes is discussed.

### Longitudinal Modes

Recall the fourth order linearized longitudinal dynamics for rigid airplanes as

$$\begin{bmatrix} \Delta\dot{u} \\ \Delta\dot{\alpha} \\ \Delta\dot{q} \\ \Delta\dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_\alpha & 0 & -g \cos \bar{\theta} \\ \frac{Z_u}{\bar{u}} & \frac{Z_\alpha}{\bar{u}} & 1 & -\frac{g}{\bar{u}} \sin \bar{\theta} \cos \bar{\phi} \\ M_u + M_{\dot{\alpha}} \frac{Z_u}{\bar{u}} & M_\alpha + M_{\dot{\alpha}} \frac{Z_\alpha}{\bar{u}} & M_q + M_{\dot{\alpha}} & -M_{\dot{\alpha}} \frac{g}{\bar{u}} \sin \bar{\theta} \cos \bar{\phi} \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta q \\ \Delta \theta \end{bmatrix} + \begin{bmatrix} 0 & X_{\delta_t} \\ \frac{Z_{\delta_e}}{\bar{u}} & \frac{Z_{\delta_t}}{\bar{u}} \\ M_{\delta_e} + M_{\dot{\alpha}} \frac{Z_{\delta_e}}{\bar{u}} & M_{\delta_t} + M_{\dot{\alpha}} \frac{Z_{\delta_t}}{\bar{u}} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \delta_e \\ \Delta \delta_t \end{bmatrix} \quad (3.363)$$

where  $\Delta w \approx \bar{u}\Delta\alpha$  may also be substituted with

$$M_\alpha = \bar{u}M_w, \quad Z_\alpha = \bar{u}Z_w, \quad M_{\dot{\alpha}} = \bar{u}M_{\dot{w}} \quad (3.364)$$

There are two oscillatory modes that dominate the free response in the longitudinal plane: the **long-period mode** and the **short-period mode**. Recall from Lyapunov stability theory, an airplane is **longitudinally Lyapunov stable** if and only if both modes are stable which occurs when the real part of the eigenvalues of the state matrix  $A$  are negative, i.e. the modes are in the left-half plane (LHP) of the complex plane.

Because the exact analytical expressions for eigenvalues are complicated, it is useful to derive approximate eigenvalues with further simplifications to the longitudinal state-space representation which is possible as each mode primarily affects a different pair of states, i.e. are weakly decoupled. Furthermore, to derive simpler expressions, first, assume that  $\bar{\theta} = 0$ , then the longitudinal state matrix becomes

$$A_{long} = \begin{bmatrix} X_u & X_\alpha & 0 & -g \\ \frac{Z_u}{\bar{u}} & \frac{Z_\alpha}{\bar{u}} & 1 & 0 \\ M_u + M_{\dot{\alpha}} \frac{Z_u}{\bar{u}} & M_\alpha + M_{\dot{\alpha}} \frac{Z_\alpha}{\bar{u}} & M_q + M_{\dot{\alpha}} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.365)$$

The long-period mode can be approximated by ignoring the pitching moment equation  $\dot{q}$  and assuming  $\Delta\alpha = 0$ , i.e.  $\alpha$  responds quickly compared to this mode. Then, one has

$$\begin{bmatrix} \Delta\dot{u} \\ 0 \\ \Delta\dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_w & 0 & -g \\ \frac{Z_u}{\bar{u}} & \frac{Z_w}{\bar{u}} & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ 0 \\ \Delta q \\ \Delta \theta \end{bmatrix} \quad (3.366)$$

Solving the second equation, one has

$$0 = \frac{Z_u}{\bar{u}} \Delta u + \Delta q \quad (3.367)$$

Rearranging and noting  $\Delta q = \Delta \dot{\theta}$ , one has

$$\Delta \dot{\theta} = -\frac{Z_u}{\bar{u}} \Delta u \quad (3.368)$$

Finally, by substitution into the third equation, one obtains the second order state-space system for the **long-period mode approximation** as

$$\begin{bmatrix} \Delta \ddot{u} \\ \Delta \dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & -g \\ -\frac{Z_u}{\bar{u}} & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \theta \end{bmatrix} \quad (3.369)$$

It should be noted that if one assumes  $\Delta \alpha = 0$ , then one can also use the following relationship

$$\Delta \dot{h} = \bar{u} \Delta \theta \quad (3.370)$$

Thus, the long-period mode can be understood as the gradual interchange of kinetic and potential energy through the varying velocity and altitude, respectively. Because the long-period mode is typically highly oscillatory, most applications require it to be corrected during flight, often manually, but these corrections can be fatiguing for pilots if the damping ratio is too low. This mode is also known as the **phugoid mode**, which was derived from the Greek words, *phuge* and *eidos*, for “flight-like.”

The short-period mode can be approximated by ignoring the  $\dot{u}$  equation and assuming  $\Delta u = 0$ , i.e. the perturbed airspeed doesn't vary quickly. Then, one has

$$\begin{bmatrix} \Delta \dot{\alpha} \\ \Delta \dot{q} \\ \Delta \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{Z_\alpha}{\bar{u}} & 1 & 0 \\ M_\alpha + M_{\dot{\alpha}} \frac{Z_\alpha}{\bar{u}} & M_q + M_{\dot{\alpha}} & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta \alpha \\ \Delta q \\ \Delta \theta \end{bmatrix} \quad (3.371)$$

and by ignoring  $\Delta \theta$ , one obtains the second order LTI state-space system for the **short-period mode approximation** as

$$\begin{bmatrix} \Delta \dot{\alpha} \\ \Delta \dot{q} \end{bmatrix} = \begin{bmatrix} \frac{Z_\alpha}{\bar{u}} & 1 \\ M_\alpha + M_{\dot{\alpha}} \frac{Z_\alpha}{\bar{u}} & M_q + M_{\dot{\alpha}} \end{bmatrix} \begin{bmatrix} \Delta \alpha \\ \Delta q \end{bmatrix} \quad (3.372)$$

The short-period is more important to design of airplanes than the long-period. One typically balances a desire for a high natural frequency, i.e. a quick response to input, but also heavy damping, i.e. little overshoot.

## Lateral-Directional Modes

Recall the lateral-directional LTI state-space system for the linearized EOM as derived in this course

$$\begin{bmatrix} \Delta\dot{\beta} \\ \Delta\dot{p} \\ \Delta\dot{r} \\ \Delta\dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{Y_\beta}{\bar{u}} & \frac{Y_p}{\bar{u}} & \frac{Y_r}{\bar{u}} - 1 & \frac{g}{\bar{u}} \cos \bar{\theta} \cos \bar{\phi} \\ L_\beta^* + \frac{I_{xz}}{I_{xx}} N_\beta^* & L_p^* + \frac{I_{xz}}{I_{xx}} N_p^* & L_r^* + \frac{I_{xz}}{I_{xx}} N_r^* & 0 \\ N_\beta^* + \frac{I_{xz}}{I_{zz}} L_\beta^* & N_p^* + \frac{I_{xz}}{I_{zz}} L_p^* & N_r^* + \frac{I_{xz}}{I_{zz}} L_r^* & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\beta \\ \Delta p \\ \Delta r \\ \Delta\phi \end{bmatrix} + \begin{bmatrix} 0 & \frac{Y_{\delta_r}}{\bar{u}} \\ L_{\delta_a}^* + \frac{I_{xz}}{I_{xx}} N_{\delta_a}^* & L_{\delta_r}^* + \frac{I_{xz}}{I_{xx}} N_{\delta_r}^* \\ N_{\delta_a}^* + \frac{I_{xz}}{I_{zz}} L_{\delta_a}^* & N_{\delta_r}^* + \frac{I_{xz}}{I_{zz}} L_{\delta_r}^* \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\delta_a \\ \Delta\delta_r \end{bmatrix} \quad (3.373)$$

where  $\Delta v \approx \bar{u}\Delta\beta$  may also be substituted with

$$Y_\beta = \bar{u}Y_v, \quad L_\beta = \bar{u}L_v, \quad N_\beta = \bar{u}N_v \quad (3.374)$$

There are two exponentially decaying modes and one oscillatory mode that dominate the free response in the lateral-directional planes: the **roll mode**, the **spiral mode**, and the **dutch roll mode**. Recall from linear stability theory, an airplane is **lateral-directionally Lyapunov stable** if and only if all three modes are stable which occurs when the real part of the eigenvalues of the state matrix  $A$  are negative, i.e. the modes are in the left-half plane (LHP) of the complex plane.

Because the exact analytical expressions for eigenvalues are complicated, it is useful to derive approximate eigenvalues with further simplifications to the state-space system which is possible as each mode primarily affects a different pair of states. Furthermore, to derive simpler expressions, first, assume that  $\bar{\theta} = \bar{\phi} = 0$  and  $I_{xz} = 0$ , then the lateral-directional state matrix becomes

$$A_{lat} = \begin{bmatrix} \frac{Y_\beta}{\bar{u}} & \frac{Y_p}{\bar{u}} & \frac{Y_r}{\bar{u}} - 1 & \frac{g}{\bar{u}} \\ L_\beta & L_p & L_r & 0 \\ N_\beta & N_p & N_r & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (3.375)$$

The roll mode can be approximated by assuming a pure rolling equation of motion, i.e.

$$\Delta\dot{p} = L_p\Delta p \quad (3.376)$$

and rearranging into the standard ODE form, one obtains the first order **roll mode approximation** as

$$\Delta\dot{p} - L_p\Delta p = 0 \quad (3.377)$$

where if  $L_p < 0$ , then the roll mode is stable. The value of  $L_p$  depends primarily on the size of the wing and tail surfaces. Typically this mode decays rapidly compared to the other lateral-directional modes.

Though the entire motion of the spiral mode is characterized by  $\beta$ ,  $p$ , and  $r$ ,  $p$  is usually low and often heavily damped. Thus, the spiral mode can be approximated by ignoring  $p$  and  $\phi$  and assuming that  $\Delta\dot{\beta} = 0$  (i.e.  $\beta$  responds quickly compared to this mode). Then, one has

$$\begin{bmatrix} 0 \\ \Delta\dot{r} \end{bmatrix} = \begin{bmatrix} L_\beta\Delta\beta + L_r\Delta r \\ N_\beta\Delta\beta + N_r\Delta r \end{bmatrix} \quad (3.378)$$

Substituting the first equation into the second using  $\Delta\beta$  and rearranging into standard ODE form, one obtains the first order **spiral mode approximation** as

$$\Delta\dot{r} + \frac{L_r N_\beta - L_\beta N_r}{L_\beta} \Delta r = 0 \quad (3.379)$$

where if  $\frac{L_r N_\beta - L_\beta N_r}{L_\beta} < 0$ , then the spiral mode is stable. Typically this mode decays slowly compared to the other lateral-directional modes.

The motion corresponding to the dutch roll mode was named for its side-to-side motion which alludes to dutch figure skaters using repetitive right-and-left skating on the outer edge of their skates to maintain speed. The spiral mode can be approximated by assuming sideslip and yawing motions, although there is typically a slight rolling action as well. Then, one obtains the second order LTI state-space system for the **dutch roll mode approximation** as

$$\begin{bmatrix} \Delta\dot{\beta} \\ \Delta\dot{r} \end{bmatrix} = \begin{bmatrix} \frac{Y_\beta}{\tilde{u}} & \frac{Y_r}{\tilde{u}} - 1 \\ N_\beta & N_r \end{bmatrix} \begin{bmatrix} \Delta\beta \\ \Delta r \end{bmatrix} \quad (3.380)$$

Lastly, it should be noted that the spiral and dutch roll modes are closely related and the previous formulas provide only a rough estimate of these modes. Often these two can only be accurately represented by a coupled 3-DOF LTI state-space system. It should also be noted that the airplane design characteristics affect the spiral and dutch roll modes in opposite ways. In particular, increasing the dihedral effect makes the dutch roll mode less stable and the spiral mode more stable. Conversely, increasing the directional stability makes the dutch roll mode more stable and the spiral mode less stable. Thus, often one must use **stability augmentation systems (SAS)** to sufficiently stabilize both the spiral and dutch roll modes. The subsequent controls portion of this part of the textbook will address how to design SAS to accomplish this.

## Flying Qualities

When designing airplanes, the **flying qualities**, i.e. the handling of the airplane by a pilot, are closely related to the modal characteristics of the airplane. However, as pilots are generally charged with performing various tasks or **missions**, the flying qualities are generally specified by airplane pilots according to the following three subjective levels:

- **Level 1 (Good):** Flying qualities clearly adequate for the mission flight phase.
- **Level 2 (Acceptable):** Flying qualities adequate to accomplish the mission flight phase, but with some increase in pilot workload and/or degradation in mission effectiveness or both.
- **Level 3 (Poor):** Flying qualities such that the airplane can be controlled safely, but pilot workload is excessive and/or mission effectiveness is inadequate or both.

which also depend on the three generalized flight phase categories:

- **Category A:** nonterminal flight phases that require rapid maneuvering, precision tracking, or highly accurate flight-path control.

- **Category B:** nonterminal flight phases that are normally accomplished using gradual maneuvers and without precision tracking, although accurate flight-path control may be required.
- **Category C:** terminal flight phases that are normally accomplished using gradual maneuvers and usually require accurate flight-path control.

Notably, for Level 3 flying qualities, Category A flight phases can be terminated safely and Category B and C flight phases can be completed. Lastly, another important part of assessing these flying qualities is by class of airplane which are generally classified as

- **Class I:** Small, light airplanes
- **Class II:** Medium-weight, low-to-medium maneuverability airplanes
- **Class III:** Large, heavy, low-to-medium maneuverability airplanes
- **Class IV:** High-maneuverability airplanes

The following table of modal characteristics for different flying quality levels, flight phase categories, and airplane classes.

Mode	Level	Category	Class	Characteristic(s)
Phugoid	1	All	All	$\zeta > 0.04$
	2	All	All	$\zeta > 0$
	3	All	All	$T > 55s$
Short-Period	1	A and C	All	$0.35 \leq \zeta \leq 1.3$
		B	All	$0.3 \leq \zeta \leq 2.0$
	2	A and C	All	$0.25 \leq \zeta \leq 2.0$
		B	All	$0.2 \leq \zeta \leq 2.0$
	3	All	All	$0.15 \leq \zeta$
Roll	1	A and C	I, IV II, III	$\tau \leq 1.0$ sec $\tau \leq 1.4$ sec
		B	All	$\tau \leq 1.4$ sec
	2	A and C	I, IV II, III	$\tau \leq 1.4$ sec $\tau \leq 3.0$ sec
		B	All	$\tau \leq 3.0$ sec
	3	All	All	$\tau \leq 10$ sec
Spiral	1	A	I, IV II, III	Double amplitude $\geq 12$ sec Double amplitude $\geq 20$ sec
		B and C	All	Double amplitude $\geq 20$ sec
	2	All	All	Double amplitude $\geq 12$ sec
	3	All	All	Double amplitude $\geq 4$ sec
Dutch Roll	1	A	I, IV II, III	$\zeta\omega_n \geq 0.35$ rad/s, $\zeta \geq 0.19$ , $\omega_n > 1.0$ rad/s $\zeta\omega_n \geq 0.35$ rad/s, $\zeta \geq 0.19$ , $\omega_n > 0.4$ rad/s
		B	All	$\zeta\omega_n \geq 0.15$ rad/s, $\zeta \geq 0.08$ , $\omega_n > 0.4$ rad/s
		C	I, II-C, IV II-L, III	$\zeta\omega_n \geq 0.15$ rad/s, $\zeta \geq 0.08$ , $\omega_n > 1.0$ rad/s $\zeta\omega_n \geq 0.15$ rad/s, $\zeta \geq 0.08$ , $\omega_n > 0.4$ rad/s
	2	All	All	$\zeta\omega_n \geq 0.05$ rad/s, $\zeta \geq 0.02$ , $\omega_n > 0.4$ rad/s
	3	All	All	$\zeta \geq 0.02$ , $\omega_n \geq 0.4$ rad/s

where C and L denote carrier- or land-based airplanes. Note that the spiral mode requirements are for doubling of amplitude for an *unstable* spiral mode.

Lastly, it is important to note that the **Cooper-Harper Rating Scale (CHRS)** is another standard used by pilots and flight test engineers to rate the flying qualities of the airplane design as a whole. The CHRS scale goes from 1 to 10 with lower numbers corresponding to better flying qualities. The description for each rating is shown in the following table.

Pilot Rating	Aircraft Characteristic	Demand of Pilot	Overall Assessment
1	Excellent, highly desirable	Pilot compensation not a factor for desired performance	Good
2	Good, negligible deficiencies	Pilot compensation not a factor for desired performance	Good
3	Fair, some mildly unpleasant deficiencies	Minimal pilot compensation required for desired performance	Good
4	Minor, but annoying deficiencies	Desired performance requires moderate pilot compensation	Acceptable
5	Moderately objectionable deficiencies	Adequate performance requires considerable pilot compensation	Acceptable
6	Very objectionable, but tolerable deficiencies	Adequate performance requires extensive pilot compensation	Acceptable
7	Major deficiencies	Adequate performance not attainable with maximum tolerable compensation	Poor
8	Major deficiencies	Considerable pilot compensation is required for control	Poor
9	Major deficiencies	Intense pilot compensation is required for control	Poor
10	Major deficiencies	Control will be lost during some portion of required operation	Unacceptable

## Example Problem 1

Given: the following longitudinal state-space equation

$$\begin{bmatrix} \Delta\dot{u} \\ \Delta\dot{\alpha} \\ \Delta\dot{q} \\ \Delta\dot{\theta} \end{bmatrix} = \begin{bmatrix} -0.045 & 0.036 & 0 & -32.1 \\ -0.369 & -2.02 & 176 & -2.80 \\ 0.0019 & -0.0396 & -2.948 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta q \\ \Delta \theta \end{bmatrix} \quad (3.381)$$

Determine:

- a ) approximate period of the long-period mode
- b ) approximate period of the short-period mode
- c ) the ratio of the approximate settling time of the short-period and the long-period modes

Solution:

- a) For the long-period/phugoid mode

$$\omega_{n,phugoid} = \sqrt{\frac{-Z_u g}{\bar{u}}} = \sqrt{\frac{-(-0.369)(32.1)}{176}} = 0.26 \text{ rad/s} \quad (3.382)$$

$$\zeta_{phugoid} = \frac{-X_u}{2\omega_{n,phugoid}} = \frac{-(-0.045)}{2(0.26)} = 0.087 \quad (3.383)$$

$$T_{phugoid} = \frac{2\pi}{\omega_n \sqrt{1 - \zeta^2}} = \frac{2\pi}{0.26 \sqrt{1 - 0.087^2}} \quad (3.384)$$

$$\underline{T_{phugoid} = 24.2 \text{ s}} \quad (3.385)$$

b) For the short-period

$$\omega_{n,short} = \sqrt{M_q Z_\alpha - M_\alpha \bar{u}} = \sqrt{-2.02(-2.948) - -0.0396(176)} = 3.6 \text{ rad/s} \quad (3.386)$$

$$\zeta_{short} = -\frac{M_q + M_\alpha \bar{u} + Z_\alpha}{2\omega_{n,short}} = -\frac{-2.948 - 2.02}{2(3.6)} = 0.69 \quad (3.387)$$

$$T_{short} = \frac{2\pi}{\omega_{n,short} \sqrt{1 - \zeta_{short}^2}} = \frac{2\pi}{3.6 \sqrt{1 - 0.69^2}} \quad (3.388)$$

$$\underline{T_{short} = 2.4 \text{ s}} \quad (3.389)$$

c) The settling time for  $\zeta < 0.8$  is approximately

$$t_s = \frac{3}{\zeta \omega_n} \quad (3.390)$$

For the long-period/phugoid mode

$$t_{s,phugoid} = \frac{3}{0.087(0.26)} = 132.6 \text{ s} \quad (3.391)$$

and the short-period mode

$$t_{s,short} = \frac{3}{0.69(3.6)} = 1.2 \text{ s} \quad (3.392)$$

which provides a ratio of 1%. This infers that the significant effects of the short-period happen in the first 1% of the significant time period for the long-period/phugoid.

## Example Problem 2

Given: for a general aviation airplane, the following has been calculated for the lateral-directional state-space system (without control)

$$\begin{bmatrix} \Delta \dot{\beta} \\ \Delta \dot{p} \\ \Delta \dot{r} \\ \Delta \dot{\phi} \end{bmatrix} = \begin{bmatrix} -0.254 & 0 & -177.8 & 32.36 \\ -0.0901 & -8.4 & 2.19 & 0 \\ 0.0252 & -0.350 & -0.760 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \beta \\ \Delta p \\ \Delta r \\ \Delta \phi \end{bmatrix} \quad (3.393)$$

Determine: the difference between the true time constants and the approximate time constants

a) roll mode

- b) spiral mode
- c) dutch roll mode

**Solution:**

The eigenvalues using a computer provide

$$\begin{aligned}\lambda &= -0.00891 && \text{Spiral mode} \\ \lambda &= -8.433 && \text{Roll mode} \\ \lambda &= -0.486 \pm 2.34 && \text{Dutch mode}\end{aligned}\quad (3.394)$$

which correspond to

1.  $\tau_{roll} = 0.119$  sec
2.  $\tau_{spiral} = 112$  sec
3.  $\omega_n = 2.39$  rad/s and  $\zeta = 0.20$  for the Dutch roll mode

a) The approximate roll mode is given by

$$\tau_{roll} = -\frac{1}{L_p} \quad (3.395)$$

$$\tau = -\frac{1}{-8.4} \quad (3.396)$$

$$\tau = 0.119 \text{ s} \quad (3.397)$$

which is the same as the computed time constant.

b) Spiral Mode:

$$\tau = \frac{L_v}{L_r N_v - L_v N_r} \quad (3.398)$$

$$\tau = \frac{0.0901}{(2.19)(0.252) - (-0.0901)(-0.760)} \quad (3.399)$$

$$\tau = 6.84 \text{ s} \quad (3.400)$$

which is much faster than the computed time constant.

c) Dutch Roll Mode:

$$\omega_n = \sqrt{Y_v N_r - N_v (Y_r - \bar{u})} \quad (3.401)$$

$$\omega_n = \sqrt{(-0.254)(-0.760) - (0.0252)(-177.8)} \quad (3.402)$$

$$\omega_n = 2.16 \text{ rad/s} \quad (3.403)$$

Then,

$$\zeta = -\frac{Y_v + N_r}{2\omega_n} \quad (3.404)$$

$$\zeta = -\frac{-0.254 + -0.76}{2(2.16)} \quad (3.405)$$

$$\zeta = 0.254 \quad (3.406)$$

which is close to the computed time constant.

## Chapter 4

# Introduction to Control Systems

### 4.1 Open- and Closed-Loop Control Systems

Previously, this part of the textbook has demonstrated how to analyze flight dynamics through the linearization of the equations of motion about trimmed steady flight. For airplanes, this was done in terms of the longitudinal and lateral-directional modal characteristics. The remainder of this part of the textbook will develop linear control theory for designing the control systems for flight vehicles to achieve commanded output responses. At the core of a control system is the design of a **control law**, i.e. a mathematical formula for a system input as a function of the commanded output.

As a motivating example, this section will consider the approximate lateral-directional roll mode for an airplane with the aileron input included, i.e.

$$\dot{p} - L_p p = L_{\delta_a} \delta_a \quad (4.1)$$

where the  $\Delta$ 's have been dropped for ease of notation. Recall that for a unit step response on a first order system, the steady-state output is given by  $y_{ss} = \frac{b_0}{a_0}$ . Thus, one has for the steady-state roll rate

$$p_{ss} = \frac{L_{\delta_a}}{L_p} \quad (4.2)$$

#### Open-Loop Control

In order to control the airplane, one needs to be able to set  $\delta_a(t)$  such that one can achieve a commanded roll rate,  $p_c$ , e.g. to perform a steady turn. A simple solution would be to set

$$\delta_a = \frac{L_p}{L_{\delta_a}} p_c \quad (4.3)$$

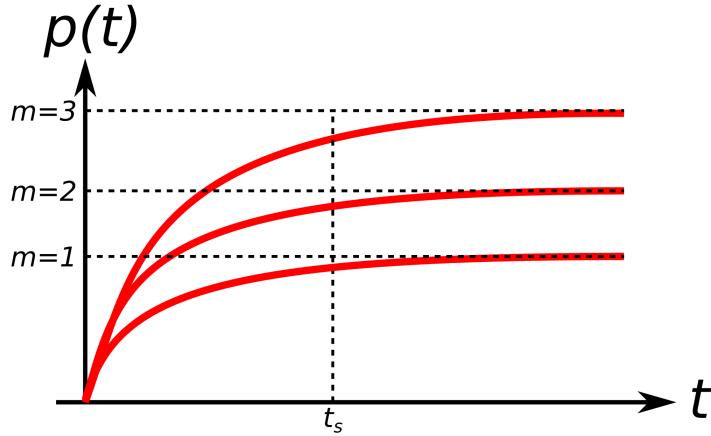
where  $\frac{L_p}{L_{\delta_a}}$  can be considered as the **control gain**. This provides the following ODE by substitution

$$\dot{p} - L_p p = L_p p_c \quad (4.4)$$

and for a step input of magnitude  $m$ , one obtains a solution of

$$p(t) = m + e^{L_p t} (p(0) - m) \quad (4.5)$$

Plotting this for  $m = 1, 2, 3$  and  $p(0) = 0$ ,

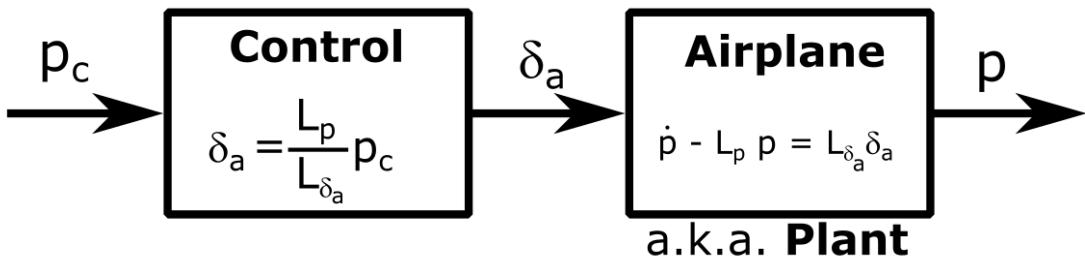


Note that  $t_s$  is the same for each response, i.e. 5% of  $p_{ss}$ , and that the time constant is  $\tau = \frac{1}{L_p}$ .

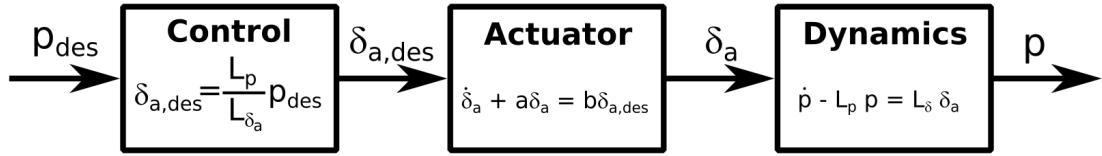
This type of control design is called **open-loop control**. In general, this logically flows as the following

1. the user/pilot specifies a commanded output  $y_c$
2. the control law sets the input  $u$  as a function of that commanded output
3. this dynamically changes the output  $y(t)$

For the open-loop control, one would have a simple block diagram as

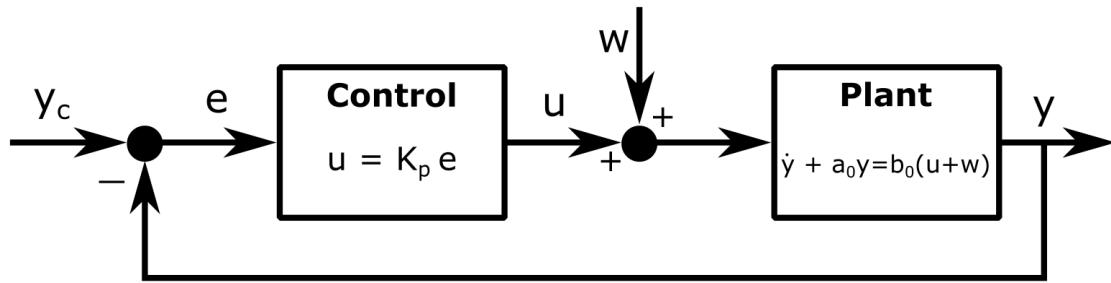


Note that the dynamics block is also known as the **plant** which is short for “powerplant” from the electrical engineering origins of control theory. It should also be noted that in reality  $\delta_a$  will undergo some dynamics, i.e. it won’t instantaneously reach  $\delta_a$ , but is actuated using some electro-mechanical interface which one can represent by an **actuator**. This additional consideration would result in the following block diagram.



### Closed-Loop Control

However, one consideration for real systems is the fact that one will have additional disturbances (e.g. unsteady airflow) and inaccuracies in the dynamics model (e.g. linearization error). These effects can be modeled as an additive disturbance signal,  $w$ , to the control input  $u$ , which for the motivating example  $u = \delta_a$  and results in a block diagram as



From this block diagram, one can infer the overall ODE as

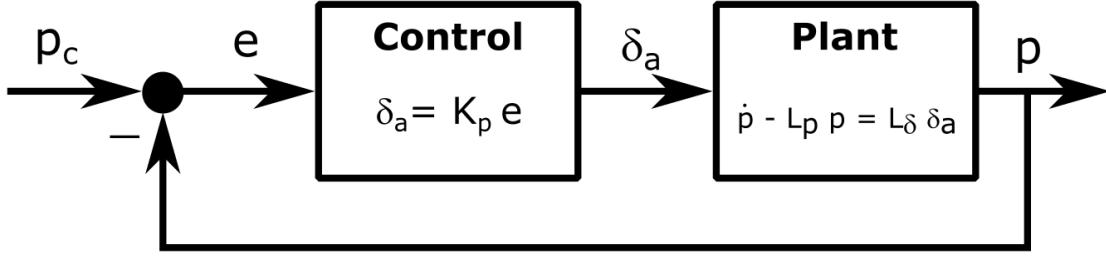
$$\dot{p} - L_p p = L_p p_c + L_{\delta_a} w \quad (4.6)$$

and if  $d(t)$  is some constant,  $\bar{d}$ , then

$$p_{ss} = p_c + \frac{L_{\delta_a}}{L_p} \bar{w} \quad (4.7)$$

where notably, the steady-state output will not reach the commanded roll rate, but has some **steady-state error**,  $e_{ss}$ .

Thus, to reject these disturbances and account for system uncertainties, one must use alternative control strategies, in particular, **closed-loop control**. The simplest form of closed-loop control is **proportional control (P-control)** which can be represented in block diagram form as



where  $e(t)$  is the **tracking error** defined as

$$e(t) = p_c(t) - p(t) \quad (4.8)$$

and  $K_p$  is the **proportional gain**,  $K_p$ , which is multiplied by the tracking error to obtain the proportional control law as

$$\delta_a(t) = K_p e(t) \quad (4.9)$$

In this way, one can use an output of the plant,  $p(t)$ , to compute the tracking error,  $e(t)$ , which directly affects the control input to the plant,  $\delta_a(t)$ , which, in turn, affects the plant output and so on, as time continues. It should be noted that this type of control strategy is also known as **feedback control**, as a plant output signal is fed back to the control law. Lastly, it should be noted that similar to the input signal being actuated in real systems, the output signal must be measured by a **sensor** which estimates the output of the system using physical phenomena that can be converted into a measured signal. This consideration will be addressed later in this course.

One can derive the closed-loop ODE for this feedback control system with a proportional control law from the ODEs for the controller and the plant, i.e.

$$\begin{aligned} \dot{p} - L_p p &= L_{\delta_a} \delta_a \\ \delta_a &= K_p (p_c - p) \end{aligned} \quad (4.10)$$

Combining the two equations, one has

$$\dot{p} - L_p p = L_{\delta_a} K_p (p_c - p) \quad (4.11)$$

and rearranging provides the first order ODE for the control system as

$$\dot{p} + (L_{\delta_a} K_p - L_p) p = L_{\delta_a} K_p p_c \quad (4.12)$$

with equivalent transfer function for the control system as

$$\frac{p(s)}{p_c(s)} = \frac{L_{\delta_a} K_p}{s + L_{\delta_a} K_p - L_p} \quad (4.13)$$

First, note that this control system has a root/pole at  $L_{\delta_a} K_p - L_p$  which depends on the value of  $K_p$ . Thus, one method to visually assess the feedback control system stability characteristics over different potential

values of  $K_p$  for any control system is to plot the closed-loop roots/poles as function of the gain  $K_p$  in the complex plane, a plot known as the **root locus**.

Second, note that the time constant of the first order ODE has been changed by the inclusion of the proportional control law to be

$$\tau = \frac{1}{L_{\delta_a} K_p - L_p} \quad (4.14)$$

Thus, by increasing  $K_p$ , one can reduce  $\tau$ , i.e. make the system converge faster to the steady-state output.

Third, note that the steady-state output is now

$$p_{ss} = \frac{L_{\delta_a} K_p}{L_{\delta_a} K_p - L_p} p_c \quad (4.15)$$

or

$$p_{ss} = \frac{1}{1 - \frac{L_p}{L_{\delta_a} K_p}} p_c \quad (4.16)$$

Thus, for any value of  $K_p$ ,  $p_{ss}$  will not converge exactly to  $p_c$ , but have a steady-state error

$$e_{ss} = p_c - p_{ss} \quad (4.17)$$

or by substitution

$$e_{ss} = \frac{-L_p}{L_{\delta_a} K_p - L_p} p_c \quad (4.18)$$

Thus, one can make  $e_{ss}$  as small as necessary by choosing  $K_p$  sufficiently large.

Now, consider a general first order ODE, i.e.

$$\dot{y} + a_0 y = b_0 u \quad (4.19)$$

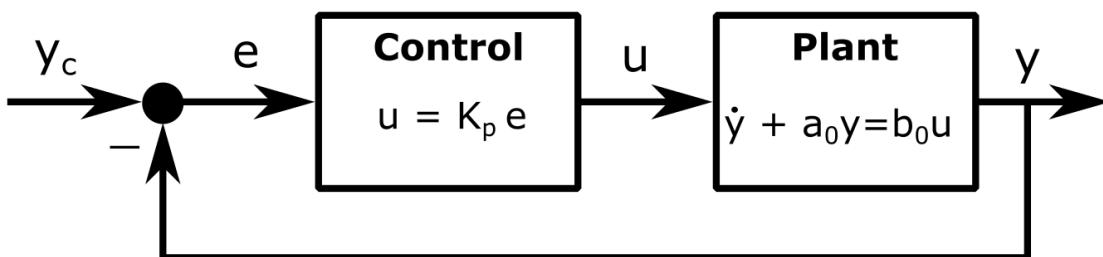
with a control law

$$u = K_p (y_c - y) \quad (4.20)$$

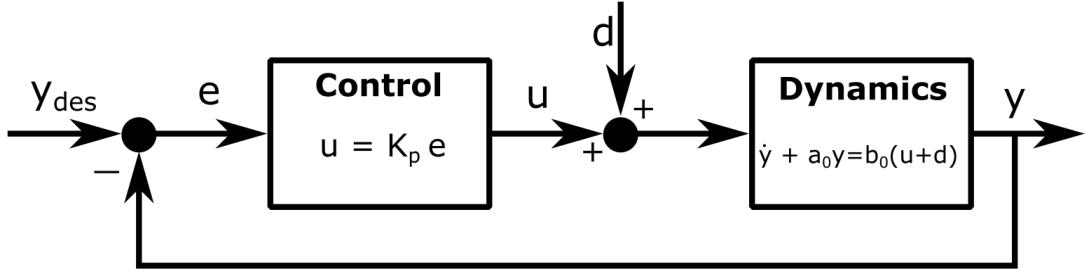
This can be described as the closed-loop ODE

$$\dot{y} + (a_0 + b_0 K_p) y = b_0 K_p y_c \quad (4.21)$$

In block diagram form, one has



Next, consider adding a disturbance signal,  $w$ , to the control input, i.e.



provides the following closed-loop ODE

$$\dot{y} + (a_0 + b_0 K_p) y = b_0 K_p y_c + b_0 w \quad (4.22)$$

where the steady-state error for a constant disturbance,  $w(t) = \bar{w}$ , can be computed as

$$e_{ss} = y_c - y_{ss} \quad (4.23)$$

$$e_{ss} = \frac{a_0}{a_0 + b_0 K_p} y_c - \frac{b_0}{a_0 + b_0 K_p} \bar{w} \quad (4.24)$$

Thus, a proportional control law will more significantly reject disturbances by increasing  $K_p$ . However, though increasing the proportional gain,  $K_p$ , will decrease the settling time and steady-state error even with disturbances, one cannot in reality achieve an arbitrary control input  $\delta_a$  instantaneously, e.g. the mechanical and aerodynamic limits of the aileron deflection angle as well as rate limits. Thus, one must test control laws on a higher fidelity model and/or on the true plant to make sure the simplifications to the plant model for the control design are feasible in practice, especially for control design of linearized plants.

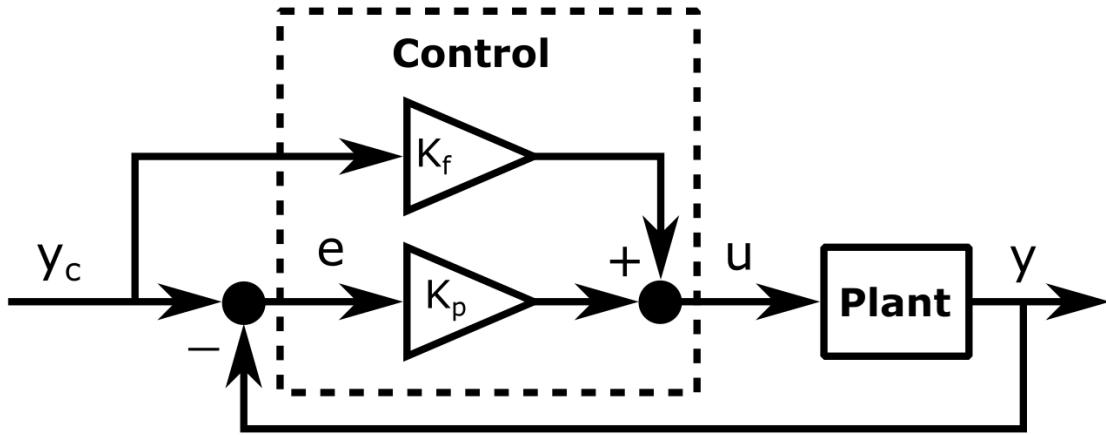
An additional control strategy used to overcome the coupling between the time constant and steady-state error design is **feedforward control** which essentially combines an open-loop control law with a closed-loop closed-loop control, e.g.

$$u = K_p(y_c - y) + K_f y_c \quad (4.25)$$

where  $K_f$  is the **feedforward gain**. This results in a closed-loop ODE of

$$\dot{y} + (a_0 + b_0 K_p) y = b_0(K_p + K_f) y_c \quad (4.26)$$

or in block diagram representation



Note that here the multiplicative gains have been represented as triangles which is the standard representation in block diagrams. In this case, the time constant can be adjusted by changing  $K_p$ , i.e.

$$\tau = \frac{1}{a_0 + b_0 K_p} \quad (4.27)$$

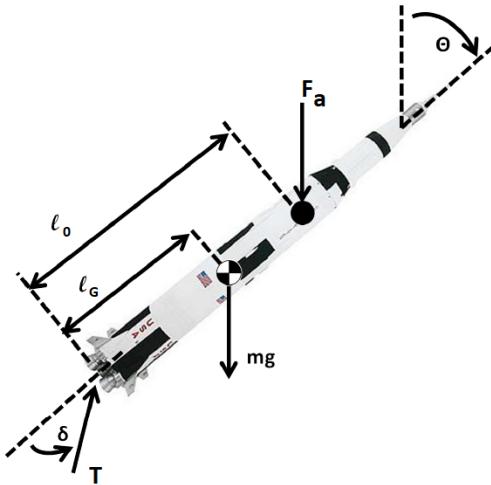
and steady-state error can be adjusted by changing  $K_f$ , i.e.

$$e_{ss} = \frac{a_0 - b_0 K_f}{a_0 + b_0 K_p} y_c \quad (4.28)$$

However, it should be noted that the addition of feedforward control will not improve the disturbance rejection.

## Example Problem

Given: a rocket with thrust vectoring modeled as



for which the linearized dynamics about trim point  $\bar{\theta} = \bar{\delta} = 0$  is

$$\ddot{\theta} - 0.1225\theta = 6.3163\delta \quad (4.29)$$

and a proportional controller:

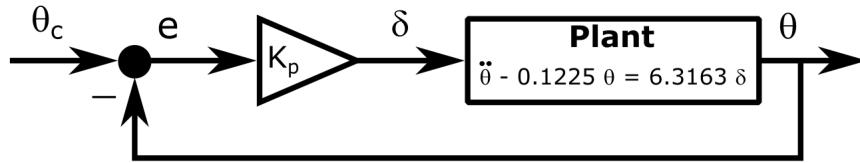
$$\delta(t) = K_p(\theta_c(t) - \theta(t)) \quad (4.30)$$

Determine:

- the block diagram and derive an ODE for the closed-loop model with the proportional controller
- the root locus plot for  $K_p$ . On this plot, label the locations of the closed-loop roots/poles for  $K_p < 0$ ,  $K_p = 0$ , and  $K_p > 0$ .
- if one can stabilize the LTI system using a proportional controller.

Solution:

a)



The system dynamics are

$$\ddot{\theta} - 0.1225\theta = 6.3163\delta \quad (4.31)$$

and with

$$\delta = K_p(\theta_c - \theta) \quad (4.32)$$

the closed-loop model ODE is

$$\ddot{\theta} + (6.3163K_p - 0.1225)\theta = 6.3163K_p\theta_c \quad (4.33)$$

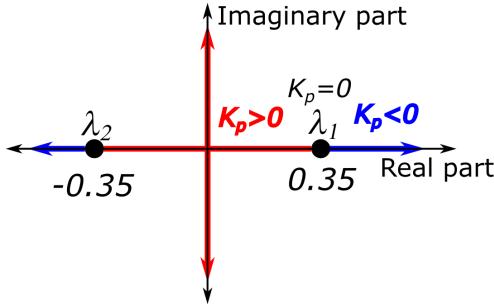
b) The characteristic equation is

$$\lambda^2 + 6.3163K_p\lambda - 0.1225 = 0 \quad (4.34)$$

whose roots are

$$\lambda_{1,2} = \pm\sqrt{-6.3163K_p + 0.1225} \quad (4.35)$$

and can be plotted in the complex plane as



c) Because the system is either marginally stable, i.e. purely imaginary, or unstable, one root in the RHP, one cannot stabilize the system using a proportional controller.

## 4.2 MIMO Feedback Control and Stability Augmentation Systems

The use of proportional control for SISO LTI control system design can be generalized to MIMO LTI systems through **state-space feedback control**, which can be implemented in as state or output feedback. For developing the two variants of state-space feedback, consider the simplified continuous-time LTI state-space representation

$$\begin{aligned}\dot{\vec{x}} &= A\vec{x} + B\vec{u} \\ \vec{y} &= C\vec{x}\end{aligned}\tag{4.36}$$

A general **state feedback control** law can be defined

$$\vec{u} = K_r \vec{r} - K_x \vec{x}\tag{4.37}$$

where  $K_x$  is the feedback gain matrix which acts on the state  $\vec{x}$  being supplied to the controller,  $\vec{r}$  is the reference command signal to the system, and  $K_r$  is the reference gain. In this case, the feedback control system state-space representation is given by

$$\begin{aligned}\dot{\vec{x}} &= A\vec{x} + B(K_r \vec{r} - K_x \vec{x}) \\ \vec{y} &= C\vec{x}\end{aligned}\tag{4.38}$$

or

$$\begin{aligned}\dot{\vec{x}} &= (A - BK_x)\vec{x} + BK_r \vec{r} \\ \vec{y} &= C\vec{x}\end{aligned}\tag{4.39}$$

A general **output feedback control** law can be defined as

$$\vec{u} = K_r \vec{r} - K_y \vec{y}\tag{4.40}$$

where  $K_y$  is the feedback gain matrix which acts on the output  $\vec{y}$  being supplied to the controller. In this case, the feedback control system state-space representation for  $D = 0$  is given by

$$\begin{aligned}\dot{\vec{x}} &= A\vec{x} + B(K_r \vec{r} - K_y \vec{y}) \\ \vec{y} &= C\vec{x}\end{aligned}\tag{4.41}$$

$$\begin{aligned}\dot{\vec{x}} &= A\vec{x} - BK(y)\vec{y} + BK_r\vec{r} \\ \vec{y} &= C\vec{x}\end{aligned}\tag{4.42}$$

$$\begin{aligned}\dot{\vec{x}} &= A\vec{x} - BK_yC\vec{x} + BK_r\vec{r} \\ \vec{y} &= C\vec{x}\end{aligned}\tag{4.43}$$

or

$$\begin{aligned}\dot{\vec{x}} &= (A - BK_yC)\vec{x} + BK_r\vec{r} \\ \vec{y} &= C\vec{x}\end{aligned}\tag{4.44}$$

In both cases, the state matrix of the feedback control system has been altered, as either  $A - BK_x$  for state feedback or  $A - BK_yC$  for output feedback. Thus, one can design  $K_x$  or  $K_y$  to change the stability characteristics of the feedback control system. Subsequently,  $K_r$  can be designed for achieve some steady-state output,  $\vec{y}_{ss}$ , which occurs when  $\dot{\vec{x}} = 0$ . Thus, for state feedback, one has

$$\vec{y}_{ss} = C(A - BK_x)^{-1}BK_r\vec{r}\tag{4.45}$$

and for output feedback, one has

$$\vec{y}_{ss} = C(A - BK_yC)^{-1}BK_r\vec{r}\tag{4.46}$$

Furthermore, using the newly defined state matrices for state and output feedback, one has the **eigenvalue equation** for state feedback as

$$\lambda\vec{v} = (A + BK_x)\vec{v}\tag{4.47}$$

and for output feedback as

$$\lambda\vec{v} = (A + BK_yC)\vec{v}\tag{4.48}$$

where  $\lambda$  is the **eigenvalue** and  $\vec{v}$  is the corresponding **eigenvector**.

Thus, one could “place” the eigenvalues of the feedback control system in the complex plane by choosing the feedback gain matrix appropriately, thereby setting the modal characteristics for the system response. This technique is called **eigenvalue placement**, also known as **pole placement**, and can be considered as a generalization of the root locus technique for selecting a single gain parameter in a SISO LTI control system to selecting a single gain matrix in a MIMO LTI control system and its effect on the system’s root/poles/eigenvalues. However, for MIMO LTI systems, one may not be able to set the eigenvalues arbitrarily as the eigenvalue equations also depend on the plant state matrix,  $A$ , and input matrix,  $B$ , for state feedback, as well as the plant output matrix,  $C$ , for output feedback. The ability to arbitrarily set the eigenvalues leads to the concepts of controllability and observability for the plant which are briefly discussed here and later parts of this textbook develop these ideas more in-depth.

Informally, a plant is **controllable** if for some initial state,  $\vec{x}_0$ , and some final state,  $\vec{x}_f$ , there exists an input function to transfer the plant state from  $\vec{x}_0$  to  $\vec{x}_f$  in a finite amount of time. A consequence of this property is that one can arbitrarily place the eigenvalues of a state feedback control system. There are various tests for controllability, but one test for checking the controllability is through the **controllability matrix**, i.e.

$$C = [B \ AB \ \dots \ A^{n-1}B]\tag{4.49}$$

where  $n$  is the state vector dimension. Then, the test for plant controllability is done by checking if the rank of the controllability matrix has full rank, i.e.  $\text{rank}(C) = n$ , then the plant is controllable. Recall that the **rank of a matrix** is defined as the number of linearly independent rows/columns of that matrix.

Informally, a plant is **observable** if for any possible time history of  $\vec{x}$  and  $\vec{u}$ , the current state can be reconstructed exactly, i.e. “observed,” through knowledge of  $\vec{y}$  and  $\vec{u}$ . A consequence of this property is that one can arbitrarily place the eigenvalues of an output feedback control system. There are various tests for observability, but one test for checking the observability is through the **observability matrix**, i.e.

$$O = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (4.50)$$

Then, the test for observability is done by checking if the rank of the observability matrix has full rank, i.e.  $\text{rank}(O) = n$ , then the plant is observable.

It should be noted that there are computer algorithms to solve the eigenvalue placement problem, e.g. `place` in MATLAB, but derivations for these generic algorithms are beyond the scope of this textbook. However, for the single-input case the state feedback gain matrix (a  $1 \times n$  row vector in this case) can be computed by **Ackermann’s formula**

$$K_x = [0 \ 0 \ \cdots \ 0 \ 1] C^{-1} \Psi_{des}(A) \quad (4.51)$$

where  $\Psi_{des}(A)$  is the desired characteristic equation for the closed-loop system evaluated at  $\lambda = A$ . Recall that the characteristic equation for state feedback as

$$\det(\lambda I - A + BK_x) = 0 \quad (4.52)$$

and for output feedback as

$$\det(\lambda I - A + BK_y C) = 0 \quad (4.53)$$

Also, note that as the controllability matrix is inverted, this assumes that the plant is controllable. In MATLAB, this formula can be used by the function `acker` which is also called by `place` for single-input systems.

Alternatively to tracking some reference command, a feedback control system can be designed only to alter the modal characteristics of the plant, e.g. improving or providing stability to the plant. Thus, this type of control system is called a **stability augmentation system (SAS)** and can be represented by the MIMO control law with state feedback as

$$\vec{u} = \vec{u}_c - K_x \vec{x} \quad (4.54)$$

or for output feedback

$$\vec{u} = \vec{u}_c - K_y \vec{y} \quad (4.55)$$

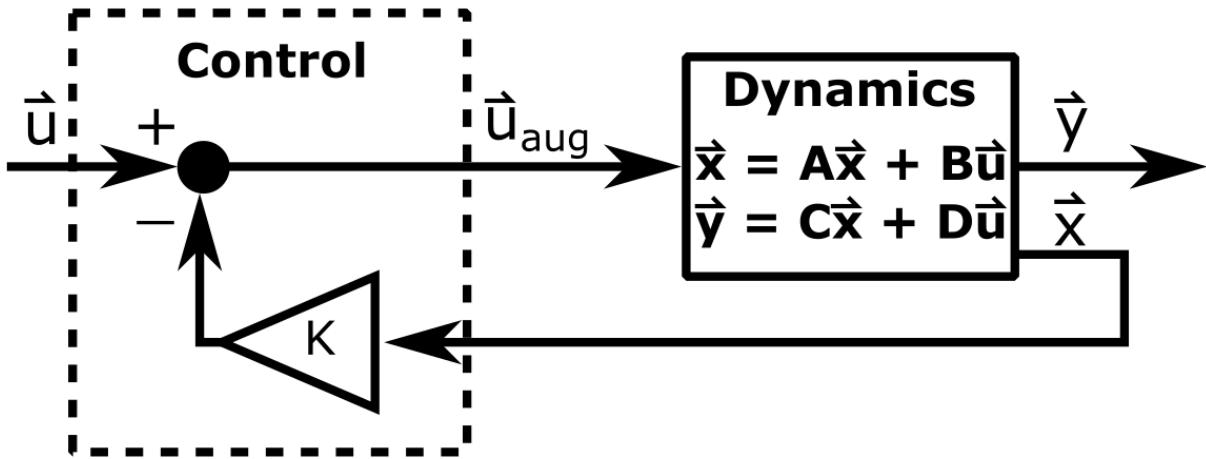
which only use the feedback gain matrices. Accordingly, the state feedback SAS state-space representation for  $D = 0$  is given by

$$\begin{aligned} \vec{x} &= (A - BK_x) \vec{x} + B \vec{u}_c \\ \vec{y} &= C \vec{x} \end{aligned} \quad (4.56)$$

and the output feedback SAS state-space representation for  $D = 0$  is given by

$$\begin{aligned}\dot{\vec{x}} &= (A - BK_y C) \vec{x} + B \vec{u}_c \\ \vec{y} &= C \vec{x}\end{aligned}\quad (4.57)$$

An SAS is typically represented in block diagram form as



### Example Problem

Given: the second order LTI ODE

$$\ddot{y} + a_1 \dot{y} + a_0 y = b_0 u \quad (4.58)$$

with a control law

$$u = K_p(y_c - y) - K_d \dot{y} \quad (4.59)$$

Determine: the control gains,  $K_p$  and  $K_d$ , to place the eigenvalues at  $\alpha \pm \beta j$ .

Solution:

One can formulate this using LTI ODE as a state-space system using the state substitution as

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \dot{y} \\ y \end{bmatrix} \quad (4.60)$$

to get the LTI state-space form

$$\begin{aligned}\dot{\vec{x}} &= \begin{bmatrix} -a_1 & -a_0 \\ 1 & 0 \end{bmatrix} \vec{x} + \begin{bmatrix} b_0 \\ 0 \end{bmatrix} u \\ \vec{y} &= \begin{bmatrix} 0 & 1 \end{bmatrix} \vec{x}\end{aligned}\quad (4.61)$$

as well as rewriting the control as

$$u = K_p y_c - [K_d \quad K_p] \vec{x} \quad (4.62)$$

then, the new LTI state-space system would become

$$\begin{aligned}\dot{\vec{x}} &= \begin{bmatrix} -a_1 & -a_0 \\ 1 & 0 \end{bmatrix} \vec{x} + \begin{bmatrix} b_0 \\ 0 \end{bmatrix} (K_p y_c - [K_d \quad K_p] \vec{x}) \\ \vec{y} &= [0 \quad 1] \vec{x}\end{aligned}\quad (4.63)$$

$$\begin{aligned}\dot{\vec{x}} &= \begin{bmatrix} -a_1 & -a_0 \\ 1 & 0 \end{bmatrix} \vec{x} + \begin{bmatrix} b_0 \\ 0 \end{bmatrix} K_p y_c - \begin{bmatrix} b_0 K_d & b_0 K_p \\ 0 & 0 \end{bmatrix} \vec{x} \\ \vec{y} &= [0 \quad 1] \vec{x}\end{aligned}\quad (4.64)$$

$$\begin{aligned}\dot{\vec{x}} &= \begin{bmatrix} -a_1 - b_0 K_d & -a_0 - b_0 K_p \\ 1 & 0 \end{bmatrix} \vec{x} + \begin{bmatrix} b_0 \\ 0 \end{bmatrix} K_p y_c \\ \vec{y} &= [0 \quad 1] \vec{x}\end{aligned}\quad (4.65)$$

Then eigenvalues/poles are given by the determinant

$$\det \left( \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} - \begin{bmatrix} -a_1 - b_0 K_d & -a_0 - b_0 K_p \\ 1 & 0 \end{bmatrix} \right) = 0 \quad (4.66)$$

which provides the characteristic equation as

$$\lambda^2 + (a_1 + b_0 K_d)\lambda + a_0 + b_0 K_p = 0 \quad (4.67)$$

which provides the eigenvalues as

$$\lambda_{1,2} = \frac{a_1 + b_0 K_d}{2} \pm \frac{1}{2} \sqrt{(a_1 + b_0 K_d)^2 - 4(a_0 + b_0 K_p)} \quad (4.68)$$

Thus, any eigenvalues can be placed at  $\alpha \pm \beta j$  in the complex plane by setting  $K_d$  and  $K_p$  appropriately. In particular,

$$\alpha = \frac{a_1 + b_0 K_d}{2} \quad (4.69)$$

or

$$K_d = \frac{2\alpha - a_1}{b_0} \quad (4.70)$$

and

$$\beta j = \frac{1}{2} \sqrt{(a_1 + b_0 K_d)^2 - 4(a_0 + b_0 K_p)} \quad (4.71)$$

or

$$-4\beta^2 = \frac{1}{2} \sqrt{(a_1 + b_0 K_d)^2 - 4(a_0 + b_0 K_p)} \quad (4.72)$$

$$4(a_0 + b_0 K_p) = \left( a_1 + b_0 \frac{2\alpha - a_1}{b_0} \right)^2 + 4\beta^2 \quad (4.73)$$

$$K_p = \frac{1}{4b_0} \left( a_1 + b_0 \frac{2\alpha - a_1}{b_0} \right)^2 + \frac{\beta^2 - a_0}{b_0} \quad (4.74)$$

Note that one could also have used the closed-loop ODE

$$\ddot{y} + (a_1 + b_0 K_d)\dot{y} + (a_0 + b_0 K_p)y = K_p y_c \quad (4.75)$$

to get the characteristic equation.

### 4.3 Proportional-Integral Control

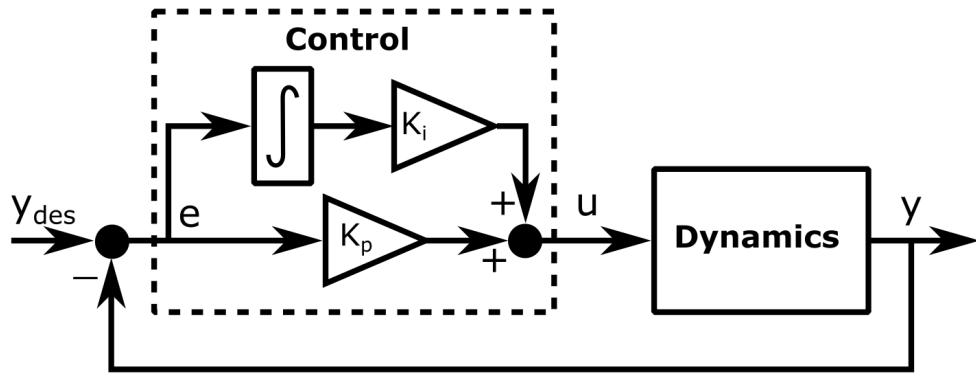
A **proportional-integral (PI)** control law for a SISO system is defined as

$$u(t) = K_p (y_c(t) - y(t)) + K_i \int_0^t [y_c(\tau) - y(\tau)] d\tau \quad (4.76)$$

where the added second term depends on the *integral* of the tracking error,  $e(t) = y_c(t) - y(t)$ . Thus, this control law can be rewritten as

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau \quad (4.77)$$

and can be drawn in block diagram form as



where an integrator block has been included in the block diagram. PI-controllers eliminate steady-state errors due to *any* constant commanded output,  $y_c$ , since as  $t$  increases, the integral contribution in the control law will increase until  $e = 0$ , then the integral contribution remains constant and the control input will also remain constant at the steady-state input,  $u_{ss}$ .

To illustrate this concept, consider the closed-loop ODE for a first order LTI system. Substituting this PI controller into a general first order system results in the ODE

$$\dot{y} + a_0 y = b_0 K_p (y_c - y) + b_0 K_i \int_0^t [y_c(\tau) - y(\tau)] d\tau \quad (4.78)$$

and differentiating with respect to  $t$ , one has

$$\ddot{y} + a_0 \dot{y} = b_0 K_p (\dot{y}_c - \dot{y}) + b_0 K_i (y_c - y) \quad (4.79)$$

Then, rearranging and assuming  $y_c$  is constant in time, one has

$$\ddot{y} + (a_0 + b_0 K_p) \dot{y} + b_0 K_i y = b_0 K_i y_c \quad (4.80)$$

which is a second order LTI system which is stable if and only if

$$\begin{aligned} a_0 + b_0 K_p &> 0 \\ b_0 K_i &> 0 \end{aligned} \quad (4.81)$$

which can be rewritten for  $b_0 > 0$ , then

$$\begin{aligned} K_p &> -\frac{a_0}{b_0} \\ K_i &> 0 \end{aligned} \quad (4.82)$$

else for  $b_0 < 0$ , then

$$\begin{aligned} K_p &< -\frac{a_0}{b_0} \\ K_i &< 0 \end{aligned} \quad (4.83)$$

Furthermore, if the system is stable and  $y_c$  is constant, then if  $\ddot{y} = \dot{y} = 0$ , then one has

$$K_i y_{ss} = K_i y_c \quad (4.84)$$

$$y_{ss} = y_c \quad (4.85)$$

and finally

$$e_{ss} = 0 \quad (4.86)$$

which was the purpose of adding the integral term to the control law.

To analyze the effects of the control gains on the system response characteristics, recall the closed-loop model as

$$\ddot{y} + (a_0 + b_0 K_p) \dot{y} + b_0 K_i y = b_0 K_i y_c \quad (4.87)$$

which has a characteristic equation of

$$\lambda^2 + (a_0 + b_0 K_p) \lambda + b_0 K_i = 0 \quad (4.88)$$

and recalling the alternative characteristic equation form for second order systems as

$$\lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2 = 0 \quad (4.89)$$

this system has roots/poles of

$$\begin{aligned} \lambda_1 &= -\zeta\omega_n + \omega_n \sqrt{\zeta^2 - 1} \\ \lambda_2 &= -\zeta\omega_n - \omega_n \sqrt{\zeta^2 - 1} \end{aligned} \quad (4.90)$$

Then, one can compute the undamped natural frequency as

$$\omega_n = \sqrt{b_0 K_i} \quad (4.91)$$

and the damping ratio as

$$\zeta = \frac{a_0 + b_0 K_p}{2\sqrt{b_0 K_i}} \quad (4.92)$$

Thus, by inspection if  $b_0 > 0$ , then one can conclude

1.  $\uparrow K_p \rightarrow \uparrow \zeta \text{ & } \uparrow \zeta \omega_n$
2.  $\uparrow K_i \rightarrow \uparrow \omega_n \text{ & } \downarrow \zeta$ , but no effect on  $\zeta \omega_n$

else if  $b_0 < 0$ , then one can conclude

1.  $\downarrow K_p$  (i.e. more negative)  $\rightarrow \uparrow \zeta \text{ & } \uparrow \zeta \omega_n$
2.  $\downarrow K_i$  (i.e. more negative)  $\rightarrow \uparrow \omega_n \text{ & } \downarrow \zeta$ , but no effect on  $\zeta \omega_n$

From this analysis, the step response for a first order LTI system can be designed completely by selecting appropriate  $K_p$  and  $K_i$  as these gains will directly impact the settling time,  $t_s$ , and maximum overshoot,  $M_p$ . In particular, for underdamped systems

$$M_p = e^{-\left(\frac{\zeta}{\sqrt{1-\zeta^2}}\right)\pi} \quad (4.93)$$

and for  $\zeta < 0.8$

$$t_s \approx \frac{3}{\zeta \omega_n} \quad (4.94)$$

Thus, if  $b_0 > 0$ , then one can conclude

1.  $\uparrow K_p \rightarrow \downarrow M_p \text{ & } \downarrow t_s$
2.  $\uparrow K_i \rightarrow \uparrow M_p \text{ & } \text{no effect on } t_s$

else if  $b_0 < 0$ , then one can conclude

1.  $\downarrow K_p \rightarrow \downarrow M_p \text{ & } \downarrow t_s$
2.  $\downarrow K_i \rightarrow \uparrow M_p \text{ & } \text{no effect on } t_s$

For overdamped systems, as  $\zeta \uparrow$  the slower root at  $r_1 \rightarrow 0$ , i.e. gets slower. Thus, for  $b_0 > 0$ , then one can conclude

1.  $\uparrow K_p \rightarrow \uparrow t_s$
2.  $\uparrow K_i \rightarrow \downarrow t_s$

else if  $b_0 < 0$ , then one can conclude

1.  $\downarrow K_p \rightarrow \uparrow t_s$
2.  $\downarrow K_i \rightarrow \downarrow t_s$

Next, consider a second order dynamic equation

$$\ddot{y} + 2\dot{y} + 101y = 101u \quad (4.95)$$

which is stable since coefficients  $> 0$ ). The characteristic equation is

$$\lambda^2 + 2\lambda + 101 = 0 \quad (4.96)$$

whose roots/poles are

$$\lambda_{1,2} = -1 \pm 10j \quad (4.97)$$

the undamped natural frequency is

$$\omega_n = \sqrt{101} \quad (4.98)$$

and the damping ratio is

$$\zeta = \frac{2}{2\omega_n} = \frac{1}{\sqrt{101}} \approx 0.0995 \ll 1 \quad (4.99)$$

Note that this system is very underdamped and thus the step response will have a large maximum overshoot, i.e.

$$M_p = e^{\frac{-\zeta}{\sqrt{1-\zeta^2}} \pi} \quad (4.100)$$

$$M_p = e^{\frac{-0.0995}{\sqrt{1-0.0995^2}} \pi} \quad (4.101)$$

$$M_p = 0.73 \quad (4.102)$$

and a long settling time, i.e.

$$t_s = \frac{3}{\zeta\omega_n} \quad (4.103)$$

$$t_s = \frac{3}{\frac{1}{\sqrt{101}}\sqrt{101}} \quad (4.104)$$

$$t_s = 3 \text{ s} \quad (4.105)$$

Suppose that one would like to design a controller that would make the closed-loop response be more damped and reduce the settling time. First, consider a proportional control law

$$u(t) = K_p(y_c - y) \quad (4.106)$$

Then, the closed-loop model is

$$\ddot{y} + 2\dot{y} + 101x = 101 [K_p(y_c - y)] \quad (4.107)$$

$$\ddot{y} + 2\dot{y} + [101 + 101K_p]y = 101K_p(y_c) \quad (4.108)$$

The characteristic equation is

$$\lambda^2 + 2\lambda + [101 + 101K_p] = 0 \quad (4.109)$$

Recall: the closed-loop system is stable if the roots/poles have a negative real part (i.e. in the left half plane). For second order systems, this is true if both coefficients are positive, i.e.

$$\begin{aligned} 2 &> 0 \\ 101 + 101K_p &> 0 \end{aligned} \quad (4.110)$$

which is true, if

$$\underline{K_p > -1} \quad (4.111)$$

The natural frequency is

$$\omega_n = \sqrt{101 + 101K_p} \quad (4.112)$$

The damping is

$$\zeta = \frac{2}{2\sqrt{101 + 101K_p}} \quad (4.113)$$

$$\zeta = \frac{1}{\sqrt{101 + 101K_p}} \quad (4.114)$$

The settling time is

$$t_s = \frac{3}{\zeta \omega_n} \quad (4.115)$$

which is constant at

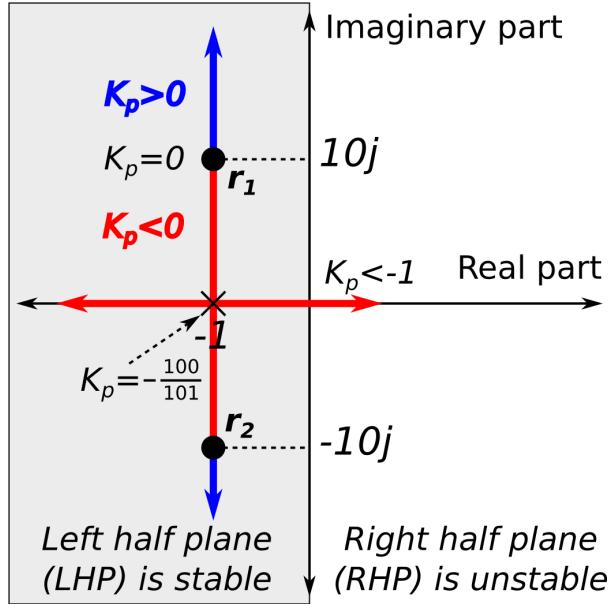
$$t_s = 3 \quad (4.116)$$

Thus, one can change  $\zeta$  using P-control, but can't affect  $t_s$ . Furthermore, assuming the system is stable, i.e. the roots/poles of the system are in the left-half plane (LHP), then the roots can be solved as

$$r_{1,2} = \frac{-2}{2} \pm \frac{\sqrt{4 - 4(101 + 101K_p)}}{2} \quad (4.117)$$

$$r_{1,2} = -1 \pm \sqrt{-100 - 101K_p} \quad (4.118)$$

and the root locus can be drawn as



From the root locus, one can deduce the following:

1.  $K_p > -\frac{100}{101}$ , then  $\zeta < 1$  and  $\zeta \omega_n = -1$ .
2.  $K_p = -\frac{100}{101}$ , then  $\zeta = 1$  and both roots/poles are at  $-1$ .
3.  $-1 < K_p < -\frac{100}{101}$ , then  $\zeta > 1$  and there is one real pole  $< -1$  and one real pole between  $-1$  and  $0$ .
4.  $K_p = -1$ , then one root/pole is at  $r = 0$  and the other is at  $-2$ . The closed-loop is marginally stable.
5.  $K_p < -1$ , there is a root/pole with a positive real part. The closed-loop is unstable.

It should be noted that in order to increase  $\zeta$ ,  $-1 < K_p < 0$  which creates an adverse impact on the steady-state value. Recall,

$$y_{ss} = \frac{K_p}{1 + K_p} y_c \quad (4.119)$$

which would be  $< 0$  when  $y_c > 0$ . One could reverse the sign for  $y_c$ , but the steady-state error is still large. Here, adding integral control would create a third order system for which analytical equations have not been explicitly derived in this textbook. However, an integral term would eliminate the steady-state error, but it wouldn't be sufficient as an integral term only typically affects long-term behavior. Thus, an alternative control strategy is needed to increase the damping of the system while also decreasing the settling time which is introduced in the next section.

### Example Problem

Given: the plant

$$\dot{y} + y = u \quad (4.120)$$

Determine: a feedback controller such that

1. The closed-loop is stable.
2. The unit step response satisfies:
  - a)  $t_s \leq 2$  sec,
  - b)  $|e_{ss}| \leq 0.01$ ,
  - c)  $|u| \leq 5$ , and
  - d) no overshoot.

Solution:

First, consider a proportional control law

$$u(t) = K_p [y_c - y] \quad (4.121)$$

Then, the closed-loop model is

$$\dot{y} + y = K_p [y_c - y] \quad (4.122)$$

$$\dot{y} + [1 + K_p] y = K_p y_c \quad (4.123)$$

To satisfy (1), the closed-loop is stable if

$$1 + K_p > 0 \quad (4.124)$$

which requires

$$\underline{K_p > -1} \quad (4.125)$$

To satisfy (2a), the time constant,  $\tau = \frac{1}{1+K_p}$ , and

$$t_s = \frac{3}{1 + K_p} \leq 2 \quad (4.126)$$

which requires

$$\underline{K_p \geq \frac{1}{2}} \quad (4.127)$$

To satisfy (2b) with  $y_c = 1$ , consider

$$y_{ss} = \frac{K_p}{K_p + 1} \quad (4.128)$$

or

$$e_{ss} = \frac{1}{K_p + 1} \quad (4.129)$$

which for  $|e_{ss}| \leq 0.01$  and  $K_p > -1$

$$\frac{1}{K_p + 1} \leq 0.01 \quad (4.130)$$

$$\underline{K_p \geq 99} \quad (4.131)$$

To satisfy (2c), the input is given by

$$u = K_p [y_c - y] \leq 5 \quad (4.132)$$

For a first order response that is initially at  $y(0) = 0$  for a unit step input and converges to  $y_{ss} = 1$ , the largest  $[y_c - y]$  will occur at 0 and be equal to  $e(0) = 1$ . Thus,

$$\underline{K_p \leq 5} \quad (4.133)$$

(2d) is naturally satisfied as a first order system has no overshoot.

However, due to the conflicting requirements for (2b) and (2c), P-control is unsatisfactory.

Next, consider a proportional-integral control law, i.e.

$$u = K_p (y_c - y) + K_i \int_0^t [y_c(\tau) - y(\tau)] d\tau \quad (4.134)$$

which results in a closed-loop ODE as

$$\dot{y} + y = K_p (y_c - y) + K_i \int_0^t [y_c(\tau) - y(\tau)] d\tau \quad (4.135)$$

and differentiating, one has

$$\ddot{y} + \dot{y} = K_p (\dot{y}_{des} - \dot{y}) + K_i (y_c - y) \quad (4.136)$$

and rearranging results in

$$\ddot{y} + (1 + K_p)\dot{y} + K_i y = K_p \dot{y}_{des} + K_i y_c \quad (4.137)$$

To satisfy (1), the closed-loop is stable if

$$\begin{aligned} 1 + K_p &> 0 \\ K_i &> 0 \end{aligned} \quad (4.138)$$

$$\frac{K_p > -1}{\underline{K_i > 0}} \quad (4.139)$$

To satisfy (2b),  $e_{ss} = 0$  is automatically satisfied if  $\underline{K_i > 0}$  because integral control is being used.

(2d) implies that the closed-loop system must be overdamped ( $\zeta > 1$ ) or critically damped ( $\zeta = 1$ ). It will be fastest if  $\zeta = 1$ , so one can choose  $K_p$  and  $K_i$  accordingly.

To satisfy (2a), recall that for second order critically damped systems,

$$t_s \approx \frac{4.744}{\omega_n} \quad (4.140)$$

or

$$\frac{4.744}{\omega_n} \leq 2 \quad (4.141)$$

which requires

$$\underline{\omega_n \geq 2.372} \quad (4.142)$$

for the fastest response. Then, for the integral gain, one has

$$K_i = \omega_n^2 \quad (4.143)$$

which results in

$$\underline{K_i \geq 5.626} \quad (4.144)$$

and for the proportional gain, one has

$$1 + K_p = 2\zeta\omega_n \quad (4.145)$$

or

$$K_p = 2\omega_n - 1 \quad (4.146)$$

which results in

$$\underline{K_p \geq 3.744} \quad (4.147)$$

To satisfy (2c), since the proposed system is critically damped, there will be no overshoot and the error,  $e$ , from  $y(0) = 0$  to  $y_c = 1$  will exponentially decay from  $1 \rightarrow 0$ . Thus, at  $t = 0$

$$u(0) = K_p(1) + \int_0^0 e(\tau)d\tau \quad (4.148)$$

or

$$u(0) = K_p \quad (4.149)$$

and the maximum contribution from the proportional term is  $K_p$  since as  $e \rightarrow 0$ , the proportional term will decrease. However, the integral term increases from  $0 \rightarrow u_{ss}$  which occurs when  $e = 0$ . To find  $u_{ss}$ , first solve for  $y_{ss}$  when  $\dot{y} = 0$ , i.e.

$$0 + y_{ss} = u_{ss} \quad (4.150)$$

Thus, for  $y_c = 1$

$$u_{ss} = 1 \quad (4.151)$$

Thus, one requires

$$\underline{K_p \leq 5} \quad (4.152)$$

From this example, it should be noted that for most PI gains, the largest input during a unit step input is given by  $K_p$ .

This analysis has shown that a PI-controller is satisfactory with control gains:

$$\begin{aligned} \underline{K_p = 3.744} \\ \underline{K_i = 5.626} \end{aligned} \quad (4.153)$$

Note that if one could not find gains to satisfy all design requirements, then it would have been necessary to relax the requirements (e.g. increase  $t_s$ ) or try a more advanced control algorithm.

## 4.4 Proportional-Integral-Derivative Control

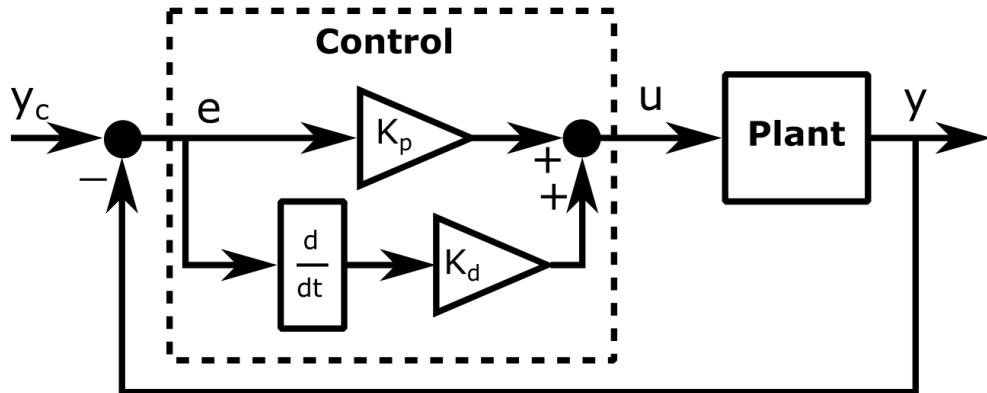
A proportional-derivative (PD) control law for a SISO system is defined as

$$u(t) = K_p(y_{des}(t) - y(t)) + K_d \frac{d}{dt}(y_{des}(t) - y(t)) \quad (4.154)$$

or in terms of the tracking error

$$u(t) = K_p e(t) + K_d \dot{e}(t) \quad (4.155)$$

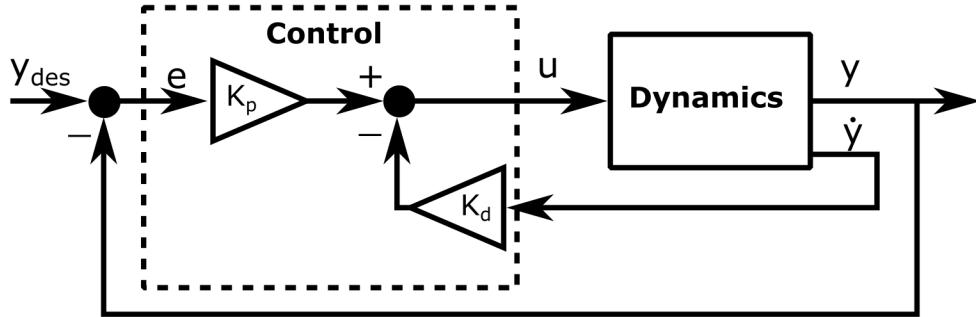
where it will be shown that the addition of the derivative term allows one to alter the damping and speed of response (e.g. settling time). This can be represented in block diagram form as:



In real applications, it is difficult to accurately differentiate the tracking error signal since it will be subject to noise/uncertainty, so often the derivative term is alternatively implemented using **rate feedback control** where the derivative of the output, i.e. “rate,” is fed back. Thus, for a proportional control law with rate feedback

$$u(t) = K_p(y_{des}(t) - y(t)) - K_d \dot{y}(t) \quad (4.156)$$

This can be represented in block diagram form as:



However, this may be understood as a SIMO feedback control system as two separate outputs are fed back to the controller, the error and the output rate.

As an example of adding derivative action to a control law, recall the second order LTI system from the previous section, i.e.

$$\ddot{y} + 2\dot{y} + 101y = 101u \quad (4.157)$$

and using proportional control with rate feedback, i.e.

$$u = K_p(y_{des} - y) - K_d\dot{y} \quad (4.158)$$

the closed-loop ODE becomes

$$\ddot{y} + 2\dot{y} + 101y = 101 [K_p(y_{des} - y) - K_d\dot{y}] \quad (4.159)$$

$$\ddot{y} + (2 + 101K_d)\dot{y} + (101 + 101K_p)y = 101K_p y_{des} \quad (4.160)$$

It should be noted that in PD-control, one would have a  $\dot{y}_{des}$  term. This would add an additional zero to the system which can be analyzed either in direct simulation or through other control design methods.

The characteristic equation for this closed-loop system is

$$\lambda^2 + (2 + 101K_d)\lambda + (101 + 101K_p) = 0 \quad (4.161)$$

For second order systems, this is stable if both coefficients are positive, i.e.

$$\begin{aligned} 2 + 101K_d &> 0 \\ 101 + 101K_p &> 0 \end{aligned} \quad (4.162)$$

which is true, if

$$\begin{aligned} K_d &> \frac{-2}{101} \\ K_p &> -1 \end{aligned} \quad (4.163)$$

The undamped natural frequency is

$$\omega_n = \sqrt{101 + 101K_p} \quad (4.164)$$

and the damping ratio is

$$\zeta = \frac{2 + 101K_d}{2\omega_n} \quad (4.165)$$

or

$$\zeta = \frac{2 + 101K_d}{2\sqrt{101 + 101K_p}} \quad (4.166)$$

Furthermore, the settling time is

$$t_s = \frac{3}{\zeta\omega_n} \quad (4.167)$$

or

$$t_s = \frac{6}{2 + 101K_d} \quad (4.168)$$

Thus, one can change  $t_s$  directly by choosing the derivative gain appropriately. The steady-state error is defined as

$$e_{ss} = y_{des} - y_{ss} \quad (4.169)$$

$$e_{ss} = y_{des} - \frac{101K_p y_{des}}{101 + 101K_p} \quad (4.170)$$

$$e_{ss} = \frac{1}{1 + K_p} y_{des} \quad (4.171)$$

Thus,  $K_p$  directly affects the steady-state error which could be made 0 by including an integral gain as described in the next subsection.

Lastly, recall the maximum overshoot is

$$M_p = e^{-\frac{\zeta}{\sqrt{1-\zeta^2}}\pi} \quad (4.172)$$

or

$$M_p = e^{\frac{-\pi(2+101K_d)}{(2\sqrt{101+101K_p})}\sqrt{1-\left(\frac{2+101K_d}{2\sqrt{101+101K_p}}\right)^2}} \quad (4.173)$$

which is a function of both control gains.

### Proportional-Integral-Derivative (PID) Control

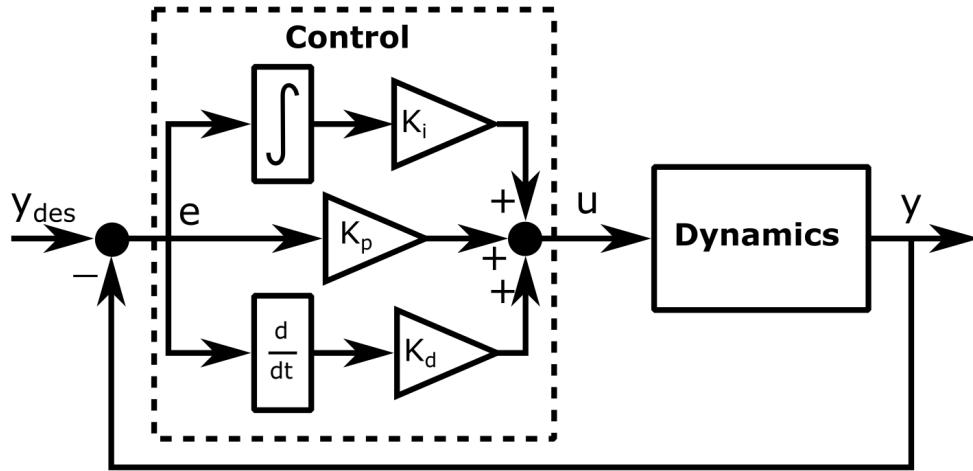
A **proportional-integral-derivative (PID)** control law for a SISO system is defined as

$$u(t) = K_p(y_{des}(t) - y(t)) + K_i \int_0^t [y_{des}(\tau) - y(\tau)] d\tau + K_d \frac{d}{dt}(y_{des}(t) - y(t)) \quad (4.174)$$

or in terms of the tracking error

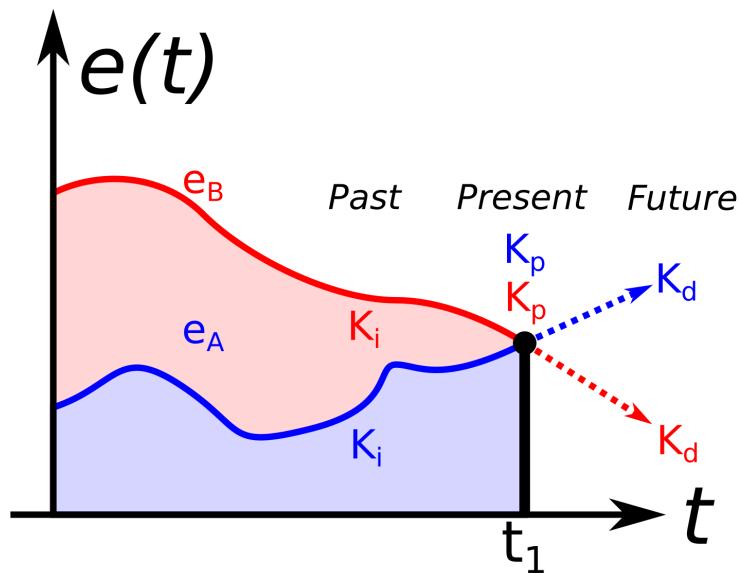
$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \dot{e}(t) \quad (4.175)$$

which can be drawn in block diagram form as



It should also be noted that rate feedback is often used instead of the derivative term which would result in a proportional-integral control with rate feedback.

An intuitive understanding of PID control is to look at the following diagram of the tracking error history for two different systems as shown below.



Here the proportional term would notably have the same correction effect here at the present time,  $t_1$ , while the integral term has a correction effect based on the *past* error and continues to grow until the controller reaches  $u_{ss}$  for  $e_{ss} = 0$ . Lastly, the derivative term has a correction effect based on the *future* trend of the tracking error, i.e. its rate of change, and thus can improve the speed of response and the damping. Note that here again, often the derivative term is replaced by a rate feedback term instead, but the effect will be similar.

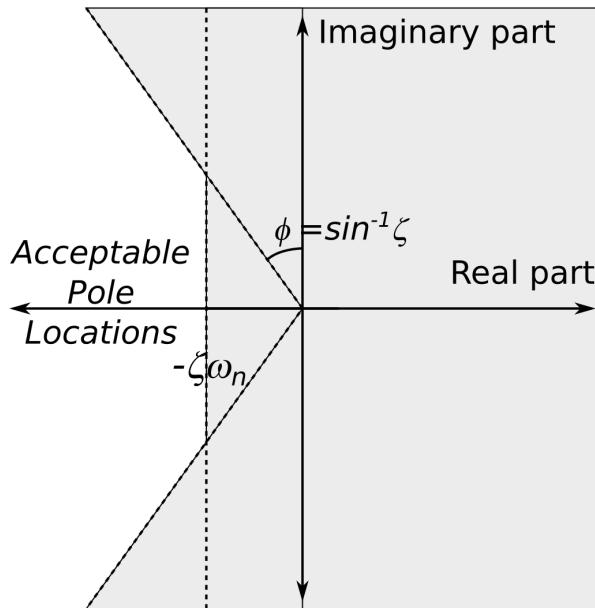
When designing PID controllers, one typically uses the unit step response (or other similar time domain responses) to specific the LTI system requirements and “tune” the control gains. For general LTI systems of any order, the requirements on maximum overshoot,  $M_p$ , and settling time,  $t_s$ , roughly correspond to requirements on  $\zeta$  and  $\zeta\omega_n$ , respectively. Considering the approximate second order LTI system relationships, i.e.

$$M_p \approx e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}} \quad (4.176)$$

and

$$t_s \approx \frac{3}{\zeta\omega_n} \quad (4.177)$$

if the individual poles of the LTI system are then plotted in the complex plane, e.g.



one should design control gains such that the poles are placed in the white region. One can also note that if the zeros and poles are not “near” the rightmost real pole or pair of complex poles, e.g.  $> 4\times$  the real part, then the rightmost pole or complex pair is “dominant” and the system is approximated well by the first or second order system of the dominant poles and zeros.

As an example of this approach to PID control design, consider the following requirements on a unit step response have been given for a control design problem:

1.  $t_s \leq 0.1$  s
2.  $e_{ss} \leq 10\%$
3.  $M_p \leq 5\%$

Starting with requirement (3)

$$M_p = e^{-\frac{\zeta}{\sqrt{1-\zeta^2}}\pi} \leq 0.05 \quad (4.178)$$

which simplifies to

$$\zeta \geq 0.69 \quad (4.179)$$

Then, choosing

$$\zeta = 0.8 \quad (4.180)$$

one has

$$t_s = \frac{3}{(0.8)\omega_n} \leq 0.1 \quad (4.181)$$

or

$$\omega_n \geq 37.5 \quad (4.182)$$

Next, choosing

$$\omega_n = 37.5 \quad (4.183)$$

one has

$$K_p = \frac{\omega_n^2}{101} - 1 \quad (4.184)$$

$$K_p = \frac{37.5^2}{101} - 1 \quad (4.185)$$

or

$$K_p = 12.92 \quad (4.186)$$

which must fulfill the requirement for  $e_{ss}$ , i.e.

$$\frac{e_{ss}}{y_{des}} = \frac{1}{1 + K_p} \leq 0.1 \quad (4.187)$$

or

$$K_p \geq 9 \quad (4.188)$$

$$12.92 \geq 9 \quad (4.189)$$

which is true. Then, computing the derivative gain

$$K_d = \frac{2(0.8)(37.5) - 2}{101} \quad (4.190)$$

$$K_d = 0.57 \quad (4.191)$$

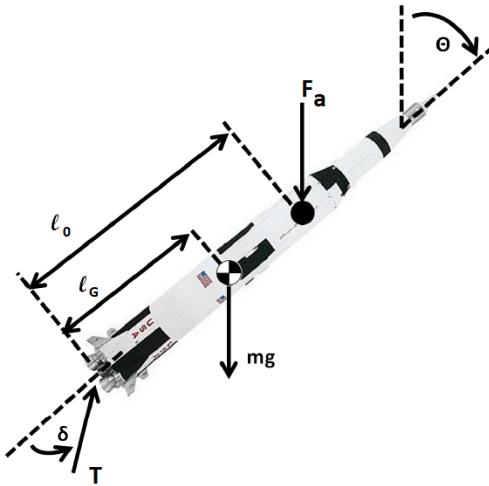
and

$$M_p = e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}} = 1.5\% \quad (4.192)$$

It should be noted that the actual unit step response has an overshoot that does not satisfy the requirements since the previous equation is an approximation. Thus, in practice, one typically simulates the higher order systems and tunes the gains appropriately until the unit step response characteristics are satisfied. For the previous example in this lecture, the true  $M_p = 19\%$  which is much larger than 1.5%, so one would need to retune the gains,  $K_p$  and  $K_d$  until a suitable response is identified. However, when using these manual PID tuning methods, one must typically have an intuition for how the three gains affect the different aspects of the response and also balance the amount of **control effort**, i.e.  $|u|$ , that is required as the control effort is usually limited in the real system. Lastly, it is important to note that automatic tools for PID tuning also exist, however, the consideration of these tools is beyond the scope of this course. Later, this course will derive an alternative to step response PID tuning for control design.

### Example Problem

Given: a rocket with thrust vectoring modeled as



for which the linearized dynamics EOM about trim point  $\bar{\theta} = \bar{\delta} = 0$  is

$$\ddot{\theta} - 0.1225\theta = 6.3163\delta \quad (4.193)$$

and a proportional controller with rate feedback:

$$\delta(t) = K_p(\theta_{des}(t) - \theta(t)) - K_d\dot{\theta}(t) \quad (4.194)$$

with control design requirements as:

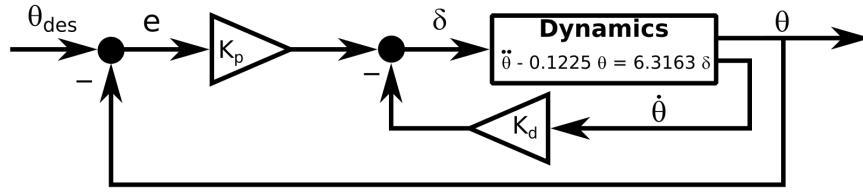
1. The closed-loop must be stable.
2. The step response for any  $\theta_{des} \leq 0.1$  rad must have:
  - a)  $t_s < 2$  s;
  - b)  $M_p < 5\%$ ; and
  - c)  $e_{ss} < 0.005$  rad.

Determine:

- a) the block diagram and the ODE for the closed-loop model with the proportional-derivative controller; and
- b) choose the gains,  $K_p$  and  $K_d$  to make the closed-loop stable and satisfy the control design requirements.

Solution:

- a)



The linearized system ODE is

$$\ddot{\theta} - 0.1225\theta = 6.3163\delta \quad (4.195)$$

and with

$$\delta = K_p(\theta_{des} - \theta) - K_d\dot{\theta} \quad (4.196)$$

the closed-loop model ODE is

$$\ddot{\theta} + 6.3163K_d\dot{\theta} + (6.3163K_p - 0.1225)\theta = 6.3163K_p\theta_{des} \quad (4.197)$$

b) For the closed-loop to be stable, the coefficients of the ODE must be  $> 0$

$$\begin{aligned} 6.3163K_d &> 0 \\ 6.3163K_p - 0.1225 &> 0 \end{aligned} \quad (4.198)$$

$$\begin{aligned} K_d &> 0 \\ K_p &> 0.0194 \end{aligned} \quad (4.199)$$

and using the second order form

$$\lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2 = 0 \quad (4.200)$$

one can calculate

$$\omega_n = \sqrt{6.3163K_p - 0.1225} \quad (4.201)$$

and

$$\zeta = \frac{6.3163K_d}{2\sqrt{6.3163K_p - 0.1225}} \quad (4.202)$$

For 2(a)  $t_s < 2$  seconds, one requires

$$\frac{3}{\zeta\omega_n} < 2 \quad (4.203)$$

$$3 < 2\zeta\omega_n \quad (4.204)$$

$$3 < 6.3163K_d \quad (4.205)$$

or

$$K_d > 0.475 \quad (4.206)$$

from which one can tentatively choose  $K_d = 0.8$ .

For 2(c),  $|e_{ss}| < 0.005$  rad for a step command of 0.1 rad, one requires

$$\frac{0.1225}{6.3163K_p - 0.1225}\theta_{des} < 0.005 \quad (4.207)$$

$$\frac{0.1225}{6.3163K_p - 0.1225}(0.1) < 0.005 \quad (4.208)$$

$$0.1225 < 0.05(6.3163K_p - 0.1225) \quad (4.209)$$

$$0.1225 < 0.05(6.3163K_p - 0.1225) \quad (4.210)$$

$$\frac{0.1225 + 0.05(0.1225)}{0.05(6.3163)} < K_p \quad (4.211)$$

or

$$K_p > 0.407 \quad (4.212)$$

from which one can tentatively choose  $\underline{K_p} = 1.5$ .

For 2(b)  $M_p < 5\%$ , one requires

$$e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}} < 0.05 \quad (4.213)$$

or

$$\zeta > 0.7 \quad (4.214)$$

For the tentatively chosen control gains

$$\zeta = 0.83 \quad (4.215)$$

Thus, these gains satisfy the three requirements.

# Chapter 5

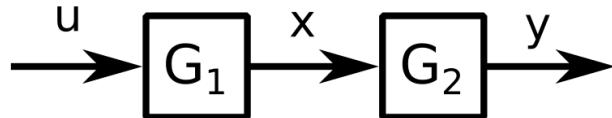
## SISO LTI Control System Robustness

### 5.1 SISO LTI Control System Analysis

As flight vehicles inherently have nonlinear dynamics and typically encounter significant disturbances in their flight conditions, this chapter of the textbook will discuss aspects of **system robustness** for SISO LTI control systems which are necessary to consider when designing SISO feedback control systems for flight vehicles. A later part of this textbook will extend this type of robustness analysis to MIMO LTI systems. However, before continuing, one must understand some of the block diagram algebra for computing the transfer functions of interconnected systems.

#### Interconnected Systems

For building an overall system transfer function, it is useful to consider the input and output to various interconnected systems represented by transfer functions, in particular the cascade/serial, parallel, and feedback interconnections. A **cascade interconnection**, also known as a **serial interconnection** can be represented as



and in the Laplace domain

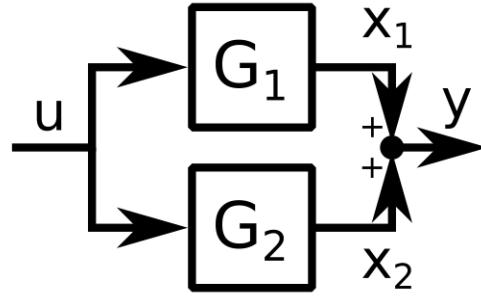
$$y(s) = G_2(s)x(s) \quad (5.1)$$

$$X(s) = G_1(s)u(s) \quad (5.2)$$

or for an overall system transfer function

$$y(s) = G(s)u(s) = [G_2(s)G_1(s)]u(s) \quad (5.3)$$

A **parallel interconnection** can be represented as



and in the Laplace domain

$$x_1(s) = G_1(s)u(s) \quad (5.4)$$

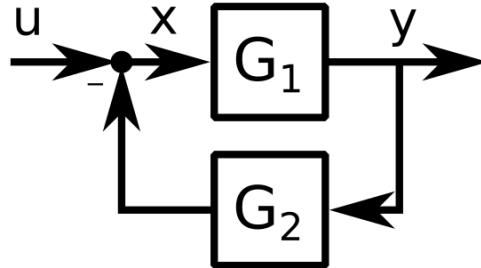
$$x_2(s) = G_2(s)u(s) \quad (5.5)$$

$$y(s) = x_1(s) + x_2(s) \quad (5.6)$$

or for an overall system transfer function

$$y(s) = G(s)u(s) = [G_1(s) + G_2(s)]u(s) \quad (5.7)$$

A **feedback interconnection** can be represented as



and in the Laplace domain

$$y(s) = G_1(s)x(s) \quad (5.8)$$

$$X(s) = u(s) - G_2(s)y(s) \quad (5.9)$$

$$y(s) = G_1(s)(u(s) - G_2(s)y(s)) \quad (5.10)$$

$$(1 + G_1(s)G_2(s))y(s) = G_1(s)u(s) \quad (5.11)$$

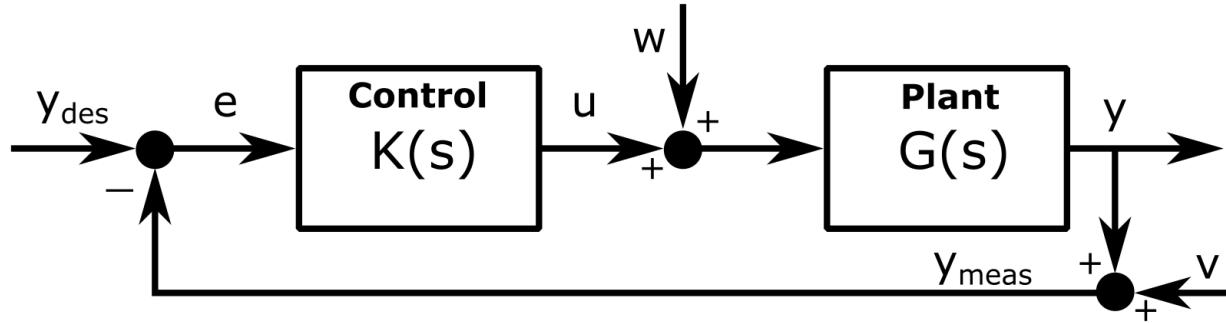
or for an overall system transfer function

$$y(s) = G(s)u(s) = \left[ \frac{G_1(s)}{1 + G_1(s)G_2(s)} \right] u(s) \quad (5.12)$$

Lastly, it is important to note that when forming transfer functions for interconnected systems, it is important that to take note of any **pole-zero cancellations**, i.e. the factors of the numerators of the transfer functions matching and thereby canceling with factors in the denominators, as this may result in an unstable feedback system due to the dynamics of interior signals of the system.

### General SISO Feedback Control System

When designing SISO feedback control systems, one usually considers the following general model



where  $G(s)$  is the dynamical system to be controlled, also known as the **plant** which derives from the original study of control of “power plants,”  $K(s)$  is the control system,  $y_c$  is the commanded output signal,  $e$  is the tracking error signal,  $u$  is the control input signal,  $w$  is a noise signal to the dynamic system input, e.g. disturbance,  $y$  is the system output signal,  $v$  is a noise signal on the dynamic system output, e.g. sensor noise,  $y_{meas}$  is the measured output signal. where the control design will typically be driven by direct requirements related to the different signals. The remainder of this course will consider a new design procedure for SISO feedback control systems to consider these signals and systems.

First, note that this feedback control system has 3 input signals (external/independent):  $y_c, w, v$ , and 4 output signals (internal/dependent):  $e, u, y, y_{meas}$ . Then, using the previously described rules for building transfer functions, each possible input-output pair has an associated transfer function for a total of twelve and can be represented as a matrix

$$\begin{bmatrix} y(s) \\ y_{meas}(s) \\ e(s) \\ u(s) \end{bmatrix} = \begin{bmatrix} \frac{GK}{1+GK} & \frac{G}{1+GK} & -\frac{GK}{1+GK} \\ \frac{GK}{1+GK} & \frac{G}{1+GK} & \frac{1}{1+GK} \\ \frac{K}{1+GK} & -\frac{GK}{1+GK} & -\frac{K}{1+GK} \\ \frac{1}{1+GK} & -\frac{G}{1+GK} & -\frac{1}{1+GK} \end{bmatrix} \begin{bmatrix} y_c(s) \\ w(s) \\ v(s) \end{bmatrix} \quad (5.13)$$

from which one can see there are four fundamental transfer functions (ignoring the sign)

$$\frac{1}{1+GK}, \frac{GK}{1+GK}, \frac{G}{1+GK}, \frac{K}{1+GK} \quad (5.14)$$

Thus, selecting  $K(s)$  to affect these four fundamental transfer functions is one method for control design. First, this section will develop new methods for analyzing signals and systems.

### General Feedback Control System Stability

When designing a control system, at a minimum, one requires that the feedback system is stable. In this model, this will only occur if and only if *all* the system transfer functions are stable. However, one can

simplify this criteria by considering the numerator/denominator polynomials of  $G(s)$  and  $K(s)$ , i.e.

$$G(s) = \frac{n_G(s)}{d_G(s)} \quad (5.15)$$

$$K(s) = \frac{n_K(s)}{d_K(s)} \quad (5.16)$$

Then, the four fundamental transfer functions can be written

$$\begin{aligned} \frac{1}{1+GK} &= \frac{1}{1 + \frac{n_G n_K}{d_G d_K}} = \frac{d_G d_K}{d_G d_K + n_G n_K} \\ \frac{GK}{1+GK} &= \frac{\frac{n_G n_K}{d_G d_K}}{1 + \frac{n_G n_K}{d_G d_K}} = \frac{n_G n_K}{d_G d_K + n_G n_K} \\ \frac{G}{1+GK} &= \frac{\frac{n_G}{d_G}}{1 + \frac{n_G n_K}{d_G d_K}} = \frac{n_G d_K}{d_G d_K + n_G n_K} \\ \frac{K}{1+GK} &= \frac{\frac{n_K}{d_K}}{1 + \frac{n_G n_K}{d_G d_K}} = \frac{d_G n_K}{d_G d_K + n_G n_K} \end{aligned} \quad (5.17)$$

where notably  $d_G d_K + n_G n_K$  is the denominator for each one. Thus, the **SISO feedback control system characteristic equation** is

$$d_G(s)d_K(s) + n_G(s)n_K(s) = 0 \quad (5.18)$$

and if the roots/poles of this equation are in the left half of the complex plane, the feedback system is stable. This is true even in the case of pole-zero cancellation which may occur between  $G(s)$  and  $K(s)$ . As a proof, let  $p_0$  be a pole such that a pole/zero cancellation occurs between  $G(s)$  and  $K(s)$ , i.e.

$$d_G(p_0) = n_K(p_0) = 0 \quad (5.19)$$

then recalling the characteristic equation

$$d_G(s)d_K(s) + n_G(s)n_K(s) = 0 \quad (5.20)$$

and substituting  $s = p_0$

$$d_G(p_0)d_K(p_0) + n_G(p_0)n_K(p_0) = 0 \quad (5.21)$$

one has

$$(0)d_K(p_0) + n_G(p_0)(0) = 0 \quad (5.22)$$

which equates to

$$0 = 0 \quad (5.23)$$

In addition, notice that  $1 + GK$  appears in the denominator of each fundamental transfer function which can be rewritten as

$$1 + G(s)K(s) = 1 + \frac{n_G n_K}{d_G d_K} = \frac{d_G(s)d_K(s) + n_G(s)n_K(s)}{d_G(s)d_K(s)} \quad (5.24)$$

Thus, one can see that the numerator of  $1 + GK$  is the feedback control system characteristic equation. However, as  $1 + GK$  can still be affected by pole-zero cancellations, one can equivalently declare that one has a **stable SISO feedback control system** if and only if

1.  $1 + G(s)K(s)$  has no zeros in the RHP and
2. there are no RHP pole-zero cancellations when forming  $G(s)K(s)$

Later sections will look at how to design  $K(s)$  to achieve certain control system performance requirements in the frequency domain which is related to the  $s$ -domain as will be shown in the next lecture.

### Example Problem 1

Given: the two LTI systems

$$\dot{x} + 4x = 3u \quad (5.25)$$

$$3\dot{y} + 6y = 5x \quad (5.26)$$

Determine: the transfer function from  $u(s)$  to  $y(s)$

Solution:

From inspection, one can see that this is a cascade interconnection, thus using the transfer function definitions, note that

$$\dot{x} + 4x = 3u \rightarrow G_1(s) = \frac{3}{s+4} \quad (5.27)$$

$$3\dot{y} + 6y = 5x \rightarrow G_2(s) = \frac{5}{3s+6} \quad (5.28)$$

then, the transfer function from  $u(s)$  to  $y(s)$  is

$$G(s) = G_2(s)G_1(s) = \left(\frac{5}{3s+6}\right)\left(\frac{3}{s+4}\right) \quad (5.29)$$

$$G(s) = \frac{15}{3s^2 + 18s + 24} \quad (5.30)$$

and transforming back to the time domain using  $G(s) = \frac{y(s)}{u(s)}$ , one has

$$3\ddot{y} + 18\dot{y} + 24y = 15u \quad (5.31)$$

Note that one could also compute the model directly from the ODEs. Solving for  $x$  and  $\dot{x}$  in terms of  $y$ , one has

$$3\dot{y} + 6y = 5x \quad (5.32)$$

provides

$$x = \frac{3}{5}\dot{y} + \frac{6}{5}y \quad (5.33)$$

$$\dot{x} = \frac{3}{5}\ddot{y} + \frac{6}{5}\dot{y} \quad (5.34)$$

Then, substituting for  $x$  and  $\dot{x}$ , one has

$$\left(\frac{3}{5}\ddot{y} + \frac{6}{5}\dot{y}\right) + 4\left(\frac{3}{5}\dot{y} + \frac{6}{5}y\right) = 3u \quad (5.35)$$

or

$$3\ddot{y} + 18\dot{y} + 24y = 15u \quad (5.36)$$

**Example Problem 2**Given: the following system

$$\dot{y} + 2y = 3u \quad (5.37)$$

and a PI feedback controller

$$u = 4e + 5 \int_0^t e(\tau) d\tau \quad (5.38)$$

where

$$e = y_c - y \quad (5.39)$$

Determine:  $G_{y_c \rightarrow y}(s)$  for the closed-loop model using the interconnected system rules.Solution:

Let

$$G(s) = \frac{y(s)}{u(s)} = \frac{3}{s+2} \quad (5.40)$$

and

$$K(s) = \frac{u(s)}{E(s)} = 4 + \frac{5}{s} = \frac{4s+5}{s} \quad (5.41)$$

The transfer function for the forward path is

$$G_{e \rightarrow y} = \frac{y(s)}{E(s)} = G(s)K(s) = \frac{3}{s+2} \left( \frac{4s+5}{s} \right) = \frac{12s+15}{s^2+2s} \quad (5.42)$$

The transfer function in the feedback path is simply 1, i.e. unity feedback, thus

$$G_{y_c \rightarrow y}(s) = \frac{G_{e \rightarrow y}}{1 + G_{e \rightarrow y}(1)} \quad (5.43)$$

$$G_{y_c \rightarrow y}(s) = \frac{\frac{12s+15}{s^2+2s}}{1 + \frac{12s+15}{s^2+2s}} \quad (5.44)$$

$$G_{y_c \rightarrow y}(s) = \frac{12s+15}{s^2+14s+15} \quad (5.45)$$

**Example Problem 3**Given:

$$G_1 = \frac{X(s)}{u(s)} = \frac{1}{s-1} \quad (5.46)$$

$$G_2 = \frac{y(s)}{X(s)} = \frac{s-1}{s+2} \quad (5.47)$$

Determine: the stability of the cascaded systemSolution:

By definition,

$$y(s) = \left( \frac{1}{s-1} \right) \left( \frac{s-1}{s+2} \right) u(s) \quad (5.48)$$

$$y(s) = \frac{1}{s+2}u(s) \quad (5.49)$$

which is a stable system, i.e. pole at  $s = -2$  (LHP). However, the internal signal,  $X(s)$ , is still

$$X(s) = \frac{1}{s-1}u(s) \quad (5.50)$$

which is unstable, i.e. pole at  $s = 1$  (RHP). Thus, this system is unstable due to the requirement that all transfer functions be stable.

### Example Problem 4

Given: a feedback control system with

$$G(s) = \frac{1}{s-10} \quad (5.51)$$

and

$$K(s) = \frac{s-10}{s+5} \quad (5.52)$$

Determine: the stability of the feedback control system

Solution:

The feedback control system characteristic equation is

$$d_G(s)d_K(s) + n_G(s)n_K(s) = 0 \quad (5.53)$$

$$(s-10)(s+5) + (1)(s-10) = 0 \quad (5.54)$$

$$(s-10)(s+6) = 0 \quad (5.55)$$

The roots are at  $s = 1$  and  $s = -3$ . Thus, the feedback control system is unstable.

Alternatively, compute the product

$$G(s)K(s) = \left(\frac{1}{s-10}\right)\left(\frac{s-10}{s+5}\right) = \frac{1}{s+2} \quad (5.56)$$

which notably involves a RHP pole/zero cancellation. Then, looking at the four fundamental transfer functions as an alternative provides

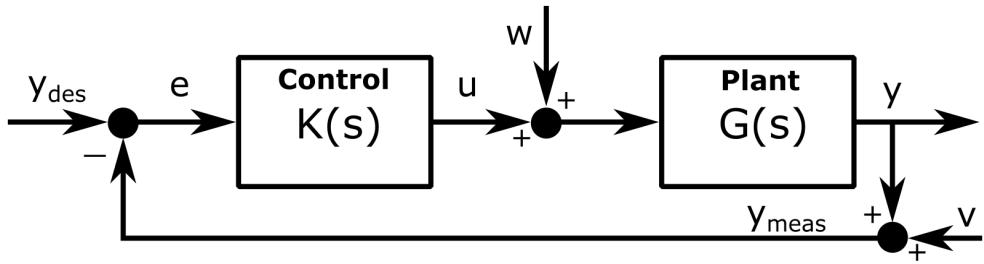
$$\begin{aligned} \frac{1}{1+GK} &= \frac{1}{1+\frac{1}{s+5}} = \frac{1}{s+6} \\ \frac{GK}{1+GK} &= \frac{\frac{1}{s+5}}{1+\frac{1}{s+5}} = \frac{s+5}{s+6} \\ \frac{G}{1+GK} &= \frac{\frac{1}{s-10}}{1+\frac{1}{s+5}} = \frac{s+5}{(s+6)(s-10)} \\ \frac{K}{1+GK} &= \frac{\frac{s-10}{s+5}}{1+\frac{1}{s+5}} = \frac{s-10}{s+6} \end{aligned} \quad (5.57)$$

one can see that  $\frac{G}{1+GK}$  is unstable (pole at  $s = 10$  in RHP), thus the feedback control system is unstable through this analysis.

As an aside, note that  $\frac{G}{1+GK}$  was the transfer function from  $w \rightarrow y$ , thus a disturbance on the control input will cause  $y$  to go unstable.

## 5.2 Open-Loop Transfer Function

Recall the general feedback control system



with the objective is to design a controller  $K(s)$  such that:

1. the feedback control system is stable;
2.  $e$  is small (i.e. good tracking);
3.  $w$  has small effect (i.e. disturbance rejection);
4.  $v$  has a small effect (i.e. sensor noise filtered out);
5.  $|u|$  is not too large; and
6. robustness to uncertainty in plant  $G$ .

Satisfying objective 1 was considered in a previous lecture using either the four fundamental transfer functions, the numerator of  $1 + GK$ , or the roots of the characteristic equation. For control design objectives 2-5, this part of the textbook will use **loop-shaping** control design for SISO systems which shapes the SISO **open-loop transfer function**, i.e.

$$L(s) = G(s)K(s) \quad (5.58)$$

to have certain characteristics by designing  $K(s)$ . This section will establish the connections between these objectives and the open-loop transfer function  $L(s)$ . As before with PID tuning, design iteration is typically required to achieve all objectives using the Bode plot as the primary analysis plot. This method can very useful for higher order systems as the Bode plot allows one to visualize the individual poles and zeros, i.e. modes, of the SISO system. Furthermore, through loop-shaping design stages, one can shape individual sections of  $L(s)$  such that specific requirements are met by specific stages. Lastly, for the robustness of requirement 6, subsequent sections will look at using the Bode and Nyquist plots for developing a notion of system robustness and subsequent stability margins.

### Loop Transfer Function Requirements

Objective 2 requires the feedback control system to have good tracking, i.e. keep  $e$  small. Recalling the previously derived transfer function for  $y_c \rightarrow e$ , i.e.

$$\frac{e(s)}{y_c(s)} = \frac{1}{1 + GK} = \frac{1}{1 + L} \quad (5.59)$$

one requires that  $|\frac{1}{1+L}| \ll 1$  which occurs if  $|L| \gg 1$ .

Objective 3 requires the feedback control system to reject disturbances, i.e.  $w$  should have little effect on  $y_{meas}$ . Recalling the previously derived transfer function for  $w \rightarrow y_{meas}$ , i.e.

$$\frac{y_{meas}(s)}{w(s)} = \frac{G}{1+GK} = \frac{G}{1+L} \quad (5.60)$$

one requires that  $|\frac{G}{1+L}| \ll 1$  which occurs if  $|L| \gg 1$ . Note that the effect of the disturbance with feedback control should be smaller than with no control, i.e.  $|\frac{G}{1+L}| \ll |G|$  or  $|\frac{1}{1+L}| \ll 1$ . This occurs if  $|L| \gg 1$ .

Objective 4 requires the feedback control system to filter out the sensor noise, i.e.  $v$  should have little effect on  $e$ . Recalling the previously derived transfer function for  $v \rightarrow e$ , i.e.

$$\frac{e(s)}{v(s)} = \frac{-GK}{1+GK} = \frac{-L}{1+L} \quad (5.61)$$

one requires that  $|\frac{L}{1+L}| \ll 1$  which occurs if  $|L| \ll 1$ .

Objective 5 requires the feedback control system to keep  $u$  small enough. Recalling the previously derived transfer function for  $y_c \rightarrow u$ , i.e.

$$\frac{u(s)}{y_c(s)} = \frac{K}{1+GK} \quad (5.62)$$

one requires that  $|\frac{K}{1+L}| \ll 1$ . Because  $G$  is fixed, one has 2 cases:

1. If  $|G| \gg 1$ , then all  $|K|$  will provide  $|\frac{K}{1+GK}| \ll 1$
2. If  $|G| \ll 1$ , then  $|K| \ll 1$  will provide  $|\frac{K}{1+GK}| \ll 1$

Summarizing these rough guidelines results in the following table.

General Requirement	Closed-Loop TF Requirement	Loop TF Requirement
1. Good Tracking	$ \frac{1}{1+GK}  \ll 1$	$ L  \gg 1$
2. Disturbance Rejection	$ \frac{G}{1+GK}  \ll 1$	$ L  \gg 1$
3. Noise Filtering	$ \frac{GK}{1+GK}  \ll 1$	$ L  \ll 1$
4. Small control effort	$ \frac{K}{1+GK}  \ll 1$	If $ G  \ll 1$ , then $ K  \ll 1$

Note that one cannot simultaneously satisfy objectives 1 and 2 with 3. However, the *key idea* in frequency domain control design is that desired output commands are designed to be low frequency signals relative to any high frequency noise signals on the actual output.

## Feedback Control System Sensitivity

Two of the four fundamental transfer functions have particular names. The first is the **error transfer function** defined as

$$S(s) = \frac{1}{1 + G(s)K(s)} \quad (5.63)$$

and is the transfer function from the closed-loop commanded output,  $y_c(s)$ , to the tracking error  $e(s)$ . The second is the **closed-loop transfer function** defined as

$$T(s) = \frac{G(s)K(s)}{1 + G(s)K(s)} \quad (5.64)$$

and is the transfer function from the closed-loop commanded output,  $y_c(s)$ , to the actual output  $y(s)$ .

$S(s)$  is also known as the **sensitivity transfer function** as it characterizes how sensitive the system is to small changes in  $G(s)$ , an important consideration in real systems due to neglected dynamics, e.g. nonlinearities, or changing characteristics, e.g. component aging. To see this, consider the transfer function from  $y_c \rightarrow y$ , i.e.

$$\frac{Y(s)}{Y_c(s)} = \frac{G(s)K(s)}{1 + G(s)K(s)} = T(s) \quad (5.65)$$

Now consider the ratio of a small change in  $T$  to a small change in  $G$ , i.e.

$$\frac{\frac{\Delta T}{T}}{\frac{\Delta G}{G}} = \left[ \frac{\Delta T}{\Delta G} \right] \left[ \frac{G}{T} \right] \quad (5.66)$$

using the derivative for small  $\Delta$ 's

$$\frac{\frac{\Delta T}{T}}{\frac{\Delta G}{G}} = \left[ \frac{dT}{dG} \right] \left[ \frac{G}{T} \right] \quad (5.67)$$

$$\frac{\frac{\Delta T}{T}}{\frac{\Delta G}{G}} = \left[ \frac{d}{dG} \left( \frac{GK}{1 + GK} \right) \right] \left[ \frac{G}{\frac{GK}{1+GK}} \right] \quad (5.68)$$

$$\frac{\frac{\Delta T}{T}}{\frac{\Delta G}{G}} = \left[ \left( \frac{(1 + GK)K - GK^2}{(1 + GK)^2} \right) \right] \left[ \frac{1 + GK}{K} \right] \quad (5.69)$$

$$\frac{\frac{\Delta T}{T}}{\frac{\Delta G}{G}} = \left[ \left( \frac{K}{(1 + GK)^2} \right) \right] \left[ \frac{1 + GK}{K} \right] \quad (5.70)$$

$$\frac{\frac{\Delta T}{T}}{\frac{\Delta G}{G}} = \frac{1}{(1 + GK)} \quad (5.71)$$

$$\frac{\frac{\Delta T}{T}}{\frac{\Delta G}{G}} = S \quad (5.72)$$

$T(s)$  is also known as the **complementary sensitivity transfer function** as one can write that

$$S(s) + T(s) = \frac{1}{1 + G(s)K(s)} + \frac{G(s)K(s)}{1 + G(s)K(s)} \quad (5.73)$$

or

$$S(s) + T(s) = 1 \quad \forall s \in \mathbb{C} \quad (5.74)$$

Furthermore, using these definitions, one can also write out the feedback control system objectives as

1.  $|S| \ll 1$  provides good tracking and good disturbance rejection
2.  $|T| \ll 1$  provides good noise filtering
3.  $|KS| \ll 1$  provides small control effort

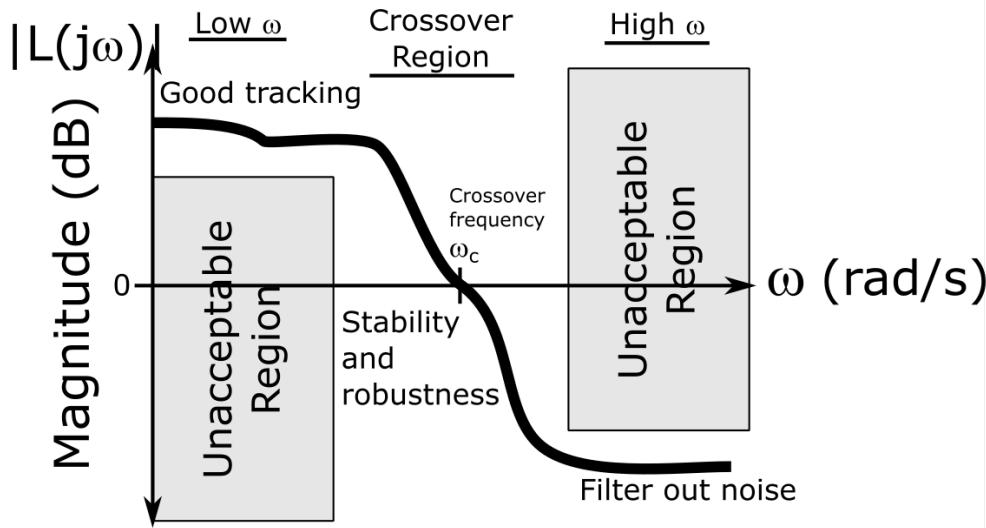
Thus, the requirement that  $S + T = 1$  demonstrates more rigorously that one cannot satisfy all control objectives at all frequencies. Thus, one requires the feedback control system satisfies

1.  $|S(j\omega)| \ll 1$  at low  $\omega$
2.  $|T(j\omega)| \ll 1$  at high  $\omega$
3.  $|K(j\omega)S(j\omega)| \ll 1 \quad \forall \omega$

Then, translating 1 and 2 to the open-loop transfer function,  $L(j\omega)$ , provides the loop-shaping design criteria

1.  $|L(j\omega)| \gg 1$  at low  $\omega$
2.  $|L(j\omega)| \ll 1$  at high  $\omega$

which can be analyzed visually using the magnitude/gain subplot of the Bode plot of  $L(j\omega)$ , i.e.



where connections to the phase plot will be made later primarily for stability and robustness design alongside the crossover region loop-shaping, essentially this crossover cannot be too steep. Lastly, for the control effort requirement, one must analyze the Bode plot of  $K(j\omega)S(j\omega)$  in parallel with the loop-shaping of  $L(s)$ . As a general rule, one typically should design  $K(j\omega)$  to not be too large where  $G(j\omega)$  is small.

### Final Value Theorem (FVT)

An important result that one can use with the transfer function model for LTI systems is the **Final Value Theorem (FVT)** which states if every pole of a transfer function  $F(s)$  is either in the open left half plane (i.e. cannot be purely imaginary), or at the origin, *and* that  $F(s)$  has, at most, a single pole at the origin. Then,

$$\lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} sF(s) = L \quad (5.75)$$

where  $s \rightarrow 0$  denotes  $s$  approaching through the positive numbers.

For feedback control systems, it is often useful to check the steady-state error for the system by the following

$$e_{ss} = \lim_{s \rightarrow 0} sE(s) \quad (5.76)$$

or using the definition of the sensitivity transfer function,  $S(s)$ , one has

$$e_{ss} = \lim_{s \rightarrow 0} sS(s)Y_c(s) \quad (5.77)$$

### First Order System with P Control

Consider a first order system

$$\dot{y} + y = u \quad (5.78)$$

with proportional control law

$$u = K_p(y_c - y) \quad (5.79)$$

The closed-loop ODE is

$$\dot{y} + (K_p + 1)y = K_p y_c \quad (5.80)$$

which has been analyzed in this course with PID tuning in the time domain. In the frequency domain, the system transfer function is

$$G(s) = \frac{1}{s + 1} \quad (5.81)$$

The controller transfer function is

$$K(s) = K_p \quad (5.82)$$

The sensitivity transfer function is

$$S(s) = \frac{1}{1 + G(s)K(s)} = \frac{s + 1}{s + K_p + 1} \quad (5.83)$$

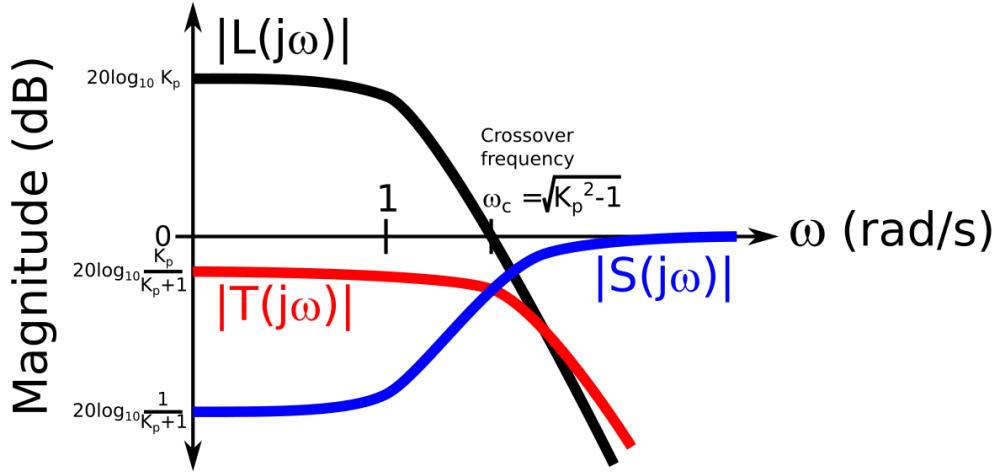
The complementary sensitivity transfer function is

$$T(s) = \frac{G(s)K(s)}{1 + G(s)K(s)} = \frac{K_p}{s + K_p + 1} \quad (5.84)$$

The open-loop transfer function is

$$L(s) = G(s)K(s) = \frac{K_p}{s + 1} \quad (5.85)$$

Plotting  $L(s)$ ,  $S(s)$ , and  $T(s)$  on the same Bode magnitude plot.



At low  $\omega$ :  $|S| \ll 1$ ,  $|T| \approx 1$ ,  $|L| \gg 1$ , thus one has good tracking and disturbance rejection yet poor noise filtering.

The crossover frequency is  $\omega_c = \sqrt{K_p^2 - 1}$ , thus increasing  $K_p$  will increase  $\omega_c$ , i.e. increase the speed of response and reduce the settling time,  $t_s$ .

At high  $\omega$ :  $|S| \approx 1$ ,  $|T| \ll 1$ ,  $|L| \ll 1$ , thus one has poor tracking and disturbance rejection yet good noise filtering.

Recalling that  $S(s)$  is transfer function from the desired output  $Y_c(s)$  to error  $E(s)$  and if  $Y_c(s)$  is the unit step, i.e.  $\frac{1}{s}$  in the Laplace domain, then the final value theorem states

$$e_{ss} = \lim_{s \rightarrow 0} sE(s) \quad (5.86)$$

$$e_{ss} = \lim_{s \rightarrow 0} sS(s)Y_c(s) \quad (5.87)$$

$$e_{ss} = \lim_{s \rightarrow 0} sS(s) \frac{1}{s} \quad (5.88)$$

$$e_{ss} = \lim_{s \rightarrow 0} S(s) \quad (5.89)$$

$$e_{ss} = S(0) \quad (5.90)$$

Thus,  $e_{ss} = S(0) = \frac{1}{K_p+1}$ .

### 5.3 Nyquist Plots and Stability

The **Nyquist plot** is a common tool used to understand the stability and robustness of a feedback control system. Similar to the Bode plot, the Nyquist plot can be used to analyze the frequency response of a transfer function, i.e.  $G(s)$  with  $s = j\omega$ . However, opposed to the Bode plot, the Nyquist plot visualizes the real and imaginary parts of  $G(j\omega)$  as a single curve with the real part on the horizontal axis and the imaginary on

the vertical axis. It should also be noted the convention for the Nyquist plot is to plot over  $-\infty < \omega < \infty$  as opposed to  $\omega \geq 0$  for the Bode plot. This convention results in a reflected curve about the real axis for  $\omega < 0$  with respect to  $\omega > 0$  as a transfer function value  $G(j\omega) = \alpha + j\beta$  simply provides the complex conjugate for  $\omega < 0$  with respect to  $\omega > 0$ .

As an example of a Nyquist plot, consider the standard form of a first order LTI system, i.e.

$$\dot{y} + a_0 y = b_0 u \quad (5.91)$$

and assume that  $a_0 \neq 0$ . The transfer function for this system is  $G(s)$

$$G(s) = \frac{b_0}{s + a_0} \quad (5.92)$$

and has a pole at  $s = -a_0$ .

The Nyquist plot of the frequency response requires computing  $G(j\omega)$ , i.e.

$$G(j\omega) = \frac{b_0}{j\omega + a_0} \quad (5.93)$$

$$G(j\omega) = \frac{b_0}{j\omega + a_0} \frac{a_0 - j\omega}{a_0 - j\omega} \quad (5.94)$$

$$G(j\omega) = \frac{a_0 b_0}{a_0^2 + \omega^2} - j \frac{\omega b_0}{a_0^2 + \omega^2} \quad (5.95)$$

where

$$\operatorname{Re}\{G(j\omega)\} = \frac{a_0 b_0}{a_0^2 + \omega^2} \quad (5.96)$$

and

$$\operatorname{Im}\{G(j\omega)\} = \frac{-\omega b_0}{a_0^2 + \omega^2} \quad (5.97)$$

Next, noting that

$$\operatorname{Re}\{G(j\omega)\}^2 = \frac{a_0^2 b_0^2}{(a_0^2 + \omega^2)^2} \quad (5.98)$$

$$\operatorname{Im}\{G(j\omega)\}^2 = \frac{\omega^2 b_0^2}{(a_0^2 + \omega^2)^2} \quad (5.99)$$

and adding, one has

$$\operatorname{Re}\{G(j\omega)\}^2 + \operatorname{Im}\{G(j\omega)\}^2 = \frac{a_0^2 b_0^2}{(a_0^2 + \omega^2)^2} + \frac{\omega^2 b_0^2}{(a_0^2 + \omega^2)^2} \quad (5.100)$$

or

$$\operatorname{Re}\{G(j\omega)\}^2 + \operatorname{Im}\{G(j\omega)\}^2 = \frac{b_0^2}{(a_0^2 + \omega^2)} \quad (5.101)$$

Then rearranging, one has

$$\operatorname{Re}\{G(j\omega)\}^2 - \frac{b_0^2}{(a_0^2 + \omega^2)} + \operatorname{Im}\{G(j\omega)\}^2 = 0 \quad (5.102)$$

Then, adding  $\left(\frac{b_0}{4a_0}\right)^2$  to both sides, one has

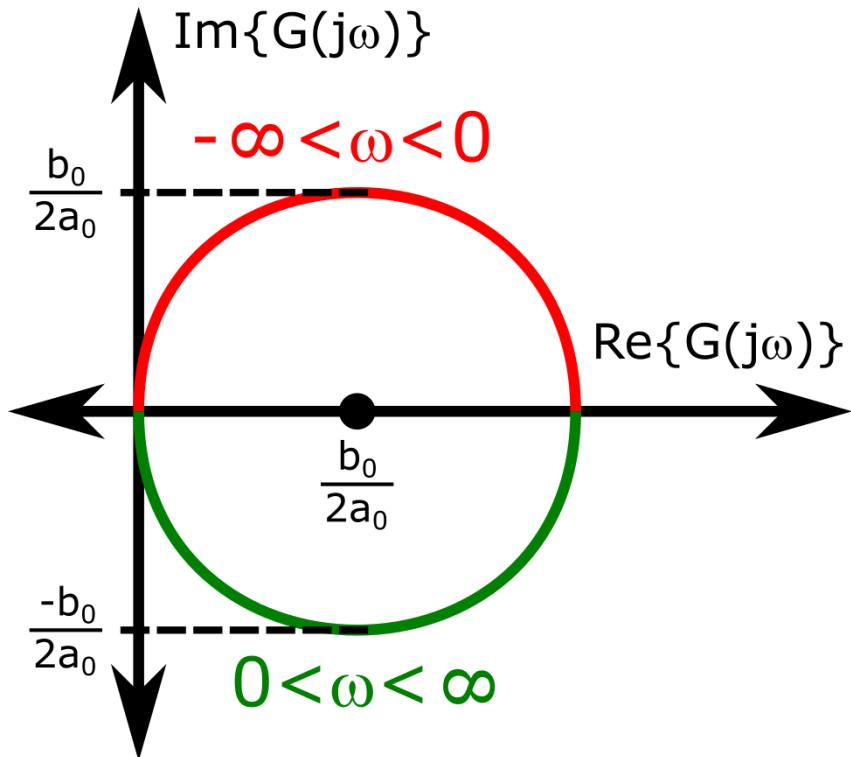
$$\operatorname{Re}\{G(j\omega)\}^2 - \frac{b_0^2}{(a_0^2 + \omega^2)} + \left(\frac{b_0}{4a_0}\right)^2 + \operatorname{Im}\{G(j\omega)\}^2 = \left(\frac{b_0}{4a_0}\right)^2 \quad (5.103)$$

and recalling Equation 5.98, one has

$$\left(\operatorname{Re}\{G(j\omega)\} - \frac{b_0}{4a_0}\right)^2 + \operatorname{Im}\{G(j\omega)\}^2 = \left(\frac{b_0}{4a_0}\right)^2 \quad (5.104)$$

which is the equation of a circle of radius  $\frac{b_0}{4a_0}$  centered at  $\left(\frac{b_0}{4a_0}, 0\right)$ .

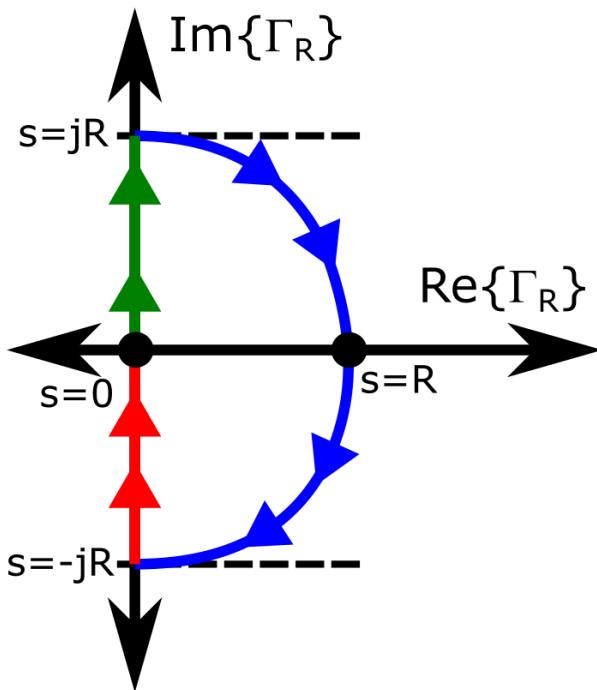
For plotting the variation with  $\omega$ , consider three particular points. At  $\omega = 0$ ,  $G(0) = \frac{b_0}{a_0}$ . As  $\omega \rightarrow \infty$ ,  $G(j\omega) \rightarrow -j\frac{b_0}{\omega}$ . At  $\omega = a_0$ :  $G(ja_0) = (1 - j)\frac{b_0}{2a_0}$ . Using these results, one can draw the Nyquist plot as



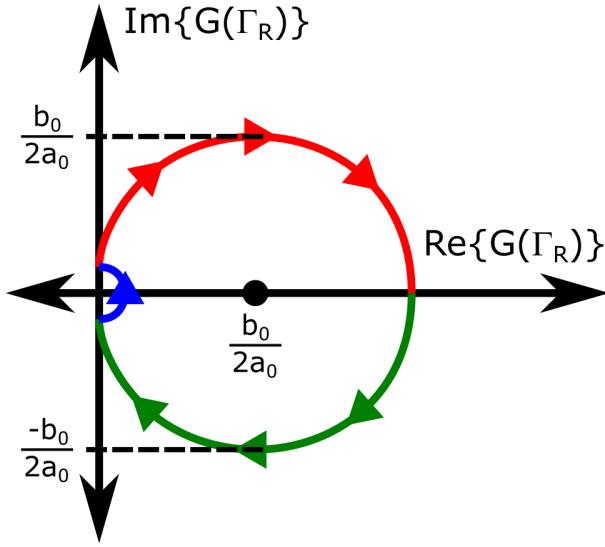
### Cauchy's Argument Principle

The Nyquist plot of the open-loop transfer function  $L(s)$  can be used to state a theorem concerning the stability of a feedback control system. However, to derive this analysis tool, one requires a result from complex analysis, Cauchy's argument principle, which will be summarized as follows. To begin, let  $G(s)$  be a transfer function of a system and let  $\Gamma$  be a simple, closed curve in the complex plane where a “simple” curve is one that does not intersect itself and a closed curve is completely connected. Here, the notation  $G(\Gamma)$  is the curve obtained by mapping each complex number  $s \in \Gamma$  to another complex number  $G(s)$ . In general,  $G(\Gamma)$  will be closed but need not be simple.

For example, consider the curve  $\Gamma_R$  shown as



which results in the mapping to  $G(\Gamma_R)$  for a first order system as



and it can be shown that  $G(\Gamma_R)$  converges to the Nyquist plot of  $G(s)$  as  $R \rightarrow \infty$ .

Next, define  $N_p$  and  $N_z$  as the number of poles and zeros of  $G(s)$  inside a general curve  $\Gamma$ , respectively. Then, **Cauchy's argument principle** states that if  $\Gamma$  does not pass through any poles or zeros of  $G(s)$ , then the closed curve  $G(\Gamma)$  encircles the origin  $N_z - N_p$  times. Furthermore, if  $N_z - N_p > 0$ , the  $G(\Gamma)$  encircles the origin clockwise and if  $N_z - N_p < 0$ , then  $G(\Gamma)$  encircles the origin counter-clockwise.

Finally, it should also be noted that if  $\Gamma$  passes through a pole of  $G(s)$ , then  $G(\Gamma)$  diverges to  $\infty$  and if  $\Gamma$  passes through a zero of  $G(s)$ , then  $G(\Gamma)$  intersects the origin. In either case, Cauchy's argument principle would not apply. The proof of this principle can be found elsewhere, e.g. in textbooks on complex analysis, and is left for the reader.

### Nyquist Stability Criterion

To develop the stability criteria based on the Nyquist plot of the open-loop transfer function  $L(j\omega)$ , first note that  $s = -1$  is a critical point for the stability of the open-loop transfer function  $L(s)$ . Recall from previous analysis that the stability of the feedback control system could be assessed through the zeros of  $1 + G(s)K(s)$ , i.e.  $1 + L(s)$  as this expression appeared in the denominator of all four fundamental transfer functions. Thus, if the frequency response of the open-loop transfer function, i.e.  $L(j\omega)$ , passes through  $j\omega = -1$  on the Nyquist plot for some  $\omega = \omega_0$ , then the feedback control system will be unstable, as any signal that contains frequency  $\omega_0$  would cause one or more of the fundamental transfer functions to have infinite gain, e.g. as  $S(s) = \frac{E(s)}{Y_c(s)} = \frac{1}{1+L(s)}$ , then  $|S(j\omega_0)| = \infty$ .

Furthermore, recall the stability conditions that if no pole-zero cancellations exist between  $G(s)$  and  $K(s)$ , then the feedback control system is stable if and only if  $1 + L(s)$  has no zeros in the RHP. Thus, one can infer that the critical  $s = -1$  point also plays a role in the Nyquist theorem on the feedback control system stability.

Next, define  $P_{c-l}(\epsilon)$  as the number of poles of the closed-loop feedback control system with real part greater than or equal to  $\epsilon$ ,  $P_{o-l}(\epsilon)$  as the number of poles of the open-loop transfer function  $L(s)$  with real

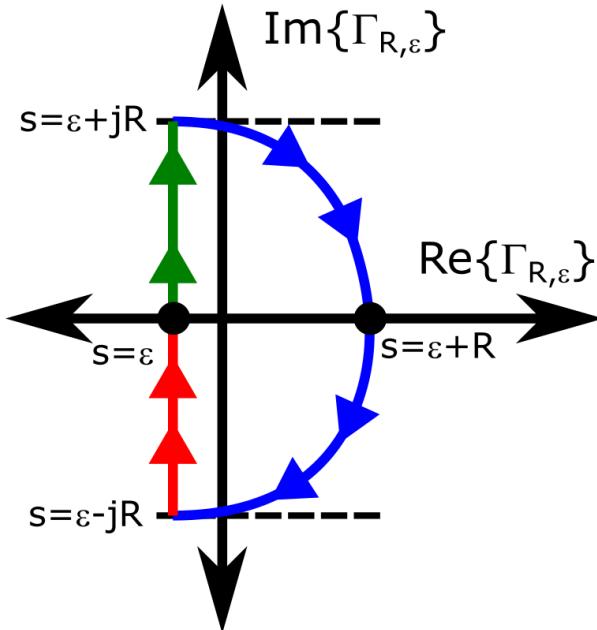
part greater than or equal to  $\epsilon$ , and  $N_{ccw}(\epsilon)$  as the number of times the Nyquist curve of  $L(\epsilon + j\omega)$  encircles the critical  $s = -1$  point in the counterclockwise direction.

Then, the **Nyquist stability criterion** states that if  $L(s)$  has no poles with real part equal to  $\epsilon$ , then  $P_{c-l}(\epsilon) = P_{o-l}(\epsilon) - N_{ccw}(\epsilon)$ . Thus,  $P_{c-l}(\epsilon) = 0$  if and only if  $N_{ccw}(\epsilon) = P_{o-l}(\epsilon)$ .

A notable special case of this theorem, e.g. **simplified Nyquist stability criterion**, states that if  $\epsilon = 0$ , i.e.  $L(s)$  has no poles on the imaginary axis, then  $P_{c-l}$  and  $P_{o-l}$  are the number of RHP poles for the closed- and open-loops, respectively (including the imaginary axis) and  $N_{ccw}$  is the number of times the Nyquist plot of  $L(j\omega)$ , i.e. the frequency response of  $L(s)$ , encircles  $s = -1$  in the counterclockwise direction. In this case, the Nyquist theorem states that  $P_{c-l} = P_{o-l} - N_{ccw}$  and the closed-loop is stable if and only if  $N_{ccw} = P_{o-l}$ . Note that this is the case in MATLAB for assessing the Nyquist plot of  $L(j\omega)$ . Thus, if  $L(j\omega)$  has a pole on the imaginary axis, e.g.  $K(s)$  has an integral component  $\frac{1}{s}$ , then care must be taken in the stability analysis.

However, in either case, the Nyquist plot of the open-loop transfer function  $L(j\omega)$  provides a visual method for determining the stability of the closed-loop feedback control system, as well as a measure of “how much” stability there is for the system, i.e. its robustness, which will be discussed in the subsequent section.

To prove the general Nyquist theorem, one requires defining a “perturbed” simple, closed curve  $\Gamma_{R,\epsilon}$  which is a shifted  $\Gamma_R$  curve by some amount  $\epsilon$  along the real axis as shown below for some  $\epsilon < 0$ .



Note that as  $\epsilon \rightarrow 0$ ,  $\Gamma_{R,\epsilon} \rightarrow \Gamma_R$ . Next, defining the transfer function  $H(s) = 1 + L(s)$  and  $N_p$  and  $N_z$  as the number of poles and zeros of  $H(s)$  inside the simple, closed curve  $\Gamma_{R,\epsilon}$ , then Cauchy’s argument principle states that  $H(\Gamma_{R,\epsilon})$  has  $N_p - N_z$  counterclockwise encirclements of the origin. Then, note the following three observations.

1. As  $L(s) = H(s) - 1$ ,  $L(\Gamma_{R,\epsilon})$  is the curve  $H(\Gamma_{R,\epsilon})$  shifted to the left by one unit. Thus,  $H(\Gamma_{R,\epsilon})$  encircling the origin is equivalent to  $L(\Gamma_{R,\epsilon})$  encircling  $s = -1$ . Furthermore, as  $R \rightarrow \infty$  and  $\epsilon \rightarrow 0$ ,  $L(\Gamma_{R,\epsilon})$  converges to the Nyquist plot of  $L(s)$ . Thus,  $N_{ccw} = N_p - N_z$ .

2.  $\Gamma_{R,\epsilon}$  contains the entire RHP as  $R \rightarrow \infty$  and  $\epsilon \rightarrow 0$ . Thus, if  $R$  is sufficiently large and  $\epsilon$  is sufficiently small, then the RHP zeros of  $H(s) = 1 + L(s)$  are precisely the closed-loop RHP poles, i.e.  $N_z = P_{c-l}$ .

3. The RHP poles of  $H(s)$  are precisely the RHP poles of the open-loop transfer function  $L(s)$ , i.e.  $N_p = P_{o-l}$ . This follows from the assumption that  $L(s)$  contains no pole-zero cancellations, and thus,  $|H(s_0)| = \infty$  for some  $s_0$  in the RHP if and only if  $|L(s_0)| = \infty$ .

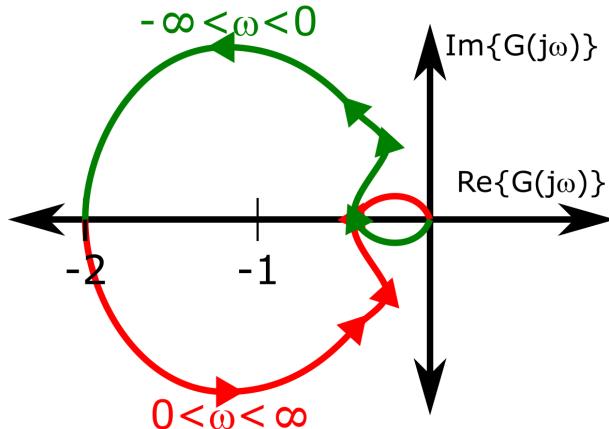
Combining these three observations, one has that  $N_{ccw} = P_{o-l} - P_{c-l}$  or rearranging  $P_{c-l} = P_{o-l} - N_{ccw}$  which is what was needed to be proved.

### Example Problem 1

Given: the open-loop transfer function

$$L(s) = \frac{200}{(s-1)(s^2+5s+100)} \quad (5.105)$$

and the Nyquist plot of  $L(j\omega)$  of an feedback control system.



Determine: the stability of the feedback control system.

Solution:

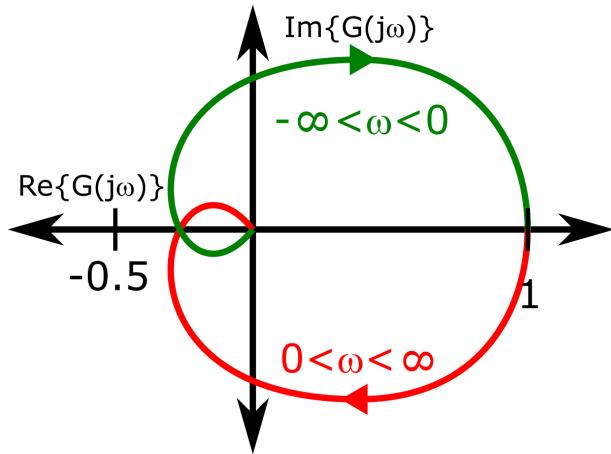
$L(s)$  has one pole in the RHP at  $s = -1$ , thus  $P_{o-l} = 1$ . The Nyquist curve for  $L(j\omega)$  encircles the critical  $s = -1$  point once in the counterclockwise direction, thus,  $N_{ccw} = 1$ . As  $L(s)$  has no poles on the imaginary axis, by the simplified Nyquist theorem,  $P_{c-l} = P_{o-l} - N_{ccw} = 0$  Thus, the feedback control system is stable.

### Example Problem 2

Given: the open-loop transfer function

$$L(s) = \frac{1}{(s+1)^5} \quad (5.106)$$

and the Nyquist plot of  $L(j\omega)$  of an feedback control system.



Determine: the stability of the feedback control system.

Solution:

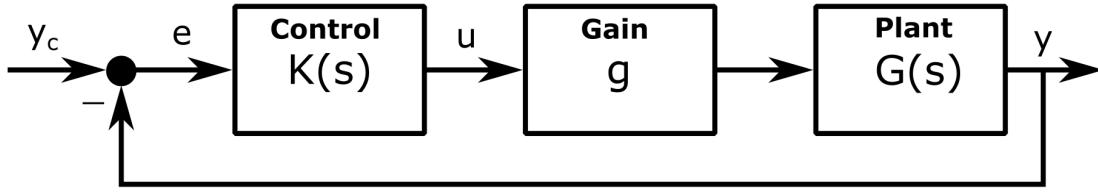
$L(s)$  has no poles in the RHP, thus  $P_{o-l} = 0$ . The Nyquist curve for  $L(j\omega)$  does not encircle the critical  $s = -1$  point, thus  $N_{ccw} = 0$ . As  $L(s)$  has no poles on the imaginary axis, by the simplified Nyquist theorem,  $P_{c-l} = P_{o-l} - N_{ccw} = 0$  Thus, the feedback control system is stable.

## 5.4 SISO LTI System Stability Margins and Robustness

As mentioned previously, the plant model in a feedback control system is typically only an approximation due to model uncertainty, e.g. linearization error, model parameter variability, simplified dynamics modeling, etc. This is especially true for FDC. To account for this, control engineers use safety factors when designing feedback control systems. These safety factors are known as **stability margins** and quantify the **feedback control system robustness**, i.e. a measure of the feedback control system will also perform adequately well under a different set of assumptions for the system model. This section will consider four primary SISO stability margins: gain margin, phase margin, delay margin, and disk margin.

### Gain Margin

The **gain margin** is the amount of gain on the nominal plant model that can be added before closed-loop instability. The gain margin can be derived by considering a scaling perturbation to the plant dynamics model,  $G(s)$ , which can be modeled in block diagram form as



where  $g > 0$  is some scalar constant. Then, defining the perturbed open-loop transfer function as  $gL(s)$ , the gain margin is determined by assessing for what values of  $g$  the feedback control system will go unstable. It should be noted that this analysis is similar to a root locus analysis.

Recalling that the feedback control system is stable if (1) there are no pole-zero cancellations and (2) the zeros of  $1 + L(s) = 0$  are only in the LHP. Then, a critical gains,  $g_0$ , occurs when the zeros of  $1 + g_0L(j\omega_0)$  are on the imaginary axis for some critical frequency  $\omega_0$ , i.e.

$$1 + g_0L(j\omega_0) = 0 \quad (5.107)$$

Rewriting this expression, one has

$$L(j\omega_0) = -\frac{1}{g_0} \quad (5.108)$$

or in polar form (i.e. gain and phase), one has

$$|L(j\omega_0)|e^{j\angle L(j\omega_0)} = -\frac{1}{g_0} \quad (5.109)$$

or the equivalent magnitude/gain and phase requirements become

$$g_0 = \frac{1}{|L(j\omega_0)|} \quad (5.110)$$

and

$$\angle L(j\omega_0) = \pm 180^\circ \quad (5.111)$$

Thus, one can identify the gain margins from the Bode plot by:

1. identifying all critical frequencies  $\omega_{0,i}$  where  $\angle L(j\omega_{0,i}) = \pm 180^\circ$ ;
2. calculating all associated critical gain  $g_{0,i}$  using the inverse of the gain at  $\omega_{0,i}$ , i.e.

$$g_{0,i} = \frac{1}{|L(j\omega_{0,i})|} \quad (5.112)$$

3. setting the **upper gain margin**,  $\bar{g}$ , as  $\min_i g_{0,i} > 1$ ; and
4. setting the **lower gain margin**,  $\underline{g}$ , as  $\max_i g_{0,i} < 1$ .

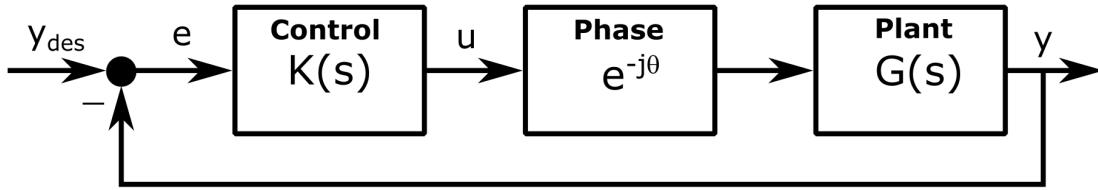
Then, one may state that the feedback control system is stable for

$$\underline{g} \leq g \leq \bar{g} \quad (5.113)$$

where it should be noted that not all systems will have an upper *and* lower gain margin. For FDC, a feedback control system is generally considered sufficiently robust if  $\underline{g} \leq 0.5$  and  $\bar{g} \geq 2$ , i.e.  $\underline{g} \leq -6$  dB and  $\bar{g} \geq 6$  dB, for the gain margin.

### Phase and Delay Margin

The **phase margin** is the amount of phase lead/lag that can be added to the input before closed-loop instability. The phase margin can be derived by considering a phase perturbation to the plant dynamics model which can be modeled in block diagram form as



Then, defining the perturbed open-loop transfer function as  $e^{-j\theta}L(s)$  the phase margin is determined by assessing for what values of  $\theta$  the feedback control system will go unstable.

Recalling that the feedback control system is stable if (1) there are no pole-zero cancellations and (2) the zeros of  $1+L(s)=0$  are only in the LHP. Then, a critical phase,  $\theta_0$ , occurs when the zeros of  $1+e^{-j\theta}L(j\omega_0)$  are on the imaginary axis for some critical frequency  $\omega_0$ , i.e.

$$1 + e^{-j\theta_0}L(j\omega_0) = 0 \quad (5.114)$$

Rewriting this expression, one has

$$L(j\omega_0) = -e^{j\theta_0} \quad (5.115)$$

or in polar form (i.e. gain and phase), one has

$$|L(j\omega_0)|e^{j\angle L(\omega_0)} = -e^{j\theta_0} \quad (5.116)$$

or the equivalent magnitude/gain and phase requirements become

$$|L(j\omega_0)| = 1 \quad (5.117)$$

and

$$\theta_0 = \pm 180^\circ + \angle L(j\omega_0) \quad (5.118)$$

where  $\theta_0$  is typically represented as a positive number as the phase margin can be referenced to either  $\pm 180^\circ$  and is symmetric for positive and negative  $\theta$ . Thus, one can identify the phase margin from the Bode plot by:

1. identifying critical frequencies  $\omega_{0,i}$  where  $|L(j\omega_{0,i})| = 1$ ;
2. calculating associated critical phase  $\theta_{0,i}$  using distance from  $180^\circ$ , i.e.

$$\theta_{0,i} = \pm 180^\circ + \angle L(j\omega_0) \quad (5.119)$$

3. setting the phase margin,  $\bar{\theta}$ , as  $\min_i |\theta_{0,i}|$

Then, one may state that the feedback control system is stable for

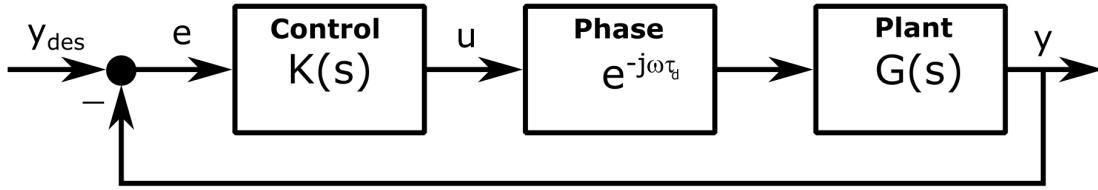
$$-\bar{\theta} \leq \theta \leq \bar{\theta} \quad (5.120)$$

For FDC, a feedback control system is generally considered sufficiently robust if  $\bar{\theta} \geq 45^\circ$  for the phase margin.

Alternatively, the phase margin can be rewritten in terms of a (time) delay margin. The **(time) delay margin** is the amount of time delay that can be added to the input before closed-loop instability. To which uses the fact that the critical time delay,  $\tau_d$ , is related to  $\theta_0$  by

$$\tau_d = \frac{\theta_0}{\omega_0} \quad (5.121)$$

which can also be modeled in block diagram form as



### Disk Margin

The **disk margin** is the minimum frequency response perturbation “distance” allowable before closed-loop instability. The disk margin is a general margin for the feedback control system based on a stability perturbation analysis of the frequency response of  $L(j\omega)$  using the Nyquist plot. Recall that the feedback control system will be unstable for some perturbation  $-\Delta L(j\omega_0)$  at a critical frequency  $\omega_0$  if

$$1 + L(j\omega_0) - \Delta L(j\omega_0) = 0 \quad (5.122)$$

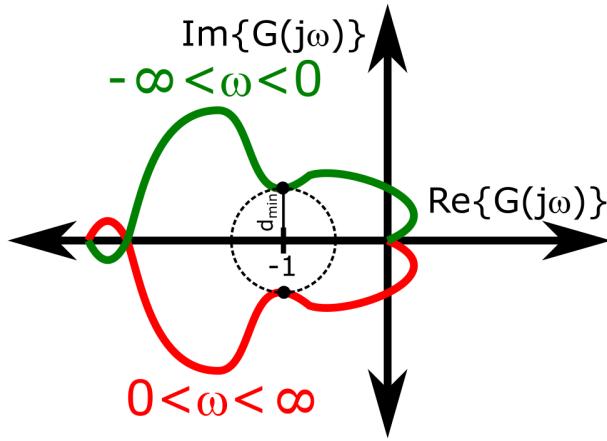
or rearranging, one has

$$\Delta L(j\omega_0) = 1 + L(j\omega_0) \quad (5.123)$$

Thus, one can identify the disk margin,  $d_{min}$  from the Nyquist plot by calculating

$$d_{min} = \min_{0 \leq \omega < \infty} |1 + L(j\omega)| \quad (5.124)$$

which can be considered as a “distance” from  $s = -1$  in the complex domain. Note that the term “disk” derives from the idea that one is forming the minimum circular disk about the critical  $s = -1$  point, i.e.



Also, note that this can be written in terms of the sensitivity function as

$$\frac{1}{d_{\min}} = \max_{0 \leq \omega < \infty} |S(j\omega)| \quad (5.125)$$

For FDC, a feedback control system is generally considered sufficiently robust if  $d_{\min} \geq 0.4$  for the disk margin.

Furthermore, the gain and phase margins are two different “directions” in the complex domain that one is assessing the distance from critical  $s = -1$  point. In particular, by noting for the different critical frequencies, the gain margin is

$$L(j\omega_0) = -\frac{1}{g_0} \quad (5.126)$$

which is a real number and the phase margin is

$$L(j\omega_0) = -e^{j\theta_0} \quad (5.127)$$

which is a circle of radius 1 centered at the origin in the complex plane. Then, noting the disk margin criterion as

$$L(j\omega_0) = 1 + \Delta L(j\omega_0) \quad (5.128)$$

one can see that (1) the gain margin assesses the distance from  $s = -1$  along the real axis, i.e.

$$g_0 = -\frac{1}{1 + \Delta L(j\omega_0)} \quad (5.129)$$

where  $\Delta L(j\omega_0)$  must be a real number perturbation and (2) the phase margin assesses the distance from  $s = -1$  along the unit circle, i.e.

$$\theta_0 = \pm 180 + \angle \Delta L(j\omega_0) \quad (5.130)$$

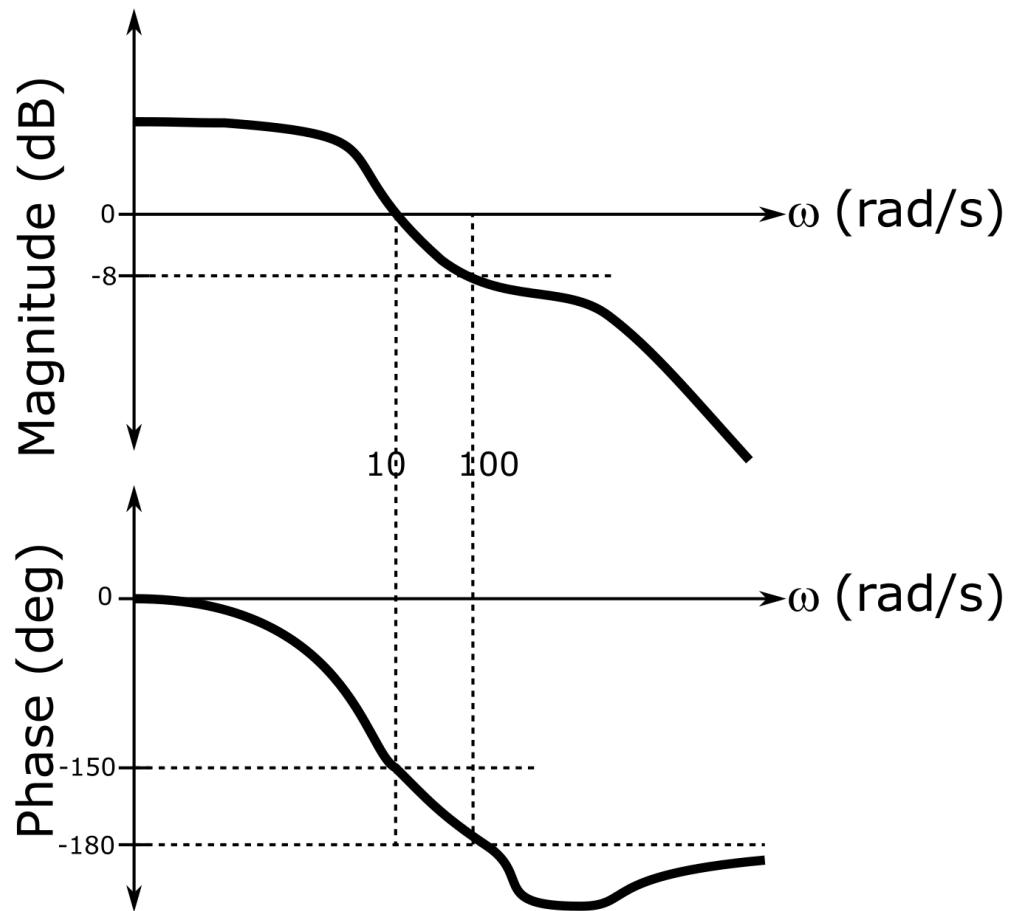
where recall that

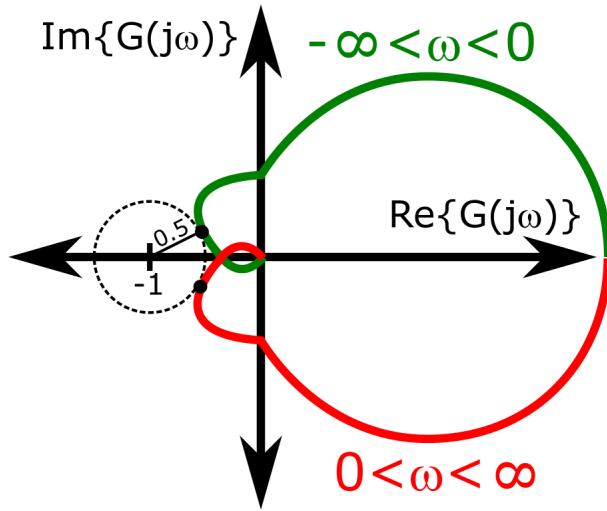
$$\angle \Delta L(j\omega_0) = \tan^{-1} \frac{\text{Im}\{\Delta L(j\omega_0)\}}{\text{Re}\{\Delta L(j\omega_0)\}} \quad (5.131)$$

With these definitions, a disk margin of 0.4 corresponds to gain margins of  $\underline{g} = 0.71$  and  $\bar{g} = 1.67$  and a phase margin of  $\bar{\theta} = 23^\circ$ . However, when considering the disk margin in the Nyquist domain, one typically will get the margin not in these directions. Thus, the traditional gain and phase margins are typically evaluated without considering all possible “directions” in the complex plane. Thus, why considering only these stability margins for robustness are typically more conservative than the disk margin equivalent, i.e.  $\underline{g} = 0.5$ ,  $\bar{g} = 2$ , and  $\bar{\theta} = 45^\circ$ .

**Example Problem**

Given: the Bode and Nyquist plots for  $L(j\omega)$  as





Determine:

- the gain margin
- the phase margin
- the time delay margin
- the disk margin

Solution:

a)  $\angle L(j\omega_0) = -180^\circ$  only once at  $\omega_0 = 100$  rad/s. Here,  $|L(j\omega_0)| = -8$  dB. Thus, the gain margin is given by

$$\frac{1}{\bar{g}} = -8 \text{ dB} \quad (5.132)$$

or

$$\bar{g} = \frac{1}{-8 \text{ dB}} \quad (5.133)$$

$$\underline{\bar{g} = 8 \text{ dB}} \quad (5.134)$$

b)  $|L(j\omega_0)| = 0$  dB only once at  $\omega_0 = 10$  rad/s. Here,  $\angle L(j\omega_0) = -150^\circ$ . Thus, the phase margin is given by the minimum

$$\theta_0 = \pm 180^\circ - 150^\circ \quad (5.135)$$

or

$$\underline{\theta_0 = 30^\circ} \quad (5.136)$$

c) The time delay margin is related to the phase margin by

$$\tau_d = \frac{\theta_0}{\omega_0} \quad (5.137)$$

or

$$\tau_d = \frac{30^\circ \left( \frac{\pi}{180^\circ} \right)}{10 \text{ rad/s}} \quad (5.138)$$

$$\underline{\tau_d = 0.0524 \text{ s}} \quad (5.139)$$

d) The disk margin is computed using

$$d_{min} = \min_{0 \leq \omega < \infty} |1 + L(j\omega)| \quad (5.140)$$

By inspection one can see the minimum occurs for the dashed disk shown, i.e.

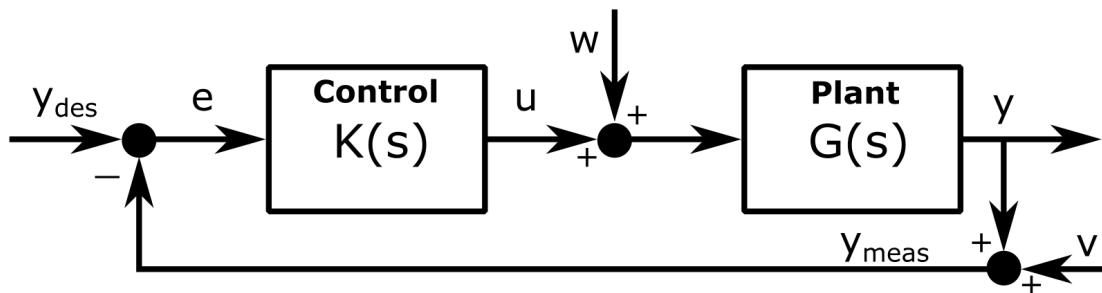
$$\underline{d_{min} = 0.5} \quad (5.141)$$

# Chapter 6

## SISO Loop-Shaping Robust Control

### 6.1 SISO Loop-Shaping Control Stages

Recall the standard feedback control system block diagram



where in loop-shaping control design,  $K(s)$  is chosen so that the open-loop transfer function  $L(s) = G(s)K(s)$  meets certain specifications.

The primary aspect of loop-shaping is that  $K(s)$  has an additive effect on the magnitude subplot of the Bode plot when considered in decibels, i.e.

$$20 \log_{10} |L(j\omega)| = 20 \log_{10} |G(j\omega)K(j\omega)| = 20 \log_{10} |G(j\omega)| + 20 \log_{10} |K(j\omega)| \quad (6.1)$$

Thus, to form a suitable  $L(s)$ , one can use **loop-shaping control stages** for different regions of the Bode plot using this additive property for shaping  $L(s)$ . Then, multiplying each stage together will provide the full controller transfer function  $K(s)$ . This part of the textbook will consider the following control stages:

1. Proportional Gain
2. Integral Boost
3. Low Frequency Boost
4. Lag
5. Rolloff
6. Lead

where it should be noted that all control stages are not necessary for a sufficient controller and typically the least number of stages, i.e. less complexity, is preferred by control engineers.

### Control Stage: Proportional Gain

This control stage is used to shift  $L(s)$  up or down, i.e. to set the crossover frequency,  $\omega_c$ , a.k.a. the bandwidth, to have the system respond as fast as necessary. The transfer function for this control stage is defined as

$$K(s) = \beta \quad (6.2)$$

where  $\beta$  is the gain. Note that  $|\beta| > 1$  shifts  $L(s)$  up while  $|\beta| < 1$  shifts  $L(s)$  down. Note also that  $\beta > 0$  or  $\beta < 0$  should be chosen so that the feedback control system is stabilized while also not being too large as to violate any control effort constraints. As an example, consider the following system

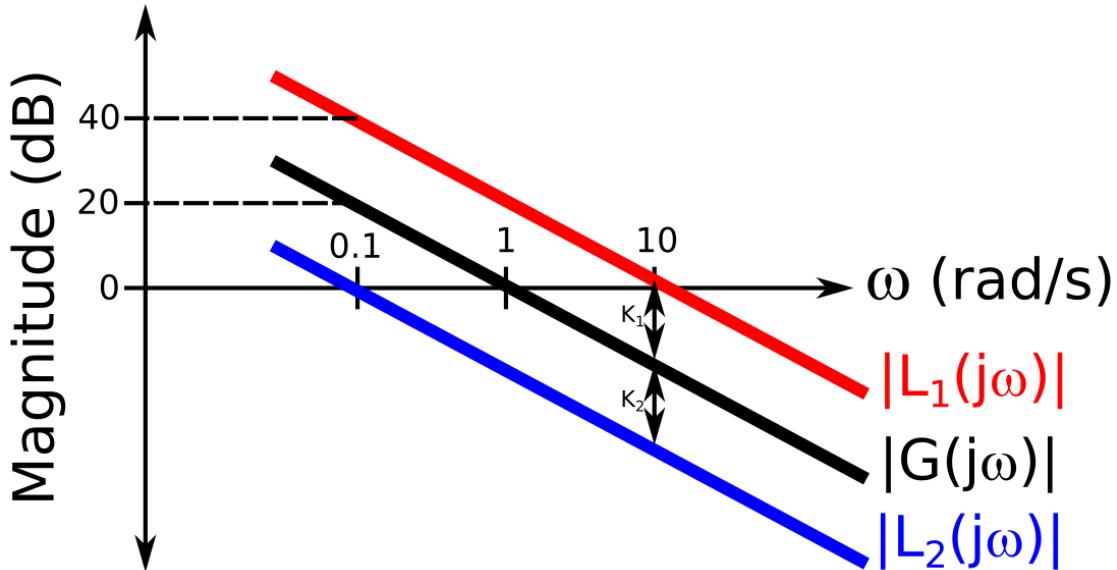
$$G(s) = \frac{1}{s} \quad (6.3)$$

$$K_1(s) = 10 \quad (6.4)$$

$$K_2(s) = 0.1 \quad (6.5)$$

$$L_1(s) = \frac{10}{s} \quad (6.6)$$

$$L_2(s) = \frac{0.1}{s} \quad (6.7)$$



### Control Stage: Integral Boost

This control stage is used to increase the gain of  $L(s)$  for the region  $\omega < \bar{\omega}$ , i.e. to have good tracking at low frequencies. The transfer function for this control stage is defined as

$$K(s) = \frac{s + \bar{\omega}}{s} \quad (6.8)$$

where  $\bar{\omega}$  is the chosen corner frequency/bandwidth to start the gain increase. This stage has the following notable properties:

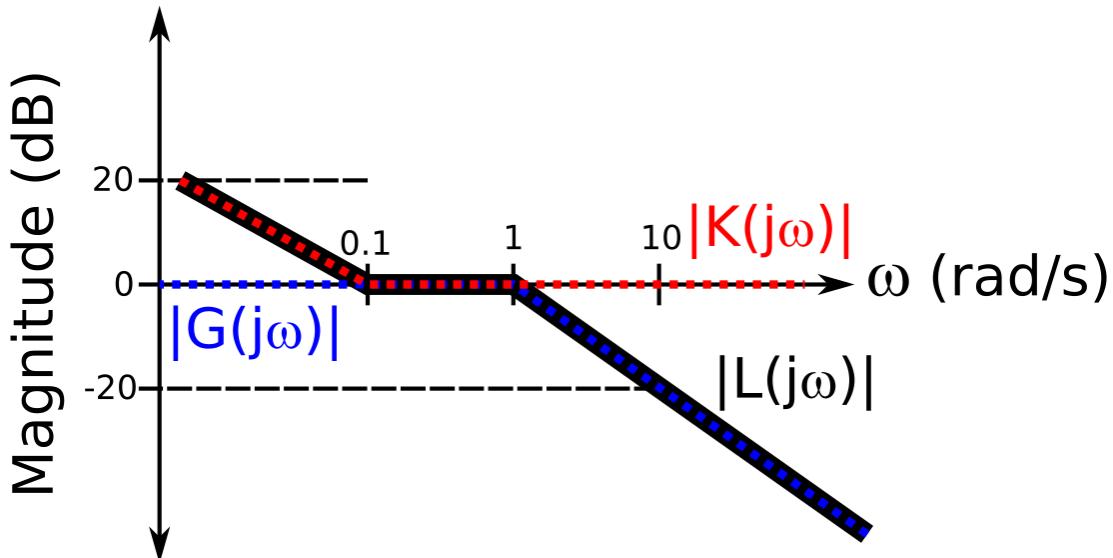
1.  $K(s)$  has a pole at  $s = 0$  and a zero at  $-\bar{\omega}$ .
2.  $K(j\omega) \rightarrow +\infty$  as  $\omega \rightarrow 0$ , thus  $e_{ss} = 0$ .
3. At low  $\omega$ ,  $K(j\omega) \approx \frac{\bar{\omega}}{j\omega}$ , i.e. for every  $10\omega \rightarrow K(j\omega)/10$  or a -20 dB/10ω slope.
4. At high  $\omega$ ,  $K(j\omega) \approx 1$ , i.e. no effect (0 dB).
5.  $K(s)$  corresponds to a PI controller with  $K_p = 1$  and  $K_i = \bar{\omega}$ .

As an example, consider the following system

$$G(s) = \frac{1}{s + 1} \quad (6.9)$$

$$K(s) = \frac{s + 0.1}{s} \quad (\bar{\omega} = 0.1) \quad (6.10)$$

$$L(s) = \frac{s + 0.1}{s^2 + s} \quad (6.11)$$



### Control Stage: Low Frequency Boost

This control stage is used to increase the gain of  $L(s)$  for the region  $\omega < \bar{\omega}$ , i.e. to have good tracking at low frequencies. The transfer function for this control stage is defined as

$$K(s) = \frac{s + \bar{\omega}}{s + \frac{\bar{\omega}}{\beta}} \quad (6.12)$$

where  $\bar{\omega}$  is the corner frequency/bandwidth and  $\beta > 1$  is the low frequency gain. This stage has the following notable properties:

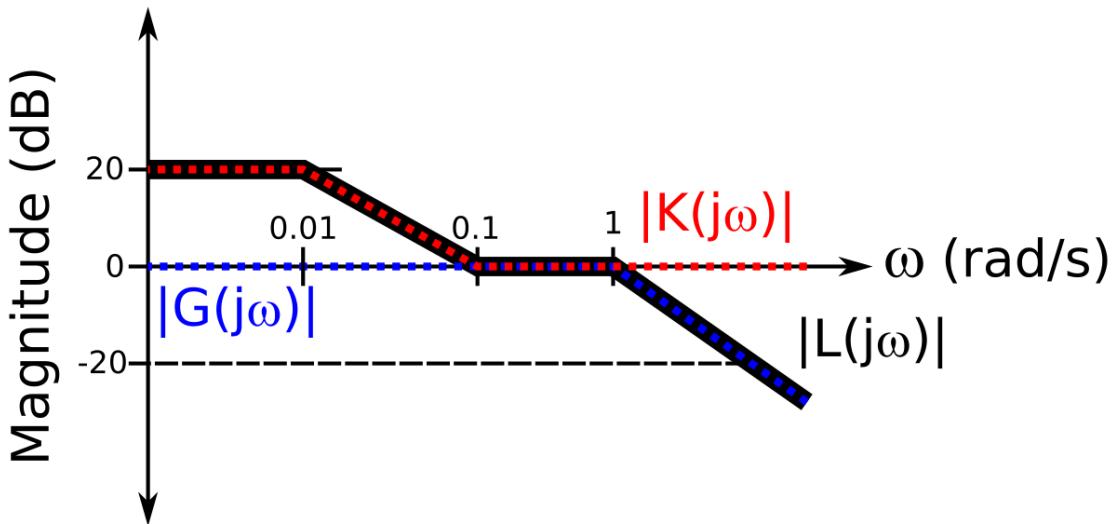
1.  $K(s)$  has a pole at  $s = -\frac{\bar{\omega}}{\beta}$  and a zero at  $-\bar{\omega}$ .
2. At low  $\omega$ ,  $K(j\omega) \approx \beta$  or  $20 \log_{10} \beta$  dB.
3. At high  $\omega$ ,  $K(j\omega) \approx 1$  or 0 dB.
4.  $K(s)$  is similar to integral boost with  $\bar{\omega}$ , but levels off at the low frequency gain  $\beta$ , i.e. as  $\beta \rightarrow \infty$ , this stage approaches an integral boost stage.

As an example, consider the following

$$G(s) = \frac{1}{s + 1} \quad (6.13)$$

$$K(s) = \frac{s + 0.1}{s + 0.01} \quad (\bar{\omega} = 0.1, \beta = 10) \quad (6.14)$$

$$L(s) = \frac{s + 0.1}{(s^2 + 1.01s + 0.01)} \quad (6.15)$$



### Control Stage: Lag

This control stage is used to increase the gain for the region  $\omega < \bar{\omega}$ , i.e. to have good tracking at low frequencies, while also simultaneously shifting  $L(s)$  down. The transfer function for this control stage is defined as

$$K(s) = \frac{\frac{1}{\beta}s + \bar{\omega}}{s + \frac{\bar{\omega}}{\beta}} \quad (6.16)$$

where  $\bar{\omega}$  and  $\beta > 1$  are chosen constants. This stage has the following notable properties:

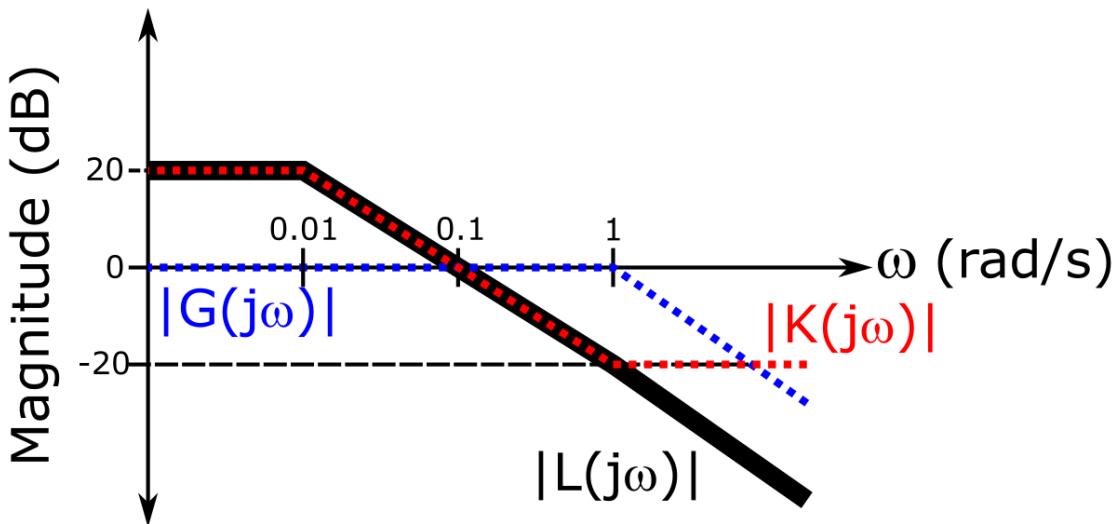
1.  $K(s)$  has a pole at  $s = -\beta\bar{\omega}$  and a zero at  $-\frac{\bar{\omega}}{\beta}$ .
2. At low  $\omega$ ,  $K(j\omega) \approx \beta$  or  $20 \log_{10} \beta$  dB.
3. At high  $\omega$ ,  $K(j\omega) \approx \frac{1}{\beta}$  or  $-20 \log_{10} \beta$  dB.
4. At  $\omega = \bar{\omega}$ ,  $|K(j\omega)| = 1$  or 0 dB.
5.  $K(s)$  is similar to a PI controller with  $K_p = \frac{1}{\beta}$  and  $K_i = \bar{\omega}$ , except this stage levels off at  $\frac{\bar{\omega}}{\beta}$ .

As an example, consider the following

$$G(s) = \frac{1}{s+1} \quad (6.17)$$

$$K(s) = \frac{0.1s + 0.1}{s + 0.01} \quad (\bar{\omega} = 0.1, \beta = 10) \quad (6.18)$$

$$L(s) = \frac{0.1s + 0.1}{(s^2 + 1.01s + 0.01)} \quad (6.19)$$



### Control Stage: Lead

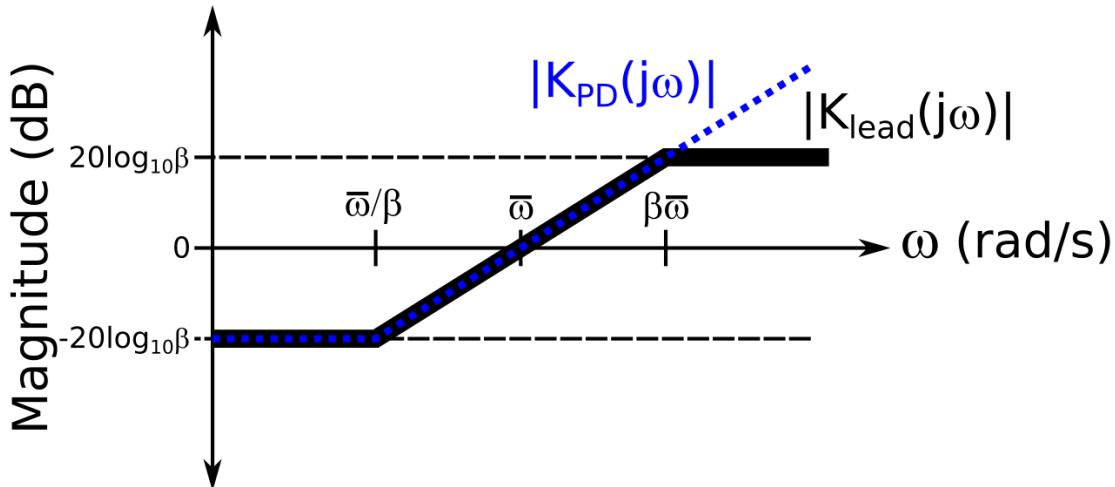
This control stage is used to increase slope of  $L(s)$  at the region around  $\bar{\omega}$ , i.e. to maintain stability and robustness in the crossover region. The transfer function for this control stage is defined as

$$K(s) = \frac{\beta s + \bar{\omega}}{s + \beta\bar{\omega}} \quad (6.20)$$

where  $\bar{\omega}$  and  $\beta > 1$  are chosen constants. This stage has the following notable properties:

1.  $K(s)$  has a pole at  $s = -\beta\bar{\omega}$  and a zero at  $-\frac{\bar{\omega}}{\beta}$ .
2. At low  $\omega$ ,  $|K(j\omega)| \approx \frac{1}{\beta}$  or  $-20 \log_{10} \beta$  dB.
3. At high  $\omega$ ,  $|K(j\omega)| \approx \beta$  or  $20 \log_{10} \beta$  dB.
4. At  $\omega = \bar{\omega}$ ,  $|K(j\omega)| = 1$  or 0 dB.
5. This is similar to a PD controller with  $K_p = \bar{\omega}$  and  $K_d = \beta$ , except this stage levels off at  $\beta\bar{\omega}$ , thus the lead control stage is better for keeping  $L(s)$  low at high frequencies than a pure derivative gain (as previously discussed).
6.  $K(s)$  is the converse of a lag control stage, i.e. if  $\beta < 1$  here  $K(s)$  becomes a lag control stage.

As an example, consider the following



It should be noted that a common simple control strategy in the frequency domain is **lead-lag control** which uses the lead stage for stability and robustness and the lag stage for having good tracking at low frequencies. This may have a proportional boost as well for the controller. Thus, lead-lag control with a proportional boost can be considered a variant of PID control as the lag stage corresponds to a type of PI controller and the lead stage corresponds to a type of PD controller but with each having leveling.

### Control Stage: Rolloff

This control stage is used to decrease the gain of  $L(j\omega)$  for the region  $\omega > \bar{\omega}$ , i.e. to have noise filtering at high frequencies. The transfer function for this control stage is defined as

$$K(s) = \frac{\bar{\omega}}{s + \bar{\omega}} \quad (6.21)$$

where  $\bar{\omega}$  is the corner frequency/bandwidth. This stage has the following notable properties:

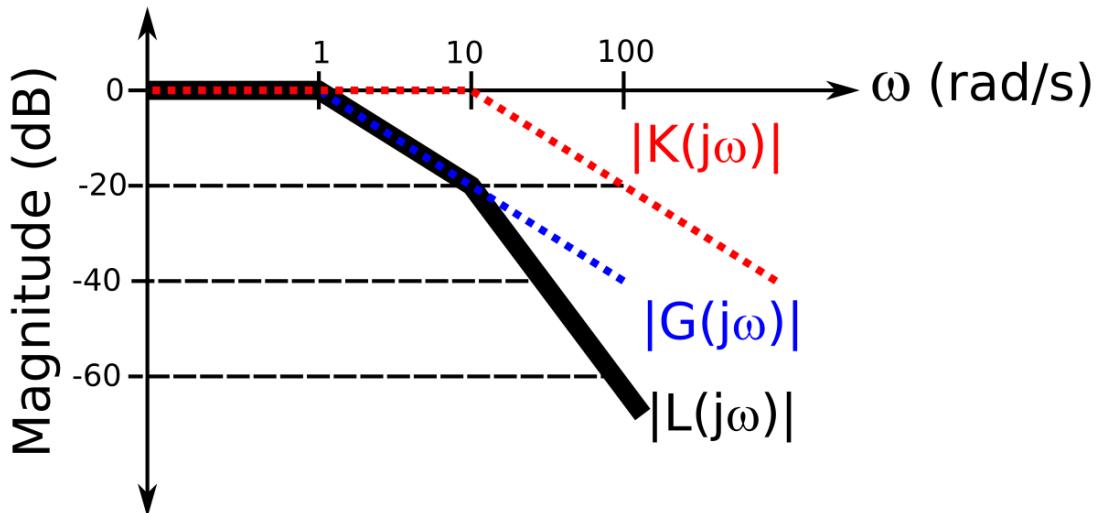
1.  $K(s)$  has a pole at  $s = -\bar{\omega}$ .
2. At low  $\omega$ ,  $K(j\omega) \approx 1$  or 0 dB.
3. At high  $\omega$ ,  $K(j\omega) \approx \frac{\bar{\omega}}{j\omega}$ , i.e. for every  $10\omega \rightarrow K(j\omega)/10$  or a -20 dB/10ω slope.

As an example, consider the following

$$G(s) = \frac{1}{s + 1} \quad (6.22)$$

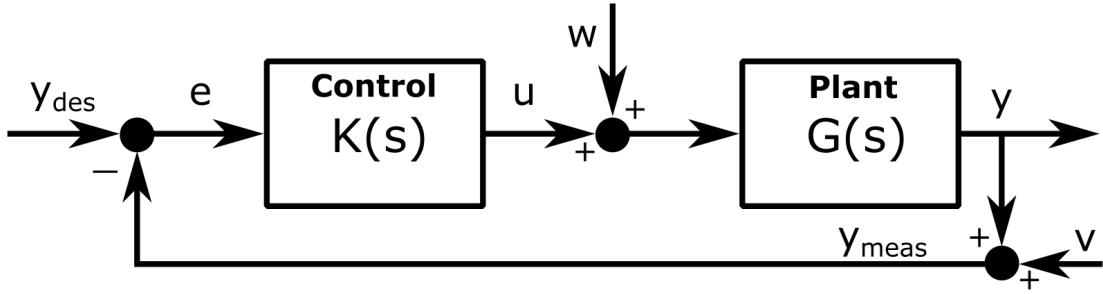
$$K(s) = \frac{10}{s + 10} \quad (\bar{\omega} = 10) \quad (6.23)$$

$$L(s) = \frac{10}{(s^2 + 11s + 10)} \quad (6.24)$$

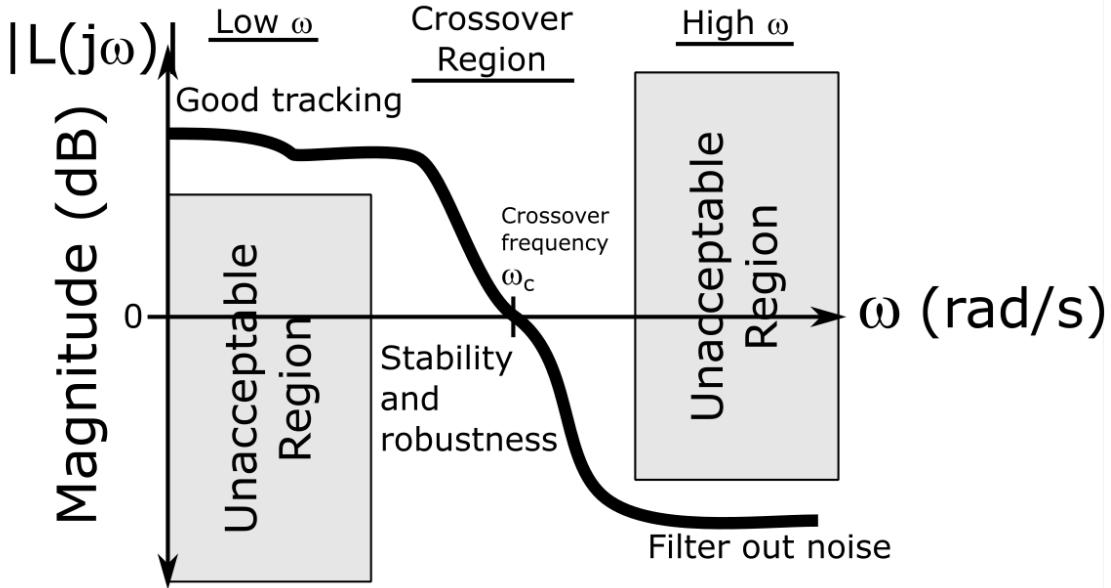


## 6.2 SISO Loop-Shaping Control Design

Recall the standard feedback control system block diagram



and the associated frequency response of the open-loop transfer function:



This section will discuss the crossover region requirements as well as relate control design criteria defined in the frequency response to those defined in the step response.

### Crossover Region Requirements

At a bare minimum, the feedback control system must be stable, i.e. the roots/poles of the characteristic equation must be in the LHP, regardless of whether one designs the control law using the time domain or frequency domain. In addition, one typically also requires some stability margin for the system due to uncertainty in the plant model  $G(s)$ . To provide some guidance on connecting stability margins to the loop-shaping control design method consider the following derivation.

The **Bode gain-phase formula** states that if  $L(0) > 0$  and has all its poles and zeros in LHP, then the

phase of  $L(j\omega)$  (in radians) at  $\omega$  is

$$\angle L(j\omega) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{d|L|_{\text{dB}}(\zeta)}{d\nu} \log_{10} \coth \left| \frac{\nu}{2} \right| d\nu \quad (6.25)$$

where  $\nu = \log_{10} \frac{\zeta}{\omega}$ . Next, noting that

$$\log_{10} \coth \left| \frac{\nu}{2} \right| \approx \frac{1}{2} \pi^2 \delta(\nu) \quad (6.26)$$

one has the approximation

$$\angle L(j\omega) \approx \frac{\pi}{2} \left( \frac{d|L|_{\text{dB}}(\zeta)}{d\nu} \right)_{\zeta=\omega} \quad (6.27)$$

or in degrees

$$\angle L(j\omega) \approx \frac{90^\circ}{20 \text{ dB/decade}} (|L(j\omega)|_{\text{dB}}/\text{decade}) \quad (6.28)$$

Thus, if  $|L(j\omega)|_{\text{dB}}/\text{decade} = -30 \text{ dB/decade}$  over roughly 1 decade of  $\omega$ , then the phase approximately satisfies

$$\angle L(j\omega) \approx \frac{90^\circ}{20 \text{ dB/decade}} (-30 \text{ dB/decade}) \quad (6.29)$$

or

$$\angle L(j\omega) \approx -135^\circ \quad (6.30)$$

Then, recall for FDC that one typically requires  $\bar{\theta} \geq 45^\circ$  (which is the phase difference between  $\pm 180^\circ$  and  $\angle L(j\omega_c)$  where  $\omega_c$  defines  $|L(j\omega_c)| = 1 = 0 \text{ dB}$ ) and combining with the gain margin requirements of  $\pm 6 \text{ dB}$  (which occur at  $\angle L(j\omega) = \pm 180^\circ$ ), the open-loop transfer function,  $L(s)$ , should roughly satisfy the following:

1. no poles or zeros in the RHP;
2.  $|L(0)| > 0$ ;
3. a single gain crossover frequency  $\omega_c$ ;
4.  $|L(j\omega)|_{\text{dB}}/\text{decade} \geq -30 \text{ dB/decade}$  for  $\frac{\omega_c}{\sqrt{10}} < \omega < \sqrt{10}\omega_c$ ;
5.  $|L|_{\text{dB}} \geq 6 \text{ dB}$  for  $\omega \leq \frac{\omega_c}{\sqrt{10}}$ ; and
6.  $|L|_{\text{dB}} \leq -6 \text{ dB}$  for  $\omega \geq \sqrt{10}\omega_c$ .

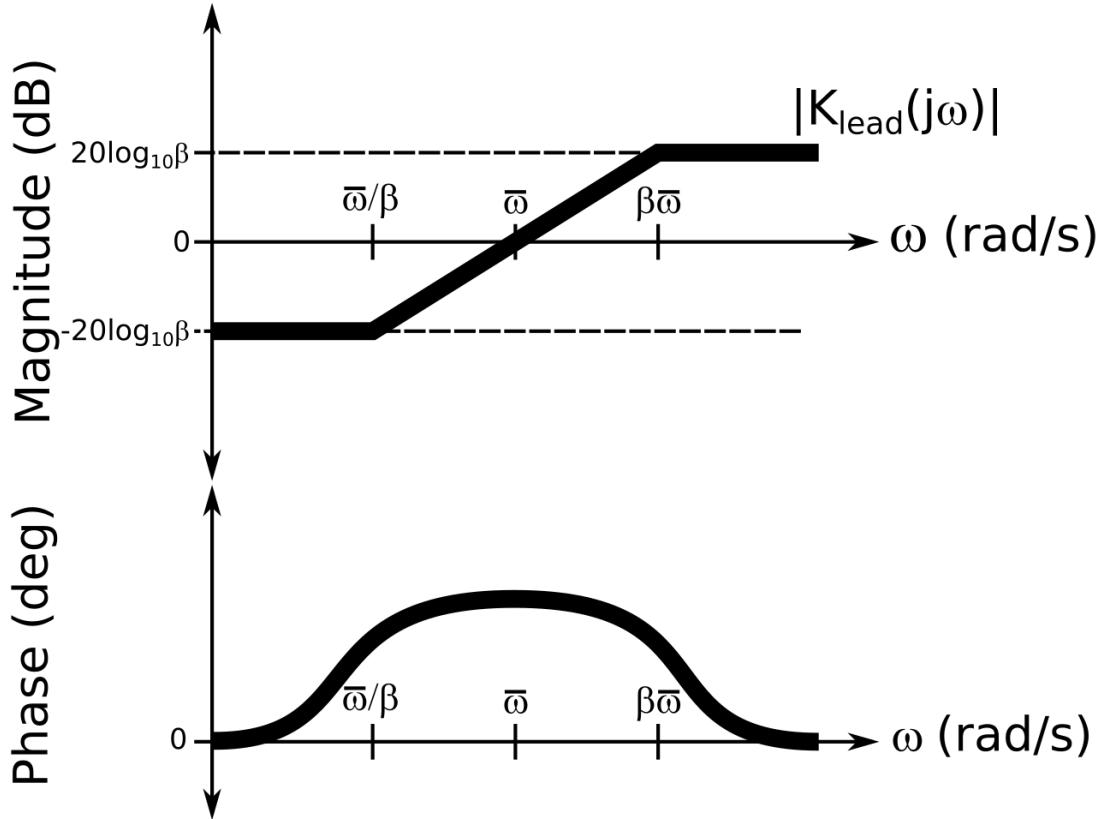
Then, one can confidently claim that the feedback control system is stable and achieves good classical margins (i.e.  $\pm 6 \text{ dB}$ ,  $\pm 45^\circ$ ).

Though these assumptions do not hold for all LTI systems, this approximation is generally decent and provides at least a good starting point for how much to limit the gain slope about  $\omega_c$ . It should be noted that if  $L(s)$  has zeros or poles in the RHP, then there may not exist a suitable  $K(s)$ , but at the very least there will be some additional phase introduced at  $\omega_c$  which will require a larger slope than  $-30 \text{ dB/decade}$ . This additional phase is why a system with no poles or zeros in the RHP is called a **minimum phase system**. For non-minimum phase systems, additional control design must be performed case-by-case. Lastly, it should be noted that the single gain crossover frequency  $\omega_c$  for  $|L(j\omega_c)| = 0 \text{ dB}$  in this case is also called the **loop bandwidth**.

To increase the slope of  $|L(j\omega)|$  in loop-shaping control design, recall that one typically uses a lead control stage which has a transfer function

$$K_{lead}(s) = \frac{\beta s + \bar{\omega}}{s + \beta\bar{\omega}} \quad (6.31)$$

where  $\bar{\omega}$  and  $\beta > 1$  are chosen constants. The Bode plot of this controller is



where it should be noted that around  $\bar{\omega}$ ,  $K_{lead}(j\omega)$  has a positive slope and therefore a positive phase by the Bode gain-phase formula and as  $\beta \rightarrow \infty$ ,  $\angle K_{lead}(j\omega) \rightarrow +90^\circ$ . Furthermore, by choosing  $\bar{\omega} = \omega_c$  and  $\beta$ , one can increase the slope of  $|L(j\omega)|$  to satisfy the stability and any phase margin requirement, but no more than  $90^\circ$ .

## Time to Frequency Domain Conversion of Requirements

For PID control design in the time domain, in addition to stability, the control law was typically required to have certain unit step response characteristics, i.e.

1.  $t_s \leq t_{s,req}$

2.  $|e_{ss}| \leq e_{ss,req}$
3.  $|u| \leq u_{req}$

where  $t_{s,req}$ ,  $e_{ss,req}$ , and  $u_{req}$  are some values where  $e_{ss,req}$  and  $u_{req}$  may alternatively be percentages.

For loop-shaping control design in the frequency domain, one places requirements on the frequency response of the open-loop transfer function, i.e. the Bode magnitude plot of  $L(j\omega)$ . However, these two ideas are connected and this section will show how unit step response requirements can be translated to frequency response requirement.

For requirement 1,  $t_s \leq t_{s,req}$ , as a rough approximation, the loop bandwidth is related to the settling time roughly by

$$t_s \approx \frac{3}{\omega_c} \quad (6.32)$$

Thus, one can define the approximate relationship that

$$t_s \leq t_{s,req} \rightarrow \omega_c \geq \frac{3}{t_{s,req}} \quad (6.33)$$

However, for some systems this approximation can be significantly off, but in general increasing  $\omega_c$  will decrease  $t_s$ .

For requirement 2,  $|e_{ss}| \leq e_{ss,req}$ , the transfer function from  $Y_c(s) \rightarrow E(s)$  is the sensitivity function, i.e.

$$S(s) = \frac{E(s)}{Y_c(s)} \quad (6.34)$$

and using the final value theorem

$$e_{ss} = \lim_{s \rightarrow 0} sS(s)Y_c(s) \quad (6.35)$$

Thus, for  $|e_{ss}| \leq e_{ss,req}$  and a unit step, i.e.  $s = 0$  and  $Y_c = \frac{1}{s}$ , one has

$$|e_{ss}| = \left| \lim_{s \rightarrow 0} sS(s) \frac{1}{s} \right| \leq e_{ss,req} \quad (6.36)$$

or

$$|S(0)| \leq e_{ss,req} \quad (6.37)$$

and since

$$|S(s)| = \left| \frac{1}{1 + G(s)K(s)} \right| = \left| \frac{1}{1 + L(s)} \right| \quad (6.38)$$

one has

$$|1 + L(s)| = \left| \frac{1}{S(s)} \right| \quad (6.39)$$

or by the triangle inequality (whose proof is left to the reader)

$$|L(s)| \geq \left| 1 + \frac{1}{S(s)} \right| \quad (6.40)$$

Thus, for a requirement on  $|S(s)| \leq e_{ss,req}$ , one can define the approximate relationship that

$$|e_{ss}| \leq e_{ss,req} \rightarrow |L(0)| \geq 1 + \frac{1}{e_{ss,req}} \quad (6.41)$$

where it should be noted that typically one includes additional low frequency requirements on the error besides only at  $\omega = 0$ .

For requirement 3,  $|u| \leq u_{req}$ , the transfer function from  $Y_c(s) \rightarrow U(s)$  is

$$\frac{U(s)}{Y_c(s)} = \frac{K(s)}{1 + G(s)K(s)} \quad (6.42)$$

Thus, for a unit step constraint on  $|u| \leq u_{req}y_c$ ,

$$\left| \frac{K(j\omega)}{1 + G(j\omega)K(j\omega)} \right| \leq u_{req} \quad \forall \omega \quad (6.43)$$

Note that as  $|K(j\omega)| \rightarrow \infty$ ,

$$\left| \frac{K(j\omega)}{1 + G(j\omega)K(j\omega)} \right| \rightarrow \left| \frac{1}{G(j\omega)} \right| \quad (6.44)$$

Thus, this requirement is satisfied for any  $\omega$  where

$$|G(j\omega)| \geq \frac{1}{u_{req}} \quad (6.45)$$

Thus, one is only concerned with regions where  $G(j\omega)$  is small. Here, then  $K(j\omega)$  should not be large such that

$$\left| \frac{K(j\omega)}{1 + L(j\omega)} \right| \leq u_{req} \quad \forall \omega \quad (6.46)$$

Thus, in loop-shaping control design

1.  $\omega_c$  cannot be arbitrarily higher than where  $|G|$  starts to naturally rolloff.
2. Use a rolloff control stage at high frequencies for  $K(s)$ . This is usually done for high frequency noise rejection anyway.

Summarizing, the equivalent  $L(j\omega)$  requirements are:

1.  $t_s \leq t_{s,req} \rightarrow \omega_c$  of  $L(j\omega) = 0$  dB  $\geq \frac{3}{t_{s,req}}$
2.  $|e_{ss}| \leq e_{ss,req} \rightarrow |L(0)| \geq 1 + \frac{1}{e_{ss,req}}$
3.  $|u| \leq u_{req} \rightarrow \left| \frac{K(j\omega)}{1 + L(j\omega)} \right| \leq u_{req}$

Note that typically there are additional frequency response requirements

For example, one typically requires filtering out noise at high frequencies, i.e.

$$|T(j\omega)| \leq g \quad \forall \omega > \omega_{high} \quad (6.47)$$

But by definition

$$T(j\omega) = \frac{L(j\omega)}{1 + L(j\omega)} \quad (6.48)$$

Thus, one requires that

$$\left| \frac{1 + L(j\omega)}{L(j\omega)} \right| \leq \frac{1}{g} \quad (6.49)$$

or if

$$\left| \frac{1}{L(j\omega)} \right| \geq \frac{1}{g} + 1 \quad (6.50)$$

then

$$\left| \frac{1 + L(j\omega)}{L(j\omega)} \right| \geq \left| \frac{1}{L(j\omega)} \right| - 1 \geq \frac{1}{g} \quad (6.51)$$

by the triangle inequality. Then, these type of requirements can be converted to

$$\left| \frac{1}{L(j\omega)} \right| \leq \frac{1+g}{g} \quad (6.52)$$

or

$$|L(j\omega)| \leq \frac{g}{1+g} \quad \forall \omega > \omega_{high} \quad (6.53)$$

Note, this high frequency requirement also typically serves to keep the control effort  $|K(j\omega)|$  low as  $G(j\omega)$  will be low past its natural bandwidth.

### Loop-Shaping Design Procedure

With this in mind, for the general loop-shaping method, one can consider the following design procedure for  $K(s)$ .

1. Use a proportional gain control stage to set the desired loop bandwidth,  $\omega_c$ 
  - May need to obtain requirements on  $\omega_c$  from  $t_s$
2. Use an integral or low frequency boost control stage to increase  $|L(j\omega)|$  at low  $\omega$ , i.e. good tracking.
  - May need to obtain requirements on  $L(j\omega)$  from  $S(j\omega)$  or  $e_{ss}$
3. Use a high frequency rolloff control stage to decrease  $|L(j\omega)|$  at high  $\omega$ , i.e. good noise filtering.
  - May need to obtain requirements on  $L(j\omega)$  from  $T(j\omega)$
4. Use a lead control stage to reduce  $|L(j\omega)|$  slope about  $\omega_c$  to  $> -30 \text{ dB}/10\omega$  for stability and robustness.
5. Iterate until all requirements are satisfied.

The primary iteration must be used typically because that  $K_{lead}(s)$  decreases the gain for  $\omega$  lower than  $\bar{\omega}$  and increases the gain for  $\omega$  higher than  $\bar{\omega}$ . An example of this loop-shaping procedure will be demonstrated in the next section.

### Example Problem

Given: the single integrator plant model

$$G(s) = \frac{10}{s} \quad (6.54)$$

Determine: a feedback controller,  $K(s)$ , such that a standard feedback control system satisfies requirements:

1. stability
2. loop bandwidth near 1 rad/s
3.  $\leq 4\%$  tracking error for frequencies below 0.1 rad/s
4.  $\leq 4\%$  noise gain for frequencies above 10 rad/s

#### Solution:

For requirement 2, one can set the loop bandwidth  $\omega_c$  with a proportional gain control stage, i.e.

$$K_1(s) = \beta \quad (6.55)$$

The loop bandwidth is defined as

$$1 = |L(j\omega_c)| = |G(j\omega_c)K_1(j\omega_c)| = \left| \frac{10}{j\omega_c} \beta \right| \quad (6.56)$$

for  $\omega_c = 1$ , choose  $\beta = \frac{1}{10}$ . Thus, the current controller is

$$K_1(s) = \frac{1}{10} \quad (6.57)$$

For requirement 3, recalling the tracking error transfer function is the sensitivity function, i.e.

$$|S(j\omega)| = \left| \frac{1}{1 + L(s)} \right| \leq 0.04 \quad \rightarrow \quad |L(s)| \geq 1 + \frac{1}{0.04} = 26 \quad (6.58)$$

At  $\omega = 0.1$  rad/s, one has

$$|L(j0.1)| = |K(j0.1)G(j0.1)| = \left| \left( \frac{1}{10} \right) \left( \frac{10}{j0.1} \right) \right| = 10 \quad (< 26) \quad (6.59)$$

Thus, one needs to increase the low frequency loop gain. Using a low frequency boost control stage, i.e.

$$K_2(s) = \frac{s + \bar{\omega}}{s + \frac{\bar{\omega}}{\beta}} \quad (6.60)$$

if one sets  $\bar{\omega} = 0.3$  and  $\beta = 50$ , then

$$K_2(s) = \frac{s + 0.3}{s + 0.006} \quad (6.61)$$

the requirement is satisfied (using trial and error in MATLAB). Thus, the current controller is

$$K(s) = K_1(s)K_2(s) = \frac{s + 0.3}{10s + 0.06} \quad (6.62)$$

For requirement 4, recalling the noise filtering transfer function is the complementary sensitivity function, i.e.

$$|T(j\omega)| = \left| \frac{L(s)}{1 + L(s)} \right| \leq 0.04 \rightarrow |L(s)| \leq \frac{0.04}{0.04 + 1} = 0.0385 \quad (6.63)$$

At  $\omega = 10$  rad/s, one has

$$|L(j10)| = |K(j10)G(j10)| = \left| \left( \frac{j10 + 0.3}{j100 + 0.06} \right) \left( \frac{10}{j10} \right) \right| = 0.1 (> 0.0385) \quad (6.64)$$

Thus, one needs to decrease the high frequency loop gain. Using a rolloff control stage, i.e.

$$K_3(s) = \frac{\bar{\omega}}{s + \bar{\omega}} \quad (6.65)$$

If one sets  $\bar{\omega} = 2.75$ , then

$$K_3(s) = \frac{2.75}{s + 2.75} \quad (6.66)$$

and the requirement is satisfied (using trial and error in MATLAB). Thus, the current controller is

$$K(s) = K_1(s)K_2(s)K_3(s) = \frac{2.75s + 0.825}{10s^2 + 27.56s + 0.165} \quad (6.67)$$

For requirement 1, one can check that the zeros of  $1 + G(s)K(s)$  are in LHP and that there are no pole/zero cancellations. This can be verified using `zero(1+GK)` in MATLAB. Thus, the final controller is

$$\underline{K(s) = \frac{2.75s + 0.825}{10s^2 + 27.56s + 0.165}} \quad (6.68)$$

Note that where if this stability requirement was not satisfied, one would need to use a lead control stage. Note that this stability requirement can also be checked throughout the loop-shaping procedure.

## Chapter 7

# Airplane Guidance and Control Systems

### 7.1 Introduction to Guidance and Control Systems

For MIMO systems, such as flight vehicles, one may use state-space methods to develop suitable controllers, typically with optimal and/or robust control considerations. However, for many systems with only a few number of inputs and/or outputs, the different outputs that one desires to control may have responses on much different time scales which allow SISO methods to be applied at these different time scales through cascade control.

For example, consider the LTI systems represented by the following transfer functions:

$$G_{fast}(s) = \frac{y_{fast}(s)}{u(s)} \quad (7.1)$$

and

$$G_{slow}(s) = \frac{y_{slow}(s)}{u(s)} \quad (7.2)$$

Next, assuming that a change in  $y_{fast}$  naturally causes a change in  $y_{slow}$  with the relationship, i.e.

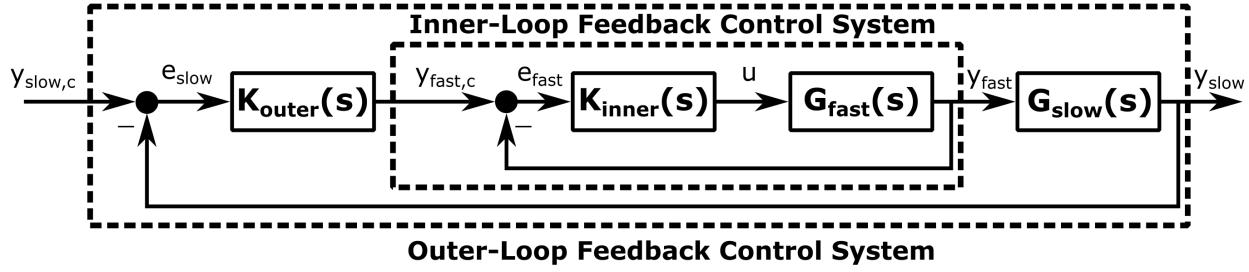
$$\frac{y_{slow}(s)}{y_{fast}(s)} = \frac{G_{slow}(s)}{G_{fast}(s)} \quad (7.3)$$

one has the fact that

$$\frac{y_{slow}(s)}{U(s)} = \frac{G_{slow}(s)}{G_{fast}(s)} G_{fast}(s) \quad (7.4)$$

Then, by assuming that  $y_{fast}$  and  $y_{slow}$  are related to one another through at least one integration, one knows that the denominator of  $\frac{G_{slow}(s)}{G_{fast}(s)}$  is at least order one higher in  $s$  than the numerator.

With this in mind, consider the following block diagram which uses two linked feedback control systems, i.e. an outer feedback control system and an inner feedback control system.



where this technique is known as **cascade-loop control**, **nested-loop control**, or **inner-outer loop control**. Although this control design may seem complicated, with proper loop-shaping, this type of feedback control system can be simpler to design than a MIMO system while also being robust with SISO LTI system stability margins.

For cascade-loop control, one designs  $K_{inner}(s)$  such that  $y_{fast}$  tracks  $y_{fast,des}$  for all frequencies up to  $\omega_{c,inner}$ . Then, note that this makes the inner-loop feedback control system have a transfer function from  $y_{fast,des}$  to  $y_{fast}$  as

$$\frac{y_{fast}(s)}{y_{fast,des}(s)} \approx \frac{\omega_{c,inner}}{s + \omega_{c,inner}} \quad (7.5)$$

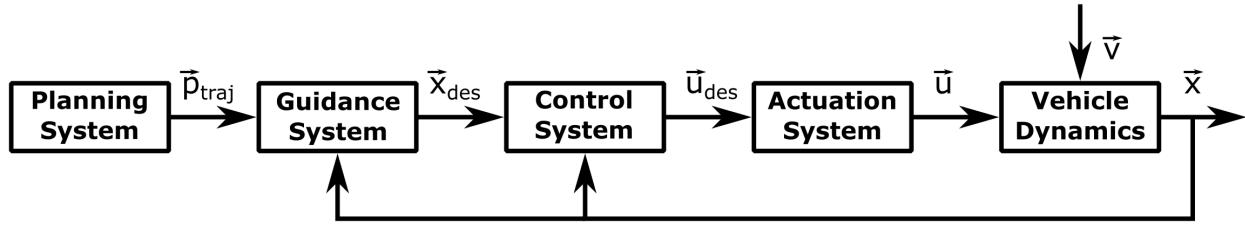
which for  $\omega < \omega_{c,inner}$ , this transfer function will be approximately unity below the inner-loop crossover frequency  $\omega_{c,inner}$ , i.e. the inner-loop bandwidth. This fact makes the preliminary design of the outer-loop much easier where assuming Equation 7.5, one can design  $K_{outer}(s)$  such that  $y_{slow}$  tracks  $y_{slow,des}$  for all frequencies up to  $\omega_{c,outer}$  which takes advantage of the simpler form of  $\frac{G_{slow}(s)}{G_{fast}(s)}$  as opposed to  $G_{slow}(s)U(s)$ . This two-step approach thus can be seen as much simpler than a one-step coupled design approach.

However, this cascade-loop control method requires that there is sufficient **crossover frequency separation** between the inner- and outer-loops, i.e.  $\omega_{c,outer} < \omega_{c,inner}$  by some amount such that the assumption of Equation 7.5 applies. For systems where there is already a natural frequency separation between  $G_{outer}(s)$  and  $G_{inner}(s)$ , this can be physically realizable, but must be conserved when performing the loop-shaping of the inner- and outer-loops. In particular, standard loop-shaping of  $L_{inner} = G_{inner}K_{inner}$  and  $L_{outer} = G_{outer}K_{outer}$  will set  $|L_{inner}|$  and  $|L_{outer}|$  large for  $\omega < \omega_{c,inner}$  and  $\omega < \omega_{c,outer}$ , respectively, and small for  $\omega > \omega_{c,inner}$  and  $\omega > \omega_{c,outer}$ , respectively. Thus, by designing  $\omega_{c,outer} < \omega_{c,inner}$  closing the outer-loop around the inner-loop will only slightly modify  $|L_{inner}|$  for  $\omega < \omega_{c,inner}$  where the magnitude was already large. In a similar manner, this type of logic can also be used extended to demonstrating that cascade-loop design will have little to no affect on the inner- and outer-loop bandwidths and stability margins using loop-shaping. It should also be noted that in this way, multiple inner-loops can be cascaded which is typical for vehicle feedback control systems.

## Planning, Guidance, and Control

Many vehicle feedback control systems use a cascade-loop control scheme which takes advantage of the physics-based relationships between position, velocity, and acceleration as well as angular versus linear states, both of which naturally have frequency separation required for cascade-loop control. As such the

different cascade control loops for vehicles have been given different names: planning, guidance, and control. From a feedback control perspective for flight vehicles, planning roughly corresponds to position feedback control, guidance to velocity and heading feedback control, and control to attitude feedback control directly through the . A block diagram for a planning, guidance, and control cascaded systems can be drawn as follows along with the actuation system,  $A(s)$ .



However, beyond this general framework, it should be noted that sometimes the guidance and control systems are combined into one feedback control system which is possible when state-space methods are used for feedback control system design to provide faster system responses in flight vehicle dynamics than is possible with a traditional cascade-loop control design. It should also be noted that when *designing* the guidance and control systems, one is typically using linearized vehicle dynamics models. Thus, vehicle feedback control systems often use some form of **adaptive control** which alters the guidance and control laws based on current operating conditions. A basic example of this is **gain scheduling** where the guidance and control “gains” change according to the “scheduled” steady flight conditions. However, when employing these types of guidance and control strategy, one typically requires that the transition from one operating condition to another is relatively “smooth,” i.e. the nonlinear dynamics are not significantly excited by the control inputs during the transitions between steady flight conditions. This is typically dealt with by designing sufficient stability margins to satisfy this requirement.

For feedback control design, one can typically model the **actuation system**,  $A(s)$ , using first or second order transfer functions which capture the primary frequency response characteristics of those systems, e.g. a first-order actuator

$$A(s) = \frac{\omega_a}{s + \omega_a} \quad (7.6)$$

or a second-order actuator

$$A(s) = \frac{\omega_a^2}{s^2 + 2\zeta\omega_a s + \omega_a^2} \quad (7.7)$$

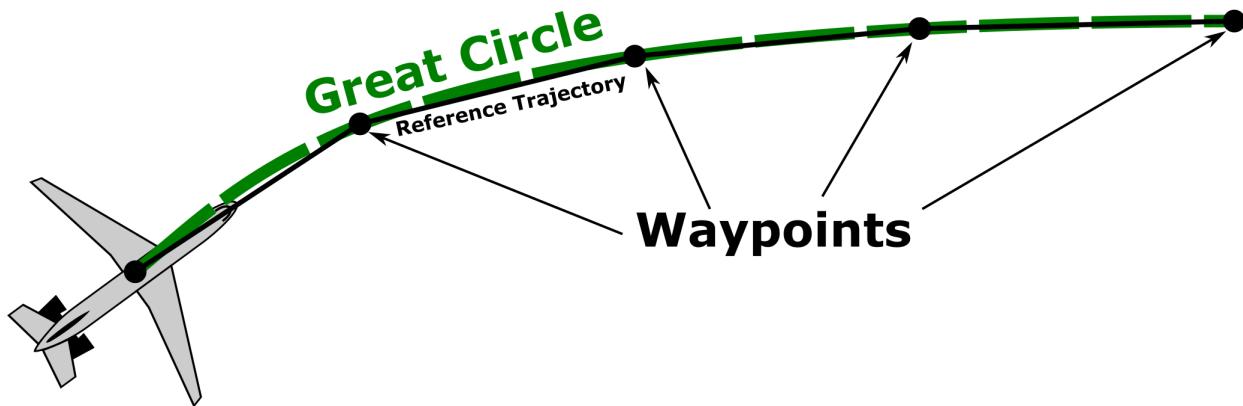
where  $\omega_a$  is the natural frequency of the actuator and  $\zeta_a$  is the damping ratio. These actuators are the primary limiting factors for the inner-loop control systems of flight vehicles. The control surfaces for airplanes are purely mechanical, hydro-mechanical, or electro-mechanical, i.e. fly-by-wire (FBW). Mechanical actuation uses a collection of mechanical parts, e.g. pushrods, tension cables, pulleys, counterweights, and chains, to transmit the forces applied to the cockpit controls directly to the control surfaces. These types of systems are used for small aircraft where the aerodynamic forces are not excessive. In this case for large aircraft, hydro-mechanical actuators are necessary to amplify the pilot applied force commands appropriately. For automatic control to occur, the actuation system for airplanes must use electro-mechanical devices which deflect the control surfaces based on the electric signals passed to the actuators by the flight computer. The

propulsion systems for airplanes control the thrust forces and the exact power control mechanisms depend on what type of engine is used, e.g. internal-combustion, jet, or electric.

### Waypoints and Guidance

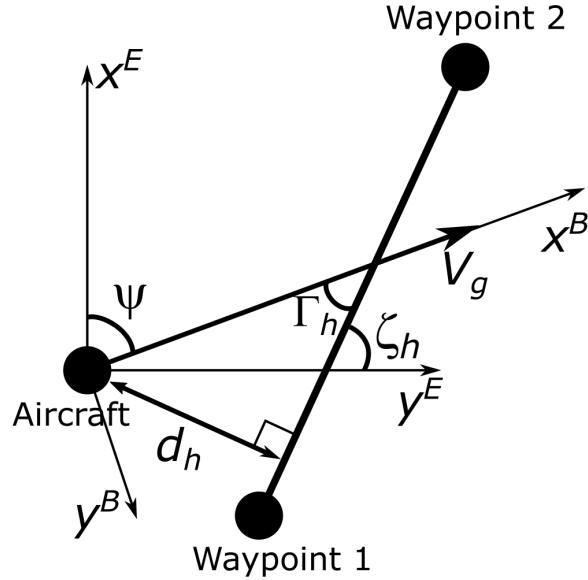
Though airplanes are inherently MIMO systems, due to the standard steady flight conditions for the majority of airplanes, the longitudinal and lateral-directional feedback control systems are typically designed separately and operated in parallel using the different control inputs for controlling the inner-loops of the airplane. In particular, the elevator and throttle are used for the longitudinal feedback control system and the rudder and ailerons are used for the lateral-directional feedback control systems. The inner-loops in both cases are first for attitude control, i.e. pitch angle (or angle of attack or pitch rate) and roll angle, respectively, which are the fundamental attitude angles to control as they directly affect the magnitude and direction of the airplane's lift vector, respectively.

Closed around the inner-loop attitude controllers are the outer-loops guidance for both longitudinal and lateral which follow the planned reference trajectory. In many cases, the reference trajectory may simply be holds on speed, altitude, or heading, or alternatively may be a sequence of **waypoints** connected by straight lines or constant turns. For jetliners these waypoints typically are plotted along the shortest distance line from one location to another, also known as a **great circle** line as it follows the curvature of the Earth's ellipsoid.



The guidance switching for following the lines connecting waypoints is performed by the planning system of the flight vehicle as it assesses the current position estimate of the flight vehicle and often uses some success and feasibility criteria checks. Then, most guidance laws for flight vehicles use a **line-following guidance law**.

For this part of the textbook, consider a straight line segment connecting two waypoints as depicted in the following figure



where  $u$  is the forward flight speed,  $\psi$  is the flight vehicle's yaw,  $d_h$  is the horizontal distance between the flight vehicle and the line segment,  $\zeta_h$  is the horizontal angle between the waypoint line segment and the reference  $x$ -axis, and  $\Gamma_h$  is the horizontal angle between the flight path of the flight vehicle relative to the waypoint line segment. Note that the flight vehicle is modeled as a point-mass in the following dynamics equations.

From the geometry, it is clear that

$$\dot{d}_h = -u \sin \Gamma_h \quad (7.8)$$

which for small angles can be rewritten as

$$\dot{d}_h = -u \Gamma_h \quad (7.9)$$

which is linear if one assumes  $u$  varies slowly compared with the flight vehicle's accelerations, i.e. assumed constant.

Next, by assuming that the waypoint line is at angle  $\zeta$

$$180 = \Gamma_h + [180 - (90 - \zeta_h)] + (90 - \psi) \quad (7.10)$$

$$\Gamma_h = \psi - \zeta_h \quad (7.11)$$

which exists only if  $u$  points through the line segment. This also means

$$\dot{\Gamma}_h = \dot{\psi} \quad (7.12)$$

Finally, by assuming  $\psi$  can be modeled sufficiently by a first order ODE, one can write

$$\dot{\psi}_y + \frac{1}{\tau_\psi} \psi = \frac{1}{\tau_\psi} \psi_{des} \quad (7.13)$$

where  $\tau_\psi$  is the approximate time constant of the yaw dynamics and can be approximated from the

$$\tau = \frac{1}{\omega_\psi} \quad (7.14)$$

where  $\omega_\psi$  is the loop bandwidth of the attitude control inner-loop.

From the previous equations and assuming  $\zeta_h = 0^\circ$ , one can write a linear continuous time state-space model of the form

$$\dot{\vec{x}} = A \vec{x} + B \vec{u} \quad (7.15)$$

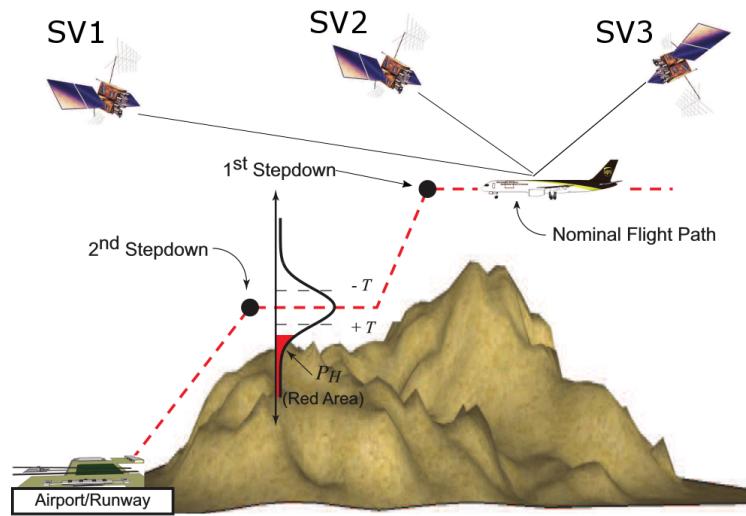
$$\begin{bmatrix} \dot{d}_h \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & -u \\ 0 & -\frac{1}{\tau_\psi} \end{bmatrix} \begin{bmatrix} d_h \\ \psi \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{\tau_\psi} \end{bmatrix} \psi_{des} \quad (7.16)$$

and for the flight vehicle to fly along the current waypoint,  $d_h$  should be controlled to zero.

First, it should be noted that these simplified guidance dynamics hold if the flight vehicle velocity is well defined by  $u$ , i.e. lateral velocity  $v \ll u$  or sideslip  $\beta$  is small. Second, if  $\zeta_h$  does not equal  $0^\circ$  then  $\psi_{des}$  in the above equation will need to be translated from the Earth reference frame to the waypoint reference frame and vice versa. Third, it should be noted that a similar method could also be used for a vertical guidance law using the pitch acceleration since the dynamics for flight vehicles are often decoupled.

## Guided Precision Approaches

In commercial aviation, guidance systems have been developed to assist an aircraft during the approach and landing phases of an aircraft's flight as these phases require the most accurate navigation information and precision guidance especially when visibility is impaired for pilots and an autopilot is engaged, also known as a **precision approach**. These systems are designed to supply this information using radio signals that are interpreted by specialized equipment onboard the aircraft which supply a heading correction that the pilot or autopilot must make to continue on the prescribed descent trajectory to that airport's runway. Thus, these systems are maintained by individual airports. Thus, the reference trajectories generated by these "guidance" systems are constant, predefined trajectories based on the topography of the area, the layout of the airport and runways, and regulations concerning safety for aircraft descent, and are not optimally solved based on techniques that we've discussed in this class.



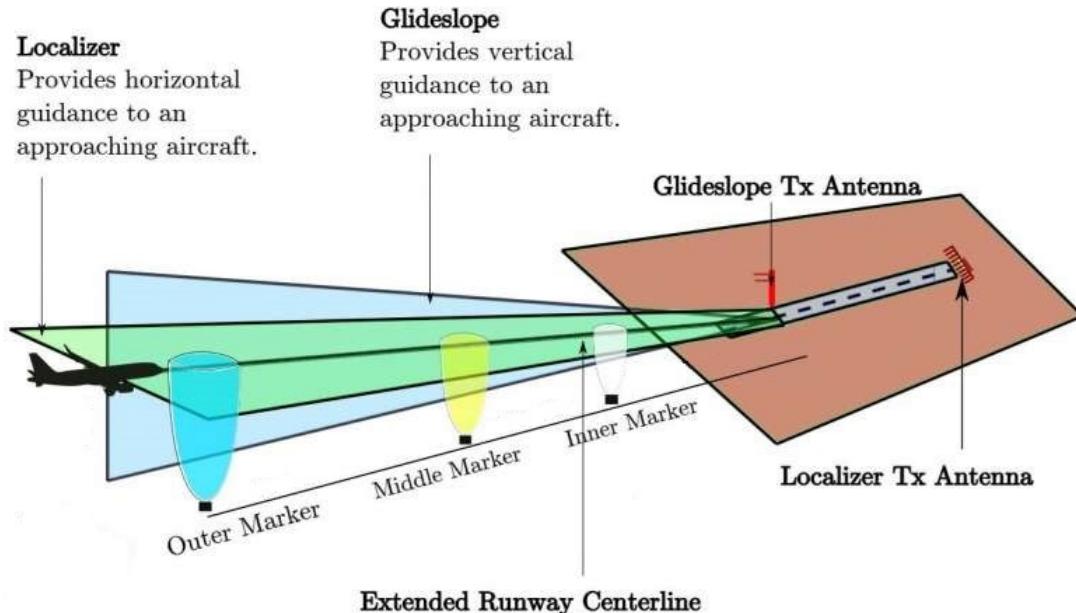
An important parameter in precision approaches is the **decision height (DH)** or **decision altitude (DA)**. Both of which are defined as the specified lowest height in the approach descent at which, if the required visual reference to continue the approach, e.g. the runway markings or runway environment, is not visible to the pilot, the pilot must initiate a missed approach maneuver and reroute to try the approach again. The visual reference is typically the runway markings or runway environment. A decision height is measured *above ground level (AGL)* while a decision altitude is measured *above mean sea level (MSL)*. The specific values for DH at a given airport are established with intention to allow a pilot sufficient time to safely re-configure an aircraft to climb and execute the missed approach procedures while avoiding terrain and obstacles. A DH denotes the height in which a missed approach procedure must be started, it does not preclude the aircraft from descending below the prescribed DH.

For autonomous guided approaches, there are four categories which allow these decision heights to be lowered.

- For Category I, the decision height is not lower than 200 feet and the runway visual range is not less than 550 meters.
- For Category II, the decision height is below 200 ft but not lower than 100 feet and the runway visual range is not less than 300 meters.
- For Category III A, the decision height is lower than 100 feet, and the runway visual range is not less than 200 meters.
- For Category III B, the decision height is lower than 50 ft, or no decision height, and the runway visual range is lower than 200 meters but not less than 75 meters.

To allow these guided precision approaches, the following technologies serve commercial aircraft systems as both navigational and guidance aids.

One of the earliest technologies for automatic guidance and is still in use today is the **Instrument Landing System (ILS)** which used radionavigation to provide aircraft with horizontal and vertical guidance information during approach and landing. At certain fixed points, it also provides the distance to the reference point of landing. An ILS uses two signals: a **localizer** and a **glideslope** which indicate the correction that the pilot or autopilot must make. As part of this system, there are also typically **marker beacons** which provide distance to the runway information at setpoints along the approach. These setpoints are published on the airport documentation for the approach.



This is displayed to pilots as a point plotted on 2 coordinate axes whose origin is “0” error. If the aircraft is to the left of the reference trajectory, the point is to the right of the origin. If the aircraft is above the set glide path, then the point is below the origin. The glide path for an ILS is typically defined as  $3^{\circ}$  above the horizontal.



Occasionally a modified ILS called a **Instrument Guidance System (IGS)**, also known as a **Localizer-type Directional Aid (LDA)** in the United States, must be used for non-straight approaches into airports with

certain topographical features which prevent normal operation. One of the most famous of these was runway 13 at Kai Tak International Airport in Hong Kong due to the mountains, apartments, and ocean surrounding the airport. **Distance measuring equipment (DME)** is also a common system available at airports that provides pilots or autopilots with a distance to runway measurement in nautical miles which can be deployed with or without an ILS and are rapidly replacing marker beacons.

Due to the rapid development of Global Navigation Satellite Systems (GNSS), in particular the Global Positioning System (GPS) run by the United States Department of Defense, the most recent advances in automatic guidance technologies for aircraft include the following technologies which enhance the basic capabilities of GNSS by reducing the errors found in the GNSS signals thereby improving the accuracy of the navigation solution. The need and details for these technologies will be discussed in later parts of this textbook. Localizer Performance with Vertical guidance (LPV) system is based on the Satellite-Based Augmentation System (SBAS) to GNSS measurements, which for GPS is also known as the Wide Area Augmentation System (WAAS). This provides greater accuracy than GPS alone which is necessary for guidance during aircraft approaches, but it is not as accurate as the next system. The Ground-Based Augmentation System (GBAS), which for GPS is also known as Local Area Augmentation System (LAAS), also augments the GNSS measurements in order to provide enhanced levels of service to support automatic guidance information during all phases of approach, landing, departure, and surface operations within radio distance. GBAS is anticipated to be able to achieve Category III B guided precision approaches and is expected to replace ILS in the future.

## 7.2 Airplane Longitudinal Guidance and Control

This section will discuss the different guidance and control loops for the longitudinal portion of an airplane's feedback control system using cascade-loop control which uses various transfer functions from the longitudinal state-space model. Recalling that the longitudinal state equation is given by

$$\begin{bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = A_{long} \begin{bmatrix} u \\ \alpha \\ q \\ \theta \end{bmatrix} + B_{long} \begin{bmatrix} \delta_e \\ \delta_t \end{bmatrix} \quad (7.17)$$

and using the Laplace transform, one has

$$\begin{bmatrix} sU(s) \\ s\alpha(s) \\ sQ(s) \\ s\Theta(s) \end{bmatrix} = A_{long} \begin{bmatrix} U(s) \\ \alpha(s) \\ Q(s) \\ \Theta(s) \end{bmatrix} + B_{long} \begin{bmatrix} \Delta_e(s) \\ \Delta_t(s) \end{bmatrix} \quad (7.18)$$

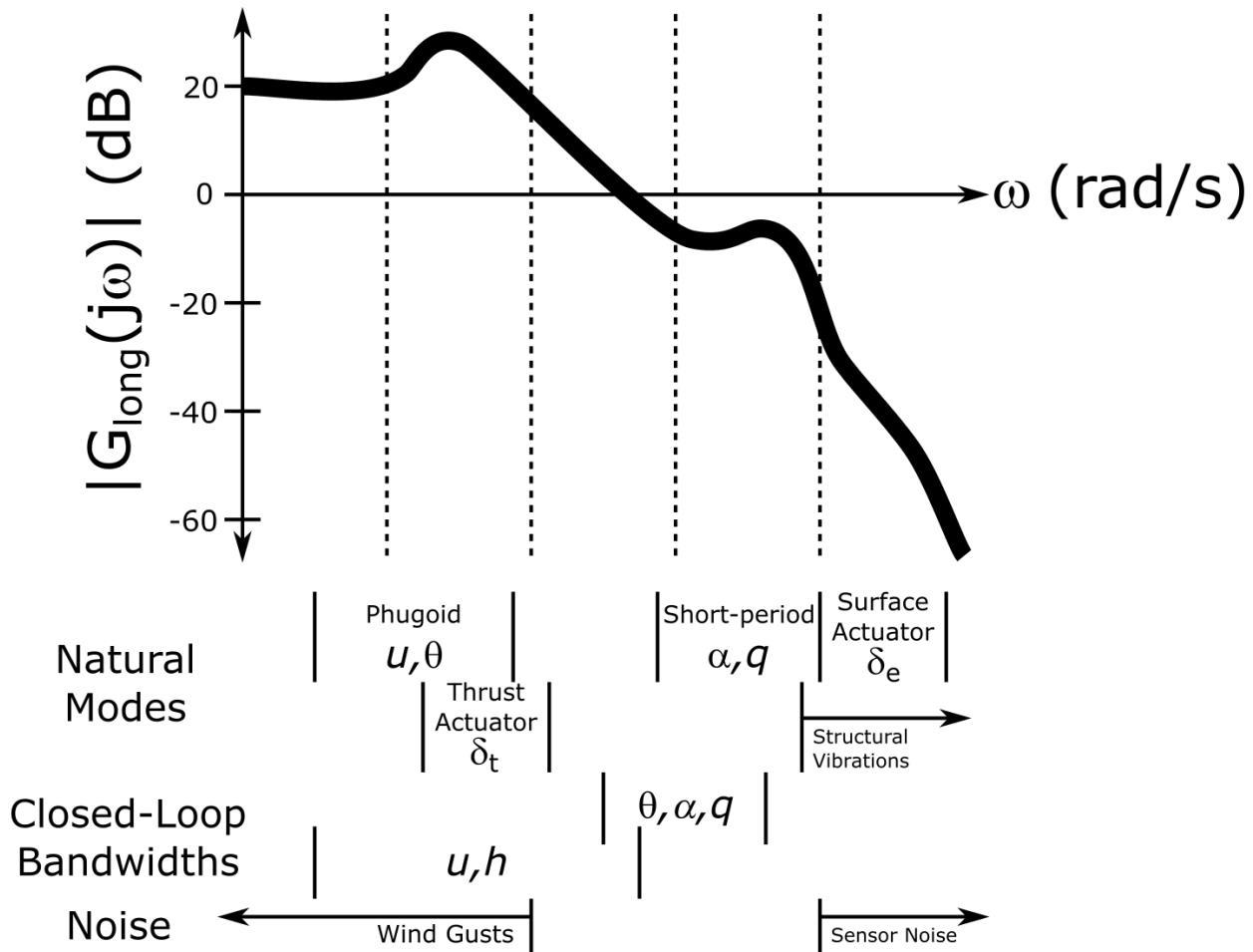
or, rearranging, one has

$$(sI_{4 \times 4} - A_{long}) \begin{bmatrix} U(s) \\ \alpha(s) \\ Q(s) \\ \Theta(s) \end{bmatrix} = B_{long} \begin{bmatrix} \Delta_e(s) \\ \Delta_t(s) \end{bmatrix} \quad (7.19)$$

and solving for the outputs, one has

$$\begin{bmatrix} U(s) \\ \alpha(s) \\ Q(s) \\ \Theta(s) \end{bmatrix} = (sI_{4 \times 4} - A_{long})^{-1} B_{long} \begin{bmatrix} \Delta_e(s) \\ \Delta_t(s) \end{bmatrix} \quad (7.20)$$

Thus, the transfer functions for each input-output pair can be computed using the particular element of  $(sI_{4 \times 4} - A_{long})^{-1} B_{long}$  where the denominators of each reflect the characteristic equation for the underdamped short-period and long-period/phugoid modes. The general frequency response for these modes, the actuation modes, and the modes of the feedback control systems, as well as the frequency bands of the noises are shown generally below.



It should be noted that the numerators of transfer functions to the states  $U(s)$ ,  $\alpha(s)$ , and  $Q(s)$  are third order polynomials in  $s$  while  $\Theta(s)$  is a second order polynomial in  $s$  (recall  $Q(s) = s\Theta(s)$ ). Furthermore,

though the linearized and decoupled transfer functions are used for airplane feedback control system design, it should be mentioned that the control inputs will generally affect all airplane states and should be simulated with the nonlinear airplane model once the guidance and control system design has been completed.

In addition, one may also use the following transfer function relationships in longitudinal feedback control design. For the flight path angle, one has the small angle approximation  $\gamma = \theta - \alpha$  or

$$\Gamma(s) = \Theta(s) - \alpha(s) \quad (7.21)$$

and for altitude  $h$ , recall (without the  $\Delta$ 's), one has

$$h(t) = \int_0^t \bar{u} \sin(\gamma(t)) \quad (7.22)$$

in the time domain which can be written in the Laplace domain as

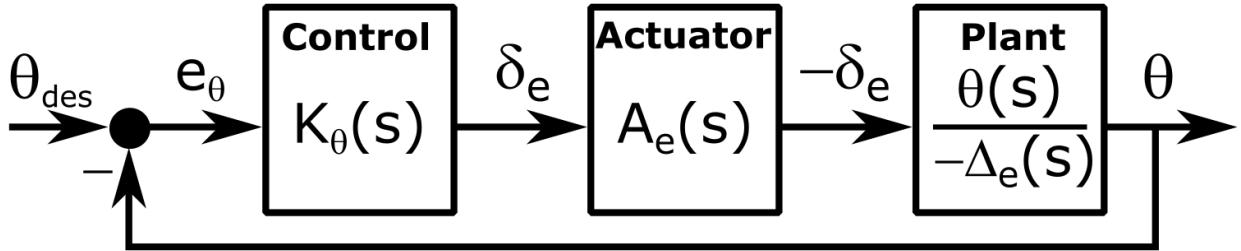
$$H(s) = \frac{1}{s} \bar{u} \sin(\Gamma(s)) \quad (7.23)$$

or assuming a small angle for  $\gamma$

$$H(s) = \frac{\bar{u}}{s} \Gamma(s) \quad (7.24)$$

### Pitch Attitude Control

The longitudinal inner-loop control system for an airplane typically uses the pitch control law,  $K_\theta(s)$ , the transfer function for the elevator actuator response,  $A_e(s)$ , and the transfer function from  $-\delta_e \rightarrow \theta$  for the plant as shown in the following block diagram.



where the plant to be controlled is obtained by

$$G_\theta(s) = A_e(s) \frac{\Theta(s)}{-\Delta_e(s)} \quad (7.25)$$

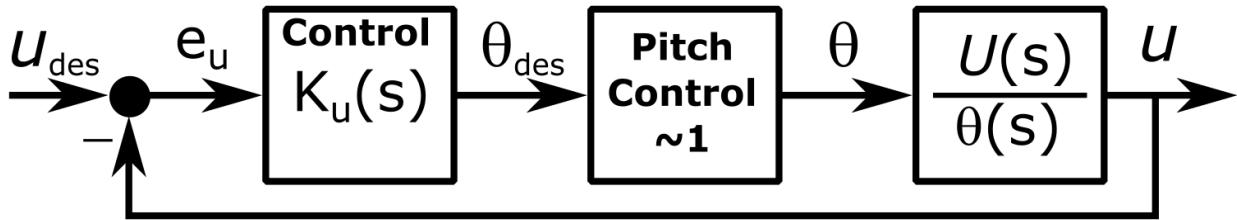
Note that the negative sign has been introduced so that a positive  $\delta_e$  produces a positive  $\theta$ . It should also be noted that other alternatives for the inner-loop include the pitch rate  $q$  and the angle of attack  $\alpha$  as the single output for the system to track with the inner-loop.

When designing the **pitch attitude control**, an important aspect to assess is whether a commanded step increase in pitch  $\theta$  leads to an increase or decrease in the steady-state flight path angle  $\gamma$ . If the steady-state

$\gamma$  is positive, then the airplane is said to be “on the front side of the power curve,” i.e. a reduction in flight velocity requires less power to maintain level flight. Thus, the airplane would continue to climb. Conversely, if the steady-state  $\gamma$  is negative, then the airplane is “on the back side of the power curve,” i.e. a reduction in flight velocity requires more power to maintain level flight. Thus, the airplane could not climb.

### Speed (Mach) Hold Guidance

One option for the longitudinal outer-loop guidance system for an airplane is a **speed hold** which is typically employed during climbing flight under air traffic control. This guidance loop uses the speed control law,  $K_u(s)$ , the pitch inner-loop control system, and the transfer function from  $\theta \rightarrow u$  for the outer-loop plant as shown in the following block diagram.



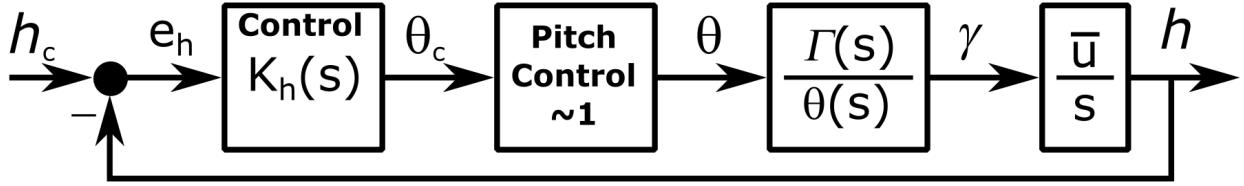
As the inner-loop of this speed hold control loop uses the elevator, one can form the outer-loop plant as

$$\frac{U(s)}{\Theta(s)} = \frac{\text{num}\left(\frac{U(s)}{\Delta_e(s)}\right)}{\text{num}\left(\frac{\Theta(s)}{\Delta_e(s)}\right)} \quad (7.26)$$

where  $\text{num}(G(s))$  represents the numerator of the transfer function  $G(s)$ . To form this model, one may use the transfer functions  $\frac{U(s)}{\Delta_e(s)}$  and  $\frac{\Theta(s)}{\Delta_e(s)}$  for the vehicle alone as long as the crossover frequency separation for the inner- and outer-loops is maintained. Typically, one also includes the pitch inner-loop control system once it has been designed. It should be noted that  $K_u(s)$  or  $\frac{U(s)}{\Theta(s)}$  may need to be negative as a positive change in pitch  $\theta$  may cause a reduction in flight velocity  $u$ .

### Altitude Hold Guidance

Another option for the longitudinal outer-loop guidance system for an airplane is an **altitude hold** which is typically employed during a cruise flight condition at a specific cruise velocity specified by the throttle setting. This guidance loop uses the altitude control law,  $K_h(s)$ , the pitch inner-loop control system, and the transfer function from  $\theta \rightarrow \gamma$  for the outer-loop plant as shown in the following block diagram.



It should be noted that the transfer function from  $\theta \rightarrow \gamma$  can be calculated using the small angle approximation for the Euler angles as

$$\Gamma(s) = \Theta(s) - \alpha(s) \quad (7.27)$$

which provides

$$\frac{\Gamma(s)}{\Theta(s)} = 1 - \frac{\alpha(s)}{\Theta(s)} \quad (7.28)$$

or

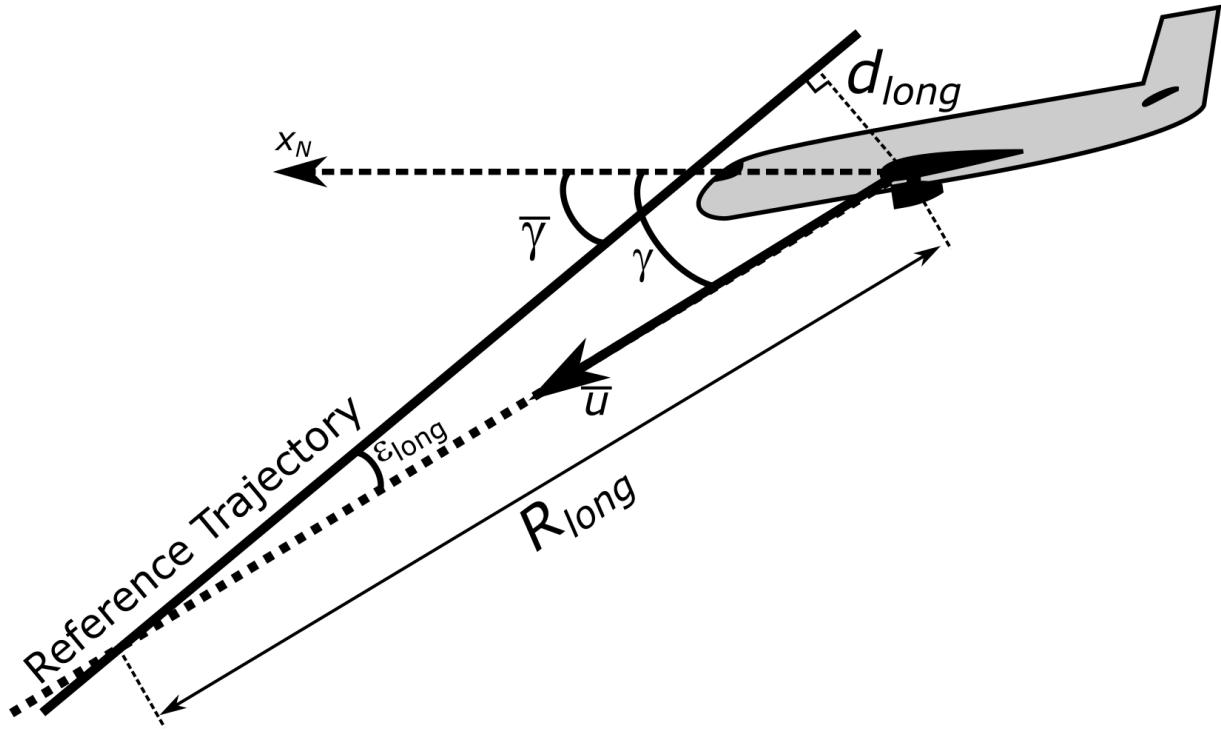
$$\frac{\Gamma(s)}{\Theta(s)} = 1 - \frac{\text{num}\left(\frac{\alpha(s)}{\Delta_e(s)}\right)}{\text{num}\left(\frac{\Theta(s)}{\Delta_e(s)}\right)} \quad (7.29)$$

To form this model, one may again use the transfer functions  $\frac{\alpha(s)}{\Delta_e(s)}$  and  $\frac{\Theta(s)}{\Delta_e(s)}$  for the vehicle alone as long as the crossover frequency separation for the inner- and outer-loops is maintained. Typically, one also includes the pitch inner-loop control system once it has been designed.

It should also be noted that here it is required that an increase in pitch angle  $\theta$  must produce a steady-state flight path angle  $\gamma$ , i.e. the airplane must be “on the front side of the power curve.”

### Longitudinal Line-Following Guidance

In general, suppose an airplane is to follow a straight line for its reference path or trajectory in the longitudinal plane as shown below



where  $d_{long}$  is the **longitudinal position deviation**,  $\epsilon_{long}$  is the longitudinal angular deviation,  $R_{long}$  is the longitudinal range to intercept,  $\bar{\gamma}$  is the reference flight path angle.

Given this figure, the longitudinal deviations can be related by

$$\sin \epsilon_{long} = \frac{d_{long}}{R_{long}} \quad (7.30)$$

which for small angular deviations is

$$\epsilon_{long} = \frac{d_{long}}{R_{long}} \quad (7.31)$$

Furthermore, as the longitudinal angular deviation is related to the flight path angle by

$$\epsilon_{long} = \bar{\gamma} - \gamma = -\Delta\gamma \quad (7.32)$$

Then, by rearrangement and substitution, one has

$$d_{long} = -R_{long}\Delta\gamma \quad (7.33)$$

then taking the derivative, one has

$$\dot{d}_{long} = -\dot{R}_{long}\Delta\gamma \quad (7.34)$$

and the instantaneous range rate is

$$\dot{R}_{long} = -\bar{u} \quad (7.35)$$

one has

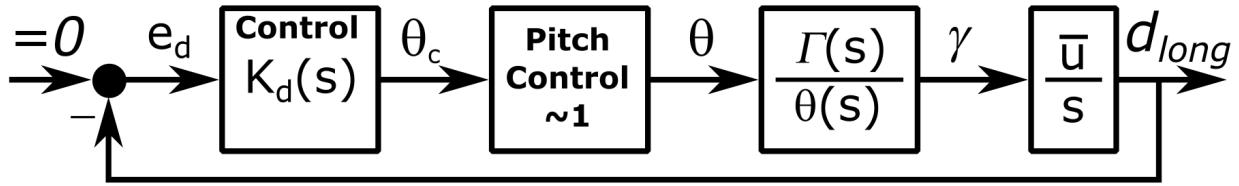
$$\dot{d}_{long} \approx \bar{u}\Delta\gamma \quad (7.36)$$

In the Laplace domain (dropping the  $\Delta$ )

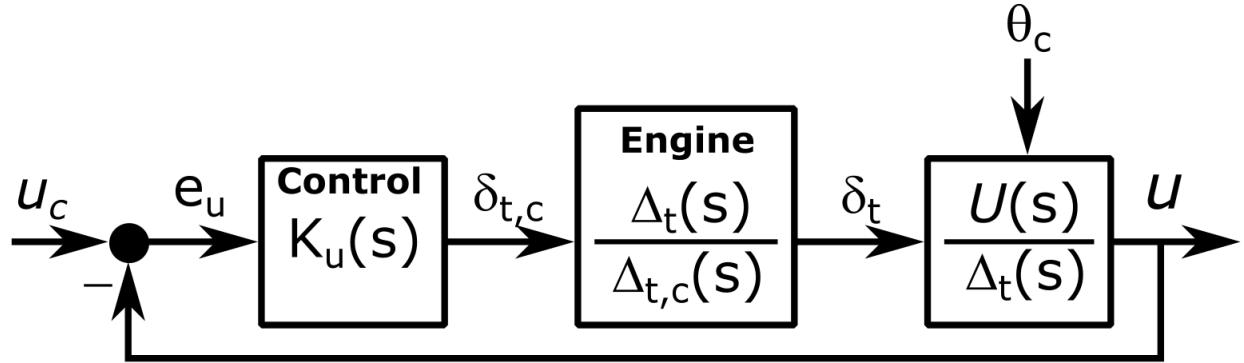
$$\frac{D_{long}(s)}{\Gamma(s)} \approx \frac{\bar{u}}{s} \quad (7.37)$$

where it should be noted that this also assumes that the body-fixed reference frame is the stability frame, i.e.  $\bar{\alpha} = 0$ , thus  $\bar{\gamma} = \bar{\theta}$ .

Thus, one can form an approximate longitudinal line-following guidance loop as



which is very similar to the altitude hold controller except that when the aircraft desires to change its flight path angle using the pitch controller, one may require a positive commanded  $\theta$  would produce a positive change in flight path angle  $\gamma$  which is not always the case. Thus, for general longitudinal guidance, an additional auto-throttle inner-loop controller is necessary to change the thrust input to adjust the velocity of the airplane to maintain the velocity required. An example of the auto-throttle inner-loop control is shown in the following figure.



In this case, the transfer function  $\frac{\Gamma(s)}{\Theta(s)}$  is not the same as the one for the altitude hold guidance due to the added inclusion of the auto-throttle inner-loop control in addition to the pitch inner-loop control. Also, as the longitudinal guidance loop will be controlled by the elevator through the pitch inner-loop control,  $u_c = \bar{u}$  for the auto-throttle.

### 7.3 Airplane Lateral-Directional Guidance and Control

This section will discuss the different guidance and control loops for the lateral-directional portion of an airplane's feedback control system using cascade-loop control which uses various transfer functions from the lateral-directional state-space model. Recalling that the lateral-directional state equation is given by

$$\begin{bmatrix} \dot{\beta} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{bmatrix} = A_{lat} \begin{bmatrix} \beta \\ p \\ r \\ \phi \end{bmatrix} + B_{lat} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad (7.38)$$

and using the Laplace transform, one has

$$\begin{bmatrix} s\beta(s) \\ sP(s) \\ sR(s) \\ s\Phi(s) \end{bmatrix} = A_{lat} \begin{bmatrix} \beta(s) \\ P(s) \\ R(s) \\ \Phi(s) \end{bmatrix} + B_{lat} \begin{bmatrix} \Delta_a(s) \\ \Delta_r(s) \end{bmatrix} \quad (7.39)$$

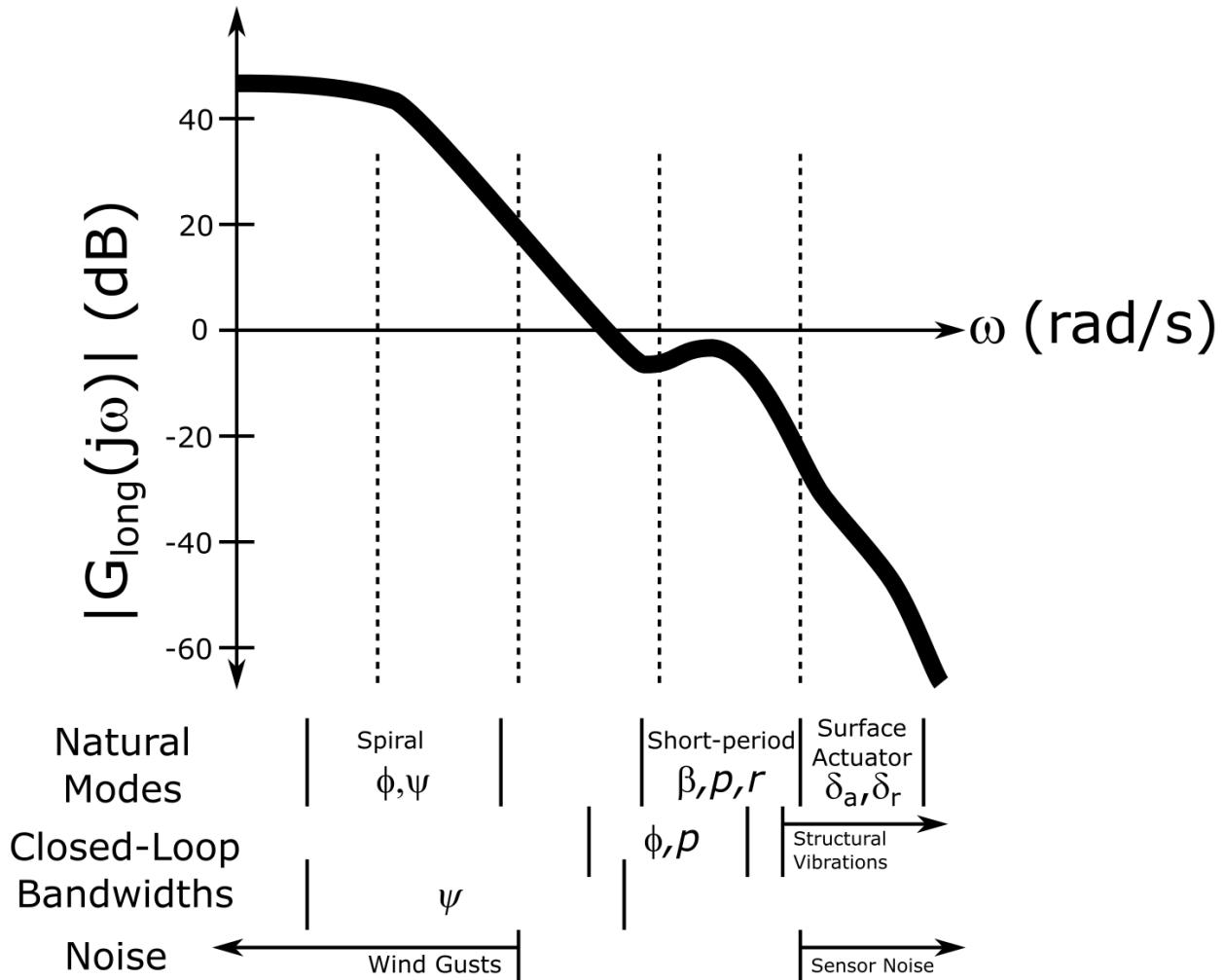
or, rearranging, one has

$$(sI_{4 \times 4} - A_{lat}) \begin{bmatrix} \beta(s) \\ P(s) \\ R(s) \\ \Phi(s) \end{bmatrix} = B_{lat} \begin{bmatrix} \Delta_a(s) \\ \Delta_r(s) \end{bmatrix} \quad (7.40)$$

and solving for the outputs, one has

$$\begin{bmatrix} \beta(s) \\ P(s) \\ R(s) \\ \Phi(s) \end{bmatrix} = (sI_{4 \times 4} - A_{lat})^{-1} B_{lat} \begin{bmatrix} \Delta_a(s) \\ \Delta_r(s) \end{bmatrix} \quad (7.41)$$

Thus, the transfer functions for each input-output pair can be computed using the particular element of  $(sI_{4 \times 4} - A_{lat})^{-1} B_{lat}$  where the denominators of each reflect the characteristic equation for the first order spiral and roll modes and underdamped dutch roll mode. The general frequency response for these modes, the actuation modes, and the modes of the feedback control systems, as well as the frequency bands of the noises are shown generally below.



It should be noted that the numerators of transfer functions to the states  $\beta(s)$ ,  $P(s)$ , and  $R(s)$  are third order polynomials in  $s$  while  $\Phi(s)$  is a second order polynomial in  $s$  (recall  $\Phi(s) = \frac{1}{s}P(s)$ ). In addition, one typically also uses the following transfer function relationship in lateral-directional feedback control design. For the heading angle, one has the small angle approximation

$$\Psi(s) = \frac{1}{s}R(s) \quad (7.42)$$

### Roll Control

As opposed to the longitudinal control inputs, the lateral-directional attitude control is performed by two control surface deflections which have strong coupling effects on the dynamics. For example, aileron deflections often cause an undesirable excitation of the dutch roll mode and/or an adverse yawing moment,  $N_{\delta_a}$ . Thus, many aircraft typically use some sort of **aileron-rudder interconnect (ARI)** to reduce these

effects. A simple way to select  $K_{ARI}$  is to cancel the adverse yaw with the rudder, i.e. forcing the effective  $N_{\delta_a} \approx 0$  by setting the yawing moment due to rudder deflection equal to the negative of the yawing moment due to aileron deflection or in mathematical terms

$$N_{\delta_r} \delta_r = -N_{\delta_a} \delta_a \quad (7.43)$$

or letting

$$\delta_r = -\frac{N_{\delta_a}}{N_{\delta_r}} \delta_a \quad (7.44)$$

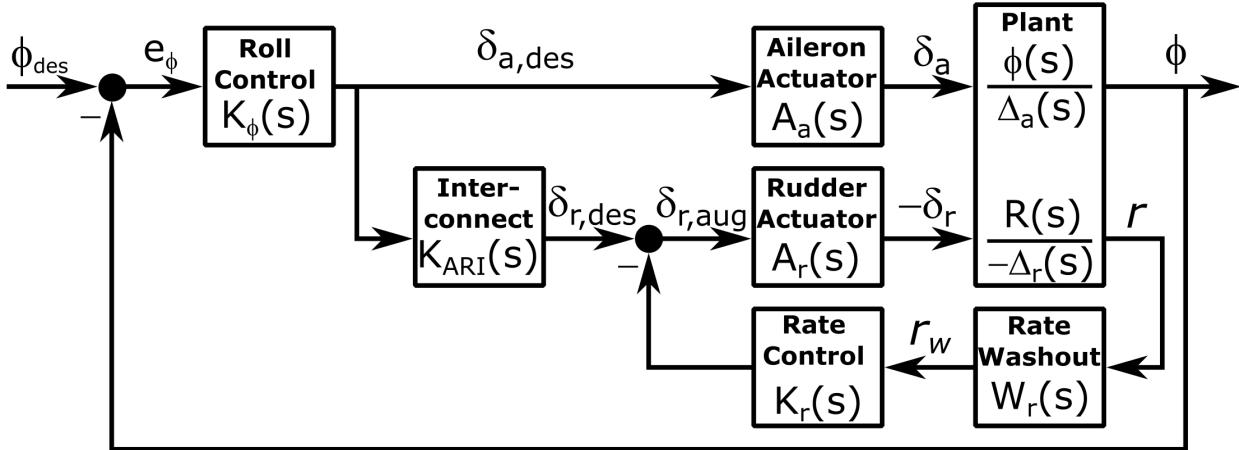
one has

$$K_{ARI} = -\frac{N_{\delta_r}}{N_{\delta_a}} \quad (7.45)$$

Thus, the lateral inner-loop control system for an airplane typically uses the roll control law,  $K_\phi(s)$ , the yaw rate control law,  $K_r(s)$ , a yaw rate washout filter,

$$WO(s) = \frac{\tau_w s}{\tau_w s + 1} \quad (7.46)$$

the ARI controller,  $K_{ARI}(s)$ , the transfer functions for the aileron and rudder actuator responses,  $A_a(s)$  and  $A_r(s)$ , and the transfer functions from  $\delta_a \rightarrow \phi$  and  $-\delta_r \rightarrow r$  for the airplane dynamics as shown in the following block diagram.



where the plant to be controlled is obtained by two separate transfer functions

$$G_r(s) = A_r(s) \frac{R(s)}{-\Delta_r(s)} \quad (7.47)$$

and

$$G_\phi(s) = A_a(s) \frac{\Phi(s)}{-\Delta_a(s)} \quad (7.48)$$

It should be noted that the washout filter must be included for the case of sustained turns where the yaw rate is not zero. Without  $W(s)$ , the yaw rate feedback (i.e. damper) would tend to “fight” the turn by maintaining a rudder deflection opposite to that desired for the turn. The effect of this filter “washes out” the yaw rate feedback control for low frequency signals, i.e. the yaw rate feedback control only affects  $\omega \geq \frac{1}{\tau_w}$  where  $\frac{1}{\tau_w}$  is typically selected well below the dutch roll natural frequency. Considering these additions to a roll angle feedback control system, one typically must form the plant model  $G_\phi(s)$  using the rudder augmentation loops before loop-shaping  $L_\phi(s) = G_\phi(s)K_\phi(s)$ .

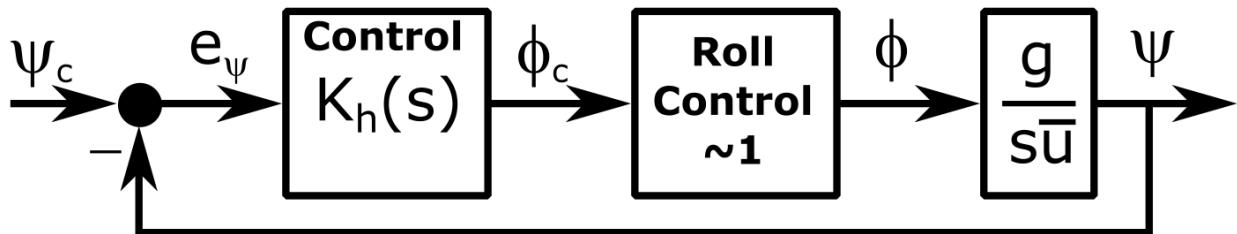
Lastly, it should be mentioned that though using an ARI is one method for controlling a coordinated turn, one may also consider feeding back to the rudder, either  $\beta$ , the lateral acceleration of the center of rotation, or the total computed yaw rate (not just the perturbed). In addition, when performing a coordinated turn one must also generate the suitable longitudinal control inputs to maintain altitude, a concept called **turn compensation**. Primarily, by rotating the lift vector of the airplane, the vertical component of lift must still counteract the weight for constant altitude. Thus, as an airplane enters the turn, the angle of attack must be increased which requires an appropriate elevator deflection given the flight velocity and bank angle. Thus, one can use a computed pitch rate to generate the desired pitch rate for a rate-command or attitude-hold feedback control system, i.e.

$$q_c = r \tan \phi \quad (7.49)$$

where these values are for the entire airplane, not the perturbation states. Alternatively, one may employ an altitude hold feedback control system in the longitudinal plane which was discussed in the previous lecture.

## Heading Hold

One option for the lateral-directional outer-loop guidance system for an airplane is a **heading hold** which is typically employed at a specific cruise velocity. This guidance loop uses the heading control law,  $K_\psi(s)$ , the roll inner-loop control system, and the transfer function from  $\phi \rightarrow \psi$  for the outer-loop plant as shown in the following block diagram.



It should be noted that this heading hold assumes that the heading and yaw are the same, i.e.  $\bar{\beta} = 0^\circ$  so  $\sigma = \psi$ , and that  $\mu \approx \phi$  which is true for the linearized lateral-directional dynamics. Furthermore, the transfer function from  $\phi \rightarrow \psi$  can be calculated using the steady flight relationship for the bank angle and the rate of change of heading/yaw, i.e.

$$\dot{\sigma} = \frac{g}{|\vec{v}|} \tan \mu \quad (7.50)$$

which, for the small angle approximation  $\tan \mu \approx \mu$  and using a stability frame for linearization, i.e.  $|\vec{v}| = \bar{u}$ , one has

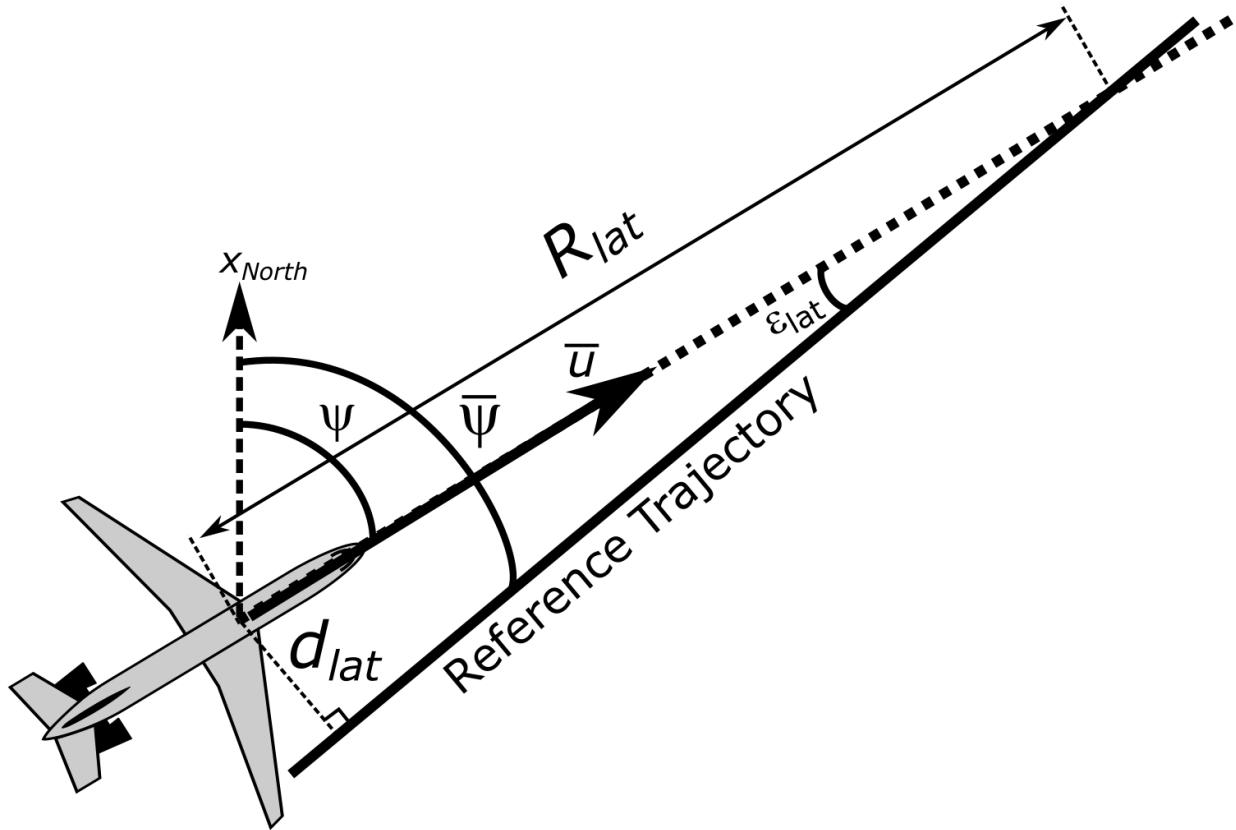
$$\Psi(s) = \frac{g}{s\bar{u}}\Phi(s) \quad (7.51)$$

after angle substitutions.

It should be noted that often the roll angle must be hard limited for particular airplane missions. Thus, often one includes a **limiter** for the desired roll which limits the max and min values to some specified values. Such an inclusion in the feedback control system will introduce a nonlinearity in the system, but generally reduces the speed of response for fast maneuvers. Performances of such systems is typically assessed through the nonlinear simulations of the feedback control systems initially designed with linear models.

## Lateral-Directional Path Guidance

In general, suppose an airplane is to follow a straight line for its reference path or trajectory in the lateral-directional plane as shown below



where  $d_{lat}$  is the **lateral-directional position deviation**,  $\epsilon_{lat}$  is the lateral-directional angular deviation,  $R_{lat}$  is the lateral-directional range to intercept,  $\bar{\psi}$  is the reference heading angle for the trajectory.

Given this figure, the lateral-directional deviations can be related by

$$\sin \epsilon_{lat} = \frac{d_{lat}}{R_{lat}} \quad (7.52)$$

which for small angular deviations is

$$\epsilon_{lat} = \frac{d_{lat}}{R_{lat}} \quad (7.53)$$

Furthermore, as the lateral angular deviation is related to the heading angle by

$$\epsilon_{lat} = \psi - \bar{\psi} \quad (7.54)$$

Then, by rearrangement and substitution, one has

$$d_{lat} = -R_{lat} (\psi - \bar{\psi}) \quad (7.55)$$

then taking the derivative, one has

$$\dot{d}_{lat} = -\dot{R}_{lat} (\psi - \bar{\psi}) \quad (7.56)$$

and the instantaneous range rate is

$$\dot{R}_{lat} = -\bar{u} \quad (7.57)$$

one has

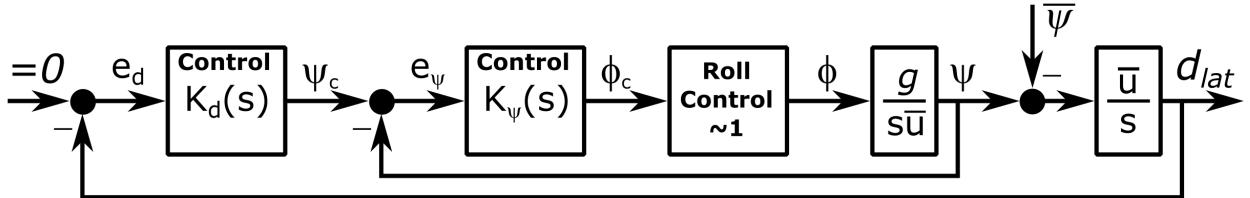
$$\dot{d}_{lat} \approx -\bar{u} (\psi - \bar{\psi}) \quad (7.58)$$

or in the Laplace domain

$$\frac{D_{lat}(s)}{\Psi(s) - \bar{\Psi}(s)} \approx \frac{\bar{u}}{s} \quad (7.59)$$

where it should be noted that this also assumes that the body-fixed reference frame is the stability frame, i.e.  $\vec{v} = u$  and that coordinated flight is occurring, i.e.  $\bar{\beta} = 0^\circ$ .

Thus, one can form an approximate lateral-directional line-following guidance loop as



which essentially uses a heading hold feedback loop with an additional outer-loop guidance for reducing the deviation from a reference heading to zero.

## **Part II**

# **Optimal Control and Estimation**

# Chapter 8

## Advanced Dynamical Systems Theory

### 8.1 Advanced Dynamical Systems Theory

A **dynamical system** can be defined as a collection of a finite or infinite number of interconnected and time-dependent quantities that evolve according to a fixed mathematical rule, also referred to as the **dynamics equation**. The system's changes in time are driven by the external environment in which the system operates. When subjected to an external time-varying dependent system input, the system generates a system output which may depend on the system's internal properties, i.e. the system state. In general, each of these signals may scalars, vectors, or of infinite dimension.

A special class of dynamical systems are those that have a dynamics equation of a finite number of coupled scalar **ordinary differential equations (ODEs)** of the general form

$$\dot{\vec{x}}(t) = f(t, \vec{x}(t), \vec{u}(t)) \quad (8.1)$$

or as vector-valued **difference equations**, also known as **recursive relations**, of the general form

$$\vec{x}[k+1] = f(k, \vec{x}[k], \vec{u}[k]) \quad (8.2)$$

where  $f : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  is a vector function,  $t \in \mathbb{R}$  is **time**,  $k \in \mathbb{R}$  is the **time step**,  $\vec{u} \in \mathbb{R}^P$  is the externally supplied **input**, and  $\vec{x} \in \mathbb{R}^n$  is the **system state**. As such, this equation is also known as the **system state equation**. These two different representations are referred to as **continuous-time** or **discrete-time** dynamics. Here,  $n$  is known as the **order of the system**.

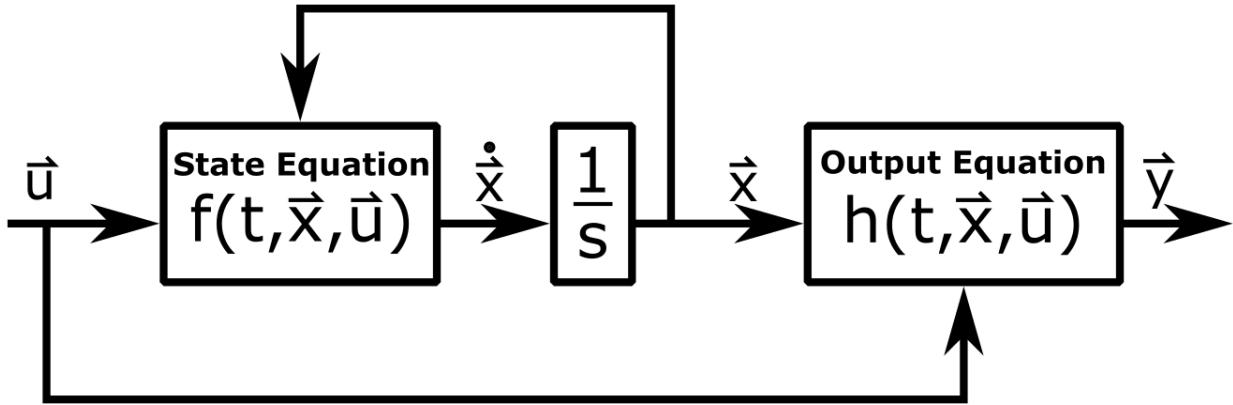
If one exists, a solution to the dynamics,  $\vec{x}$ , can be thought as corresponding to a multi-dimensional curve in the system state space  $\mathbb{R}^n$  as  $t$  or  $k$  vary from an initial value to  $\infty$ , known as the **system state trajectory**. In addition to the dynamics equation, one may also be given an algebraic **system output equation** of the general continuous-time form

$$\vec{y} = h(t, \vec{x}, \vec{u}) \quad (8.3)$$

or discrete-time form

$$\vec{y} = h(k, \vec{x}, \vec{u}) \quad (8.4)$$

where  $h : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^P$  is a vector function and  $\vec{y} \in \mathbb{R}^m$  is **system output**. Together, the state and output equations form the general **system state-space model** which can also be visualized as the following block diagram for constructing signals and systems.



Dynamical systems can be classified by the dimension of the input and output vectors. If  $p = 1$ , then the system is said to be a **single input (SI)** system, otherwise it is a **multiple inputs (MI)** system. Furthermore, if  $m = 1$ , then the system is said to be a **single output (SO)** system, otherwise it is a **multiple outputs (MO)** system. Thus, in combination, one can form the four classifications for systems as SISO, SIMO, MISO, and MIMO systems.

Dynamical systems can also be classified by the structure of the state equation. First, if the state equation does not depend explicitly on time, but only implicitly through the state and input, then one has **time-invariant dynamics** which for the continuous-time case, one has

$$\dot{\vec{x}} = f(\vec{x}(t), \vec{u}(t)) \quad (8.5)$$

and for the discrete-time case, one has

$$\vec{x}[k+1] = f(\vec{x}[k], \vec{u}[k]) \quad (8.6)$$

Second, if the state equation does not contain an input signal,  $\vec{u}$ , then one has **unforced dynamics** which for continuous-time case, one has

$$\dot{\vec{x}} = f(t, \vec{x}(t)) \quad (8.7)$$

and for discrete-time case, one has

$$\vec{x}[k+1] = f(k, \vec{x}[k]) \quad (8.8)$$

Third, if the state equation is both unforced and time-invariant, then one has **autonomous dynamics** which for continuous-time case, one has

$$\dot{\vec{x}} = f(\vec{x}) \quad (8.9)$$

and for discrete-time case, one has

$$\vec{x}[k+1] = f(\vec{x}[k]) \quad (8.10)$$

## Vector and Matrix Norms

The **vector norm**,  $\|\vec{x}\|$ , of any vector  $\vec{x} \in \mathbb{R}^n$  is a real valued function from  $\mathbb{R}^n \rightarrow \mathbb{R}$  with the following properties

1.  $\|\vec{x}\| \geq 0$ ;
2.  $\|\vec{x}\| = 0$  if and only if  $\vec{x}$  is the zero vector in  $\mathbb{R}^n$ ;
3. for any  $\lambda \in \mathbb{R}$ ,  $\|\lambda \vec{x}\| = |\lambda| \|\vec{x}\|$ ; and
4. for any  $\vec{y} \in \mathbb{R}^n$ , the **triangle inequality**

$$\|\vec{x} + \vec{y}\| \leq \|\vec{x}\| + \|\vec{y}\| \quad (8.11)$$

holds.

An important class of vector norms are the  **$L^p$ -norms**, also known as  **$p$ -norms**, defined as

$$\|\vec{x}\|_p = \left( \sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}, \quad 1 \leq p \leq \infty \quad (8.12)$$

where the subscript  $p$  will be dropped and any norm operation implicitly be a  $L^p$ -norm. Of particular interest are the  **$L^1$ -norm**, also known as the **taxicab norm**,

$$\|\vec{x}\|_1 = \sum_{i=1}^n |x_i| \quad (8.13)$$

the  **$L^2$ -norm**, also known as the **Euclidean norm**,

$$\|\vec{x}\|_2 = \left( \sum_{i=1}^n x_i^p \right)^{\frac{1}{2}} \quad (8.14)$$

and the  **$L^\infty$ -norm**, also known as the **vector maximum norm**,

$$\|\vec{x}\|_\infty = \max |x_i| \quad (8.15)$$

The **matrix norm**,  $\|A\|$ , of the matrix  $A \in \mathbb{R}^{m \times n}$  is a real valued function from  $\mathbb{R}^{m \times n} \rightarrow \mathbb{R}$  with the following properties

1.  $\|A\| \geq 0$ ;
2.  $\|A\| = 0$  if and only if  $A$  is the zero matrix in  $\mathbb{R}^{m \times n}$ ;
3. for any  $\lambda \in \mathbb{R}$ ,  $\|\lambda A\| = |\lambda| \|A\|$ ; and
4. for any  $B \in \mathbb{R}^{m \times n}$ , the **triangle inequality**

$$\|A + B\| \leq \|A\| + \|B\| \quad (8.16)$$

holds.

An important class of matrix norms are **induced matrix norms** which are defined for a matrix  $A$  and some specified norm  $\|\vec{x}\|$  as

$$\|A\| = \sup_{\vec{x} \neq 0} \frac{\|A\vec{x}\|}{\|\vec{x}\|} \quad (8.17)$$

where  $\sup$  stands for the **supremum**, also known as the **least upper bound** of the specified set. Typically one uses  $L^p$ -norms for induced matrix norms which can be written as

$$\|A\|_p = \sup_{\vec{x} \neq 0} \frac{\|A\vec{x}\|_p}{\|\vec{x}\|_p} \quad (8.18)$$

Another important class of matrix are **entry-wise matrix norms** which treats a matrix  $A$  as a vector of size  $m \times n$  and uses a vector norm. One such norm is the  $L^{p,q}$ -norms defined as

$$\|A\|_{p,q} = \left( \sum_{j=1}^n \left( \sum_{i=1}^m |a_{i,j}|^p \right)^{\frac{q}{p}} \right)^{\frac{1}{q}} \quad (8.19)$$

where  $a_{i,j}$  denotes the element of  $A$  in the  $i^{\text{th}}$  row and  $j^{\text{th}}$  column. The  $L^{2,2}$ -norm is also known as the **Frobenius norm** and can be defined as

$$\|A\|_F = \left( \sum_{i=1}^m \sum_{j=1}^n |a_{i,j}|^2 \right)^{\frac{1}{2}} \quad (8.20)$$

In addition the  $L^{\infty,\infty}$ -norm is also known as the **matrix maximum norm** and can be defined as

$$\|A\|_{\max} = \max_{i,j} |a_{i,j}| \quad (8.21)$$

### Existence and Uniqueness of Solutions

Consider the state of a continuous-time dynamical system at some time  $t_0 \geq 0$  as  $\vec{x}(t_0) = \vec{x}_0$ . Then, with either unforced or autonomous dynamics, one has the **initial value problem (IVP)** with initial condition  $(t_0, \vec{x}_0)$ , also known as the **Cauchy problem**, which may have many solutions, one unique solution, or no existing solution. This existence and uniqueness of solutions for IVPs is crucial for dynamical systems analysis. For example, if the dynamical system has been modeled to emulate a real-world process, one must know if and when the unique solution would exist, otherwise any dynamical system simulations could lead to erroneous conclusions about the dynamics and control of that process. However, there exists well-known theorems stating sufficient, though not necessary, conditions.

The **Cauchy-Peano theorem** provides sufficient conditions for the existence of an IVP solution which may or may not be unique as follows. If, for some  $T > 0$  and some  $\epsilon > 0$ ,  $f(t, \vec{x}) : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$  is continuous in a closed region  $B$  defined as

$$B = \{(t, \vec{x}) : |t - t_0| \leq T, \|\vec{x} - \vec{x}_0\| \leq \epsilon\} \subseteq \mathbb{R} \times \mathbb{R}^n \quad (8.22)$$

then there exists  $t_0 < t_1 \leq T$  such that the IVP has at least one continuously differentiable solution  $\vec{x}(t)$  on the interval  $[t_0, T]$ . The assumed continuity of  $f(t, \vec{x})$  in its arguments,  $t$  and  $\vec{x}$ , ensures that there is at least one solution of the IVP.

The Lipschitz condition can be used to derive sufficient conditions for the uniqueness of IVP solutions as follows. Defining the **Lipschitz condition** as  $f(t, \vec{x})$  satisfying the inequality

$$\|f(t, \vec{x}) - f(t, \vec{y})\| \leq L \|\vec{x} - \vec{y}\| \quad (8.23)$$

for all  $(t, \vec{x})$  and  $(t, \vec{y})$  in some “neighborhood” of  $(t_0, \vec{x}_0)$  with a finite constant  $L > 0$ , then sufficient conditions for the uniqueness of IVP solutions can be stated for three different cases.

First, a theorem for sufficient conditions for the IVP to admit the local existence and uniqueness of a solution states that if, for some  $\epsilon > 0$ ,  $f(t, \vec{x}) : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$  is piece-wise continuous in  $t$  and satisfies the Lipschitz condition, i.e.

$$\forall \vec{x}, \vec{y} \in B = \{\vec{x} \in \mathbb{R}^n : \|\vec{x} - \vec{x}_0\| \leq \epsilon\}, \quad \forall t \in [t_0, t_1] \quad (8.24)$$

then, there exists some  $\delta > 0$  such that the IVP for the state equation  $\dot{\vec{x}} = f(t, \vec{x})$  with  $\vec{x}(t_0) = \vec{x}_0$  has a unique solution over  $[t_0, t_0 + \delta]$ . Note that here the Lipschitz condition is assumed to be valid only locally in a “neighborhood” of  $(t_0, \vec{x}_0)$  from a compact set, i.e. closed and bounded,  $B$  as defined.

Second, one can extend the interval of existence and uniqueness over a given time interval  $[t_0, t_0 + \delta]$  by taking  $t'_0 = t_0 + \delta$  as the “new” initial time and  $\vec{x}'_0 = \vec{x}(t_0 + \delta)$  as the “new” initial state. If the conditions of the theorem are then satisfied at  $(t_0 + \delta, \vec{x}(t_0 + \delta))$ , then there exists  $\delta_1 > 0$  such that the IVP has a unique solution over  $[t_0 + \delta, t_0 + \delta + \delta_1]$  that passes through point  $(t_0 + \delta, \vec{x}(t_0 + \delta))$ . Furthermore, one can piece together the two solutions to establish the existence of a unique solution over the larger interval  $[t_0, t_0 + \delta + \delta_1]$ . This idea can be repeated to continue to extend the IVP solution, arriving at the maximal IVP solution defined on the maximal interval  $[t_0, t_0 + \delta_{max}]$  with finite or infinite  $\delta_{max}$ .

Third, a theorem for sufficient conditions for the IVP to admit the global existence and uniqueness of a solution states that if, for some finite  $L > 0$ ,  $f(t, \vec{x}) : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$  is piece-wise continuous in  $t$  and is globally Lipschitz in  $\vec{x}$ , i.e.

$$\|f(t, \vec{x}) - f(t, \vec{y})\| \leq L \|\vec{x} - \vec{y}\|, \quad \forall \vec{x}, \vec{y} \in \mathbb{R}^n, \quad \forall t \in [t_0, t_1] \quad (8.25)$$

then, the IVP has a unique solution over  $[t_0, t_1]$  where the final time  $t_1$  may be arbitrarily large, thus achieving global uniqueness. It should be noted that typically when using dynamical systems to model real-world processes, one is primarily interested in constructing IVPs whose solutions globally exist  $\forall t \geq t_0$  and are unique which would at least guarantee the soundness of the models. However, one would typically be required to perform **validation and verification (V&V)** of the models by correlating simulation data with experimental data of the process under consideration.

One should note that the Lipschitz condition, even when local, can be shown to be quite restrictive since the set of all Lipschitz-continuous functions represents a “meager” subset of all continuous functions. Thus, one may conjecture that only a “meager” set of IVPs have unique solutions. Fortunately, **Orlicz theorem** states that “almost all” differential equations with continuous right-hand sides have unique solutions. However, the set of IVPs for which one can formally characterize uniqueness of their solutions by the Lipschitz condition is “almost nothing,” which suggests that there are very many classes of non-Lipschitz IVPs with unique solutions that are yet to be discovered.

## Dynamical System Equilibrium

Recall that one particular type of solution for dynamical systems is an **equilibrium point** which can be defined for both autonomous and non-autonomous, unforced dynamical systems.

For autonomous dynamical systems, an equilibrium point is defined as the value of the state vector,  $\bar{x}$ , in the continuous-time case which solves

$$\dot{\vec{x}}(t) = f(\bar{x}) = 0 \quad (8.26)$$

and in the discrete-time case which solves

$$\vec{x}[k+1] = f(\bar{x}) = \bar{x} \quad (8.27)$$

Thus, by definition, if a system starts at an equilibrium point, it will remain there for all future times. A dynamical system can have multiple equilibrium points. Some may be isolated from each other while others may form a continuum of equilibrium points.

For non-autonomous, unforced dynamical systems, the origin in  $\mathbb{R}^n$  is an equilibrium point for the unforced non-autonomous system at  $t_0 = 0$  if

$$f(t, \vec{0}) = \vec{0}, \quad \forall t \geq 0 \quad (8.28)$$

Without loss of generality, this result can be extended beyond the origin and initial time by defining a nonzero vector  $\bar{x} \in \mathbb{R}^n$  to be an equilibrium point of  $\dot{\vec{x}} = f(t, \vec{x})$  at a nonzero initial time  $t = t_0$ , thus

$$f(t, \bar{x}) = \vec{0}, \quad \forall t \geq t_0 \quad (8.29)$$

then defining the new time as  $\tau = t - t_0$  and new state as  $\vec{z}(\tau) = \vec{x}(\tau + t_0) - \bar{x}$ , one has for the new system dynamics

$$\frac{d\vec{z}(\tau)}{d\tau} = \frac{\vec{x}(\tau + t_0)}{dt} = f(\tau + t_0, \vec{z}(\tau) + \bar{x}) = g(\tau, \vec{z}(\tau)) \quad (8.30)$$

with  $g(0, \vec{0}) = f(t_0, \bar{x}) = 0$ . Thus, one can shift the equilibrium point to the origin and initial time to zero. Furthermore, suppose one has a state trajectory  $\bar{x}(t)$  that starts at  $t = t_0$ , i.e.

$$\dot{\bar{x}}(t) = f(t, \bar{x}(t)), \quad \forall t \geq t_0 \quad (8.31)$$

then again defining the new time as  $\tau = t - t_0$  and the new state this time as  $\vec{z}(\tau) = \vec{x}(\tau + t_0) - \bar{x}(\tau + t_0)$ , one has for the new system dynamics

$$\begin{aligned} \frac{d\vec{z}(\tau)}{d\tau} &= \frac{\vec{x}(\tau + t_0)}{dt} - \frac{d\bar{x}(\tau + t_0)}{dt} \\ &= f(\tau + t_0, \vec{z}(\tau) + \bar{x}(\tau + t_0)) - f(\tau + t_0, \bar{x}(\tau + t_0)) = g(\tau, \vec{z}(\tau)) \end{aligned} \quad (8.32)$$

with  $g(0, \vec{0}) = \vec{0}$ . Consequently, analyzing these new dynamics around the origin, as an equilibrium point, while starting at  $t_0$ , allows one to determine the original system behavior around the original nonzero equilibrium  $\bar{x}$ , i.e. one can assess the system relative dynamics with respect to any time-dependent trajectory  $\bar{x}(t)$ , starting at an arbitrary initial time  $t_0 \geq 0$ . Note that for unforced linear dynamical systems,  $\dot{\vec{x}} = A\vec{x}$  has a single equilibrium point at  $\vec{x} = \vec{0}$  if and only if  $A$  is full rank, i.e.  $\det A \neq 0$ , otherwise the system will have a continuum of equilibrium points.

## 8.2 Advanced Linear State-Space Systems

An important class of state-space system representations is **linear state-space system**, which can generally be defined for the continuous-time representation as

$$\begin{aligned} \dot{\vec{x}}(t) &= A(t)\vec{x}(t) + B(t)\vec{u}(t) \\ \vec{y}(t) &= C(t)\vec{x}(t) + D(t)\vec{u}(t) \end{aligned} \quad (8.33)$$

where  $A \in \mathbb{R}^{n \times n}$  is the **continuous-time state matrix**,  $B \in \mathbb{R}^{n \times p}$  is the **continuous-time input matrix**,  $C \in \mathbb{R}^{m \times n}$  is the **output matrix**, and  $D \in \mathbb{R}^{m \times p}$  is the **feedthrough matrix**. The discrete-time linear state-space representation is

$$\begin{aligned}\vec{x}[k+1] &= F[k]\vec{x}[k] + G[k]\vec{u}[k] \\ \vec{y}[k] &= H[k]\vec{x}[k] + D[k]\vec{u}[k]\end{aligned}\quad (8.34)$$

where  $F \in \mathbb{R}^{n \times n}$  is the **discrete-time state matrix**,  $G \in \mathbb{R}^{n \times p}$  is the **discrete-time input matrix**, and  $H \in \mathbb{R}^{m \times n}$  is the **output matrix**. One should note that  $A$ ,  $B$ , and  $C$  are sometimes also used for denoting both continuous- and discrete-time state-space models in other materials. For linear state-space system representations, if the coefficient matrices depend on time, one has a **linear, time-varying (LTV)** system. Furthermore, if this time dependence of the coefficient matrices enters by parameter vector, then one has a **linear, parameter-varying (LPV)** state-space system. Lastly, if the coefficient matrices do not depend on time, then one has a **linear, time-invariant (LTI)** system.

### Linear State-Space System Solutions

While analytical solutions to general dynamical systems can be difficult to compute, methods from linear algebra provide an analytical solution for linear state-space systems. The solution for continuous-time linear state-space systems at some time  $t$  given the state  $\vec{x}(t_0)$  at some initial time  $t_0$  can be generalized as

$$\begin{aligned}\vec{x}(t) &= \Phi(t, t_0)\vec{x}(t_0) + \int_{t_0}^t \Phi(t, \tau)B\vec{u}(\tau)d\tau \\ \vec{y}(t) &= C\vec{x}(t) + D\vec{u}(t)\end{aligned}\quad (8.35)$$

where  $\Phi(t_2, t_1)$  is the **state-transition matrix** from time  $t_1$  to  $t_2$ , the first term is the **zero-input response** or **free response**, and the second term is the **zero-state response**.

For continuous-time LTI systems from  $t_0 = 0$ , this solution can be shown to be

$$\begin{aligned}\vec{x}(t) &= e^{At}\vec{x}(0) + \int_0^t e^{A(t-\tau)}B\vec{u}(\tau)d\tau \\ \vec{y}(t) &= C\vec{x}(t) + D\vec{u}(t)\end{aligned}\quad (8.36)$$

where  $\Phi(t_2, t_1) = e^{A(t_2-t_1)}$  and denotes the **matrix exponential** using  $t_1$  and  $t_2$ . This quantity can be defined as the series

$$e^A = \sum_{k=0}^{\infty} \frac{1}{k!} A^k \quad (8.37)$$

where  $A^0$  is the identity matrix  $I$ , i.e.

$$I = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \quad (8.38)$$

with same dimensions as  $A$ . This series is well defined and always converges so it is not difficult to compute a high fidelity numerical solution for continuous-time LTI state-space systems.

Furthermore, it should be noted that the LTI state-space solution can also be computed using the Laplace transform of the state equation as

$$s\vec{X}(s) - \vec{X}(0) = A\vec{X}(0) + B\vec{U}(s) \quad (8.39)$$

or

$$(sI - A)\vec{X}(s) = \vec{X}(0) + B\vec{U}(s) \quad (8.40)$$

which yields

$$\vec{X}(s) = (sI - A)^{-1}\vec{X}(0) + (sI - A)^{-1}B\vec{U}(s) \quad (8.41)$$

where  $(sI - A)^{-1}$  is equivalent to the Laplace transform of the matrix exponential,  $e^{At}$ , and is also known as the **resolvent matrix**. With the output equation, one has

$$\vec{Y}(s) = C\vec{X}(s) + D\vec{U}(s) \quad (8.42)$$

or

$$\vec{Y}(s) = C(sI - A)^{-1}\vec{X}(0) + \left(C(sI - A)^{-1}B + D\right)\vec{U}(s) \quad (8.43)$$

where for zero initial conditions,  $C(sI - A)^{-1}B + D$  would be the LTI system's **transfer function matrix** which maps the input vector to the output vector in the Laplace domain. This type of representation is often used for MIMO frequency domain control methods, but will not be heavily utilized in this course.

The solution for discrete-time linear state-space systems can be obtained much more simply through matrix multiplication and addition due to the fact that the state equation is a difference equation instead of a differential equation. Specifically, by stacking the input vectors at each time step, one can write the solution for the state-space system at  $k$  starting from  $\vec{x}_{k_0}$  at some initial time step  $k_0$  as

$$\begin{aligned} \vec{x}[k+1] &= (F[k] \dots F[k_0]) \vec{x}[k_0] \\ &\quad + [(F[k] \dots F[k_0+1]G[k_0]) \quad \cdots \quad (F[k]G[k-1]) \quad (G[k])] \begin{bmatrix} \vec{u}[k_0] \\ \vdots \\ \vec{u}[k-1] \\ \vec{u}[k] \end{bmatrix} \end{aligned} \quad (8.44)$$

$$\vec{y}[k] = H[k]\vec{x}[k] + D[k]\vec{u}[k]$$

For LTI systems and  $k_0 = 0$ , one can simplify this solution as

$$\begin{aligned} \vec{x}[k+1] &= F^k \vec{x}[0] + \sum_{\ell=0}^{k-1} F^{k-1-\ell} G \vec{u}[\ell] \\ \vec{y}[k] &= H\vec{x}[k] + D\vec{u}[k] \end{aligned} \quad (8.45)$$

Similar to the Laplace transform, one can use the Z-transform to obtain the discrete frequency solution from the state equation as

$$z\vec{X}(z) - z\vec{X}[0] = F\vec{X}(z) + G\vec{U}(z) \quad (8.46)$$

or

$$(zI - F)\vec{X}(z) = z\vec{X}[0] + G\vec{U}(z) \quad (8.47)$$

which yields

$$\vec{X}(z) = z(zI - F)^{-1}\vec{X}[0] + (zI - F)^{-1}G\vec{U}(z) \quad (8.48)$$

With the output equation

$$\vec{Y}(z) = H\vec{X}(z) + D\vec{U}(z) \quad (8.49)$$

or

$$\vec{Y}(z) = Hz(zI - F)^{-1}\vec{X}[0] + \left(H(zI - F)^{-1}G + D\right)\vec{U}[z] \quad (8.50)$$

### LTI Continuous- to Discrete-Time Approximation

Furthermore, one can approximate a continuous-time LTI state-space system with a discrete-time representation by defining  $t = k\Delta t$ , where  $k = 0, 1, 2, \dots$  and  $\Delta t$  is increment between time steps. Then, for the first time step, one has

$$\vec{x}(\Delta t) = e^{A\Delta t}\vec{x}(0) + \int_0^{\Delta t} e^{A(\Delta t-\tau)}B\vec{u}(\tau)d\tau \quad (8.51)$$

If one assumes a **zero-order hold** on  $\vec{u}$ , i.e.  $\vec{u}(t)$  is constant from  $t = 0$  to  $t = \Delta t$ , then

$$\vec{x}(\Delta t) = [e^{A\Delta t}] \vec{x}(0) + \left[ \int_0^{\Delta t} e^{A(\Delta t-\tau)}B(\tau)d\tau \right] \vec{u}(0) \quad (8.52)$$

which can be generalized for LTI systems for any given state at step  $k - 1$  to transition at the next time step  $k$  as

$$\vec{x}[k] = [e^{A\Delta t}] \vec{x}[k - 1] + \left[ \int_0^{\Delta t} e^{A(\Delta t-\tau)}B(\tau)d\tau \right] \vec{u}[k - 1] \quad (8.53)$$

Finally, by inspection, the approximate discrete-time state matrix can be defined as

$$F = e^{A\Delta t} \quad (8.54)$$

and the discrete-time input matrix can be defined as

$$G = \int_0^{\Delta t} e^{A(\Delta t-\tau)}B(\tau)d\tau, \quad (8.55)$$

to obtain an approximate discrete-time LTI state-space system.

### Matrix Decompositions and Definitions

A fundamental analysis of the behavior of linear state-space system solutions can be performed through an **eigenvalue decomposition** of the state matrix. An **eigenvector**,  $\vec{v}$ , of a square matrix  $A$  is defined as a solution to the **eigenvalue problem** defined as “given some  $A$ , for what values of  $\lambda$  and  $\vec{v}$  does the following equation hold:

$$A\vec{v} = \lambda\vec{v} \quad (8.56)$$

where  $\lambda$  is the scalar **eigenvalue** associated with  $\vec{v}$ . To solve this problem, one can rewrite the equation above as

$$\lambda\vec{v} - A\vec{v} = 0 \quad (8.57)$$

$$[\lambda I - A] \vec{v} = 0 \quad (8.58)$$

which for nontrivial solutions, i.e.  $\vec{v} \neq 0$ , one must solve for

$$\det(\lambda I - A) = 0 \quad (8.59)$$

to obtain the eigenvalues of  $A$ , i.e.  $\lambda(A)$ . Then, by substituting back into  $[\lambda I - A] \vec{v} = 0$ , one can obtain also obtain the corresponding eigenvectors. In general,  $\lambda$  and  $\vec{v}$  can be complex-valued even if  $A$  is real-valued. Based on the values of  $\lambda$ , one can define the following:

- $A$  is **positive definite**, denoted by  $A > 0$ , if the real parts of all  $\lambda(A)$  are  $> 0$ ;
- $A$  is **positive semi-definite**, denoted by  $A \geq 0$ , if the real parts of all  $\lambda(A)$  are  $\geq 0$ ;
- $A$  is **negative definite**, denoted by  $A < 0$ , if the real parts of all  $\lambda(A)$  are  $< 0$ ;
- $A$  is **negative semi-definite**, denoted by  $A \leq 0$ , if the real parts of all  $\lambda(A)$  are  $\leq 0$ ; and
- $A$  is **indefinite** otherwise.

It should also be noted that a positive definite  $A$  matrix is commonly also known as a **Hurwitz matrix**.

Furthermore, a square matrix  $A$  is **diagonalizable**, if an eigenvalue decomposition can be performed as

$$A = V\Lambda V^{-1} \quad (8.60)$$

where the  $n$  eigenvectors of  $A$  makeup  $V$  as

$$V = [\vec{v}_1 \vec{v}_2 \cdots \vec{v}_n] \quad (8.61)$$

and the  $n$  corresponding *non-repeated* eigenvalues of  $A$  makeup a diagonal  $\Lambda$  as

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} \quad (8.62)$$

However, if there are *repeated* eigenvalues of  $A$ , then  $A$  may not be diagonalizable. Then, a **Jordan matrix** can be used to form  $\Lambda$  in a similar fashion to an eigenvalue decomposition where a Jordan matrix,  $J$ , is defined as

$$J = \begin{bmatrix} J_1 & 0 & \cdots & 0 \\ 0 & J_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & J_k \end{bmatrix} \quad (8.63)$$

where the 0's are zero-valued matrices and the  $k$  **Jordan blocks**,  $J_k$ , are specified by dimension  $r$  and eigenvalue  $\lambda_r$ , i.e.

$$J_k(r, \lambda_r) = \begin{bmatrix} \lambda_r & 1 & 0 & \cdots & 0 & 0 \\ 0 & \lambda_r & 1 & \cdots & 0 & 0 \\ 0 & 0 & \lambda_r & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_r & 1 \\ 0 & 0 & 0 & \cdots & 0 & \lambda_r \end{bmatrix} \quad (8.64)$$

For diagonalizable matrices, the Jordan matrix is purely diagonal since each constituent Jordan block is  $1 \times 1$ . The Jordan matrix is useful in forming the **Jordan Canonical Form (JCF)** of LTI state-space systems through the substitution

$$\vec{x} = V \vec{z} \quad (8.65)$$

which allows the continuous-time LTI state-space system to be rewritten as

$$\begin{aligned} V \dot{\vec{z}}(t) &= AV \vec{z}(t) + B \vec{u}(t) \\ \dot{y}(t) &= CP \vec{z}(t) + D \vec{u}(t) \end{aligned} \quad (8.66)$$

$$\begin{aligned} \dot{\vec{z}}(t) &= V^{-1} AV \vec{z}(t) + V^{-1} B \vec{u}(t) \\ \dot{y}(t) &= CP \vec{z}(t) + D \vec{u}(t) \end{aligned} \quad (8.67)$$

$$\begin{aligned} \dot{\vec{z}}(t) &= \Lambda \vec{z}(t) + \bar{B} \vec{u}(t) \\ \dot{y}(t) &= \bar{C} \vec{z}(t) + D \vec{u}(t) \end{aligned} \quad (8.68)$$

where  $\Lambda$  is a Jordan matrix (i.e. diagonal or nearly diagonal), and  $\bar{B}$  and  $\bar{C}$  are new input and output matrices, respectively. To obtain  $V$ , one must solve for the **generalized eigenvectors** for each Jordan block,  $J_k(r, \lambda_r)$ , which solve

$$\begin{aligned} (A - \lambda_r I) \vec{v}_1 &= 0 \\ (A - \lambda_r I) \vec{v}_2 &= v_1 \\ &\vdots \\ (A - \lambda_r I) \vec{v}_r &= v_{r-1} \end{aligned} \quad (8.69)$$

There are several other important matrix definitions used in linear state-space systems analysis. If one defines  $A_{i,j}$  as the element of  $A$  in the  $i^{\text{th}}$  row and  $j^{\text{th}}$  column, then one can define the following.

- The **matrix transpose** of  $A$  is  $A^T = B$  which assigns  $B_{i,j} = A_{j,i}$ .
- The **conjugate matrix transpose** or **Hermitian transpose** of  $A$  is  $A^H = B$  which assigns  $B_{i,j} = A_{j,i}^*$ .
- The **matrix inverse** of  $A$  is the solution to  $A^{-1}A = I$ .
- $A$  is an **orthogonal matrix** if  $A^{-1} = A^T$ .
- $A$  is an **unitary matrix** if  $A^{-1} = A^*$ .
- $A$  is a **symmetric matrix** if  $A = A^T$ .
- $A$  is a **Hermitian matrix** if  $A = A^*$ .
- $A$  is a **diagonal matrix** if for  $i \neq j$ ,  $A_{i,j} = 0$ .
- $A$  is an **upper triangular matrix** if for  $i < j$ ,  $A_{i,j} = 0$ .

- $A$  is a **lower triangular matrix** if for  $i > j$ ,  $A_{i,j} = 0$ .
- $A$  is a **square matrix** if it has an equal number of rows and columns.

When considering the full system behavior from input to output, one often uses the **singular value decomposition (SVD)** defined for any  $m \times n$  real-valued matrix  $M$  as

$$M = U\Sigma V^{-1} \quad (8.70)$$

where  $U$  is an  $m \times m$  orthogonal matrix,  $\Sigma$  is a diagonal  $m \times n$  matrix with *non-negative* real numbers on the diagonal, and  $V$  is an  $n \times n$  orthogonal matrix. The diagonal entries of  $\Sigma$  are known as the **singular values** of  $M$  and there will be at most  $\min(m, n)$  distinct singular values because of its non-square nature. In addition, one can say that  $U$  and  $V$  are orthogonal,  $U^{-1} = U^T$  and  $V^{-1} = V^T$ . Thus the SVD could also be written as

$$M = U\Sigma V^T \quad (8.71)$$

which is much simpler to compute. The singular values  $\sigma_i$  for  $M$  are the non-negative real numbers for which there exists unit-length real-valued vectors  $\vec{u}$  and  $\vec{v}$  such that

$$M\vec{v}_i = \sigma_i \vec{u} \quad (8.72)$$

and

$$\vec{u}_i^T M = \sigma_i \vec{v}^T \quad \text{or} \quad M^T \vec{u}_i = \sigma_i \vec{v} \quad (8.73)$$

where  $\vec{v}_i$  is the right-singular vector for  $\sigma_i$  and all together make up the columns of  $V$  and  $\vec{u}_i$  is the left-singular vector for  $\sigma_i$  and all together make up the columns of  $U$ . Note that these two linked expressions are very similar to the eigenvalue problem except one has two related problems due to the non-square nature of  $M$ . In fact, it can be shown that the left-singular vectors of  $M$  are the eigenvectors of  $MM^T$ , the right-singular vectors of  $M$  are the eigenvectors of  $M^T M$ , and the non-negative singular values of  $M$  are the square roots of the non-negative eigenvalues of both  $M^T M$  and  $MM^T$ . It is important to point out that the SVD also can be extended to complex matrices which uses Hermitian transposes and unitary matrices instead of the transpose and orthogonal matrices.

Lastly, there are other matrix decompositions that may be used in computational algorithms for linear state-space systems. One such decomposition is the **QR decomposition** for any square matrix  $A$  defined as

$$A = QR \quad (8.74)$$

where  $Q$  is orthogonal and  $R$  is upper triangular. A second decomposition is the **Cholesky decomposition** for Hermitian positive definite matrices defined as

$$A = LL^* \quad (8.75)$$

where  $L$  is an upper triangular matrix with real and positive entries along its main diagonal. Note that if  $A$  is real-valued, then this reduces to  $A = LL^T$ .

## 8.3 Lyapunov Stability and Methods

### Lyapunov Stability of Equilibrium Points

In his dissertation, Lyapunov developed definitions for the stability of state trajectories of unforced dynamical systems. System stability can be interpreted as a continuity of the state trajectories, with respect to initial conditions, over an *infinite* time interval. This infinite time interval highlights the primary notion of stability as a continuity property of Lipschitz-continuous differential equations holding infinitely in time.

Formally, consider  $\vec{x}(t, \vec{x}_0)$  as a unique solution of  $\dot{\vec{x}} = f(t, \vec{x})$  with initial condition  $\vec{x}(t_0) = \vec{x}_0$  which exists on a finite, possibly open-ended interval  $[t_0, T]$ . The continuity property of  $\vec{x}(t, \vec{x}_0)$  due to changes in  $\vec{x}_0$  can be stated as follows. Given any constant  $\epsilon > 0$ , there must exist a sufficiently small constant  $\delta > 0$  such that for all perturbed initial conditions  $\vec{x}_0 + \Delta\vec{x}_0$  with  $\|\Delta\vec{x}_0\| \leq \delta$ , the corresponding perturbed solution  $\vec{x}(t, \vec{x}_0 + \Delta\vec{x}_0)$  deviates from the original  $\vec{x}_0$  by no more than  $\epsilon$ , i.e.  $\|\vec{x}(t, \vec{x}_0 + \Delta\vec{x}_0) - \vec{x}(t, \vec{x}_0)\| \leq \epsilon$ , for all  $t_0 \leq t < T$ . In practice, one is most often interested in analyzing state trajectories that are defined on an infinite interval  $[t, \infty)$ . In particular, if  $\vec{x}(t, \vec{x}_0)$  has this continuity property defined on an infinite interval, then  $\vec{x}(t, \vec{x}_0)$  is **Lyapunov stable**, otherwise it is unstable.

Furthermore, one can define the **Lyapunov stability** of an equilibrium point,  $\bar{x}$ , as *stable* if for any  $\epsilon > 0$  and  $t \geq 0$  there exists some  $\delta(\epsilon, t_0) > 0$  such that for all initial conditions  $\|\vec{x}_0\| < \delta$  and for all  $t \geq t_0 \geq 0$ , the corresponding state trajectories are bounded, i.e.  $\|\vec{x}(t)\| < \epsilon$ , otherwise it is *unstable*. In essence, Lyapunov stability of an equilibrium point means that given an outer “hyper-sphere”  $B_\epsilon = \{\vec{x} \in \mathbb{R}^n : \|\vec{x}\| \leq \epsilon\}$ , one can find an inner “hyper-sphere”  $B_\delta = \{\vec{x} \in \mathbb{R}^n : \|\vec{x}\| \leq \delta\}$ , such that any trajectory that starts inside  $B_\delta$  will evolve inside  $B_\epsilon$  for *all* future times. A stronger sense of Lyapunov stability is **asymptotic stability** if there also exists  $\delta_0 > 0$  such that whenever the initial conditions are within  $\delta_0$  (which may depend on  $t_0$  or  $k_0$ ), then  $\vec{x} \rightarrow \bar{x}$  as  $t \rightarrow \infty$  or  $k \rightarrow \infty$ . Furthermore, if  $\delta = \infty$  or  $\delta_0 = \infty$ , then the equilibrium point is said to be **globally stable** or **globally asymptotically stable**, respectively.

A unique feature of non-autonomous dynamical systems is the dependence of the state trajectories on the selected initial time,  $t_0$ , or time step,  $k_0$ . In general, the stability of an equilibrium point for non-autonomous systems will depend on  $t_0$  or  $k_0$ . However, the equilibrium point has **uniform stability** if it is stable and  $\delta$  does not depend on  $t_0$ . The equilibrium point has **uniform asymptotic stability** if it is uniformly stable and there exists a constant  $c > 0$  independent of  $t_0$  such that  $\vec{x} \rightarrow 0$  as  $t \rightarrow \infty$  for all  $\|\vec{x}\| \leq c$  uniformly in  $t_0$ . The equilibrium point has **global uniform asymptotic stability** if it is uniformly asymptotically stable and  $\lim_{\epsilon \rightarrow \infty} \delta(\epsilon) = \infty$ .

### Lyapunov Direct and Indirect Methods

In his dissertation, Lyapunov also developed two methods for assessing the stability of state trajectories about equilibrium points without requiring explicit computation of these state trajectories, known as the indirect and the direct methods. These two methods form the basis of control design for general dynamical systems.

The **indirect Lyapunov method** of Lyapunov states that one can determine the stability of a state trajectory about an equilibrium point for autonomous dynamical systems,  $\bar{x}$ , by linearizing the system dynamics about the equilibrium point. In order for the original nonlinear system to be locally stable in the Lyapunov sense, it is sufficient to show the system Jacobian matrix  $A = \frac{\partial f(\vec{x})}{\partial \vec{x}}|_{\vec{x}=\bar{x}}$  has *all* its eigenvalues,  $\lambda_i$ ,  $i = 1, \dots, n$  in the complex open left-half plane (LHP), i.e.  $\text{Real}(\lambda_i) < 0$ ,  $\forall i = 1, \dots, n$ . If *at least* one eigenvalue is not in the LHP, then the origin is unstable. Furthermore, if  $A$  has eigenvalues on the  $j\omega$ -axis,

then the indirect method of Lyapunov does not apply. This method justifies the use of LTI systems analysis and optimal control based on the linearization of nonlinear time-invariant systems. This will be further discussed for LTI systems in the subsections that follow.

The **direct Lyapunov method** of Lyapunov uses a Lyapunov function of the state,  $V(\vec{x})$ , which must have certain properties in order to guarantee stability. One of which is **function definiteness**.

A scalar function  $V(\vec{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$  of a vector argument  $\vec{x} \in \mathbb{R}^n$  is called **locally positive definite** if  $V(\vec{0}) = 0$  and there exists a constant  $\epsilon$  such that  $V > 0$  for all  $\vec{x} \in \mathbb{R}^n$  in the neighborhood of the origin, i.e.  $B_\epsilon = \{\vec{x} \in \mathbb{R}^n : \|\vec{x}\| \leq \epsilon\}$ , where if  $\epsilon = \infty$ , then  $V(\vec{x})$  is **globally positive definite**.

A scalar function  $V(\vec{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$  of a vector argument  $\vec{x} \in \mathbb{R}^n$  is called **locally positive semi-definite** if  $V(\vec{0}) = 0$  and there exists a constant  $\epsilon$  such that  $V \geq 0$  for all  $\vec{x} \in \mathbb{R}^n$  in the neighborhood of the origin, i.e.  $B_\epsilon = \{\vec{x} \in \mathbb{R}^n : \|\vec{x}\| \leq \epsilon\}$ , where if  $\epsilon = \infty$ , then  $V(\vec{x})$  is **globally positive semi-definite**.

A scalar function  $V(\vec{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$  of a vector argument  $\vec{x} \in \mathbb{R}^n$  is called **locally negative definite** if  $V(\vec{0}) = 0$  and there exists a constant  $\epsilon$  such that  $V < 0$  for all  $\vec{x} \in \mathbb{R}^n$  in the neighborhood of the origin, i.e.  $B_\epsilon = \{\vec{x} \in \mathbb{R}^n : \|\vec{x}\| \leq \epsilon\}$ , where if  $\epsilon = \infty$ , then  $V(\vec{x})$  is **globally negative definite**.

A scalar function  $V(\vec{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$  of a vector argument  $\vec{x} \in \mathbb{R}^n$  is called **locally negative semi-definite** if  $V(\vec{0}) = 0$  and there exists a constant  $\epsilon$  such that  $V \leq 0$  for all  $\vec{x} \in \mathbb{R}^n$  in the neighborhood of the origin, i.e.  $B_\epsilon = \{\vec{x} \in \mathbb{R}^n : \|\vec{x}\| \leq \epsilon\}$ , where if  $\epsilon = \infty$ , then  $V(\vec{x})$  is **globally negative semi-definite**.

Secondly, one must introduce the concept of the time derivative of a scalar function along the state trajectories of a differential equation, i.e. it's possible solutions. Suppose one is given a scalar continuously differentiable function  $V(\vec{x})$  where  $\vec{x}(t) \in \mathbb{R}^n$  represents a time-varying trajectory of the nonautonomous system. Then, the time derivative of  $V(\vec{x}(t))$  along the system solution  $\vec{x}(t) = [x_1(t), \dots, x_n(t)]^T$  as

$$\dot{V}(\vec{x}) = \sum_{i=1}^n \frac{\partial V}{\partial x_i} \dot{x}_i = \sum_{i=1}^n \frac{\partial V}{\partial x_i} f_i(t, \vec{x}) = \nabla V(\vec{x}) f(t, \vec{x}) \quad (8.76)$$

where  $\nabla V(\vec{x}) = [\frac{\partial V}{\partial x_1}, \dots, \frac{\partial V}{\partial x_n}]$  is the row vector gradient of  $V(\vec{x})$  with respect to  $\vec{x}$ . Note that the time derivative of  $V(\vec{x})$  depends not only on the function  $V(\vec{x})$  but also on the system dynamics under consideration. Changing the latter while keeping the same  $V(\vec{x})$  will, in general, result in a different  $\dot{V}(\vec{x})$ .

Let  $\vec{x}^* = \vec{0} \in \mathbb{R}^n$  be an equilibrium point for the nonautonomous dynamics, whose initial conditions are drawn from a domain  $D \subset \mathbb{R}^n$  with  $\vec{x}^* \in D$  and  $t_0 = 0$ . If on the domain  $D$ , there exists a continuously differentiable locally positive definite function  $V(\vec{x}) : D \rightarrow \mathbb{R}$ , whose time derivative along the system trajectories is locally negative semi-definite, i.e.

$$\dot{V}(\vec{x}) = \nabla V(\vec{x}) f(t, \vec{x}) \leq 0 \quad (8.77)$$

for all  $t \geq 0$  and for all  $\vec{x} \in D$ , then the system equilibrium  $\vec{x}^* = \vec{0}$  is locally uniformly stable. Furthermore, if  $\dot{V}(\vec{x}) < 0$  for all  $t \geq 0$ , i.e. the time derivative along the system trajectories is locally negative definite, then the origin is locally uniformly asymptotically stable. Here any locally positive definite  $V(\vec{x})$  is called a **Lyapunov function candidate** and if it satisfies the time derivative condition it is called a **Lyapunov function**.

Note though that the existence of a Lyapunov function is sufficient to claim uniform stability for the equilibrium point, if one cannot be found, nothing can be stated about the stability of the equilibrium point. Furthermore, it should be noted that Lyapunov functions are not unique. The Lyapunov function can be viewed as an “energy-like” function for testing the stability of a system. If the values of  $V$  do not increase

along the system trajectories, then the origin is uniformly stable. If  $V$  strictly decreases, then, in addition, the system trajectories will approach the origin asymptotically. Lastly, note that the uniform asymptotic stability requires a subset of  $D$  known as the **region of attraction****region of attraction**, i.e. starting there the system solutions will converge to the origin. If the region of attraction of a uniformly asymptotically stable equilibrium is  $\mathbb{R}^n$ , then the equilibrium is said to be globally uniformly asymptotically stable.

Lastly, define the property  $\lim_{\|\vec{x}\| \rightarrow \infty} V(\vec{x}) = \infty$  where  $V(\vec{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$  as a **radially unbounded** Lyapunov function candidate. In addition, define  $V_c = \{\vec{x} \in \mathbb{R}^n : V(\vec{x}) = c\}$  as a **level set**, i.e.  $V_c$  has a constant value  $c$ , of a radially unbounded Lyapunov function candidate  $V(\vec{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ , and define  $\Omega_c = \{\vec{x} \in \mathbb{R}^n : V(\vec{x}) \leq c\}$  be the union of the interior set of  $V_c$  and  $V_c$  itself. Then, consider a converging sequence  $\lim_{k \rightarrow \infty} \vec{x}_k = \vec{a}$  with all  $\vec{x}$  from  $\Omega_c$ , then the limit point  $\vec{a}$  must also be in  $\Omega_c$ . Furthermore, since  $V(\vec{x})$  is continuous on  $\mathbb{R}^n$  and  $V(\vec{x}) \leq c$  for all  $k = 1, 2, \dots$ , one has  $c \geq \lim_{k \rightarrow \infty} V(\vec{x}_k) = V(\vec{a})$ , and consequently  $\vec{a} \in \Omega_c$ . Thus, every converging sequence in  $\Omega_c$  has its limit point in the same set which defines a **closed set**.

One can also prove that  $\Omega_c$  is a **bounded set** as if it was not, then there must exist a sequence of points  $\{\vec{x}_k\} \in \Omega_c$  whose limit is  $\infty$ . However, since  $V(\vec{x})$  is continuous and radially unbounded, then  $c \geq \lim_{k \rightarrow \infty} V(\vec{x}_k) = \infty$ , which is a contradiction. Thus, since  $\Omega_c$  is closed, bounded, and belongs to  $\mathbb{R}^n$ , it is a **compact set** which allows one to state the following **Krasovskii-LaSalle theorem**. If  $\vec{x} = \vec{0}$  is an equilibrium point of  $\dot{\vec{x}} = f(t, \vec{x})$  and  $V(\vec{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$  be a radially unbounded *Lyapunov function*, then  $\vec{x}$  is a globally uniformly asymptotically stable equilibrium point. Note that a simple example of radially unbounded Lyapunov function candidates include the quadratic form  $V(\vec{x}) = \vec{x}^T P \vec{x}$  where  $P$  is a symmetric positive definite matrix, i.e.  $P = P^T > 0$ . This is the most common Lyapunov function candidate and will be used to prove stability for model reference adaptive control systems.

## Free Response for LTI State-Space Systems

For an initial condition at  $\vec{x}(0)$ , the free response dynamics of a continuous-time state-space system representation can be reduced to the following

$$\dot{\vec{x}}(t) = A \vec{x}(t) , \quad (8.78)$$

By using the JCF, i.e.

$$\vec{x}(t) = V \vec{z}(t) \quad (8.79)$$

where  $V$  is the matrix of  $n$  eigenvectors of  $A$ , then, the state-space representation can be transformed to using a new state,  $\vec{z}(t)$  as

$$\dot{\vec{z}}(t) = \Lambda \vec{z}(t) \quad (8.80)$$

where  $\Lambda$  is in Jordan form, i.e. diagonal or nearly diagonal.

Assuming  $A$  is diagonalizable, then  $\Lambda$  is diagonal and the state-space is

$$\begin{bmatrix} \dot{z}_1(t) \\ \vdots \\ \dot{z}_n(t) \end{bmatrix} = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix} \begin{bmatrix} z_1(t) \\ \vdots \\ z_n(t) \end{bmatrix} \quad (8.81)$$

$$\begin{bmatrix} \dot{z}_1(t) \\ \vdots \\ \dot{z}_n(t) \end{bmatrix} = \begin{bmatrix} \lambda_1 z_1(t) \\ \vdots \\ \lambda_n z_n(t) \end{bmatrix} \quad (8.82)$$

Then, since each component is independent, the free response is given by the homogeneous solution to a first order ODE, i.e.

$$\begin{bmatrix} z_1(t) \\ \vdots \\ z_n(t) \end{bmatrix} = \begin{bmatrix} e^{\lambda_1 t} z_1(0) \\ \vdots \\ e^{\lambda_n t} z_n(0) \end{bmatrix} \quad (8.83)$$

$$\begin{bmatrix} z_1(t) \\ \vdots \\ z_n(t) \end{bmatrix} = \begin{bmatrix} e^{\lambda_1 t} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^{\lambda_n t} \end{bmatrix} \begin{bmatrix} z_1(0) \\ \vdots \\ z_n(0) \end{bmatrix} \quad (8.84)$$

$$\vec{z}(t) = \begin{bmatrix} e^{\lambda_1 t} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^{\lambda_n t} \end{bmatrix} \vec{z}(0) \quad (8.85)$$

and using the inverse transformation,  $\vec{z}(t) = V^{-1} \vec{x}(t)$ , one can solve for the free response solution in the original state as

$$V^{-1} \vec{x}(t) = \begin{bmatrix} e^{\lambda_1 t} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^{\lambda_n t} \end{bmatrix} V^{-1} \vec{x}(0) \quad (8.86)$$

$$\vec{x}(t) = V \begin{bmatrix} e^{\lambda_1 t} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^{\lambda_n t} \end{bmatrix} V^{-1} \vec{x}(0) \quad (8.87)$$

$$\vec{x}(t) = [\vec{v}_1 \ \cdots \ \vec{v}_n] \begin{bmatrix} e^{\lambda_1 t} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^{\lambda_n t} \end{bmatrix} V^{-1} \vec{x}(0) \quad (8.88)$$

$$\vec{x}(t) = [e^{\lambda_1 t} \vec{v}_1 \ \cdots \ e^{\lambda_n t} \vec{v}_n] V^{-1} \vec{x}(0) \quad (8.89)$$

and representing

$$V^{-1} \vec{x}(0) = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} \quad (8.90)$$

where  $c_1, \dots, c_n$  are scalar constants, one has

$$\vec{x}(t) = [e^{\lambda_1 t} \vec{v}_1 \ \cdots \ e^{\lambda_n t} \vec{v}_n] \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} \quad (8.91)$$

$$\vec{x}(t) = c_1 e^{\lambda_1 t} \vec{v}_1 + \dots + c_n e^{\lambda_n t} \vec{v}_n \quad (8.92)$$

which is a similar form to the SISO LTI free response except here the eigenvectors have an additional affect. By analyzing the relative strength of each element in an eigenvector, it is simpler to observe which eigenvalues affect which states more than others. If some eigenvalues/eigenvectors are complex conjugate pairs, then this expression can be rewritten using sin and cos functions to get only the real part of the solution for  $x(t)$ . This type of analysis is often called **modal analysis** where each real eigenvalue or complex conjugate eigenvalue pair denotes one **mode** of the solution.

It is also important to note that if  $A$  is not diagonalizable, the general solution can be found using **Jordan Chains** which are related to the generalized eigenvectors for repeated eigenvalues. This will include  $t, t^2, \dots, t^{n-1}$  terms and is quite similar to the SISO LTI free response for repeated roots of the characteristic equation. This result is left to other linear algebra resources.

For an initial condition at  $\vec{x}[k]$ , the free response dynamics for discrete-time state-space can be reduced to

$$\vec{x}_{k+1} = F \vec{x}_k \quad (8.93)$$

which has the simple solution for the state at any  $k$  time step as

$$\vec{x}_k = F^k \vec{x}_0 . \quad (8.94)$$

This solution can also be analyzed similar to the continuous time case by transforming to the Jordan canonical form

$$\vec{x}_k = V \vec{z}_k \quad (8.95)$$

where  $V$  is the matrix of  $n$  eigenvectors of  $F$ . Substituting into the equation above we get

$$\vec{z}_{k+1} = \Lambda \vec{z}_k \quad (8.96)$$

where  $\Lambda$  is in Jordan Form.

Rewriting the free response, we have

$$\vec{z}_k = \Lambda^k \vec{z}_0 \quad (8.97)$$

If  $F$  is diagonalizable, then  $\Lambda$  is diagonal and the free response is

$$\vec{z}_k = \begin{bmatrix} \lambda_1^k & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n^k \end{bmatrix} \vec{z}_0 \quad (8.98)$$

and substituting the inverse transformation,  $\vec{z}(t) = V^{-1} \vec{x}(t)$ , one can solve for the free response solution in the original state as

$$V^{-1} \vec{x}_k = \begin{bmatrix} \lambda_1^k & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n^k \end{bmatrix} V^{-1} \vec{x}_0 \quad (8.99)$$

$$\vec{x}_k = V \begin{bmatrix} \lambda_1^k & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n^k \end{bmatrix} V^{-1} \vec{x}_0 \quad (8.100)$$

$$\vec{x}_k = [\vec{v}_1 \ \dots \ \vec{v}_n] \begin{bmatrix} \lambda_1^k & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_n^k \end{bmatrix} V^{-1} \vec{x}_0 \quad (8.101)$$

$$\vec{x}_k = [\lambda_1^k \vec{v}_1 \ \dots \ \lambda_n^k \vec{v}_n] V^{-1} \vec{x}_0 \quad (8.102)$$

and representing

$$V^{-1} \vec{x}_0 = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} \quad (8.103)$$

where  $c_1, \dots, c_n$  are scalar constants, one has

$$\vec{x}_k = [\lambda_1^k \vec{v}_1 \ \dots \ \lambda_n^k \vec{v}_n] \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} \quad (8.104)$$

$$\vec{x}_k = c_1 \lambda_1^k \vec{v}_1 + \dots + c_n \lambda_n^k \vec{v}_n \quad (8.105)$$

which is very similar to the continuous-time solution except the exponential terms have become power terms of the eigenvalues. The same sort of modal analysis and characteristics apply here as well.

### LTI State-Space System Stability

In particular, continuous-time LTI systems are globally stable if and only if the eigenvalues of  $A$  are in the closed left half of the complex plane (LHP), i.e. the real part of the eigenvalues  $\leq 0$ . As a partial proof, consider the free response

$$\vec{x}(t) = c_1 e^{\lambda_1 t} \vec{v}_1 + \dots + c_n e^{\lambda_n t} \vec{v}_n \quad (8.106)$$

as  $t \rightarrow \infty$ , each  $i^{\text{th}}$  component with:

- $\text{Real}(\lambda_i) < 0$  will  $\rightarrow 0$
- $\text{Real}(\lambda_i) = 0$  will remain constant if  $\lambda_i$  is purely real
- $\text{Real}(\lambda_i) = 0$  will oscillate with a constant magnitude if  $\lambda_i$  is purely imaginary

all of which are bounded (i.e. there exists some  $\epsilon$ ). A full proof must include non-diagonalizable  $A$  and can be done in a similar fashion.

Similarly, discrete-time LTI systems are globally stable if and only if the eigenvalues of  $F$  have magnitude  $\leq 1$ . As a partial proof, consider the free response

$$\vec{x}_k = c_1 \lambda_1^k \vec{v}_1 + \dots + c_n \lambda_n^k \vec{v}_n \quad (8.107)$$

as  $k \rightarrow \infty$  each  $i^{\text{th}}$  component with

- $|\lambda_i| < 1$  will go  $\rightarrow 0$

- $|\lambda_i| = 1$  will remain constant as  $c_i \vec{v}_i$

both of which are bounded. A full proof must include non-diagonalizable  $F$  and can be done in a similar fashion.

Each of these stabilities can be expanded to include two different types. Continuous-time MIMO LTI systems are **marginally stable** if  $\text{real}(\lambda_i) \leq 0 \forall i$  with  $\text{real}(\lambda_j) = 0$  for at least one index  $j$  and are **asymptotically stable** if all  $\text{real}(\lambda_i) < 0 \forall i$ . Likewise, discrete-time LTI systems are **marginally stable** if  $|\lambda_i| \leq 1 \forall i$  with  $|\lambda_j| = 1$  for at least one index  $j$  and **asymptotically stable** if all  $|\lambda_i| < 1 \forall i$ . It is also important to point out that typically *asymptotically stable* shortened to *stable* in literature.

## 8.4 Linear System Controllability and Observability

For general state-space systems, one is not only concerned with the free response or autonomous dynamics, but typically also the system effects due to the control input in the state equation as well as the form of the algebraic output equation. For linear state-space systems, the analysis of these effects can be captured by **controllability** and **observability** analysis of the system state. Similar to how analysis of the state matrix was fundamental to the stability analysis of the linear state-space system, the input matrix and output matrix are crucial for these additional analyses, in conjunction with the state matrix.

Before proceeding, a concept from linear algebra is required. The **rank** of a matrix  $M$ , denoted by  $\text{rank}(M)$ , is a measure of the “non-degenerateness” of the system of linear equations encoded by  $M$ . More formally, this can be expressed mathematically as the maximal number of linearly independent rows/columns of  $M$  or as the **dimension of the vector space** spanned by the rows/columns of  $M$  where a **vector space** is defined as a collection of vectors which may be added together and multiplied by scalars, i.e. linear operations. Vector spaces are the formal subject of linear algebra and is well characterized by its dimension, or the number of linearly independent “directions” in the space. Thus, the rank of a matrix can describe the dimension of the column space, i.e. linearly independent columns, of  $M$  as the **column rank** and the dimension of the row space, i.e. linearly independent rows, of  $M$  as the **row rank**. A fundamental theorem of linear algebra is that column rank is equal to row rank. Lastly,  $M$  is said to have **full rank** if its rank equals the lesser of the number of rows and columns, i.e.  $\text{rank}(M) = \min(m, n)$ . Conversely,  $M$  is said to be **rank deficient** if it does not have full rank.

### State Controllability for Linear State-Space Systems

**State controllability** is defined as the ability to control a system to *any* desired state using some finite input. Related to this is the concept of **state reachability** and can be stated as follows. For continuous-time dynamical systems, a state  $x^*$  is reachable if for every finite  $T > 0$ , there exists an input function  $u(t)$  with  $0 < t \leq T$  such that the state goes from  $x(0) = 0 \rightarrow x(T) = x^*$ . Similarly, for discrete-time dynamical systems, a state  $x^*$  is reachable if for every finite  $N > 0$ , there exists an input function  $u_k$  with  $0 < k \leq N$  such that the state goes from  $x_0 = 0 \rightarrow x_N = x^*$ . Thus, reachability is generally a slightly weaker notion than controllability. However, for linear state-space systems, the sequence for reaching any state can be inverted to return to zero from any initial conditions, thus state reachability is equivalent to state controllability. Using linear algebra, one can look at the conditions for state reachability which will imply state controllability for continuous- and discrete-time LTI state-space systems.

Consider the state equation for a continuous-time LTI state-space system

$$\vec{x}(t) = A\vec{x}(t) + B\vec{u}(t) \quad (8.108)$$

Given initial state  $\vec{x}(0) = 0$ , recall the general state-space solution as

$$\vec{x}(t) = e^{At}\vec{x}(0) + \int_0^t e^{A(t-\tau)}B\vec{u}(\tau)d\tau \quad (8.109)$$

which becomes

$$\vec{x}(t) = \int_0^t e^{A(t-\tau)}B\vec{u}(\tau)d\tau \quad (8.110)$$

using a change of variables of  $\tau_2 = t - \tau$

$$\vec{x}(t) = \int_0^t -e^{A(\tau_2)}B\vec{u}(t - \tau_2)d\tau_2 \quad (8.111)$$

and using the **Cayley-Hamilton definition** of the matrix exponential, i.e.

$$e^{At} = \sum_{i=0}^{n-1} A^i \alpha_i(t) \quad (8.112)$$

where  $\alpha_0, \dots, \alpha_{n-1}$  are coefficients which depend on  $A$ , we have

$$\vec{x}(t) = \int_0^t \sum_{i=0}^{n-1} -A^i \alpha_i(\tau_2) B\vec{u}(t - \tau_2)d\tau_2 \quad (8.113)$$

which can be rearranged using the properties of integrals and sums

$$\vec{x}(t) = \sum_{i=0}^{n-1} A^i B \int_0^t -\alpha_i(\tau_2) \vec{u}(t - \tau_2)d\tau_2 \quad (8.114)$$

Then, by letting  $\beta_i(t) = \int_0^t -\alpha_i(\tau_2) \vec{u}(t - \tau_2)d\tau_2$ , we have

$$\vec{x}(t) = \sum_{i=0}^{n-1} A^i B \beta_i(t) \quad (8.115)$$

and the sum can be written out explicitly as

$$\vec{x}(t) = B\beta_0(t) + AB\beta_1(t) + \dots + A^{n-1}B\beta_{n-1}(t) \quad (8.116)$$

which can also be separated into the product of two matrices

$$\vec{x}(t) = [B \ AB \ \dots \ A^{n-1}B] \begin{bmatrix} \beta_0(t) \\ \beta_1(t) \\ \vdots \\ \beta_{n-1}(t) \end{bmatrix} \quad (8.117)$$

Finally, since any control input  $u(t)$  is allowed, any  $\begin{bmatrix} \beta_0(t) \\ \beta_1(t) \\ \vdots \\ \beta_{n-1}(t) \end{bmatrix}$  can be constructed, thus any  $\vec{x}(t)$  can be reached if and only if the **controllability matrix**  $[B \ AB \ \cdots \ A^{n-1}B]$  has full rank, i.e. its dimension “spans” the state dimension  $n$ .

Similarly, consider the state equation for a discrete-time LTI state-space model

$$\vec{x}[k+1] = F\vec{x}[k] + G\vec{u}[k] \quad (8.118)$$

Given an initial state  $\vec{x}_0 = 0$ , the solution at each  $k$  can be found by iterating through the equation above, i.e.

$$\begin{aligned} \vec{x}[1] &= F\vec{x}[0] + G\vec{u}[0] = G\vec{u}[0] \\ \vec{x}[2] &= F\vec{x}[1] + G\vec{u}[1] = G\vec{u}[1] + FG\vec{u}[0] \\ &\vdots \\ \vec{x}[k] &= G\vec{u}[k-1] + FG\vec{u}[k-2] + \dots + F^{n-1}G\vec{u}[0] \end{aligned} \quad (8.119)$$

This general formula can be rewritten as the product of two matrices as

$$\vec{x}[k] = [G \ FG \ \cdots \ F^{n-1}G] \begin{bmatrix} \vec{u}[k-1] \\ \vec{u}[k-2] \\ \vdots \\ \vec{u}[0] \end{bmatrix} \quad (8.120)$$

which contains the same controllability matrix as before and therefore the same logic holds that any  $\vec{x}[k]$  can be reached if and only if  $[G \ FG \ \cdots \ F^{n-1}G]$  has full rank (i.e.  $n$ ).

A simpler, but less robust methods for identifying the controllability of an LTI state-space system can be done by inspecting the JCF of the LTI state-space system, i.e.

$$\dot{\vec{z}}(t) = \Lambda\vec{z}(t) + \bar{B}\vec{u}(t) \quad (8.121)$$

or

$$\dot{\vec{z}}[k+1] = \Lambda\vec{z}[k] + \bar{G}\vec{u}[k] \quad (8.122)$$

where

$$\vec{x} = V\vec{z} \quad (8.123)$$

where  $V$  is the matrix of  $n$  generalized eigenvectors of  $A$  or  $F$  and  $\Lambda$  is a Jordan matrix populated with the eigenvalues of  $A$  or  $F$ . If  $A$  or  $F$  is diagonalizable, then for every row of  $\bar{B} = V^{-1}B$  or  $\bar{G} = V^{-1}G$  which has only zeros, that state is uncontrollable since each element of  $\vec{z}$  is independent in the JCF due to the diagonal structure of  $\Lambda$ . However, when  $A$  or  $F$  is not diagonalizable, this visual inspection can be more difficult to use.

While the previous two tests provide a binary test for controllability, an additional test can assess the relative controllability of a system compared with another system, similar to how relative eigenvalues

provide information on the relative stability. This method uses the **controllability Gramian**, which for continuous-time can be written as

$$W(t) = \int_0^t e^{A\tau} BB^T e^{A\tau} d\tau \quad (8.124)$$

and for discrete-time as

$$W[k] = \sum_{i=0}^k F^i G G^T F^i \quad (8.125)$$

It can be shown that if and only if  $W(t)$  or  $W[k]$  is nonsingular for *any*  $t > 0$  or  $k > 0$ , respectively, then the system is controllable. The two equations above can be reduced to solving the equation for continuous-time as

$$AW + WA^T = -BB^T \quad (8.126)$$

and for discrete-time as

$$W - FWF^T = GG^T \quad (8.127)$$

then checking if  $W$  is positive definite. Note that  $W$  is an  $n \times n$  matrix, thus once one has solved for  $W$ , additional controllability assessment can be done by looking at the eigenvalue decomposition of  $W$ .

Furthermore, for continuous-time LTV systems, one can extend these results by defining the controllability Gramian using the state-transition matrix as

$$W(t_0, t_1) = \int_{t_0}^{t_1} \Phi(t_0, t) B(t) B(t)^T \Phi(t_0, t)^T dt \quad (8.128)$$

where  $W(t_0, t_1)$  is symmetric, positive semi-definite, and satisfies

$$W(t_0, t_1) = W(t_0, t) + \Phi(t_0, t) W(t, t_1) \Phi(t_0, t)^T \quad (8.129)$$

which is similar to the discrete-time case. For discrete-time LTV systems, one can use dependence on  $k$  in the rank condition matrix and/or the Grammian summation directly.

In addition to state controllability, there are three other common notions of controllability. The first is **output controllability** which is the ability of an input to move the *output* from any initial condition to any final condition in finite time. This type of analysis naturally involves the output matrix in addition to the input matrix. It is also important to point out that state and output controllability are not equivalent nor does one imply the other. The second form is **stabilizability**. A system is said to be stabilizable if all *uncontrollable* state variables can be made to have stable dynamics, thus this characteristic is a weaker statement than state controllability since only naturally unstable states must be controllable. The third form is **controllability under constraints** which may be imposed upon practical systems modeled as an LTI system. Such constraints may be inherent to the system, e.g. saturating actuator, or imposed by the control designer, e.g. due to safety-related concerns. The effect of constraints to a system is a vast larger topic in control and is introduced in nonlinear optimal control.

## State Observability for Linear State-Space Systems

**State observability** is defined as: if, for some finite  $T > 0$ , inputs  $\vec{u}(t)$ , and outputs  $\vec{y}(t)$  with  $0 < t \leq T$ , the initial state  $\vec{x}(0)$  can be determined, then one can observe the system's *past* initial state. One can

quantify this concept without loss of generality given the output sequence  $\vec{y}[k]$  for  $k = 0, 1, \dots, n - 1$ , and the simplified discrete-time LTI output equation

$$\vec{y}[k] = H\vec{x}[k] \quad (8.130)$$

since if  $D \neq 0$  and one has the input sequence  $\vec{u}[k]$  for  $k = 0, 1, \dots, n - 1$ , one could form a secondary output  $\vec{y}' = \vec{y} - D\vec{u}$  instead. Thus, one has the following sequence of output equations

$$\vec{y}[0] = H\vec{x}[0] \quad (8.131)$$

$$\vec{y}[1] = H\vec{x}[1] = H(F\vec{x}[0] + G\vec{u}[0]) = HF\vec{x}[0] + HG\vec{u}[0] \quad (8.132)$$

and

$$\vec{y}[2] = H\vec{x}_2 = H(F\vec{x}[1] + G\vec{u}[1]) \quad (8.133)$$

or more simply

$$\vec{y}[2] = HF(F\vec{x}[0] + G\vec{u}[0]) + HG\vec{u}[1] \quad (8.134)$$

$$\vec{y}[2] = HF^2\vec{x}[0] + HFG\vec{u}[0] + HG\vec{u}[1] \quad (8.135)$$

Thus, by extension one has

$$\vec{y}[n - 1] = HF^{n-1}\vec{x}[0] + HF^{n-2}G\vec{u}[0] + \cdots + HG\vec{u}[n - 2] \quad (8.136)$$

and rearranging into matrix

$$\begin{bmatrix} \vec{y}[0] \\ \vec{y}[1] \\ \vdots \\ \vec{y}[n - 1] \end{bmatrix} - \begin{bmatrix} 0 & 0 & \cdots & 0 \\ HG & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ HF^{n-2}G & HF^{n-3}G & \cdots & HG \end{bmatrix} \begin{bmatrix} \vec{u}[0] \\ \vec{u}[1] \\ \vdots \\ \vec{u}[n - 2] \end{bmatrix} = \begin{bmatrix} H \\ HF \\ \vdots \\ HF^{n-1} \end{bmatrix} \vec{x}[0] \quad (8.137)$$

which has the form

$$\bar{y} = \bar{H}\vec{x}[0] \quad (8.138)$$

Thus, by inspection one can see that *any*  $\vec{x}_0$  can be observed if and only if  $\bar{H}$  has full row rank, i.e.  $\text{rank } \bar{H} = n$ . The initial state can thus be computed by

$$\vec{x}_0 = (\bar{H}^T \bar{H})^{-1} \bar{H}^T \bar{y} \quad (8.139)$$

also known as the **pseudoinverse** of  $\bar{H}$  which exists if  $\bar{H}$  has full row rank.

In addition, one can consider the continuous-time LTI output equation

$$\vec{y}(t) = C\vec{x}(t) \quad (8.140)$$

Analogous to the discrete output sequence, one must consider the continuous output derivatives

$$\begin{aligned}
 \vec{y}(0) &= C\vec{x}_0 \\
 \dot{\vec{y}}(0) &= C\dot{\vec{x}}(0) = C(A\vec{x}_0 + B\vec{u}_0) \\
 \ddot{\vec{y}}(0) &= C\ddot{\vec{x}}(0) = C \frac{d}{dt} (A\vec{x}_0 + B\vec{u}_0) \\
 \ddot{\vec{y}}(0) &= CA(A\vec{x}_0 + B\vec{u}_0) + CB\dot{\vec{u}}_0 \\
 &\vdots = \vdots \\
 \vec{y}^{n-1}(0) &= CA^{n-1}G\vec{x}_0 + CA^{n-2}B\vec{u}^{n-2}(0) + \cdots CB\dot{\vec{u}}(0)
 \end{aligned} \tag{8.141}$$

and by stacking each of these one can form (same as discrete-time)

$$\bar{y}(0) = \bar{H}\vec{x}(0) \tag{8.142}$$

where  $\bar{H}$  is the same observability matrix. It is important to note that one cannot increase the rank of  $\bar{H}$  for  $k \geq n$ . Lastly, a weaker notion than observability is **detectability** where only the *unstable* states must be observable.

Lastly, one can use the **observability Gramian** to determine the observability for continuous-time LTI state-space systems as

$$W(t) = \int_0^t e^{A\tau} CC^T e^{A\tau} d\tau \tag{8.143}$$

and in discrete-time as

$$W[k] = \sum_{i=0}^k \left(F^T\right)^i H^T H F^i \tag{8.144}$$

which must be nonsingular for any  $t > 0$  or  $k > 0$  in order for the system to be *observable*. The Gramian can be shown to be the solution in continuous-time for the equation

$$AW + WA^T = -CC^T \tag{8.145}$$

and in discrete-time for the equation

$$F^T W F - W = -H^T H \tag{8.146}$$

where the values of the  $n \times n$  Gramian characterize the relative degree of observability.

Furthermore, for continuous-time LTV systems, one can extend these results by defining the observability Gramian using the state-transition matrix as

$$W(t_0, t_1) = \int_{t_0}^{t_1} \Phi(t, t_0)^T C(t)^T C(t) \Phi(t, t_0) dt \tag{8.147}$$

where  $W(t_0, t_1)$  is symmetric, positive semi-definite, and satisfies

$$W(t_0, t_1) = W(t_0, t) + \Phi(t, t_0)^T W(t, t_1) \Phi(t, t_0) \tag{8.148}$$

which is similar to the discrete-time case. For discrete-time LTV systems, one can use dependence on  $k$  in the rank condition matrix and/or the Grammian summation directly.

## 8.5 Uncertain Dynamical Systems and Random Variables

In the study of dynamical systems, there is typically some level of uncertainty in the mathematical rule for dynamical system. Furthermore, this uncertain dynamical system may be considered as a **deterministic**, i.e. from a given initial state, the same inputs will produce the same output, or a **random dynamical system**, i.e. from a given initial state, the same inputs may produce different outputs. Random dynamical systems are also sometimes referred to as **stochastic dynamical systems** although stochastic technically refers to the modeling approach and the random refers to the phenomena, but these two terms are often used interchangeably. For modeling deterministic systems, one can use any of the representations used previously in this textbook. However, for such random dynamical systems it is typical to use a **stochastic state-space representation** which can be defined for continuous-time as

$$\begin{aligned} d\vec{x}(t) &= f(\vec{x}(t), \vec{u}(t), d\vec{w}(t), dt) \\ d\vec{y}(t) &= h(d\vec{x}(t), d\vec{v}(t), dt) \end{aligned} \quad (8.149)$$

and for discrete-time as

$$\begin{aligned} \vec{x}_k &= f(\vec{x}_{k-1}, \vec{u}_{k-1}, \vec{w}_k, k) \\ \vec{y}_k &= h(\vec{x}_k, \vec{v}_k, k) \end{aligned} \quad (8.150)$$

where  $\vec{w}$  and  $\vec{v}$  are the **process noise** and the **measurement noise** or **observation noise**, respectively. Notably, one can also define a **linear stochastic state-space representation** for continuous-time as

$$\begin{aligned} d\vec{x}(t) &= A(t)\vec{x}(t)dt + B(t)\vec{u}(t)dt + d\vec{w}(t) \\ d\vec{y}(t) &= C(t)d\vec{x}(t) + d\vec{v}(t) \end{aligned} \quad (8.151)$$

and for discrete-time as

$$\begin{aligned} \vec{x}_k &= F_k \vec{x}_{k-1} + G_k \vec{u}_{k-1} + \vec{w}_k \\ \vec{y}_k &= H_k \vec{x}_k + \vec{v}_k \end{aligned} \quad (8.152)$$

Lastly, it should also be noted that a special type of random dynamical system is a **tychastic dynamical system** in which the system uncertainty enters through a set of random parameters,  $\vec{p}$ , which, if it was known, one would have a deterministic dynamical system. In this case, the **tychastic state-space representation** can be defined for continuous-time as

$$\begin{aligned} \vec{x}(t) &= f(\vec{x}(t), \vec{u}(t), t; \vec{p}) \\ \vec{y}(t) &= h(\vec{x}(t), t) \end{aligned} \quad (8.153)$$

and for discrete-time as

$$\begin{aligned} \vec{x}_k &= f(\vec{x}_{k-1}, \vec{u}_{k-1}, k; \vec{p}) \\ \vec{y}_k &= h(\vec{x}_k, k) \end{aligned} \quad (8.154)$$

Thus, to discuss how one can use randomness in dynamical systems, one requires the use of random variables, a fundamental topic in probability theory which will be introduced in this section. A **random variable** takes on values which depend on the outcomes of a random phenomenon. Thus, a random variable is a function that must be **measurable**, i.e. probabilities of occurrence can be assigned to sets of its potential values known as **events**. From a frequentist viewpoint of probability, the **outcomes** that a random variable depends on can be defined as all possible results of a yet-to-be performed experiment. To introduce these concepts, the next subsection will introduce discrete random variables.

## Discrete Random Variables

A **discrete random variable** depends on outcomes that take on any value in a **countable set**, i.e. one can create a rule that “assigns” a value of the natural numbers 1, 2, 3, ... to each value in the set. Thus, any **finite set**, i.e. a set with a finite number of elements  $n$ , is countable as one can assign those  $n$  elements to the first  $n$  natural numbers. However, one can also define a **countably infinite set** for a set that is countable, but also has infinite elements. For example, the set of integers and the set of rational numbers are countably infinite sets as one can use an explicit pattern to assign the values, e.g. for integers by assigning a rule that switches between positive and negative numbers and for rational numbers by assigning a rule based on a pattern of numerators and denominators.

As an example, consider rolling two dice. This experiment would have the following outcomes:

$$\begin{aligned}
 & (1, 1) \\
 & (1, \cdot) \\
 & (1, 6) \\
 & (2, 1) \\
 & (2, \cdot) \\
 & (2, 6) \\
 & (\cdot, \cdot) \\
 & (6, 6)
 \end{aligned} \tag{8.155}$$

which mathematically provides thirty-six possible outcomes for this experiment. As a random variable, consider the sum of the two dice, and the event  $A$  would be the sum of two dice being equal to 2, which has one outcome. Similarly, the event  $B$  could be assigned as the sum of dice begin equal to 3 which has two outcomes. This could be continued for all eleven possible events. To assign probabilities to these events from a frequentist view, one could perform the experiment of rolling two dice  $N$  times. Then, forming ratios of occurrences of the 11 events  $A-K$

$$\frac{N_A}{N}, \frac{N_B}{N}, \dots \tag{8.156}$$

one can assign the probabilities of each event as

$$Pr(A) = \lim_{N \rightarrow \infty} \frac{N_A}{N}, Pr(B) = \lim_{N \rightarrow \infty} \frac{N_B}{N}, \dots \tag{8.157}$$

Note that if the two dice are **fair**, i.e. each outcome is equally probable, then  $Pr(A) = \frac{1}{36}$ ,  $Pr(B) = \frac{2}{36} = \frac{1}{18}$ , ... .

This textbook will represent random variables by the uppercase letters, e.g.  $X$ , and the values or the **realizations**, by lowercase letters, e.g.  $x$ . Thus, if the random variable is equal to some value, one write  $X = x$  which differentiates between random variables and their particular realizations. For assigning probabilities to discrete random variables one typically uses **probability mass functions (PMFs)** defined as

$$p_X(x) = Pr(X = x) \tag{8.158}$$

which must also satisfy that the probabilities of every possible event sum up to 100%, i.e.

$$\sum p_X(x) = 1 \quad \forall x \quad (8.159)$$

thus, an event must always occur.

Some common probability distributions for discrete random variables include the following. The **Bernoulli distribution**, represented by  $X \sim Ber(p)$ , has a PMF of the form

$$p_X(x) = \begin{cases} p & \text{if } x = 1 \\ 1 - p & \text{if } x = 0 \end{cases} \quad (8.160)$$

and can be used to model experiments with only two events, e.g. a coin flip, but can easily be extended to a **Multi-Bernoulli distribution** with multiple events. The **Binomial distribution**, represented by  $X \sim Bin(n, p)$ , has a PMF of the form

$$p_X(x) = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k} \quad (8.161)$$

and can be used to model the number of  $n$  successes in an experiment performed  $k$  times where each event can reoccur. Lastly, the Poisson random variable, represented by  $X \sim Pois(\lambda)$ , has a PMF of the form

$$p_X(x) = \frac{\lambda^x e^{-\lambda}}{x!} \quad (8.162)$$

with  $\lambda > 0$  and can be used to model the number of events,  $x$ , that may occur in some time interval.

## Continuous Random Variables

A **continuous random variable** depends on outcomes that take on any value in an uncountable set, thus one cannot “assign” probabilities to exact values and the probability of an exact value  $x$  for a continuous random variable  $X$  is zero. However, one can assign probabilities to *ranges* of values as the event which, for continuous random variables, one typically uses **cumulative distribution functions (CDFs)** defined as

$$F_X(x) = \Pr(X \leq x) \quad (8.163)$$

From the properties of probabilities, the CDF is always a monotonically non-decreasing function, i.e. the accumulated probabilities as one moves from  $-\infty$  to  $\infty$  can never go down as an event cannot have a “negative” probability. Furthermore, the limits of the CDF must be

$$\lim_{x \rightarrow -\infty} F_X(x) = 0 \quad (8.164)$$

and

$$\lim_{x \rightarrow \infty} F_X(x) = 1 \quad (8.165)$$

thus, as one approaches  $-\infty$ , the probability that  $X \leq -\infty$  should vanish and as one approaches  $\infty$ , the probability that  $X \leq \infty$  should approach 100%. Furthermore, using the CDF, one can form the probability for  $X$  taking any realization between values  $a$  and  $b$  as

$$\Pr(a < X \leq b) = F_X(b) - F_X(a) \quad (8.166)$$

For absolutely continuous CDFs, one also typically considers the **probability density function (PDF)** of a continuous random variable defined as

$$f_X(x) = \frac{dF(x)}{dx} \quad (8.167)$$

or

$$F_X(x) = \int_{-\infty}^x f_X(\zeta) d\zeta \quad (8.168)$$

where one can intuitively think of the PDF as an infinitesimal PMF, i.e.

$$f_X(x) = \Pr(x < X \leq x + dx) \quad (8.169)$$

Two common probability distributions for continuous random variables include the following. The **Uniform distribution**, represented by  $X \sim \mathcal{U}(a, b)$ , has a PDF of the form

$$f_X(x) = \begin{cases} \frac{1}{b-a} & x \in [a, b] \\ 0 & \text{otherwise} \end{cases} \quad (8.170)$$

and can be used to model events when all values in the range  $[a, b]$  are equally likely. The **Gaussian distribution**, also known as the **Normal distribution**, represented by  $X \sim \mathcal{N}(\mu, \sigma)$ , has a PDF of the form

$$f_X(x) = \frac{1}{\sigma_x \sqrt{2\pi}} \exp \left[ -\frac{1}{2} \frac{(x - \mu)^2}{\sigma_x^2} \right] \quad (8.171)$$

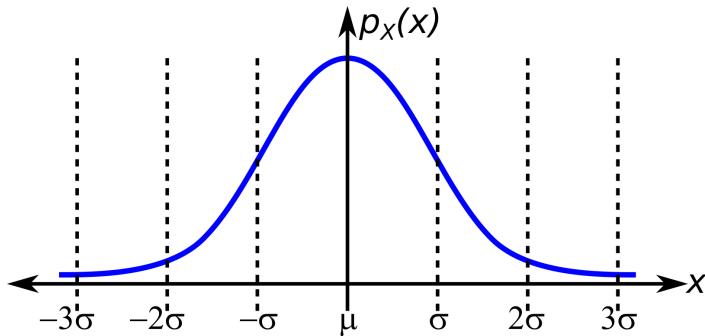
and is used extensively in probability modeling as it exhibits some convenient mathematical properties. One can also define the **standard normal distribution** where  $\mu = 0$  and  $\sigma = 1$  which provides the simpler PDF

$$f_Y(y) = \frac{1}{\sqrt{2\pi}} \exp \left[ -\frac{1}{2} y^2 \right] \quad (8.172)$$

and thereby rewrite  $f_X(x)$  in terms of  $f_Y(y)$  as

$$f_X(x) = \frac{1}{\sigma} f_Y \left( \frac{x - \mu}{\sigma} \right) \quad (8.173)$$

The Gaussian random variable is also known as the “bell curve,” due to the shape of its PDF as shown below.



## Statistics of Random Variables

Furthermore, one often desires to compute some characteristic of the values a random variable may take, known as a **statistic**. There are many different types of statistics; however, many common statistics are defined using the expectation operator  $E[.]$ , which for the random variable  $X$  is defined as

$$E[X] = \int_{-\infty}^{\infty} x f_X(x) dx \quad (8.174)$$

but can also be computed for functions of random variables. One of the most common functions are the **moments of a random variable**, where the  $i^{\text{th}}$  moment of a random variable is defined as

$$E[X^i] = \int_{-\infty}^{\infty} x^i f_X(x) dx \quad (8.175)$$

and the  $i^{\text{th}}$  **central moment of a random variable** is defined as

$$E[(X - \bar{x})^i] = \int_{-\infty}^{\infty} (x - \bar{x})^i f_X(x) dx \quad (8.176)$$

The  $i^{\text{th}}$  **standardized moment of a random variable** is defined as

$$E\left[\left(\frac{X - \bar{x}}{\sigma_x}\right)^i\right] = \int_{-\infty}^{\infty} \left(\frac{x - \bar{x}}{\sigma_x}\right)^i f_X(x) dx \quad (8.177)$$

Four moments of random variables have particular names. These are the **mean**, denoted by  $\bar{x}$ , which is the first moment, the **variance**, denoted by  $\sigma_x^2$ , which is second central moment, the **skewness** which is the third standardized moment, and the **kurtosis** which is the fourth standardized moment. Three other common statistics that are used to describe random variables are the **median**, denoted by  $m_x$ , defined by the equation

$$F_X(m_x) = 0.5 \quad (8.178)$$

the **mode**, denoted by  $Mo_x$ , defined by

$$Mo_x = \operatorname{argmax}_x f_X(x) \quad (8.179)$$

and the **standard deviation**, denoted by  $\sigma_x$ , defined by

$$\sigma_x = \sqrt{\sigma_x^2} \quad (8.180)$$

As an example, consider the Gaussian random variable  $X \sim \mathcal{N}(\mu, \sigma)$ . Using the expectation integral, one can show that for a Gaussian distribution the mean of  $X$ ,  $\bar{x}$ , is the parameter  $\mu$  (also the median and mode) and the variance of  $x$ ,  $\sigma_x^2$  is the parameter  $\sigma^2$ . Furthermore, by integrating the CDF of the Gaussian distribution, one can compute that the following probabilities hold

$$\Pr(-\sigma < X \leq \sigma) = 0.68 \quad (8.181)$$

$$\Pr(-2\sigma < X \leq 2\sigma) = 0.95 \quad (8.182)$$

$$\Pr(-3\sigma < X \leq 3\sigma) = 0.997 \quad (8.183)$$

## 8.6 Random Vectors

When considering a finite collection of  $n$  random variables, one has a **random vector** with dimension  $n$  defined as

$$\vec{X} = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} \quad (8.184)$$

with a corresponding **realization vector** that can be defined as

$$\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad (8.185)$$

### Random Vector Probability Functions

To describe the probabilities and relationships between elements of a continuous random vector, one typically uses the **joint cumulative distribution function (joint CDF)**, defined as

$$F_{X_1, \dots, X_n}(x_1, \dots, x_n) = \Pr(X_1 \leq x_1, \dots, X_n \leq x_n) \quad (8.186)$$

which can be written in vector form as

$$F_{\vec{X}}(\vec{x}) = \Pr(\vec{X} \leq \vec{x}) \quad (8.187)$$

Furthermore, by taking the  $n$  partial derivatives of the joint CDF, one can define the **joint probability density function (joint PDF)** as

$$f_{\vec{X}}(\vec{x}) = \frac{\partial^n F_{\vec{X}}(\vec{x})}{\partial x_1 \cdots \partial x_n} \quad (8.188)$$

which can be considered intuitively as

$$\Pr(x_1 < X_1 \leq x_1 + dx_1, \dots, x_n < X_n \leq x_n + dx_n) \quad (8.189)$$

Reversing this relationship, one can also relate the joint CDF and joint PDF by

$$F_{\vec{X}}(\vec{x}) = \int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_n} f_{\vec{X}}(x_1, \dots, x_n) dx_1 \dots dx_n \quad (8.190)$$

where for any PDF, including a joint PDF, the following is true.

$$\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_{\vec{X}}(x_1, \dots, x_n) dx_1 \dots dx_n = 1 \quad (8.191)$$

In optimal estimation, one will often deal with functions of random vectors, in particular, invertible functions of random variables, e.g.

$$\vec{Y} = h(\vec{X}) \quad (8.192)$$

which by the definition of the CDF and derivatives, one can show that the PDF of  $\vec{Y}$  given  $\vec{X}$  is

$$f_{\vec{Y}}(\vec{y}) = \left[ \frac{f_{\vec{X}}(\vec{x})}{\left| \det \frac{\partial h(\vec{x})}{d\vec{x}} \right|} \right]_{\vec{x}=h^{-1}(\vec{y})} \quad (8.193)$$

An important type of function is the **affine transformation**

$$\vec{Y} = H\vec{X} + \vec{b} \quad (8.194)$$

which can be written as

$$f_{\vec{Y}}(\vec{y}) = \frac{f_{\vec{X}}(H^{-1}(\vec{y} - \vec{b}))}{|\det H|} \quad (8.195)$$

### Random Vector Statistics

To characterize the statistics different random vectors, one can use the expectation operator which operates on each individual random variable within the vector, i.e.

$$E[\vec{X}] = \begin{bmatrix} E[X_1] \\ \vdots \\ E[X_n] \end{bmatrix} = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \vec{x} f_{\vec{X}}(x_1, \dots, x_n) dx_1 \cdots dx_n \quad (8.196)$$

Thus, the multivariate analog for the first moment of  $\vec{X}$  is the **mean of a random vector** is simply

$$\bar{x} = \begin{bmatrix} \bar{x}_1 \\ \vdots \\ \bar{x}_n \end{bmatrix} \quad (8.197)$$

However, higher moments of random vectors, one must consider the correlation between the random variables of the vector.

For random vectors, the second moment of  $\vec{X}$  is the **correlation matrix** defined as

$$R_X = E[\vec{X}\vec{X}^T] \quad (8.198)$$

and it be can shown that

$$R_X = C_X + E[\vec{X}]E[\vec{X}]^T \quad (8.199)$$

Furthermore, if one relates the correlation for each covariance element in  $C_x$ , one has

$$\sigma_{i,j} = \rho_{i,j}\sigma_i\sigma_j \quad (8.200)$$

where  $\rho_{i,j} \in [-1, 1]$  is called the **correlation coefficient** between  $X_i$  and  $X_j$ . Thus, the elements of  $\vec{X}$  are **uncorrelated** if  $\rho_{i,j} = \sigma_{i,j} = 0$ , or in terms of expectations

$$E[X_i X_j] = E[X_i]E[X_j] \quad \forall i \neq j \quad (8.201)$$

If the elements of  $\vec{X}$  are uncorrelated for all  $i \neq j \rightarrow$ , then the correlation matrix is a diagonal matrix. Furthermore, the elements of  $\vec{X}$  are **pairwise independent** if

$$f_{X_i, X_j}(x_i, x_j) = f_{X_i}(x_i)f_{X_j}(x_j) \quad \forall i \neq j \quad (8.202)$$

It can be shown that independence implies uncorrelatedness, however, it does not hold vice versa.

In addition to the correlation matrix, the second centralized moment of  $\vec{X}$  is the **covariance matrix**, also known as the **variance-covariance matrix**, defined as

$$\begin{aligned} C_X &= E \left[ (\vec{X} - \bar{x})(\vec{X} - \bar{x})^T \right] \\ &= \begin{bmatrix} E[(X_1 - \bar{x}_1)^2] & \cdots & E[(X_1 - \bar{x}_1)(X_n - \bar{x}_n)] \\ \vdots & \ddots & \vdots \\ E[(X_n - \bar{x}_n)(X_1 - \bar{x}_1)] & \cdots & E[(X_n - \bar{x}_n)^2] \end{bmatrix} \\ &= \begin{bmatrix} \sigma_1^2 & \cdots & \sigma_{n,1} \\ \vdots & \ddots & \vdots \\ \sigma_{1,n} & \cdots & \sigma_n^2 \end{bmatrix} \end{aligned} \quad (8.203)$$

which is a symmetric positive definite matrix. The individual elements of the covariance matrix are called the variances on the diagonal, i.e.

$$\sigma_i^2 = \int_{-\infty}^{\infty} (x_i - \bar{x}_i)^2 f_{\vec{X}}(x_1, \dots, x_n) dx_1 \dots dx_n \quad (8.204)$$

and the covariances on the off-diagonal, i.e.

$$\sigma_{i,j}^2 = \int_{-\infty}^{\infty} (x_i - \bar{x}_i)(x_j - \bar{x}_j) f_{\vec{X}}(x_1, \dots, x_n) dx_1 \dots dx_n \quad (8.205)$$

It should be noted that this concept of correlation and independence can also be extended to *different* random vectors, e.g.  $\vec{X}$  and  $\vec{Y}$ . For example, the covariance matrix can be generalized to the **cross-covariance matrix** defined as

$$E \left[ (\vec{X} - \bar{x})(\vec{Y} - \bar{y})^T \right] \quad (8.206)$$

## Bayes' Rule

The **intersection** of two events,  $A$  and  $B$ , is defined as the probability that  $A$  and  $B$  occur for one experiment, i.e.

$$\Pr(A \cap B) = \lim_{N \rightarrow \infty} \frac{N_{A \& B}}{N} \quad (8.207)$$

where  $\cap$  represents the intersect operator. Similarly, the **union** of two events,  $A$  and  $B$ , is defined as the probability that  $A$  or  $B$  occur for one experiment

$$\Pr(A \cup B) = \lim_{N \rightarrow \infty} \frac{N_A + N_B - N_{A \& B}}{N} \quad (8.208)$$

which implies that

$$\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B) \quad (8.209)$$

where the third term occurs to not double count the intersection of  $A$  and  $B$ . Keeping these definitions in mind, one can define  $A$  and  $B$  to be **mutually independent events** if  $\Pr(A \cap B) = 0$ . In addition, the **conditional probability** describes the probability that an event has happened *given* that a different event has happened. This is related to the intersection of events as

$$\Pr(A \cap B) = \lim_{N \rightarrow \infty} \left( \frac{N_{A \& B}}{N_A} \right) \left( \frac{N_A}{N} \right) = \Pr(B|A)\Pr(A) \quad (8.210)$$

or as

$$\Pr(A \cap B) = \lim_{N \rightarrow \infty} \left( \frac{N_{A \& B}}{N_B} \right) \left( \frac{N_B}{N} \right) = \Pr(A|B)\Pr(B) \quad (8.211)$$

Thus, the conditional probability also leads to the following equation known as **Bayes' Rule**

$$\Pr(B|A)\Pr(A) = \Pr(A|B)\Pr(B) \quad (8.212)$$

and the **law of total probability** for events  $A, B_1, \dots, B_p$  defined as

$$\Pr(A) = \sum_{i=1}^p \Pr(A \cap B_i) = \sum_{i=1}^p \Pr(A|B_i)\Pr(B_i) \quad (8.213)$$

where one can think of the LTP as the “weighted average” of all possible  $\vec{X} = \vec{x}$  for  $\vec{Y}$ . Bayes' Rule is used in many estimation methods known collectively as **Bayesian methods** which will be introduced for optimal parameter and state estimation in this part of the textbook.

For continuous random vectors the **marginal PDF** can be defined for any element of  $\vec{X}$  as

$$f_{X_i}(x_i) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_{\vec{X}}(\vec{x}) dx_1 \cdots dx_{i-1} dx_{i+1} \cdots dx_n \quad (8.214)$$

which can also be generalized to be any sub-vector. Thus, the general marginal PDF also leads to the random vector definition for the **conditional PDF** of  $\vec{X}$  as the probability of  $\vec{X}$  given  $\vec{Y} = \vec{y}$ , i.e.

$$f_{\vec{X}|\vec{Y}}(\vec{x}|\vec{y}) = \frac{f_{\vec{X},\vec{Y}}(\vec{x},\vec{y})}{f(\vec{y})} \quad (8.215)$$

Extending Bayes' rule for continuous random variables,  $\vec{X}$  and  $\vec{Y}$ , one has

$$f_{\vec{X}|\vec{Y}=\vec{y}}(x) = \frac{f_{\vec{Y}|\vec{X}=\vec{x}}(y)f_{\vec{X}}(\vec{x})}{f_{\vec{Y}}(\vec{y})} \quad (8.216)$$

Extending the law of **law of total probability** for continuous random variables, one can relate marginal and conditional PDFs as

$$f_{\vec{Y}}(\vec{y}) = \int_{-\infty}^{\infty} f_{\vec{Y}|\vec{X}=\vec{\zeta}}(y)f_{\vec{X}}(\vec{\zeta})d\vec{\zeta} \quad (8.217)$$

Note that by using the law of total probability for  $f_{\vec{Y}}(\vec{y})$ , one also has for Bayes' Rule

$$f_{\vec{X}|\vec{Y}=\vec{y}}(x) = \frac{f_{\vec{Y}|\vec{X}=\vec{x}}(y)f_{\vec{X}}(\vec{x})}{\int_{-\infty}^{\infty} f_{\vec{Y}|\vec{X}=\vec{\zeta}}(y)f_{\vec{X}}(\vec{\zeta})d\vec{\zeta}} \quad (8.218)$$

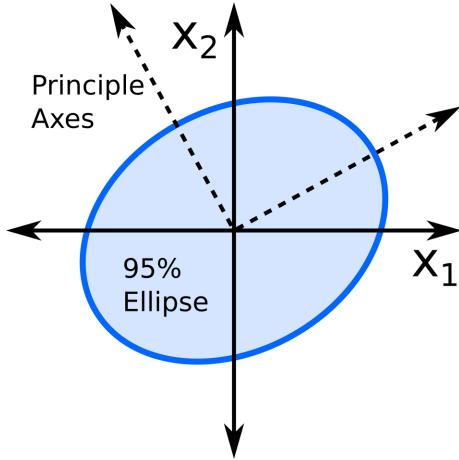
where the denominator essentially serves as a normalizing factor for the product  $f_{\vec{Y}|\vec{X}=\vec{x}}(y)f_{\vec{X}}(\vec{x})$ .

## Multivariate Gaussian Distribution

In much of optimal estimation for random vectors, one often uses the **multivariate Gaussian distribution**, i.e.  $\vec{X} \sim \mathcal{N}(\vec{\mu}, \Sigma)$ , because of its convenient mathematical properties. The joint PDF of the multivariate Gaussian is defined as

$$f_{\vec{X}}(\vec{x}) = \frac{1}{(2\pi)^{n/2} \det(\Sigma^{0.5})} \exp \left[ -\frac{1}{2} (\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu}) \right] \quad (8.219)$$

where  $\vec{\mu}$  is the mean and  $\Sigma$  is the covariance and completely characterize the random vector. The PDF can be integrated to obtain the joint Gaussian CDF for which there is no analytical solution. Typically this integration is approximated using numerical methods. In 2D, one can represent the confidence region as an **error ellipse**



Furthermore, by using the eigendecomposition of the covariance, i.e.

$$\Sigma = U \Lambda^{1/2} (U \Lambda^{1/2})^T \quad (8.220)$$

one can write the multivariate Gaussian, i.e.

$$\vec{X} \sim \mathcal{N}(\vec{\mu}, \Sigma) \quad (8.221)$$

as

$$X \sim \vec{\mu} + U \mathcal{N}(0, \Lambda) \quad (8.222)$$

or

$$X \sim \vec{\mu} + U \Lambda^{1/2} \mathcal{N}(0, I) \quad (8.223)$$

where  $U$  also defines the principle axes of the error ellipse or hyper-ellipse for  $n > 2$ .

One useful property of the multivariate Gaussian is that Gaussian marginal PDFs are also Gaussian distributed and can be formed by simply dropping the unnecessary elements from  $\vec{\mu}$  and  $\Sigma$ . As an example, consider the 2-dimensional case, i.e.

$$\vec{X} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \quad (8.224)$$

with

$$\vec{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \quad (8.225)$$

and

$$\Sigma = \begin{bmatrix} \sigma_1 & \sigma_{1,2} \\ \sigma_{1,2} & \sigma_2 \end{bmatrix} \quad (8.226)$$

Then, the marginal distribution of  $X_1 \sim \mathcal{N}(\mu_1, \sigma_1)$  where one has dropped  $\mu_2, \sigma_2, \sigma_{1,2}$  and the marginal distribution of  $X_2 \sim \mathcal{N}(\mu_2, \sigma_2)$  where one has dropped  $\mu_1, \sigma_1, \sigma_{1,2}$ .

A second useful property of the multivariate Gaussians is that independence and uncorrelatedness are equivalent, i.e. one implies the other. This can be shown by assuming

$$E[X_i X_j] = E[X_i] E[X_j] \quad \forall i \neq j \quad (8.227)$$

and reducing this to

$$f_{X_i, X_j}(x_i, x_j) = f_{X_i}(x_i) f_{X_j}(x_j) \quad \forall i \neq j \quad (8.228)$$

A third useful property of random vectors is for independent multivariate Gaussians  $\vec{X} \sim \mathcal{N}(\vec{\mu}_X, \Sigma_X)$  and  $\vec{Y} \sim \mathcal{N}(\vec{\mu}_Y, \Sigma_Y)$ , their sum defined as

$$\vec{Z} = \vec{X} + \vec{Y} \quad (8.229)$$

can be shown to be distributed according to

$$\vec{Z} \sim \mathcal{N}(\vec{\mu}_X + \vec{\mu}_Y, \Sigma_X + \Sigma_Y) \quad (8.230)$$

However, it should be noted that this is *not* true for dependent Gaussians. Similarly, the affine transformation for multivariate Gaussians is simply

$$\vec{Y} \sim \mathcal{N}(\vec{b} + H\vec{\mu}, H\Sigma H^T) \quad (8.231)$$

Lastly, a particularly relevant statistic for the multivariate Gaussian is the **Mahalanobis distance**, defined as

$$D_M = \sqrt{(\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu})} \quad (8.232)$$

which can be used to test probabilistic distance between a value  $\vec{x}$  and  $\vec{\mu}$ . This is typically used in conjunction with the **confidence region** which consists of all vectors,  $\vec{x}$ , which satisfy

$$(\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu}) \leq \chi_n^2(p) \quad (8.233)$$

where  $\chi_n^2$  is the chi-squared CDF with  $n$  degrees of freedom at the probability  $p$  where  $n$  is the dimension of  $\vec{x}$ . Using this with  $D_M$  is called **confidence testing**, i.e. the probability that  $\vec{x}$  is “close” to  $\mu$ .

## 8.7 Random Processes and Sequences

When considering a collection of random variables (or vectors) indexed by some mathematical set, one has a **random process**, also known as a **stochastic process** defined as

$$\{X_t : t \in T\} \quad (8.234)$$

where each random variable is uniquely associated with an element in the **index set**,  $T$ , of the random variables where  $T$  is typically *time*. Furthermore, the values that these random variables can take are known as the **state space**,  $S$ , which is different than the state-space representation for dynamical systems and can be either a continuous or discrete state space. Another important distinction in random processes is the **cardinality classification** where the random process is called continuous-time if  $T$  is an uncountable index set and discrete-time if  $T$  is a countable index set. A discrete-time random process is also known as a **random sequence**.

### Random Process Probability Functions

A **sample** of a random process,  $X$ , is a *single* outcome at  $t$ . For time-indexed processes, a sample is also known as a **time sample**. In contrast, the outcome of the combined sample through the index set  $T$  is the **realization** of the random process. For time-indexed processes, a realization is also known as the **sample path** or **sample trajectory**. When discussing random sequences, the difference in outcome between two random variables of the same random sequence is called the **increment** whose probabilities and statistics describe how a random process can change over a certain time period. For any times  $t_1 \leq t_2$ , the increment is represented by the difference in the indexed random variables as  $X_{t_2} - X_{t_1}$ .

To describe the probabilities and relationships between elements of a random process, one typically uses the **joint CDF** of random variables defined for continuous-time random processes as

$$F_{X_{t_1}, \dots, X_{t_n}}(x_1, \dots, x_n) \quad (8.235)$$

as well as the **joint PDF** defined as

$$f_{X_{t_1}, \dots, X_{t_n}}(x_1, \dots, x_n) \quad (8.236)$$

For the discrete-time random process, one can define the **joint PMF**

$$p_{X_{t_1}, \dots, X_{t_n}}(x_1, \dots, x_n) = \Pr(X_1 = x_1, \dots, X_n = x_n) \quad (8.237)$$

### Random Process Statistics

To characterize the statistics of a random process, one can use the **expectation function** or **mean function** is

$$E[X_t] = m_{X_t} = \sum_{-\infty}^{\infty} x p_{X_{t_1}, \dots, X_{t_n}}(x_1, \dots, x_n) dx \quad (8.238)$$

The **auto-correlation function** between any two time samples is defined as

$$R_X(t_1, t_2) = E[X_{t_1} X_{t_2}] \quad (8.239)$$

and it should be noted that for any  $t$ ,  $t_1$ , and  $t_2$

$$R_X(t, t) = E[X_t^2] \geq 0 \quad (8.240)$$

$$|R_X(t_1, t_2)| \leq \sqrt{E[X_{t_1}^2] E[X_{t_2}^2]} \quad (8.241)$$

The **auto-covariance function** between any two time samples is defined as

$$C_X(t_1, t_2) = E[(X_{t_1} - m_{X_{t_1}})(X_{t_2} - m_{X_{t_2}})] \quad (8.242)$$

and it can be shown that

$$C_X(t_1, t_2) = R_X(t_1, t_2) - m_{X_{t_1}} m_{X_{t_2}} \quad (8.243)$$

For any two random processes,  $X_t$  and  $Y_t$ , the **cross-correlation function** between two time samples is defined as

$$R_{X,Y}(t_1, t_2) = E[X_{t_1} Y_{t_2}] \quad (8.244)$$

and the **cross-covariance function** between two time samples is represented by

$$C_{X,Y}(t_1, t_2) = E[(X_{t_1} - m_{X_{t_1}})(Y_{t_2} - m_{Y_{t_2}})] \quad (8.245)$$

or

$$C_{X,Y}(t_1, t_2) = R_{X,Y}(t_1, t_2) - m_{X_{t_1}} m_{Y_{t_2}} \quad (8.246)$$

Two random processes,  $X_t$  and  $Y_t$ , with the same index set,  $T$ , are called **independent** if  $\forall n \in \mathbb{N}$  and every choice of  $t_1, \dots, t_n \in T$ ,  $[X_{t_1} \dots X_{t_n}]$  and  $[Y_{t_1} \dots Y_{t_n}]$  are independent. Similarly, two random processes,  $X_t$  and  $Y_t$ , with the same index set,  $T$ , are **uncorrelated** if the cross-covariance is zero  $\forall t_1, t_2$ , i.e.

$$C_{X,Y}(t_1, t_2) = E[(X_{t_1} - E[X_{t_1}])(Y_{t_2} - E[Y_{t_2}])] = 0 \quad \forall t_1, t_2 \quad (8.247)$$

By definition of the expectation, it is clear that independence implies uncorrelatedness, however not vice versa. In addition, two random processes,  $X_t$  and  $Y_t$ , with the same index set,  $T$ , are **orthogonal** if the cross-correlation is zero  $\forall t_1, t_2$ , i.e.

$$R_{X,Y}(t_1, t_2) = E[(X_{t_1} Y_{t_2})] = 0 \quad \forall t_1, t_2 \quad (8.248)$$

Related to the concept of independence is **strict-sense stationarity** (SSS) which is a random process where each random variable in the process is identically distributed, i.e.

$$f_{X_{t_1}}(x_1) = \dots = f_{X_{t_n}}(x_n) \quad (8.249)$$

Intuitively, this means that as time passes, the probability distribution for a single sample of a stationary random process remains constant. Another stationarity concept is **wide-sense stationarity** (WSS), also known as **covariance stationarity**. If a random process,  $X_t$ , has a finite second moment  $\forall t \in T$  and the covariance of two RVs  $X_t$  and  $X_{t+h}$  depends only on increment length  $h \forall t \in T$ ,  $X_t$  is WSS. For such random processes, one can analyze the auto-covariance and auto-correlation functions as only dependent on the difference  $(t_1 - t_2)$  though they generally depend on  $(t_1, t_2)$ . Thus, the auto-correlation at  $h = 0$

is represented by  $R_X(0)$  and is known as the **average power**. This is also where the maximum of  $R_X(h)$  occurs. In addition, it should be noted that strict-sense stationarity implies wide-sense stationarity.

Lastly, one often uses the statistics of the **ensemble average** which is obtainable from multiple outcomes of a random sequence. For example, an estimate of the mean of a random process would be given by

$$\hat{m}_{X_t} = \frac{1}{N} \sum_{i=1}^N X_{t(i)} \quad (8.250)$$

where  $X_{t(i)}$  is  $i^{\text{th}}$  outcome at time step  $t$  and  $N$  is the number of sequences. Furthermore, if a process is stationary, then it is clear that  $m_X(t) = m$  for all  $t$ , and one can consider the **time average**, e.g. for the mean over time

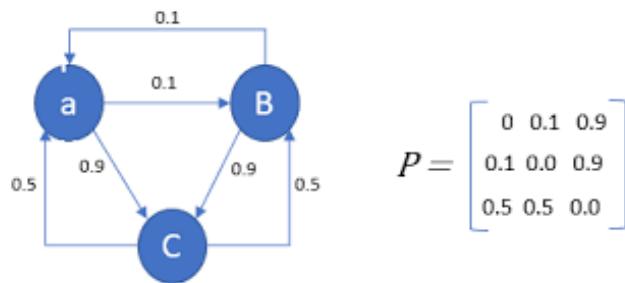
$$\langle X_t \rangle_T = \frac{1}{N} \sum_{k=1}^N X_k \quad (8.251)$$

If the time-averaged statistics converge to the true random process statistics as  $t \rightarrow \infty$ , then the random process is called **ergodic** which are a type of random process that is typically assumed in optimal estimation theory.

## Markov Property

The **Markov property** for random processes exists when the next sample of a random process depends only on the current sample. In other words, the outcome at any time is conditionally independent of past values, and future behavior conditionally depends only on the present. Thus, random processes with the Markov property are also known as a **memoryless processes**. A continuous-time random process with the Markov property is called a **Markov process** while a discrete-time random process with the Markov property is typically called a **Markov chain**. A Markov process with a *countable* state space  $S$  is also known as a **Markov jump process**.

For Markov processes or chains, the sample at any time is called the **state** and the change in the current state of the Markov process/chain is the **state transition** for which there are associated **state transition probabilities** of changing from one state to any other possible state in the state space  $S$ . For Markov chains with a finite state space  $S$ , all transition probabilities and remaining-at-current-state probabilities can be represented in a **state transition matrix**. Thus, given an initial state, one can solve for the *probability and statistics* of achieving any other state in the future



Markov processes/chains are used in many applications. Three common processes are the Bernoulli process, the Wiener process, and the random walk which are summarized below.

The **Bernoulli process** can be considered as a probability model for flipping a biased coin and is possibly the first random process studied. Its name comes from Jacob Bernoulli, one of the first mathematicians to formally study probabilities. For this model one considers a finite or infinite random sequence of independent RVs, i.e.  $X_1, X_2, X_3, \dots$  where the value of  $X_i$  is either 0 or 1  $\forall i$  and the probability that  $X_i = 1$  is the same  $\forall i$ , i.e. a random sequence of IID Bernoulli trials.

The **Wiener process**, also known as **Brownian Motion**, is a continuous time random process,  $W(t)$ , with the following properties:

- $W(0) = 0$
- $W(t)$  has independent increments, i.e. for every  $t$

$$W(t+u) - W(t), \quad u \geq 0 \quad \text{independent of} \quad W(s), \quad s \leq t \quad (8.252)$$

- $W(t)$  has Gaussian increments, i.e.

$$W(t+u) - W(t) \sim \mathcal{N}(0, u) \quad (8.253)$$

- $W(t)$  continuous in  $t$

From these properties one can also write that for  $0 \leq s_1 < t_1 \leq s_2 < t_2$ ,  $W(t_1) - W(s_1)$  and  $W(t_2) - W(s_2)$  are independent random variables. Thus, the Wiener process is used for modeling random processes with independent increments. The Wiener process has the unconditional PDF

$$f_{W(t)}(x) = \frac{1}{\sqrt{2\pi t}} \exp\left(-\frac{x^2}{2t}\right) \quad (8.254)$$

Its mean function is

$$E [W(t)] = 0 \quad (8.255)$$

Its covariance function is

$$\text{Var}(W(t)) = t \quad (8.256)$$

and its increment random variable is Gaussian, i.e.

$$W(t) = W(t) - W(0) \sim \mathcal{N}(0, t) \quad (8.257)$$

which for  $s \leq t$  has an auto-covariance function

$$\text{Cov}(W(s), W(t)) = s \quad (8.258)$$

and an auto-correlation function

$$\text{Corr}(W(s), W(t)) = \sqrt{\frac{s}{t}} \quad (8.259)$$

Related to the Wiener process is the random process called **white noise process** which is the generalized mean-square derivative of the Wiener process. It can also be thought of as the generalization of random vectors of a finite number of elements to containing infinitely many components. The mean of a white noise process does not depend on the time and is precisely equal to zero. Its auto-correlation function is

$$R_{W_t} = E [W_{k+t} W_k] \quad (8.260)$$

which only depends on  $t$  but not  $k$  and thus is nonzero only for  $t = 0$ , i.e.

$$R_{W_t} = \sigma^2 \delta_n \quad (8.261)$$

where  $\delta_n$  is the **Kronecker delta function**.

The last Markov process to be aware of in state estimation is the **random walk** which is a special Markov chain where at each step, the state can only transitions by  $\pm 1$  with equal probability. Thus, from any state, there are two possible transitions: the next integer or the previous integer. This process was first studied to prove the **gambler's ruin** problem where a gambler does the following actions

- **After win:** raise bet to fixed fraction of bankroll
- **After loss:** do not reduce bet

It can be shown using probability theory that the gambler will eventually and inevitably go broke *if* he never stops gambling. Furthermore, this fact is true even if there is a positive expected value on each bet. The rough idea of this proof is that for any starting amount of money there are two corresponding likelihoods that the gambler will either double his money or lose it all. If the gambler does double it, then there is again the two corresponding likelihoods. If the gambler never stops, then the likelihood that the gambler loses all his money approaches 1.

## Gaussian Processes

Another important random process is the **Gaussian process** which has the following property. For every finite set of indices  $t_1, \dots, t_k \in T$

$$X_{t_1, \dots, t_k} = [X_{t_1}, \dots, X_{t_k}] \quad (8.262)$$

is multivariate Gaussian random vector, i.e. every linear combination of  $X_{t_i}$  is a univariate Gaussian. These random processes are completely defined by their covariance function which commonly take one of the following forms for different applications:

- constant:  $C_X(t_1, t_2) = C$
- linear:  $C_X(t_1, t_2) = x_{t_1} x_{t_2}$
- white noise:  $C_X(t_1, t_2) = \sigma^2 \delta_{t_1, t_2}$
- Ornstein-Uhlenbeck:  $C_X(t_1, t_2) = \exp\left(-\frac{|x_{t_2} - x_{t_1}|}{\ell}\right)$

An important property of a WSS Gaussian process is that it is also guaranteed to be SSS since the covariance only depends on separation, not individual time values. It can also be shown that the Wiener process is the integral of the white noise Gaussian process with stationary increments. Lastly, the Ornstein-Uhlenbeck process is the only stationary Gaussian process and also is known as the **Brownian motion process**, or the **damped random walk**.

A random process is called a **Gauss-Markov process** if a random process  $X_t$  is both a Gaussian and a Markov process. These are required to have the three properties:

- if  $h_t$  non-zero scalar function of  $t$ , then  $Z_t = h_t X_t$  is also a Gauss-Markov process;
- if  $f_t$  non-decreasing scalar function of  $t$ , then  $Z_t = X(f_t)$  is also a Gauss-Markov process; and
- any non-degenerate mean-square continuous Gauss-Markov process can be synthesized from standard Wiener process.

# Chapter 9

## Introductory Optimal Parameter Estimation

### 9.1 Introduction to Optimal Parameter Estimation

Consider the following parameter estimation problem. Let  $\vec{y}$  be samples of a random vector  $\vec{Y}$  which depend on some deterministic, but *uncertain* parameter vector,  $\vec{\beta}$ . As such, this probabilistic relationship can be described using a conditional PDF,  $f_{\vec{Y}}(\vec{y}, \vec{\beta})$ , or as a **likelihood function**  $\mathcal{L}(\vec{\beta} | \vec{y})$  which are functionally equivalent, but view the relationships from different viewpoints. Using this information, one generally forms some mathematical law called the **parameter estimator** of  $\vec{\beta}$  based on  $\vec{y}$  and is typically denoted by  $\hat{\vec{\beta}}(\vec{y})$ . An **optimal parameter estimator** is used when one forms the mathematical law through an optimization with respect to a chosen statistic of  $f_{\vec{Y}}(\vec{y}, \vec{\beta})$ . An estimator which is optimal is also known as **efficient**.

An example of an optimal parameter estimator is the **maximum likelihood estimator (MLE)** which can be stated as

$$\hat{\vec{\beta}}^{MLE} = \underset{\vec{\beta}}{\operatorname{argmax}} \mathcal{L}(\vec{\beta} | \vec{y}) \quad (9.1)$$

which is often computed using the log-likelihood as

$$\hat{\vec{\beta}}^{MLE} = \underset{\vec{\beta}}{\operatorname{argmin}} -\ln \mathcal{L}(\vec{\beta} | \vec{y}) = \underset{\vec{\beta}}{\operatorname{argmin}} -\hat{\downarrow}(\vec{\beta} | \vec{y}) \quad (9.2)$$

The MLE exists if the Jacobian of the log-likelihood is the zero row vector. For some probability models, one can directly compute  $\hat{\vec{\beta}}^{MLE}$ , but in general no closed-form solution to the optimization problem is known and can only be found via numerical optimization. For a single observed sample,  $\vec{y}$ , the MLE has no guaranteed performance. However, the MLE has some nice properties as the number of repeated samples approaches infinity, i.e. for repeated  $\vec{y}_i$  with  $i = 1, 2, 3, \dots$ . Namely, if each sample is **independent and identically distributed (IID)**, then the combined likelihood can be written simply as a product of the conditional PDFs or likelihoods as

$$\hat{\vec{\beta}}^{MLE} = \underset{\vec{\beta}}{\operatorname{argmin}} \prod_{i=1}^N \mathcal{L}(\vec{\beta} | \vec{y}_i) \quad (9.3)$$

or, in terms of the log-likelihood, as

$$\hat{\vec{\beta}}^{MLE} = \underset{\vec{\beta}}{\operatorname{argmin}} - \frac{1}{N} \sum_{i=1}^N \hat{\downarrow}(\vec{\beta} | \vec{y}_i) \quad (9.4)$$

It can be shown that as  $N \rightarrow \infty$ ,  $\hat{\vec{\beta}}^{MLE}$  converges in probability to  $\vec{\beta}$ , a property known as **estimator consistency**. More properties of the MLE will be discussed later in this part of the textbook.

One important characteristic of any parameter estimator is the **estimator error**, i.e.  $\hat{\vec{\beta}} - \vec{\beta}$ , which is a random vector that can generally have *positive or negative* values. Thus, a common statistic for any estimator considers the expectation of the square of the error, i.e. the **mean square error (MSE)** of the estimator, defined as

$$\text{MSE}(\hat{\vec{\beta}}) = E \left[ (\hat{\vec{\beta}} - \vec{\beta})^T (\hat{\vec{\beta}} - \vec{\beta}) \right] \quad (9.5)$$

Thus, an optimal parameter estimator may be formed as

$$\hat{\vec{\beta}}^{MMSE} = \underset{\vec{\beta}}{\operatorname{argmin}} E \left[ (\hat{\vec{\beta}} - \vec{\beta})^T (\hat{\vec{\beta}} - \vec{\beta}) \right] \quad (9.6)$$

which is known as the **minimum MSE (MMSE) estimator**. For the MLE, it can be shown that as  $N \rightarrow \infty$ , the MLE becomes the MMSE.

Another statistic of an estimator is the **estimator mean**, i.e.  $E[\hat{\vec{\beta}}]$ , which may or may not match the actual parameter  $\vec{\beta}$ . Thus, one often may consider the **estimator bias**,  $\vec{b}(\hat{\vec{\beta}})$  defined as

$$\vec{b}(\hat{\vec{\beta}}) = E[\hat{\vec{\beta}}] - \vec{\beta} \quad (9.7)$$

Thus, if for some  $\hat{\vec{\beta}}$ ,  $E[\hat{\vec{\beta}}] = \vec{\beta}$ , then  $\hat{\vec{\beta}}$  is **unbiased**. A third statistic of an estimator is the **estimator variance**, i.e.

$$\text{Var}(\hat{\vec{\beta}}) = E \left[ (\hat{\vec{\beta}} - E[\hat{\vec{\beta}}])^T (\hat{\vec{\beta}} - E[\hat{\vec{\beta}}]) \right] \quad (9.8)$$

With these two characteristics in mind, an optimal parameter estimator may be formed as

$$\hat{\vec{\beta}}^{MVUE} = \underset{\substack{\text{unbiased } \vec{\beta}}}{\operatorname{argmin}} E \left[ (\hat{\vec{\beta}} - E[\hat{\vec{\beta}}])^T (\hat{\vec{\beta}} - E[\hat{\vec{\beta}}]) \right] \quad (9.9)$$

which is known as the **minimum variance unbiased estimator (MVUE)**.

Next, note that the MSE of  $\hat{\vec{\beta}}$  can be written as

$$\text{MSE}(\hat{\vec{\beta}}) = E \left[ \hat{\vec{\beta}}^T \hat{\vec{\beta}} - \hat{\vec{\beta}}^T \vec{\beta} - \vec{\beta}^T \hat{\vec{\beta}} - \vec{\beta}^T \vec{\beta} \right] \quad (9.10)$$

$$\text{MSE}(\hat{\vec{\beta}}) = E[\hat{\vec{\beta}}^T \hat{\vec{\beta}}] - E[2\hat{\vec{\beta}}^T \vec{\beta}] - E[\vec{\beta}^T \vec{\beta}] \quad (9.11)$$

or

$$\text{MSE}(\hat{\vec{\beta}}) = E[\hat{\vec{\beta}}^T \hat{\vec{\beta}}] - 2E[\hat{\vec{\beta}}^T] \vec{\beta} - \vec{\beta}^T \vec{\beta} \quad (9.12)$$

Also, note that the square of the bias can be written as

$$\vec{b}(\hat{\beta})^2 = E[\hat{\beta}]E[\hat{\beta}] - 2E[\hat{\beta}^T]\vec{\beta} - \vec{\beta}^T\vec{\beta} \quad (9.13)$$

and that the variance can be written as

$$\text{Var}(\hat{\beta}) = E\left[\hat{\beta}^T\hat{\beta} - 2E[\hat{\beta}]^T\hat{\beta} - E[\hat{\beta}]^TE[\hat{\beta}]\right] \quad (9.14)$$

or

$$\text{Var}(\hat{\beta}) = E\left[\hat{\beta}^T\hat{\beta}\right] - 2E[\hat{\beta}]^TE[\hat{\beta}] - E[\hat{\beta}]^TE[\hat{\beta}] \quad (9.15)$$

Thus, by comparing the MSE, variance, and bias, one can show that

$$\text{MSE}(\hat{\beta}) = \text{Var}(\hat{\beta}) + \vec{b}(\hat{\beta})^2 \quad (9.16)$$

which demonstrates that the MVUE is equivalent to the MMSE among unbiased estimators.

## Linear Estimators

In many cases, one wishes to use a **linear estimator**, i.e.

$$\hat{\beta} = L\vec{y} \quad (9.17)$$

where  $L$  is the **estimator gain matrix**. Then, for a MVUE that is also linear, one can form the optimization

$$\hat{\beta}^{MVUE} = \underset{\substack{\text{unbiased, linear} \\ \vec{\beta}}}{\operatorname{argmin}} E\left[(\hat{\beta} - E[\hat{\beta}])^T(\hat{\beta} - E[\hat{\beta}])\right] \quad (9.18)$$

which is known as the **best linear unbiased estimator (BLUE)**.

Now, consider a linear observation model, i.e.

$$\vec{y} = X\vec{\beta} + \vec{\epsilon} \quad (9.19)$$

where  $X$  is known as the **observation matrix** and  $\vec{\epsilon}$  is zero-mean **observation error**, i.e.

$$E[\vec{\epsilon}] = 0 \quad (9.20)$$

which is an arbitrary assumption since one can simply form a new linear observation model,  $\vec{y}'$ , with zero-mean observation error,  $\vec{\epsilon}'$ , as

$$\vec{y}' = \vec{y} - E[\vec{\epsilon}] = X\vec{\beta} + \vec{\epsilon} - E[\vec{\epsilon}] = X\vec{\beta} + \vec{\epsilon}' \quad (9.21)$$

Furthermore, assume that the covariance of  $\vec{\epsilon}$  is

$$E[\vec{\epsilon}\vec{\epsilon}^T] = \sigma^2 I \quad (9.22)$$

With this model, the mean of  $\vec{y}$  can be shown to be

$$E[\vec{y}] = E[X\vec{\beta} + \vec{\epsilon}] \quad (9.23)$$

$$E[\vec{y}] = E[X\vec{\beta}] + E[\vec{\epsilon}] \quad (9.24)$$

$$E[\vec{y}] = X\vec{\beta} \quad (9.25)$$

and the covariance of  $\vec{y}$  can be shown to be

$$\text{Cov}(\vec{y}) = E[(\vec{y} - E[\vec{y}])(\vec{y} - E[\vec{y}])^T] \quad (9.26)$$

$$\text{Cov}(\vec{y}) = E[(X\vec{\beta} + \vec{\epsilon} - X\vec{\beta})(X\vec{\beta} + \vec{\epsilon} - X\vec{\beta})^T] \quad (9.27)$$

$$\text{Cov}(\vec{y}) = E[\vec{\epsilon}\vec{\epsilon}^T] \quad (9.28)$$

$$\text{Cov}(\vec{y}) = \sigma^2 I \quad (9.29)$$

In this case, it can be shown that the BLUE is given by

$$\hat{\vec{\beta}}^{BLUE} = (X^T X)^{-1} X^T \vec{y} \quad (9.30)$$

As a proof, assume a different linear unbiased estimator exists with minimum variance, i.e.

$$\tilde{\vec{\beta}} = ((X^T X)^{-1} X^T + D) \vec{y} \quad (9.31)$$

where  $D$  is some non-zero matrix. Then, computing expectation of this estimator allows one to find the condition for which  $\tilde{\vec{\beta}}$  is unbiased, i.e.

$$E[\tilde{\vec{\beta}}] = E[((X^T X)^{-1} X^T + D) \vec{y}] \quad (9.32)$$

Substituting for  $\vec{y}$

$$E[\tilde{\vec{\beta}}] = E[((X^T X)^{-1} X^T + D)(X\vec{\beta} + \vec{\epsilon})] \quad (9.33)$$

Distributing the expectation for the random vector term

$$E[\tilde{\vec{\beta}}] = ((X^T X)^{-1} X^T + D) X\vec{\beta} + E[((X^T X)^{-1} X^T + D)\vec{\epsilon}] \quad (9.34)$$

which can be simplified since  $\vec{\epsilon}$  is zero-mean as

$$E[\tilde{\vec{\beta}}] = ((X^T X)^{-1} X^T + D) X\vec{\beta} \quad (9.35)$$

and rearranging

$$E[\tilde{\vec{\beta}}] = (X^T X)^{-1} X^T X\vec{\beta} + D X\vec{\beta} \quad (9.36)$$

or

$$E[\tilde{\vec{\beta}}] = (I + D X)\vec{\beta} \quad (9.37)$$

which is unbiased only if  $D X = 0$ .

Then, one can inspect the covariance to see if this new estimator can potentially have a lower variance

$$\text{Cov}(\tilde{\beta}) = \text{Cov}(L\vec{y}) \quad (9.38)$$

$$\text{Cov}(\tilde{\beta}) = L\text{Cov}(\vec{y})L^T \quad (9.39)$$

$$\text{Cov}(\tilde{\beta}) = \sigma^2 LL^T \quad (9.40)$$

Substituting the potential linear estimator for  $L$

$$\text{Cov}(\tilde{\beta}) = \sigma^2 \left( (X^T X)^{-1} X^T + D \right) \left( X(X^T X)^{-1} + D^T \right) \quad (9.41)$$

which can be distributed as

$$\text{Cov}(\tilde{\beta}) = \sigma^2 \left( (X^T X)^{-1} X^T X (X^T X)^{-1} + (X^T X)^{-1} X^T D^T + D X (X^T X)^{-1} + D D^T \right) \quad (9.42)$$

$$\text{Cov}(\tilde{\beta}) = \sigma^2 \left( (X^T X)^{-1} + (X^T X)^{-1} (D X)^T + D X (X^T X)^{-1} + D D^T \right) \quad (9.43)$$

and simplifies to the following since  $D X = 0$  must be zero

$$\text{Cov}(\tilde{\beta}) = \sigma^2 (X^T X)^{-1} + 0 + \sigma^2 D D^T \quad (9.44)$$

or

$$\text{Cov}(\tilde{\beta}) = \text{Cov}(\hat{\beta}) + \sigma^2 D D^T \quad (9.45)$$

and since  $D D^T$  is positive semi-definite for any matrix  $D$  by the properties of the transpose, one can say that

$$\text{Cov}(\tilde{\beta}) > \text{Cov}(\hat{\beta}) \quad (9.46)$$

which implies non-minimum variance for any other unbiased estimator than the BLUE as defined above.

## 9.2 Ordinary Least-Squares Estimation

In optimal parameter estimation, one often can form the estimator using the **least-squares (LS) problem** which can be stated for parameter estimation as

$$\vec{\beta}^{LS} = \underset{\vec{\beta} \in X}{\operatorname{argmin}} \| \vec{y} - f(\vec{x}, \vec{\beta}) \|_2^2 \quad (9.47)$$

where  $\vec{\beta}$  is the parameter vector containing any number of parameters and the difference  $\vec{y} - f(\vec{x}, \vec{\beta})$  is often called the **residual**. The term *least* in the name *least-squares* refers to a minimization while the term *squares* refers to a summation of multiple squared terms, in particular, the residuals.

If  $f()$  is a linear function, i.e.

$$f(\vec{x}, \vec{\beta}) = X\vec{\beta} \quad (9.48)$$

then one has the **ordinary least-squares (OLS) problem** which can be stated as

$$\vec{\beta}^{LS} = \underset{\vec{\beta} \in X}{\operatorname{argmin}} \| \vec{y} - X\vec{\beta} \|_2^2 \quad (9.49)$$

If  $X$  is square, then this problem can be solved by simply taking the inverse of  $X$

$$\vec{\beta}^{LS} = X^{-1} \vec{y} \quad (9.50)$$

However, if  $X$  is not square, then the true inverse of  $X$  does not exist, then one uses the **pseudoinverse**, also known as the **Moore-Penrose inverse**,  $X^+$ , which satisfies the following properties:

- $XX^+X = X$
- $X^+XX^+ = X^+$
- $(XX^+)^* = XX^+$
- $(X^+X)^* = X^+X$

It should be noted that if  $X$  is invertible, then the pseudoinverse is the inverse. It can be shown that the pseudoinverse exists for all matrices, but may not have a simple algebraic formula. However, if  $X$  has linearly independent columns (i.e. full rank), then

$$X^+ = (X^T X)^{-1} X^T \quad (9.51)$$

which is also known as the **left pseudoinverse**.

Thus, the **OLS solution** can be generalized as

$$\vec{\beta}^* = X^+ \vec{y} \quad (9.52)$$

and if  $X^T X$  is invertible

$$\vec{\beta}^{LS} = (X^T X)^{-1} X^T \vec{y} \quad (9.53)$$

As a proof of the OLS solution, let

$$f(x) = \|X\vec{\beta} - \vec{y}\|_2^2 \quad (9.54)$$

Then, writing out  $f(x)$  by component, one has

$$f(x) = \sum_{i=1}^m \left( \left( X\vec{\beta} \right)_i - b_i \right)^2 \quad (9.55)$$

or writing out by elements of  $X$  and  $\vec{\beta}$ , one has

$$f(x) = \sum_{i=1}^m \left( \left( \sum_{j=1}^n X_{i,j} x_j \right) - b_i \right)^2 \quad (9.56)$$

Then, the minimum of  $\|X\vec{\beta} - \vec{y}\|_2^2$  occurs when its gradient is equal to 0. To calculate this, first recall the definition for the gradient of a function

$$\nabla f(x) = \left( \frac{\partial f}{\partial x_1}(x), \dots, \frac{\partial f}{\partial x_n}(x) \right) \quad (9.57)$$

where each element of the gradient can be determined by

$$\frac{\partial f}{\partial x_k}(\vec{\beta}) = 2 \sum_{i=1}^m X_{i,k} \left( \left( \sum_{j=1}^n X_{i,j} x_j \right) - y_i \right) \quad (9.58)$$

$$\frac{\partial f}{\partial x_k}(\vec{\beta}) = 2 \left( X^T (X \vec{\beta} - \vec{y}) \right)_k \quad (9.59)$$

Thus, the gradient becomes

$$\nabla \|X \vec{\beta} - \vec{y}\|_2^2 = 2X^T (X \vec{\beta} - \vec{y}) \quad (9.60)$$

Setting this equal to 0 for the minimum

$$2X^T (X \vec{\beta}^{LS} - \vec{y}) = 0 \quad (9.61)$$

$$X^T X \vec{\beta}^{LS} - X^T \vec{y} = 0 \quad (9.62)$$

$$X^T X \vec{\beta}^{LS} = X^T \vec{y} \quad (9.63)$$

$$\vec{\beta}^{LS} = (X^T X)^{-1} X^T \vec{y} \quad (9.64)$$

which exists as long as  $X^T X$  is left-invertible, i.e.  $X$  has linearly independent columns.

Solving for  $X^T X \vec{\beta}^* = X^T \vec{y}$  by back substitution can suffer from numerical difficulties due to the matrix multiplication  $X^T X$  for certain matrices. Thus, most solvers use the QR decomposition which allow us to reform  $\vec{\beta}^*$  as

$$\vec{\beta}^{LS} = ((QR)^T QR)^{-1} (QR)^T \vec{y} \quad (9.65)$$

$$\vec{\beta}^{LS} = (R^T Q^T QR)^{-1} R^T Q^T \vec{y} \quad (9.66)$$

$$\vec{\beta}^{LS} = (R^T R)^{-1} R^T Q^T \vec{y} \quad (9.67)$$

$$\vec{\beta}^{LS} = R^{-1} R^{-T} R^T Q^T \vec{y} \quad (9.68)$$

$$\vec{\beta}^{LS} = R^{-1} Q^T \vec{y} \quad (9.69)$$

Finally, it should be noted that numerical algorithms for efficiently computing the OLS solution are typically performed as follows.

1. Compute the QR decomposition on  $X$ 
  - Method typically uses Householder matrices
  - Reflects vector about some hyperplane
2. Compute  $\vec{c} = Q^T \vec{y}$
3. Solve  $R \vec{\beta}^{LS} = \vec{c}$  for  $\vec{\beta}^{LS}$  by back substitution

## OLS Parameter Estimation

Consider the following regression problem where one has  $N$  sets of samples of independent data,  $\vec{x}(i)$ , and dependent data,  $\vec{y}(i)$ , with  $i = 1, \dots, N$ . In addition, consider that one desires to optimally fit this data to a chosen regression model, i.e.

$$\vec{y}(i) = f(\vec{x}(i), \vec{\beta}) \quad (9.70)$$

where the **residuals** of the samples as

$$\vec{r}(i) = \vec{y}(i) - f(\vec{x}(i), \vec{\beta}) \quad (9.71)$$

Then, with an optimality criterion of least-squares, one can form the **least-squares (LS) estimator**,  $\hat{\vec{\beta}}^{LS}$ , which minimizes the sum of squares of the residuals, i.e.

$$\hat{\vec{\beta}}^{LS} = \underset{\vec{\beta}}{\operatorname{argmin}} \sum_{i=1}^N \left( \vec{y}(i) - f(\vec{x}(i), \vec{\beta}) \right)^T \left( \vec{y}(i) - f(\vec{x}(i), \vec{\beta}) \right) \quad (9.72)$$

Furthermore, if the regression model,  $f()$ , is linear, i.e.

$$\vec{y}(i) = X(i)\vec{\beta} \quad (9.73)$$

where  $X(i)$  are matrices, then, one has a linear regression problem whose least-squares solution is the **linear least-squares (LLS) estimator**, i.e.

$$\hat{\vec{\beta}}^{LLS} = \underset{\vec{\beta}}{\operatorname{argmin}} \sum_{i=1}^N \left( \vec{y}(i) - X(i)\vec{\beta} \right)^T R^{-1}(i) \left( \vec{y}(i) - X(i)\vec{\beta} \right) \quad (9.74)$$

which minimizes the expression known as the **Mahalanobis distance** and has simple analytical solutions. Here  $R$  is a positive definite weight matrix for each sample  $i = 1, \dots, N$ . Also, note that by defining

$$\mathbf{R} = \begin{bmatrix} R(1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R(N) \end{bmatrix} \quad (9.75)$$

$$\mathbf{X} = [X(1) \quad \cdots \quad X(N)] \quad (9.76)$$

and

$$\vec{\mathbf{y}} = \begin{bmatrix} \vec{y}(1) \\ \vdots \\ \vec{y}(N) \end{bmatrix} \quad (9.77)$$

then, one can rewrite the LLS optimization above as

$$\hat{\vec{\beta}}^{LLS} = \underset{\vec{\beta}}{\operatorname{argmin}} \left( \vec{\mathbf{y}} - \mathbf{X}\vec{\beta} \right)^T \mathbf{R}^{-1} \left( \vec{\mathbf{y}} - \mathbf{X}\vec{\beta} \right) \quad (9.78)$$

Furthermore, if  $R = I$ , then  $\hat{\beta}^{LLS}$  is the **OLS parameter estimator**, i.e.

$$\hat{\beta}^{OLS} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \vec{\mathbf{y}} \quad (9.79)$$

Thus, if one considers this regression problem as an optimal parameter estimation problem with a linear observation model

$$\vec{\mathbf{y}}(i) = H(i) \vec{\beta} + \vec{\epsilon}(i) \quad (9.80)$$

where the observation error,  $\vec{\epsilon}(i)$ , are zero-mean with equal variance and are uncorrelated with respect to  $i$ , then the OLS estimator is the BLUE. This result is known as the **Gauss-Markov theorem**.

In the practice of parameter estimation, often one does not know if the observations are uncorrelated with respect to  $i$ . One method for checking this auto-correlation for one observation can be checked by estimating the **auto-correlation function** of the observed residuals, i.e.

$$\hat{R}_{\vec{r}\vec{r}}(i) = \sum_{j=1}^{N-i} \vec{r}(j) \vec{r}(i+j) \quad (9.81)$$

for  $j = 0, \dots, N - 1$ , though this can become very inaccurate for large  $j$  values which make  $N - j$  small. If the residuals are completely uncorrelated, then it should be that  $\hat{R}(k) = 0$ ,  $k \neq 0$ .

The covariance matrix of the OLS estimator error can be shown to be

$$\text{Cov}(\hat{\beta}) = E[(\hat{\beta} - E[\hat{\beta}])(\hat{\beta} - E[\hat{\beta}])^T] \quad (9.82)$$

$$\text{Cov}(\hat{\beta}) = E[(\hat{\beta} - \vec{\beta})(\hat{\beta} - \vec{\beta})^T] \quad (9.83)$$

$$\text{Cov}(\hat{\beta}) = E[(H^T H)^{-1} H^T (\vec{\mathbf{y}} - \vec{\beta})(\vec{\mathbf{y}} - \vec{\beta})^T H (H^T H)^{-1}] \quad (9.84)$$

$$\text{Cov}(\hat{\beta}) = (H^T H)^{-1} H^T E[\vec{\epsilon} \vec{\epsilon}^T] H (H^T H)^{-1} \quad (9.85)$$

Furthermore, if the observation error is uncorrelated and has constant variance  $\sigma^2$  across  $i$ , i.e.  $E[\vec{\epsilon} \vec{\epsilon}^T] = \sigma^2 I$ , then one has

$$\text{Cov}(\hat{\beta}) = \sigma^2 (H^T H)^{-1} \quad (9.86)$$

where notably  $(H^T H)^{-1}$  was also required to compute the OLS parameter estimate.

Second, recall that the diagonal elements of this covariance matrix represent the parameter variances,  $\sigma^2(\hat{\beta}_i)$ , i.e. the standard deviations squared, while the off-diagonal elements represent the covariance between any two parameter estimates,  $\sigma^2(\hat{\beta}_1, \hat{\beta}_2)$ , and can be related to the correlation coefficient between any two parameter estimates by diving by the standard deviations of each parameter, i.e.

$$\rho(\hat{\beta}_1, \hat{\beta}_2) = \frac{\sigma^2(\hat{\beta}_1, \hat{\beta}_2)}{\sigma(\hat{\beta}_1)\sigma(\hat{\beta}_2)} \quad (9.87)$$

Second, note that the calculation of the error covariance matrix requires one knows the observation error variance,  $\sigma^2$ , and that it is constant across all samples. However, in practice,  $\sigma^2$  is usually not known *a*

*priori* and therefore must be estimated from the measured data. An unbiased estimator of  $\sigma^2$  for  $N_r$  repeated observations under the *same* sampling conditions is

$$\hat{\sigma}^2 = \frac{1}{N_r - 1} \sum_{i=1}^{N_r} [\vec{y}_r(i) - \bar{y}_r]^2 \quad (9.88)$$

where the observations,  $\vec{y}_r$ , have an observed mean value  $\bar{y}_r$  defined as

$$\bar{y}_r = \frac{1}{N_r} \sum_{i=1}^{N_r} \vec{y}_r(i) \quad (9.89)$$

Furthermore, it is unlikely that the same sampling conditions will be exactly repeated, thus,  $\sigma^2$  can also be estimated independently using smoothing methods where . Assuming the model structure is adequate, an unbiased estimator of  $\sigma^2$  can be computed using the residuals as

$$\hat{\sigma}^2 = \frac{\vec{r}^T \vec{r}}{N_r - n_\beta} = \frac{\sum_{i=1}^{N_r} [\vec{y}_i - \hat{\vec{\beta}}_i]^2}{N_r - n_\beta} \quad (9.90)$$

where  $n_\beta$  is the dimension of  $\vec{\beta}$  and  $\sqrt{\hat{\sigma}^2}$  is called the **fit error** which indicates how close the estimates  $\hat{\vec{\beta}}(i)$  are to the measured values  $\vec{y}(i)$ . Note that this estimator depends on the model through  $\hat{\vec{\beta}}_i$ .

Third, it should be pointed out that the output residuals,  $\vec{r}$ , can be interpreted as samples of the observation error  $\vec{\epsilon}$ . Note that for the linear regression

$$\vec{r} = \vec{z} - \mathbf{X}^T \hat{\vec{\beta}} \quad (9.91)$$

where any departure from the underlying assumptions on the errors should be seen in the residuals. Using the estimate for  $\hat{\vec{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \vec{y}$ , the residuals can also be written as

$$\vec{r} = (I - \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T) \vec{y} \quad (9.92)$$

Analysis of the residuals is an effective method for discovering various types of observation model deficiencies, but may not be a simple process. Typically a consistency check is used on the standard deviations of the residuals which can be computed from the diagonal elements of the covariance matrix of the residuals. For OLS estimation, this residual covariance can be shown to be

$$\text{Cov}(\vec{r}) = \sigma^2 (I - \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T) \quad (9.93)$$

Then, for a Gaussian distribution assumption on the statistics, the 95% confidence intervals on the residuals is equal to  $\pm 2\sigma(\vec{r})$ . However, the proof of this is beyond the scope of this textbook.

Lastly, another metric that quantifies the “nearness” of  $\hat{\vec{\beta}}(i)$  to  $\vec{y}(i)$  is the **coefficient of determination**,  $R^2$ . Its definition for OLS estimation follows from partitioning the total sum of squared variations in the measured output  $\vec{y}$  about its mean value, a quantity denoted by  $SS_T$  and defined as

$$SS_T = \sum_{i=1}^N [\vec{y}(i) - \bar{y}]^2 = \vec{y}^T \vec{y} - N\bar{y}^2 \quad (9.94)$$

where

$$\bar{y} = \frac{1}{N} \sum_{i=1}^N \vec{y}(i) \quad (9.95)$$

This can be partitioned into the sum of squared variations of the estimate  $\hat{\beta}$  about the same mean value, denoted by  $SS_R$  and defined as

$$SS_R = \sum_{i=1}^N [\hat{\beta}(i) - \bar{y}]^2 \quad (9.96)$$

plus the sum of squared variations of the observation  $\vec{y}$  about the estimate  $\hat{\beta}$ , denoted by  $SS_E$  and defined as

$$SS_E = \sum_{i=1}^N [\vec{y}(i) - \hat{\beta}(i)]^2 = (\vec{y} - \mathbf{X}\hat{\beta})^T (\vec{y} - \mathbf{X}\hat{\beta}) \quad (9.97)$$

$$SS_E = \vec{y}^T \vec{y} - 2\hat{\beta}^T \mathbf{X}^T \vec{y} + \hat{\beta}^T \mathbf{X}^T \mathbf{X} \hat{\beta} \quad (9.98)$$

$$SS_E = \vec{y}^T \vec{y} - 2\hat{\beta}^T \mathbf{X}^T \vec{y} + \hat{\beta}^T \mathbf{X}^T \mathbf{X} \hat{\beta} \quad (9.99)$$

$$SS_E = \vec{y}^T \vec{y} - \hat{\beta}^T \mathbf{X}^T \vec{y} \quad (9.100)$$

Then, by inspection, one can see that

$$SS_T = SS_R + SS_E \quad (9.101)$$

For good models,  $SS_E$  will only include the observation error and  $SS_R$  will be large relative to  $SS_E$ . Here the coefficient of determination,  $R^2$ , represents the proportion of the variation in the measured output that is explained by the model, i.e.

$$R^2 = \frac{SS_R}{SS_T} = 1 - \frac{SS_E}{SS_T} = \frac{\hat{\beta}^T \mathbf{X}^T \vec{y} - N\bar{y}^2}{\vec{y}^T \vec{y} - N\bar{y}^2} \quad (9.102)$$

where values of  $R^2$  vary from 0 to 1, where 1 represents a perfect fit to the data. Note that  $R^2$  is usually expressed as a percentage in linear regression analysis.

## 9.3 Generalized and Bayesian Least-Squares Estimation

### Generalized Least-Squares Parameter Estimation

Consider the linear regression model, i.e.

$$\vec{y}(i) = X(i)\hat{\beta} \quad (9.103)$$

then, by defining

$$\mathbf{R} = \begin{bmatrix} R(1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R(N) \end{bmatrix} \quad (9.104)$$

$$\mathbf{X} = [X(1) \quad \cdots \quad X(N)] \quad (9.105)$$

and

$$\vec{y} = \begin{bmatrix} \vec{y}(1) \\ \vdots \\ \vec{y}(N) \end{bmatrix} \quad (9.106)$$

then, the linear regression solution is the linear least-squares (LLS) estimator

$$\hat{\beta}^{LLS} = \underset{\vec{\beta}}{\operatorname{argmin}} \left( \vec{y} - \mathbf{X}\vec{\beta} \right)^T \mathbf{R}^{-1} \left( \vec{y} - \mathbf{X}\vec{\beta} \right) \quad (9.107)$$

Furthermore, for arbitrary  $R$ , then  $\hat{\beta}^{GLS}$  is the **GLS parameter estimator**, i.e.

$$\hat{\beta}^{GLS} = (\mathbf{X}^T \mathbf{R}^{-1} \mathbf{X}) \mathbf{X}^{-1} \mathbf{R}^{-1} \vec{y} \quad (9.108)$$

Thus, if one considers this regression problem as an optimal parameter estimation problem with a linear observation model

$$\vec{y}(i) = X(i)\vec{\beta} + \vec{\epsilon}(i) \quad (9.109)$$

where the observation error,  $\vec{\epsilon}(i)$ , are zero-mean with covariances  $R(i)$  and are uncorrelated with respect to  $i$ , then the GLS estimator is the BLUE.

Thus, the GLS estimate can be considered as the OLS on scaled and “de-correlated” observations. To see this consider the Cholesky decomposition (exists for all positive definite matrices) of  $R$ , i.e.

$$R = CC^T \quad (9.110)$$

Next, multiply the observation model by  $C^{-1}$  on both sides, i.e.

$$C^{-1}\vec{y} = C^{-1}\mathbf{X}\vec{\beta} + C^{-1}\vec{\epsilon} \quad (9.111)$$

Then, one can form a “new” model

$$\vec{y}' = \mathbf{X}'\vec{\beta} + \vec{\epsilon}' \quad (9.112)$$

where

$$\vec{y}' = C^{-1}\vec{y} \quad (9.113)$$

$$\mathbf{X}' = C^{-1}\mathbf{X} \quad (9.114)$$

$$\vec{\epsilon}' = C^{-1}\vec{\epsilon} \quad (9.115)$$

For this model, one can show that

$$\operatorname{Cov}(\vec{\epsilon}') = \operatorname{Cov}(C^{-1}\vec{\epsilon}) \quad (9.116)$$

$$\operatorname{Cov}(\vec{\epsilon}') = C^{-1}\operatorname{Cov}(\vec{\epsilon})C^{-T} \quad (9.117)$$

or

$$\operatorname{Cov}(\vec{\epsilon}') = C^{-1}RC^{-T} \quad (9.118)$$

and substituting for  $R$  from the Cholesky decomposition, one has

$$\operatorname{Cov}(\vec{\epsilon}') = C^{-1}CC^TC^{-T} \quad (9.119)$$

$$\text{Cov}(\vec{\mathbf{v}}') = I \quad (9.120)$$

Thus, substituting back for the “new” parameters into the expression for  $\hat{\beta}^{OLS}$ , one has

$$\hat{\beta}^{GLS} = (\mathbf{X}^T C^{-T} C^{-1} \mathbf{X})^{-1} \mathbf{X}^T C^{-T} C^{-1} \vec{\mathbf{y}} \quad (9.121)$$

or

$$\hat{\beta}^{GLS} = (\mathbf{X}^T R^{-1} \mathbf{X})^{-1} \mathbf{X}^T R^{-1} \vec{\mathbf{y}} \quad (9.122)$$

Secondly, one can show that the covariance of the GLS estimator is given by

$$\text{Cov}(\hat{\beta}) = (X^T R^{-1} X)^{-1} \quad (9.123)$$

and likewise to the previous problem  $X$  must be full rank.

### Bayesian Least-Squares Parameter Estimation

Another approach to the parameter estimation problem is from a Bayesian inference perspective. Here one considers that  $\vec{\mathbf{y}}$  are samples of a random vector  $\vec{Y}$  which depend on some *random* parameter vector,  $\vec{B}$ , which has taken some realization  $\vec{\beta}$ . In this case, one must consider the joint probabilities of  $\vec{Y}$  and  $\vec{\beta}$  which can be modeled by some joint PDF,  $f_{\vec{Y}, \vec{B}}(\vec{y}, \vec{\beta})$ . However, as one observes the realizations of  $\vec{Y}$ , one is particularly concerned with the *a posteriori* conditional PDF, i.e.  $f_{\vec{B}|\vec{Y}}(\vec{\beta}|\vec{y})$ , which is related to the joint PDF by the equation

$$f_{\vec{B}|\vec{Y}}(\vec{\beta}|\vec{y}) = \frac{f_{\vec{Y}, \vec{B}}(\vec{y}, \vec{\beta})}{f_{\vec{Y}}(\vec{y})} \quad (9.124)$$

which simply restates the definition of conditional probabilities. Furthermore, one can rewrite this equation as

$$f_{\vec{B}|\vec{Y}}(\vec{\beta}|\vec{y}) = \frac{f_{\vec{Y}|\vec{B}}(\vec{y}|\vec{\beta}) f_{\vec{B}}(\vec{\beta})}{f_{\vec{Y}}(\vec{y})} \quad (9.125)$$

which, by the law of total probability, i.e.

$$f_{\vec{Y}}(\vec{y}) = \int f_{\vec{Y}|\vec{B}}(\vec{y}|\vec{\beta}) f_{\vec{B}}(\vec{\beta}) d\vec{\beta} \quad (9.126)$$

one has by Bayes' rule

$$f_{\vec{B}|\vec{Y}}(\vec{\beta}|\vec{y}) = \frac{f_{\vec{Y}|\vec{B}}(\vec{y}|\vec{\beta}) f_{\vec{B}}(\vec{\beta})}{\int f_{\vec{Y}|\vec{B}}(\vec{y}|\vec{\beta}) f_{\vec{B}}(\vec{\beta}) d\vec{\beta}} \quad (9.127)$$

Thus, Bayesian parameter estimation can be seen as using three sources of information, the *a priori* PDF of the parameter vector,  $f_{\vec{B}}(\vec{\beta})$ , the likelihood function,  $f_{\vec{Y}|\vec{B}}(\vec{y}|\vec{\beta})$ ,  $\mathcal{L}(\vec{\beta}|\vec{y})$ , and the **model evidence**,  $f_{\vec{Y}}(\vec{y})$ .

An example of an optimal parameter estimator using Bayesian inference is the **maximum *a posteriori* (MAP) estimator** which can be stated as

$$\hat{\beta}^{MAP} = \underset{\vec{\beta}}{\operatorname{argmax}} f_{\vec{B}|\vec{Y}}(\vec{\beta}|\vec{y}) \quad (9.128)$$

which by Bayes' rule can be rewritten as

$$\hat{\vec{\beta}}^{MAP} = \operatorname{argmax}_{\vec{\beta}} \frac{f_{\vec{Y}|\vec{B}}(\vec{y}|\vec{\beta})f_{\vec{B}}(\vec{\beta})}{\int f_{\vec{Y}|\vec{B}}(\vec{y}|\vec{\beta})f_{\vec{B}}(\vec{\beta})d\vec{\beta}} \quad (9.129)$$

which is equivalent to

$$\hat{\vec{\beta}}^{MAP} = \operatorname{argmax}_{\vec{\beta}} f_{\vec{Y}|\vec{B}}(\vec{y}|\vec{\beta})f_{\vec{B}}(\vec{\beta}) \quad (9.130)$$

It should be noted the MAP estimator is equivalent to the MLE if the *a priori* PDF is uniform across all values of  $\vec{\beta}$ . An alternative statistic that can be used as an estimator is the mean of the *a posteriori* PDF.

In addition, one can consider a least-squares parameter estimator from a Bayesian inference perspective, also known as a **Bayesian least-squares (BLS) estimator** which considers an observation model

$$\vec{y} = f(\vec{x}, \vec{\beta}) + \vec{\epsilon} \quad (9.131)$$

with the  $\mathbb{E}[\vec{\epsilon}] = 0$  and covariance  $\mathbb{E}[\vec{\epsilon}\vec{\epsilon}^T] = R$ , but also has some *a priori* information about the possible values of  $\vec{\beta}$  represented by the *a priori* PDF defined as  $\vec{\beta}_0 \sim f_{\vec{B}}(\vec{\beta})$  and can be regarded as user-defined prior knowledge or due to previous parameter estimates. Due to this statistical nature, the objective in Bayesian least-squares is to minimize the sum of the estimator errors, i.e.

$$\hat{\vec{\beta}}^{BLS} = \operatorname{argmin}_{\hat{\vec{\beta}}} E[(\vec{\beta} - \hat{\vec{\beta}})^T(\vec{\beta} - \hat{\vec{\beta}})] \quad (9.132)$$

which can also be written as

$$\hat{\vec{\beta}}^{BLS} = \operatorname{argmin}_{\hat{\vec{\beta}}} E[\operatorname{Tr}((\vec{\beta} - \hat{\vec{\beta}})(\vec{\beta} - \hat{\vec{\beta}})^T)] \quad (9.133)$$

which implicitly takes into account the probability distributions of  $\vec{\epsilon}$  and  $\vec{\beta}$ . It should be noted that BLS is also known as **recursive least-squares (RLS)** due to the sequential updating of  $\hat{\vec{\beta}}$  based on “new” observational data sets,  $\vec{y}_i$  with  $i = 1, 2, \dots$  and can also be done as more data sets become available.

A common analytical solution for Bayesian parameter estimation occurs when one has a linear observation model

$$\vec{y} = X\vec{\beta} + \vec{\epsilon} \quad (9.134)$$

with zero-mean multivariate Gaussian errors,  $\vec{\epsilon} \sim \mathcal{N}(0, R)$ , and a multivariate Gaussian *a priori* PDF,  $\vec{\beta} \sim \mathcal{N}(\hat{\vec{\beta}}_0, \Sigma_0)$ , where the mean of the *a priori* PDF is the current best estimate of  $\vec{\beta}$  with regards to the maximum likelihood, mean, and minimum variance. It can be shown that the *a posteriori* PDF of this model is a multivariate Gaussian,  $\vec{\beta}|\vec{y} \sim \mathcal{N}(\hat{\vec{\beta}}^{BLS}, \Sigma_{BLS})$ . Thus, for this observation model, the maximum, mean, and minimum variance of the *a posteriori* PDF and the BLS solution are all the same value.

This solution can be obtained by considering the linear estimator of the form

$$\hat{\vec{\beta}}^{BLS} = \hat{\vec{\beta}}_0 + L(\vec{y} - X\hat{\vec{\beta}}_0) \quad (9.135)$$

which can also be written as

$$\hat{\vec{\beta}}^{BLS} = (I - LX)\hat{\vec{\beta}}_0 + L\vec{y} \quad (9.136)$$

which has an estimator error given

$$E \left[ \vec{\beta} - \hat{\vec{\beta}}^{BLS} \right] = E \left[ \vec{\beta} - (I - LX)\hat{\vec{\beta}}_0 - L\vec{y} \right] \quad (9.137)$$

$$E \left[ \vec{\beta} - \hat{\vec{\beta}}^{BLS} \right] = E \left[ \vec{\beta} - LX\vec{\beta} - \vec{\epsilon} \right] - (I - LX)\hat{\vec{\beta}}_0 \quad (9.138)$$

$$E \left[ \vec{\beta} - \hat{\vec{\beta}}^{BLS} \right] = (I - LX) \left( E \left[ \vec{\beta} \right] - \hat{\vec{\beta}}_0 \right) - E \left[ \vec{\epsilon} \right] \quad (9.139)$$

$$E \left[ \vec{\beta} - \hat{\vec{\beta}}^{BLS} \right] = 0 \quad (9.140)$$

and an estimator covariance given by

$$\begin{aligned} E \left[ (\vec{\beta} - \hat{\vec{\beta}}^{BLS})^T (\vec{\beta} - \hat{\vec{\beta}}^{BLS}) \right] &= (I - LX)E \left[ (\vec{\beta} - \hat{\vec{\beta}}^{BLS}) \right] (I - LX)^T - LE \left[ \vec{\epsilon}(\vec{\beta} - \hat{\vec{\beta}}^{BLS})^T \right] (I - LX)^T \\ &\quad - (I - LX)E \left[ (\vec{\beta} - \hat{\vec{\beta}}^{BLS}) \vec{\epsilon}^T \right] L^T - LE \left[ \vec{\epsilon} \vec{\epsilon}^T \right] L^T \end{aligned} \quad (9.141)$$

$$E \left[ (\vec{\beta} - \hat{\vec{\beta}}^{BLS})^T (\vec{\beta} - \hat{\vec{\beta}}^{BLS}) \right] = (I - LX)\Sigma_0(I - LX)^T + LRL^T \quad (9.142)$$

Then, the BLS solution can be found by solving for when the derivative of the trace of this covariance equals zero, i.e.

$$2(I - LX)\Sigma_0(-X^T) + 2LR = 0 \quad (9.143)$$

$$LX\Sigma_0 - X^T + LR = \Sigma_0 X^T \quad (9.144)$$

$$L = \Sigma_0 X^T (X\Sigma_0 X^T + R)^{-1} \quad (9.145)$$

Thus, for the BLS parameter estimator for linear, Gaussian observation model, one has

$$\hat{\vec{\beta}}^{BLS} = \hat{\vec{\beta}}_0 + \Sigma_0 X^T (X\Sigma_0 X^T + R)^{-1} (\vec{y} - X\hat{\vec{\beta}}_0) \quad (9.146)$$

with a covariance

$$\Sigma_{BLS} = (I - \Sigma_0 X^T (X\Sigma_0 X^T + R)^{-1} H) \Sigma_0 \quad (9.147)$$

which also corresponds to mean and covariance of the multivariate Gaussian *a posteriori* PDF.

## 9.4 Nonlinear and Constrained Least-Squares Estimation

### Nonlinear Least-Squares Parameter Estimation

Consider the following regression problem where one has  $N$  sets of samples, i.e. vector-pairs  $\vec{x}(i), \vec{y}(i)$  with  $i = 1, \dots, N$ , and desires to optimally fit this data to a chosen regression model, i.e.

$$\vec{y}(i) = f(\vec{x}(i), \vec{\beta}) \quad (9.148)$$

with residuals

$$\vec{r}(i) = \vec{y}(i) - f(\vec{x}(i), \vec{\beta}) \quad (9.149)$$

which can be redefined as the stacked vectors

$$\vec{y} = \begin{bmatrix} \vec{y}(1) \\ \vdots \\ \vec{y}(N) \end{bmatrix} \quad (9.150)$$

$$\vec{x} = \begin{bmatrix} \vec{x}(1) \\ \vdots \\ \vec{x}(N) \end{bmatrix} \quad (9.151)$$

$$\mathbf{f}(\vec{x}, \vec{\beta}) = \begin{bmatrix} f(\vec{x}(1), \vec{\beta}) \\ \vdots \\ f(\vec{x}(N), \vec{\beta}) \end{bmatrix} \quad (9.152)$$

and

$$\vec{r} = \begin{bmatrix} \vec{r}(1) \\ \vdots \\ \vec{r}(N) \end{bmatrix} \quad (9.153)$$

which results in the regression model

$$\vec{y} = \mathbf{f}(\vec{x}, \vec{\beta}) \quad (9.154)$$

For nonlinear  $f()$  and with an optimality criterion of least-squares, one can form the **nonlinear least-squares (NLS) estimator**,  $\hat{\vec{\beta}}^{NLS}$ , which minimizes the sum of squares of the residuals, i.e.

$$\hat{\vec{\beta}}^{NLS} = \underset{\vec{\beta}}{\operatorname{argmin}} \vec{r}^T \vec{r} \quad (9.155)$$

Any local minimum for this optimization, if one exists, is found by setting the derivative of the squared residuals to zero, i.e.

$$2 \vec{r}^T \frac{\partial \vec{r}}{\partial \vec{\beta}} = 0 \quad (9.156)$$

or, in terms of the observation model Jacobian,  $\frac{\partial \mathbf{f}(\vec{x}, \vec{\beta})}{\partial \vec{\beta}}$ , one has

$$-2 \vec{r}^T \frac{\partial \mathbf{f}(\vec{x}, \vec{\beta})}{\partial \vec{\beta}} = 0 \quad (9.157)$$

where the optimal estimate,  $\hat{\vec{\beta}}^{NLS}$ , will be the one (or multiple) values of  $\beta$  which accomplish this.

However, one cannot often find an analytical solution to the NLS estimation problem, thus, one must use numerical methods to find the optimal value. Such methods begin with some initial estimate,  $\hat{\vec{\beta}}_0$ , and then use an iterative procedure to refine the parameter estimate, i.e.

$$\hat{\vec{\beta}}^{k+1} = \hat{\vec{\beta}}^k + \vec{\Delta}_k \quad (9.158)$$

where  $k$  is the iteration number and  $\vec{\Delta}_k$  is the **search vector** at iteration  $k$ . Then, after some convergence criteria for the search vector, e.g. for some chosen  $\delta \lll 1$ , one can approximate the NLE estimator by

$$\hat{\vec{\beta}}^{NLS} \approx \hat{\vec{\beta}}^k \quad \text{for } \frac{\vec{\Delta}_k \vec{\beta}^T \vec{\Delta}_k}{\vec{\beta}^k} < \delta \quad (9.159)$$

where a typical value for  $\delta$  would be 0.0001 which would require a precision of 0.1%. When used in this form, the NLS estimator is a type of **iterative least-squares (ILS) estimator**.

One of the simplest methods for computing  $\vec{\Delta}_k$  is the **Gauss-Newton algorithm (GNA)** which approximates the observation model by a first-order Taylor series expansion, i.e.

$$\mathbf{f}(\vec{x}, \vec{\beta}) \approx \mathbf{f}(\vec{x}, \hat{\vec{\beta}}_k) + \frac{\partial \mathbf{f}(\vec{x}, \vec{\beta})}{\partial \vec{\beta}} \Big|_{\vec{\beta}=\hat{\vec{\beta}}_k} \vec{\Delta}_k \vec{\beta} \quad (9.160)$$

where

$$\vec{\Delta}_k = \vec{\beta} - \hat{\vec{\beta}}_k \quad (9.161)$$

and thus

$$\frac{\partial \mathbf{f}(\vec{x}, \vec{\beta})}{\partial \vec{\beta}} = \frac{\partial \mathbf{f}(\vec{x}, \vec{\beta})}{\partial \vec{\beta}} \Big|_{\vec{\beta}=\hat{\vec{\beta}}_k} \quad (9.162)$$

Next, defining the Jacobian of the regression model as

$$\mathbf{J} = \frac{\partial \mathbf{f}(\vec{x}, \vec{\beta})}{\partial \vec{\beta}} \Big|_{\vec{\beta}=\hat{\vec{\beta}}_k} \quad (9.163)$$

one can rewrite  $\vec{r} = \vec{y} - \mathbf{f}(\vec{x}, \vec{\beta})$  as

$$\vec{r} = \vec{y} - \mathbf{f}(\vec{x}, \hat{\vec{\beta}}_k) + \mathbf{f}(\vec{x}, \hat{\vec{\beta}}_k) - \mathbf{f}(\vec{x}, \vec{\beta}) \quad (9.164)$$

or

$$\vec{r} = \vec{y} - \mathbf{f}(\vec{x}, \hat{\vec{\beta}}_k) - \mathbf{J} \vec{\Delta}_k \quad (9.165)$$

Then, setting the gradient of this approximation for the squared residuals to zero, one has

$$-2 \vec{r}^T \frac{\partial f}{\partial \vec{\beta}} = -2 \left( \vec{y} - \mathbf{f}(\vec{x}, \hat{\vec{\beta}}_k) - \mathbf{J} \vec{\Delta}_k \right)^T \mathbf{J} = 0 \quad (9.166)$$

which can be rewritten as

$$\mathbf{J}^T \left( \vec{y} - \mathbf{f}(\vec{x}, \hat{\vec{\beta}}_k) - \mathbf{J} \vec{\Delta}_k \right) = 0 \quad (9.167)$$

and by rearranging, one obtains the OLS solution for the search vector

$$\vec{\Delta}_k = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T (\vec{\mathbf{y}} - \mathbf{f}(\vec{\mathbf{x}}, \vec{\beta}_k)) \quad (9.168)$$

Furthermore, if the residuals have an expected covariance matrix,  $\mathbf{R}$ , then one can alternatively use the GLS solution as

$$\vec{\Delta}_k = (\mathbf{J}^T \mathbf{R} \mathbf{J})^{-1} \mathbf{J}^T \mathbf{R} (\vec{\mathbf{y}} - \mathbf{f}(\vec{\mathbf{x}}, \vec{\beta}_k)) \quad (9.169)$$

An important part of this numerical method is the initial parameter estimates and the problem of divergence. Since divergence in the GNA often occurs, one often uses the **Levenburg-Marquardt algorithm (LMA)**, also known as **damped least-squares (DLS)**, as a trust-region augmentation to the GNA. In this case, one includes a damping factor,  $\zeta_k$ , also known as **Marquardt parameter** in the search vector calculation as

$$\vec{\Delta}_k = (\mathbf{J}^T \mathbf{R} \mathbf{J} + \zeta_k I)^{-1} \mathbf{J}^T \mathbf{R} (\vec{\mathbf{y}} - \mathbf{f}(\vec{\mathbf{x}}, \vec{\beta}_k)) \quad (9.170)$$

where this second term combines the GNA with the direction of **steepest gradient**, i.e.

$$\vec{\Delta}_k = \frac{1}{\zeta_k} \mathbf{J}^T \mathbf{R} (\vec{\mathbf{y}} - \mathbf{f}(\vec{\mathbf{x}}, \vec{\beta}_k)) \quad (9.171)$$

which the LMA approximates if  $\mathbf{J}^T \mathbf{R} \mathbf{J} \ll \zeta_k I$ . The LMA can be further improved through Fletcher's **modified LMA**

$$\vec{\Delta}_k = (\mathbf{J}^T \mathbf{R} \mathbf{J} + \zeta_k \text{diag}(\mathbf{J}^T \mathbf{R} \mathbf{J}))^{-1} \mathbf{J}^T \mathbf{R} (\vec{\mathbf{y}} - \mathbf{f}(\vec{\mathbf{x}}, \vec{\beta}_k)) \quad (9.172)$$

where  $\text{diag}(\mathbf{J}^T \mathbf{R} \mathbf{J})$  selects the diagonal elements of  $\mathbf{J}^T \mathbf{J}$ . This modification slows down convergence in the direction of small gradients by adjusting more where the gradient of  $f()$  is smaller.

In either case of LMA,  $\zeta_k$  is heuristically adjusted at each iteration which is intrinsic to the trust-region method. An effective heuristic known as **delayed gratification** consists of increasing  $\zeta_k$  by a small amount for iterations where  $\vec{\mathbf{r}}^T \vec{\mathbf{r}}$  increases, and decreasing  $\zeta_k$  by a large amount for iterations where  $\vec{\mathbf{r}}^T \vec{\mathbf{r}}$  decreases. This heuristic slows down convergence which avoids converging too quickly in the beginning of optimization. An increase by a factor of 2 and a decrease by a factor of 3 are effective for most problems. Other trust-region methods can also be used which solve a sub-problem for the optimal damping factor which are discussed in the next section.

## Constrained Least-Squares Parameter Estimation

**Trust-region methods**, also known as **restricted-step methods**, can be used for both unconstrained and constrained least-squares problems. For least-squares parameter estimation, the **constrained least-squares (CLS) problem** can be stated as

$$\begin{aligned} \hat{\vec{\beta}}^{CLS} &= \underset{\vec{\beta} \in X}{\operatorname{argmin}} \quad \| \vec{\mathbf{y}} - \mathbf{f}(\vec{\mathbf{x}}, \vec{\beta}) \|_2^2 \\ &\text{subject to: } l_\beta \leq \vec{\beta} \leq u_\beta \end{aligned} \quad (9.173)$$

where  $l_\beta$  and  $u_\beta$  are the lower and upper bounds on the parameter estimate, respectively. This section will discuss how trust-region methods solve the CLS problem at a high level.

To understand trust-region methods, consider the general minimization search problem where one desires to improve, i.e. move from  $\hat{\vec{\beta}}_k$  to some  $\hat{\vec{\beta}}_{k+1}$  which reduces the least-squares function. Trust-region methods approximate the least-squares function,  $\left\| \vec{y} - f(\vec{x}, \hat{\vec{\beta}}_k) \right\|_2^2$  with a simpler function,  $q(\vec{y}, \vec{x}, \hat{\vec{\beta}}_k)$  in some neighborhood about  $\hat{\vec{\beta}}_k$ , then solving for the least-squares function approximation does decrease for optimal **trial step vector**,  $\vec{\Delta}_k$ , within that neighborhood, i.e. **trust-region**. However, one must also check that the obtained trial step vector can be “trusted,” i.e. that  $\left\| \vec{y} - f(\vec{x}, \hat{\vec{\beta}}^k + \vec{\Delta}_k) \right\|_2^2 < q(\vec{y}, \vec{x}, \hat{\vec{\beta}}_k + \vec{\Delta}_k)$ , where if this fails to hold, then the trust-region must be adjusted and a new trial step vector must be found. Thus, different trust-region methods use different approximations,  $q()$ , different trial step vector solvers in the trust-region, and different trust-region adjustments. Standard trust-region methods use a quadratic form for the least-squares approximation, e.g. the LMA, which defines an ellipsoidal trust-region, quadratic programming solvers, and standard trust-region adjustments similar to the delayed gratification heuristic above.

## Quadratic Programming

As an aside, the **quadratic programming (QP) problem** can be stated as

$$\begin{aligned} \vec{\beta}^{QP} = \underset{\vec{\beta}}{\operatorname{argmin}} \quad & \frac{1}{2} \vec{\beta}^T Q \vec{\beta} + \vec{c}^T \vec{\beta} \\ \text{subject to: } & A \vec{\beta} \leq \vec{b} \end{aligned} \tag{9.174}$$

where the constraint is a vector-defined component-wise inequality. QP problems appear in different problems. One is minimizing a function  $f()$  in a local neighborhood about a point  $\vec{\beta}_0$ , e.g. a trust-region, one can set  $Q$  to the Hessian matrix at that point,  $H(\vec{\beta}_0)$ , and  $\vec{c}$  to its gradient,  $\nabla f(\vec{\beta}_0)$ .

As a special case, when  $Q = Q^T > 0$ , a QP problem is equivalent to the LLS problem. To see this, consider the LLS minimization written as

$$\underset{\vec{\beta}}{\operatorname{argmin}} \quad \left( \vec{y} - X \vec{\beta} \right)^T \left( \vec{y} - X \vec{\beta} \right) \tag{9.175}$$

$$\vec{\beta}^{QP} = \underset{\vec{\beta}}{\operatorname{argmin}} \quad \vec{y}^T \vec{y} - \vec{y}^T X \vec{\beta} - \vec{\beta}^T X^T \vec{y} + \vec{\beta}^T X^T X \vec{\beta} \tag{9.176}$$

which is equivalent to minimizing only over terms with  $\vec{\beta}$ , multiplying the expression by  $\frac{1}{2}$ , and that the order of the middle terms produces the same result, i.e.

$$\underset{\vec{\beta}}{\operatorname{argmin}} \quad \frac{1}{2} \vec{\beta}^T X^T X \vec{\beta} - \vec{y}^T X \vec{\beta} \tag{9.177}$$

which by the Cholesky decomposition  $Q = X^T X$  and defining  $\vec{c} = -X^T \vec{y}$ , one has

$$\underset{\vec{\beta}}{\operatorname{argmin}} \quad \frac{1}{2} \vec{\beta}^T Q \vec{\beta} + \vec{c}^T \vec{\beta} \tag{9.178}$$

QP programming solvers can be derived explicitly for equality constraints and for  $Q > 0$  similar to the least-squares solver, i.e. the ellipsoid method solves the problem in (weakly) polynomial time. However if  $Q$  is indefinite or has even one negative eigenvalue, then the problem is NP-hard.

# Chapter 10

## Introductory Optimal Control

### 10.1 Introduction to Optimal Control

**Optimal control** is a control framework where the controller is selected as the optimal control law with respect to some chosen objective for the dynamical system. This optimization over time is critical to optimal control theory and can be formulated for both continuous-time and discrete-time dynamical systems. To solve for this optimal control law, one uses methods from **mathematical optimization**, also known as **mathematical programming**, which formulates the selection of the “best” or optimal element of a set of possible elements with respect to some criteria, i.e. objective. Typically, the best is defined as the minimum or maximum with respect to the objective. These sets of possible elements can be continuous, discrete, and/or constrained. The objective in optimal control can be imposed superficially by the control designer or dictated by the real system process. Furthermore, this optimization procedure is also referred to as solving the **optimal control problem (OCP)** and is typically stated as a minimization.

The OCP for continuous-time dynamical systems can be generally written as

$$\begin{aligned} \vec{u}^{\text{opt}}(t) = & \underset{u(t) \forall t \in [0, t_f]}{\operatorname{argmin}} J = \mathcal{E}(\vec{x}(t_f), t_f) + \int_0^{t_f} \mathcal{L}(\vec{x}(t), \vec{u}(t)) dt \\ & \text{subject to:} \\ & \text{dynamics } \dot{\vec{x}}(t) = f(\vec{x}(t), \vec{u}(t), t) \\ & \text{initial condition } \vec{x}(0) = \vec{x}_0 \\ & \text{constraints } c(\vec{x}(t), \vec{u}(t), t) \leq 0 \end{aligned} \tag{10.1}$$

where  $\vec{u}^{\text{opt}}(t)$  is the **optimal control function**,  $\operatorname{argmin}$  stands for “the argument which minimizes” the following expression,  $t_f$  is the final time (also known as the **time horizon** since  $t$  starts at zero without loss of generality),  $J$  is the **objective functional**, also known as the **cost functional**,  $\mathcal{E}$  is the **endpoint cost** or **terminal cost**, and  $\mathcal{L}$  is the Lagrangian or **running cost** which must be nonnegative. The optimization is also subject to some dynamics for the state of the system, an initial condition for the system, and some constraints on the state, inputs, and time.

Similarly, the OCP for discrete-time dynamical systems can be generally written as

$$\begin{aligned} \vec{u}^{\text{opt}}[k] = \underset{\vec{u}[k] \text{ for } k=0, \dots, N-1}{\operatorname{argmin}} J &= \mathcal{E}(\vec{x}[N], N) + \sum_{k=0}^{N-1} \mathcal{L}(\vec{x}[k], \vec{u}[k]) \\ \text{subject to:} \\ \text{discrete dynamics} \quad \vec{x}[k+1] &= f(\vec{x}[k], \vec{u}[k], k) \\ \text{initial condition} \quad \vec{x}[0] &= \vec{x}_0 \\ \text{constraints} \quad c(\vec{x}[k], \vec{u}[k], k) &\leq 0 \end{aligned} \tag{10.2}$$

where  $\vec{u}^{\text{opt}}[k]$  is the **optimal control sequence**,  $N$  is the final time step (also known as the **time horizon** since  $k$  starts at zero without loss of generality), and  $J$  is the **objective function**, also known as the **cost function**. Comparing this discrete-time OCP to the continuous-time OCP, the integration became a summation and the differential equation became a difference equation.

It is important to note a few things about the OCP formulation. First, the term **functional** is a mathematical definition for a “function of a function.” Second, the cost functional or function,  $J$ , is determined by the control designer and depends on the control and state over time. Third, the initial condition is sometimes further generalized to **boundary conditions** which can be for *any* time instant in the considered time horizon, not just zero. This would then fall under the optimization framework as a **boundary value problem (BVP)**, a subject which has been studied in-depth by many mathematicians. Fourth, there may be multiple solutions to the OCP depending on the cost functional or function and the state dynamics. Fifth, often one is not necessarily interested in finding the truly optimal solution to the OCP, but an near-optimal solution for complex OCPs, a topic which will be considered later.

Within the general OCP framework, there are different versions of the OCP that can be further characterized. First, one can optimize for finite or infinite time horizons,  $t_f$  and  $N$ , typically shortened to finite- or infinite-horizon OCPs. In this case, the integral or summation in the cost functional or function is taken to infinity. In this case the OCPs may be easier to solve than finite-horizon OCPs as the optimal control law will not depend on any specific time, but only on the state, thus becoming a *fixed-gain* control law, which may be desirable. Second, the dynamics of the system may be linear or nonlinear which also allows for simpler solvers. Third, OCP solvers may be simplified if the OCP is unconstrained or constrained. The effects of constraints on the solving method typically revolves around whether there are state constraints, input constraints, or both. Finally, the cost functional or function may characterized as linear, quadratic, convex, or non of these, each requiring more complex solvers. This is due to the fact that for a finite minimum cost to exist, the simplest model would be a linear cost for constrained OCPs and a quadratic cost for unconstrained OCPs. Beyond these cases, one can also make a more general distinction between convex and non-convex costs since convexity implies a local minimum is a global minimum which simplifies many numerical methods which rely on searches.

Mathematical optimization algorithms can be characterized in one of two ways, either as indirect methods or direct methods. The remainder of this section will discuss an indirect method for continuous-time OCPs using a generalization of the calculus of variation as well as dynamic programming for discrete-time OCPs which introduces the principal of optimality.

## Generalized Calculus of Variations

One indirect optimization method is a generalization of the **calculus of variations** which can be used to solve the unconstrained continuous-time OCP. These indirect methods begin by augmenting the cost functional with the **costate** vector, also known as the **adjoint** vector,  $\vec{\lambda}(t)$ , which is analogous to Lagrange multiplier problems in other mathematics. This results in the following equation for the augmented cost functional

$$\bar{J} = \mathcal{E}(\vec{x}(t_f), t_f) + \int_0^{t_f} \left( \mathcal{L}(\vec{x}(t), \vec{u}(t)) + \vec{\lambda}^T(f(\vec{x}(t), \vec{u}(t), t) - \dot{\vec{x}}(t)) dt \right) \quad (10.3)$$

where  $\vec{\lambda}(t)$  can be *any* vector because the state dynamics require that

$$\dot{\vec{x}}(t) = f(\vec{x}(t), \vec{u}(t), t) \quad (10.4)$$

holds for all time, which means  $\vec{\lambda}(t)$  is being multiplied by zero in the expression above. Next, one can form the **variation** of  $\bar{J}$  as

$$\delta \bar{J} = \mathcal{E}_x \delta \vec{x}(t_f) + \int_0^{t_f} \left( \mathcal{L}_x \delta \vec{x} + \mathcal{L}_u \delta \vec{u} + \vec{\lambda}^T f_x \delta \vec{x} + \vec{\lambda}^T f_u \delta \vec{u} - \vec{\lambda}^T \delta \dot{\vec{x}} dt \right) \quad (10.5)$$

where the subscripts denote partial derivatives for the functionals above. By expanding the last term using integration by parts, one obtains

$$-\int_0^{t_f} \vec{\lambda}^T \delta \dot{\vec{x}} dt = -\vec{\lambda}^T(t_f) \delta \vec{x}(t_f) + \vec{\lambda}^T(0) \delta \vec{x}(0) + \int_0^{t_f} \dot{\vec{\lambda}}^T \delta \vec{x} dt \quad (10.6)$$

By substitution and rearrangement, one can separate  $\delta \bar{J}$  into four different components as

$$\delta \bar{J} = \left( \mathcal{E}_x - \vec{\lambda}^T(t_f) \right) \delta \vec{x}(t_f) + \int_0^{t_f} \left( \mathcal{L}_x + \vec{\lambda}^T f_x + \dot{\vec{\lambda}}^T \right) \delta \vec{x} + \left( \mathcal{L}_u + \vec{\lambda}^T f_u \right) \delta \vec{u} dt + \vec{\lambda}^T(t_0) \delta \vec{x}(0) \quad (10.7)$$

If one chooses  $J$  to be continuous in all  $\vec{x}$ ,  $\vec{u}$ ,  $t$ ,  $\delta J$  (and  $\delta \bar{J}$ ), then for  $\delta \bar{J}$  to be zero, i.e the optimal solution by the definition of the variation, the first three components of  $\delta \bar{J}$  must independently be zero. This occurs because one can vary  $\vec{x}$ ,  $\vec{u}$  and  $\vec{x}(t_f)$ , but not  $\vec{x}(0)$ . Then, the following three equations must hold

$$\begin{cases} \mathcal{E}_x - \vec{\lambda}^T(t_f) &= 0 \\ \mathcal{L}_x + \vec{\lambda}^T f_x + \dot{\vec{\lambda}}^T &= 0 \\ \mathcal{L}_u + \vec{\lambda}^T f_u &= 0 \end{cases} \quad (10.8)$$

or rewriting, one has

$$\begin{cases} \vec{\lambda}(t_f) &= \mathcal{E}_x^T \\ \dot{\vec{\lambda}} &= -\mathcal{L}_x^T - f_x^T \vec{\lambda} \\ 0 &= \mathcal{L}_u + \vec{\lambda}^T f_u \end{cases} \quad (10.9)$$

of which the first two equations provide the final condition and dynamics for  $\vec{\lambda}$  while the third provides the constraints on  $\vec{\lambda}$ .

These equations are solvable in reverse time, but can be difficult for large states and complex cost functionals. Beyond simple problems where an analytical solution can be solved directly, these equations for  $\vec{\lambda}$  are typically numerically solved using methods which rely on iteration techniques. An example method would be something like

1. Start with initial guess:  $u(t)$
2. Iterate the following until solution converges to  $\delta u(t) = 0$ 
  - a) Propagate  $\dot{\vec{x}} = f(\vec{x}(t), \vec{u}(t), t)$  forward in time
  - b) Evaluate  $\mathcal{E}_x(x(t_f))$
  - c) Propagate  $\dot{\vec{\lambda}}^T = -\mathcal{L}_x - \vec{\lambda}^T f_x$  backward in time
  - d) Choose  $\delta u(t) = -K(\mathcal{L}_u + \vec{\lambda}^T f_u)$  where  $K$  is positive definite
  - e) Let  $u(t) = u(t) + \delta u(t)$

Such methods are typically susceptible to poor initial guesses for  $u(t)$  and typically require a well-performing gain  $K$  which often takes the form of a heuristic for different types of optimizations.

## Dynamic Programming

A key concept in optimization over time is one can perform the optimization in stages. In essence, one is balancing the lowest possible cost at the present stage against the impact this would have for costs at future stages. The optimal control action minimizes the sum of the cost incurred at the current stage and the least total cost that can be incurred from all subsequent stages, consequent on this decision. This is known as the **principle of optimality** which states that from any point on an optimal trajectory, the remaining trajectory is optimal for the corresponding problem initiated at that point. Formally for discrete-time OCPs, this can be stated by defining the **cost-to-go function** at time step  $k$  as

$$J_k(\vec{x}[k]) = \mathcal{E}(\vec{x}[N]) + \sum_{i=k}^{N-1} \mathcal{L}(\vec{x}[i], \vec{u}[i]) \quad (10.10)$$

which has a corresponding sub-optimization problem

$$V_k(\vec{x}[k]) = \min_{\vec{u}[i] \text{ for } i=k, \dots, N-1} J_k = \mathcal{E}(\vec{x}[N], N) + \sum_{i=k}^{N-1} \mathcal{L}(\vec{x}[i], \vec{u}[i]) \quad (10.11)$$

subject to the dynamics, initial conditions, and constraints. Considering  $V_k$  is called the **value function** as a function of  $k$  and  $\vec{x}[k]$  with the following properties

$$V_N(\vec{x}[N]) = \mathcal{E}(\vec{x}[N], N) \quad (10.12)$$

and

$$V_0(\vec{x}[0]) = \min_{\vec{u}[k] \text{ for } k=0, \dots, N-1} J \quad (10.13)$$

Furthermore, this formalization allows one to write the sub-optimization as a recursive relationship

$$V_k(\vec{x}[k]) = \min_{\vec{u}[i] \text{ for } i=k, \dots, N-1} \mathcal{L}(\vec{x}[k], \vec{u}[k]) + V_{k+1}(\vec{x}[k+1]) \quad (10.14)$$

which is the OCP form of the **Bellman equation**, also known as the **optimality equation**. Thus, one can begin at  $V_N(\vec{x}[N])$  and calculate the value function at previous time steps by working backwards using the Bellman equation and finally obtaining  $V_0(\vec{x}[0])$  as the value of the optimal cost. Then, the optimal control

sequence,  $\vec{u}^{\text{opt}}[k]$ , can then be recovered by tracing back the calculations already performed for the value function.

The Bellman equation serves as the fundamental result of dynamic programming (DP) and is often referred to as the **dynamic programming equation**. By definition, **dynamic programming (DP)** solves an optimization problem by recursively solving simpler sub-problems which make up the overall problem. In general, not all optimization problems allow dynamic programming methods, however optimization problems over a sequence of time steps often allow recursive sub-problems to be nested inside the overall problem. This time sequencing is the origin of the term “dynamic” in dynamic programming. Mathematically, these sub-problems done by defining a sequence of value functions, as shown previously for discrete-time OCPs. The value function at any time step is valued based on future time steps and thus can be used to compute the minimum cost-to-go function. Lastly, it should also be noted that a continuous-time version of the Bellman equation is the **Hamilton-Jacobi-Bellman equation** which is a partial differential equation (PDE) that can be used to solve for the optimal cost of continuous-time OCPs.

A classic problem whose solvers typically implement dynamic programming is the **shortest-path problem** which has several versions. This problem can be stated formally using a **graph** which is a discrete mathematical structures used to model pairwise relations between objects where each object is represented by **nodes** connected by **edges** from one node to another. If these edges are defined asymmetrically between nodes, one has a **directed graph**. In addition, if one assigns weights to each edge, one has a **weighted graph**. For the shortest-path problem, these weights can be used to correspond to the “distance” between the two nodes and is typically constrained to be non-negative. With these definitions in mind, the **single-pair shortest-path problem** seeks to solve for the shortest path starting from a single, specified node, called the **source node** and ending at another single, specified node, called the **destination node**. Here, the **path** can be defined as the sequence of nodes which one visits to travel from the source node to the destination node.

One common method for the **single-pair shortest-path problem** is **Dijkstra's algorithm** which uses a **cumulative distance** for each node from the source node.

1. Create a set of all nodes called the *unvisited set*.
2. Assign to the source node a cumulative distance value of zero and infinity for all other nodes. Set the source node as *current node*.
3. For the current node, consider all unvisited neighbors, i.e. nodes to which it has connected edges, and calculate a new cumulative distance through the current node and compare to the current assigned cumulative distance and assign the smaller one.
  - For example, if the current node,  $A$ , is marked with a distance of 6, and the edge connecting it with a neighbor,  $B$ , has length 2, then the distance to  $B$  through  $A$  will be  $6 + 2 = 8$ . If  $B$  was previously marked with a cumulative distance greater than 8 then change it to 8. Otherwise, the current value will be kept.
4. Once all of the unvisited neighbors of the current node have been checked, remove the current node from the unvisited set. A visited node will never be checked again.
5. If the destination node has the smallest tentative distance among all “unvisited” nodes or if the smallest tentative distance among the nodes in the unvisited set is not infinity, then, select the unvisited node with the smallest tentative distance, set it as the new current node, and go back to step 3.
6. Else, stop as the algorithm has finished.

It should be noted that Dijkstra's algorithm makes no attempt of "smart" exploration towards the destination node. The only consideration in determining the next "current" node is the cumulative distance from the source node and expands outward from the source node, considering *every* node that is closer in terms of cumulative distance until it reaches the destination node. Another algorithm for solving the single-pair shortest-path problem is the **A<sup>opt</sup> search algorithm** which is an extension of Dijkstra's algorithm, but which uses a heuristic exploration term, i.e. attempts to explore directly towards the destination node, to improve the speed at which one explores the graph. It can be shown that Dijkstra's algorithm is equivalent to the A<sup>opt</sup> search with a heuristic equal to zero. A<sup>opt</sup> search often appears in real-time path planning of dynamical systems, for which it was originally developed, and is often part of the reference command signal generation for automatic control systems.

This shortest-path problem can also be scaled to three other generalizations which arise in other discrete optimization methods and applications. These generalizations of the single-pair shortest-path problem have significantly more efficient algorithms than the simplistic approach of running a single-pair shortest-path algorithm on all relevant pairs of nodes. First, the **single-source shortest-path problem** seeks to solve for the shortest paths from a single source node to all other nodes. It can be shown that a variant of Dijkstra's algorithm is the asymptotically fastest algorithm for the single-source shortest-path problem with unbounded non-negative weights. Second, the **single-destination shortest-path problem** seeks to solve for the shortest paths from all other nodes to a single destination node. This can be reduced to the single-source shortest path problem by reversing the edges in a directed graph. Third, the **all-pairs shortest-path problem** seeks to solve for the shortest paths from all nodes to every other node. Lastly, it should be noted that the **Euclidean shortest-path problem** can be defined continuously-valued space as opposed discrete graphs.

## 10.2 Unconstrained Linear-Quadratic Regulator

One of the fundamental OCPs is the **linear-quadratic** OCP, defined as having *linear* dynamics and a *quadratic* cost function/functional, with or without constraints. In particular, for continuous-time one has dynamics of the form

$$\dot{\vec{x}}(t) = A(t)x(t) + B(t)u(t) \quad (10.15)$$

and a cost functional of the form

$$J = \frac{1}{2}x^T(t_f)Ex(t_f) + \frac{1}{2} \int_{t_0}^{t_f} \left( x^T(t)Q(t)x(t) + u^T(t)R(t)u(t) + 2x^T(t)S(t)u(t)dt \right) \quad (10.16)$$

and for discrete-time, one has dynamics of the form

$$\vec{x}_k = F_k x_{k-1} + G_k u_{k-1} \quad (10.17)$$

and a cost function of the form

$$J = \frac{1}{2}\vec{x}[N]^T E \vec{x}[N] + \frac{1}{2} \sum_{k=0}^{N-1} \left( \vec{x}[k]^T Q[k] \vec{x}[k] + \vec{u}[k]^T R[k] \vec{u}[k] + 2\vec{x}[k]^T S[k] \vec{u}[k] \right) \quad (10.18)$$

Both of these formulations contain **cost/weight matrices** where *E* is the **endpoint cost/weight matrix** or **terminal cost/weight matrix**, *Q* is the **state cost/weight matrix**, *R* is the **input cost/weight matrix**, and

$S$  is the **cross-cost/weight matrix**. The control designer selects the values of these matrices to balance the costs of large values of  $\vec{x}$  and  $\vec{u}$ . Furthermore, for the unconstrained case,  $E$ ,  $Q$ , and  $R$  are all symmetric, positive semi-definite matrices so that the cost function/functional is non-negative. The controller which solves the LQ OCP is called the **linear-quadratic regulator (LQR)**, i.e.  $\vec{u}^{\text{opt}}(t)$  or  $\vec{u}^{\text{opt}}[k]$ . The term **regulator** denotes that this controller steers the system state to 0, or more generally to a single constant input. Furthermore, it should be noted that  $A/F$ ,  $B/G$ ,  $Q$ ,  $R$ , and  $S$  can all vary with  $t/k$ ; however, the focus of this lecture will be on LTI systems for the LQ OCP, for both finite- and infinite-horizons.

### Unconstrained Continuous-Time LQR

The **unconstrained finite-horizon continuous-time LQ OCP** can be stated as

$$\begin{aligned}\vec{u}^{\text{opt}}(t) &= \underset{u(t) \forall t \in [0, t_f]}{\text{argmin}} J = x^T(t_f)Ex(t_f) + \int_0^{t_f} x^T(t)Qx(t) + u^T(t)Ru(t) + 2x^T(t)Su(t)dt \\ \text{subject to: } &\dot{\vec{x}}(t) = Ax(t) + Bu(t) \\ \text{initial condition: } &\vec{x}(0)\end{aligned}\tag{10.19}$$

where the unconstrained finite-horizon continuous-time LQR is the optimal control function,  $\vec{u}^{\text{opt}}(t)$ , which minimizes the quadratic cost functional,  $J$ . Recalling the generalized calculus of variations solution method, one can assign the following for this continuous-time OCP.

$$\begin{aligned}E_x &= x^T(t_f)E \\ L_x &= x^T(t)Q + u^T(t)S^T \\ L_u &= u^T(t)R + x^T(t)S \\ f_x &= A \\ f_u &= B\end{aligned}\tag{10.20}$$

which allows one to define the costate equations of the LQ OCP as

$$\begin{cases} \vec{\lambda}(t_f) &= E_x^T \\ \dot{\vec{\lambda}} &= -\mathcal{L}_x^T - f_x^T \vec{\lambda} \\ 0 &= \mathcal{L}_u + \vec{\lambda}^T f_u \end{cases}\tag{10.21}$$

which, by substitution and dropping the explicit  $t$ , one has

$$\begin{cases} \dot{\vec{\lambda}} &= -\mathcal{L}_x^T - f_x^T \vec{\lambda} \\ 0 &= \mathcal{L}_u + \vec{\lambda}^T f_u \\ \vec{\lambda}(t_f) &= E_x^T \end{cases}\tag{10.22}$$

To solve this OCP, assume the costate has a linear form, i.e.

$$\vec{\lambda}(t) = P(t) \vec{x}(t)\tag{10.23}$$

where  $P(t)$  is a symmetric matrix. Substituting this into the costate equations and including the state dynamics equation, one has

$$\begin{cases} \dot{\vec{x}} &= A\vec{x} + B\vec{u} \\ \frac{d}{dt}(P\vec{x}) = \dot{P}\vec{x} + P\dot{\vec{x}} &= -Q\vec{x} - S\vec{u} - A^T P\vec{x} \\ 0 &= \vec{u}^T R + \vec{x}^T S + \vec{x}^T P B \\ P(t_f)\vec{x}(t_f) &= E\vec{x}(t_f) \end{cases} \quad (10.24)$$

Next, substituting the first equation into the second equation results in

$$\dot{P}\vec{x} + PA\vec{x} + PB\vec{u} = -Q\vec{x} - S\vec{u} - A^T P\vec{x} \quad (10.25)$$

Then, rewriting the third equation, one has

$$\vec{u} = -R^{-1}(B^T P + S^T)\vec{x} \quad (10.26)$$

By substitution for  $\vec{u}$  into the newly derived equation, one has

$$\dot{P}\vec{x} + PA\vec{x} - PBR^{-1}(B^T P + S^T)\vec{x} = -Q\vec{x} + SR^{-1}(B^T P + S^T)\vec{x} - A^T P\vec{x} \quad (10.27)$$

By removing the common  $\vec{x}$  term and rearranging to obtain, one has

$$\dot{P} = -PA - A^T P + (PB + S)R^{-1}(B^T P + S^T) - Q \quad (10.28)$$

which is known as the **Riccati differential equation** and describes the dynamics of  $P(t)$  and can be solved using the fourth equation from the costate equations (i.e. the boundary condition)

$$P(t_f) = E \quad (10.29)$$

Thus, the Lagrangian multiplier problem has been reduced to a matrix-valued ODE which must be solved in reverse time from the end condition.

Finally, the **unconstrained finite-horizon continuous-time LQR** has the form

$$\vec{u}^{\text{opt}}(t) = -K(t)\vec{x}(t) \quad (10.30)$$

where

$$K(t) = R^{-1}(B^T P(t) + S^T) \quad (10.31)$$

which notably results in closed-loop state-space dynamics represented by

$$\dot{\vec{x}}(t) = (A - BK(t))\vec{x}(t) \quad (10.32)$$

Furthermore, if one considers the **unconstrained infinite-horizon continuous-time LQ OCP** which sets  $t_f = \infty$ , the **unconstrained infinite-horizon continuous-time LQR** is the steady-state solution of the Riccati differential equation, i.e.

$$0 = PA + A^T P - (PB + S)R^{-1}(B^T P + S^T) + Q \quad (10.33)$$

which is also known as the **continuous algebraic Riccati equation (CARE)**. This results in a fixed-gain continuous-time LQR which has the property  $\vec{x} \rightarrow 0$  as  $t \rightarrow \infty$  as  $J$  must be finite-valued.

Lastly, it should be noted that if one has LTV dynamics and/or time-varying cost matrices, the Riccati differential equation still applies, but may be more difficult to solve and a steady-state solution for the infinite-horizon LQ OCP may not exist.

### Unconstrained Discrete-Time LQR

The **unconstrained finite-horizon discrete-time LQ OCP** can be stated as

$$\begin{aligned} \vec{u}^{\text{opt}}[k] = \underset{\vec{u}[k] \text{ for } k=0, \dots, N-1}{\operatorname{argmin}} \quad J = \vec{x}[N]^T E \vec{x}[N] + \sum_{k=0}^{N-1} \vec{x}[k]^T Q \vec{x}[k] + \vec{u}[k]^T R \vec{u}[k] + 2 \vec{x}[k]^T S \vec{u}[k] \\ \text{subject to: } \vec{x}[k+1] = F \vec{x}[k] + G \vec{u}[k] \\ \text{initial condition: } \vec{x}_0 \end{aligned} \tag{10.34}$$

where the unconstrained finite-horizon discrete-time LQR is the optimal control sequence,  $\vec{u}^{\text{opt}}[k]$ , which minimizes the quadratic cost function,  $J$ . Recalling the dynamic programming solution method, one can define the following value function for this discrete-time OCP:

$$V_k(\vec{x}[k]) = \min_{\vec{u}[k] \text{ for } k=0, \dots, N-1} \vec{x}_N^T E \vec{x}_N + \sum_{\tau=k}^{N-1} \vec{x}_{\tau}^T Q \vec{x}_{\tau} + \vec{u}_{\tau}^T R \vec{u}_{\tau} + 2 \vec{x}_{\tau}^T S \vec{u}_{\tau} \tag{10.35}$$

which has the boundary condition

$$V_k(\vec{x}) = \vec{x}^T E \vec{x} \tag{10.36}$$

Moreover, the incurred cost for the LQ OCP can be defined as

$$\vec{x}^T Q \vec{x} + \vec{u}^T R \vec{u} \tag{10.37}$$

and the cost-to-go from time step  $k$  due to “next” state governed by the linear dynamics of the LQR OCP can be identified as

$$V_k(F \vec{x} + G \vec{u}) \tag{10.38}$$

which, by the Bellman equation, one has

$$V_k(\vec{x}) = \min_{\vec{u}} \left( \vec{x}^T Q \vec{x} + \vec{u}^T R \vec{u} + 2 \vec{x}^T S \vec{u} + V_k(F \vec{x} + G \vec{u}) \right) \tag{10.39}$$

To solve this OCP, assume the value function has a quadratic form, i.e.

$$V_k(\vec{x}) = \vec{x}^T P[k] \vec{x} \tag{10.40}$$

with  $P[k]$  being a symmetric and positive semi-definite matrix and with the condition

$$P[N] = E \tag{10.41}$$

and by substitution into the Bellman equation, one has

$$V_{k-1}(\vec{x}) = \vec{x}^T Q \vec{x} + \vec{u}^T R \vec{u} + 2 \vec{x}^T S \vec{u} + (F \vec{x} + G \vec{u})^T P[k] (F \vec{x} + G \vec{u}) \tag{10.42}$$

and setting

$$\frac{dV_{k-1}}{d\vec{u}} = 0 \tag{10.43}$$

one has

$$2\vec{u}^{*T}R + 2\vec{x}^TS + 2(F\vec{x} + G\vec{u}^{*\text{opt}})^TP[k]G = 0 \quad (10.44)$$

Dividing by 2 and rearranging, one has

$$\vec{u}^{*\text{opt} T}R + \vec{u}^{*\text{opt} T}G^TP[k]G = -\vec{x}^T(F^TP[k]G + S) \quad (10.45)$$

and taking the transpose

$$R\vec{u}^{*\text{opt}} + G^TP[k]G\vec{u}^{*\text{opt}} = -(G^TP[k]F + S^T)\vec{x} \quad (10.46)$$

and the inverse

$$\vec{u}^{*\text{opt}} = -\left(G^TP[k]G + R\right)^{-1}\left(G^TP[k]F + S^T\right)\vec{x} \quad (10.47)$$

one obtains the optimal control input for a single time step and can be summarized for entire sequence as

$$\vec{u}^{*\text{opt}}[k] = -K[k]\vec{x}[k] \quad (10.48)$$

with

$$K[k] = \left(G^TP[k]G + R\right)^{-1}\left(G^TP[k]F + S^T\right) \quad (10.49)$$

where one still must solve for  $P[k]$  to get an explicit solution.

First, using  $\vec{u}^{*\text{opt}}$  and

$$2\vec{x}^TS\vec{u}^{*\text{opt}} = \vec{x}^TS\vec{u}^{*\text{opt}} + \vec{u}^{*\text{opt} T}S\vec{x} \quad (10.50)$$

one can rewrite the Bellman equation as

$$J_{k-1}^{\text{opt}}(\vec{x}) = \vec{x}^TQ\vec{x} + \vec{u}^{*\text{opt} T}R\vec{u}^{*\text{opt}} + \vec{x}^TS\vec{u}^{*\text{opt}} + \vec{u}^{*\text{opt} T}S\vec{x} + (F\vec{x} + G\vec{u}^{*\text{opt}})^TP[k](F\vec{x} + G\vec{u}^{*\text{opt}}) \quad (10.51)$$

and distributing out the quadratic components, one has four terms

$$\begin{aligned} V_{k-1}(\vec{x}) &= \vec{x}^T(Q + F^TP[k]F)\vec{x} \\ &\quad + \vec{u}^{*\text{opt} T}(G^TP[k]G + R)\vec{u}^{*\text{opt}} \\ &\quad + \vec{x}^T(F^TP[k]G + S)\vec{u}^{*\text{opt}} \\ &\quad + \vec{u}^{*\text{opt} T}(G^TP[k]F + S^T)\vec{x} \end{aligned} \quad (10.52)$$

Next, one can substitute for  $\vec{u}^{*\text{opt}}$  from the solution earlier and by using the equivalent

$$\vec{u}^{*\text{opt} T} = -\vec{x}^T(F^TP[k]G + S)(G^TP[k]G + R)^{-1} \quad (10.53)$$

the Bellman equation becomes

$$\begin{aligned}
 V_{k-1}(\vec{x}) &= \vec{x}^T \left( Q + F^T P[k] F \right) \vec{x} \\
 &\quad + \vec{x}^T \left( F^T P[k] G + S \right) \left( G^T P[k] G + R \right)^{-1} \\
 &\quad \left( G^T P[k] F + S^T \right) \vec{x} \\
 &\quad - \vec{x}^T \left( F^T P[k] G + S \right) \left( G^T P[k] G + R \right)^{-1} \\
 &\quad \left( G^T P[k] F + S^T \right) \vec{x} \\
 &\quad - \vec{x}^T \left( F^T P[k] G + S \right) \left( G^T P[k] G + R \right)^{-1} \\
 &\quad \left( G^T P[k] F + S^T \right) \vec{x}
 \end{aligned} \tag{10.54}$$

and by simplifying

$$\begin{aligned}
 V_{k-1}(\vec{x}) &= \vec{x}^T \left[ F^T P[k] F - \left( F^T P[k] G + S \right) \right. \\
 &\quad \left. \left( G^T P[k] G + R \right)^{-1} \left( G^T P[k] F + S^T \right) \right] \vec{x}
 \end{aligned} \tag{10.55}$$

Finally, the quadratic form for the value function provides have derived a recursive equation for  $P[k-1]$  given  $P[k]$  as

$$P[k-1] = F^T P[k] F + Q - \left( F^T P[k] G + S \right) \left( G^T P[k] G + R \right)^{-1} \left( G^T P[k] F + S^T \right) \tag{10.56}$$

which is known as the **dynamic Riccati equation** which is a backwards recursion formula to solve for  $P[k]$  over some time horizon  $N$  starting from  $P[N] = E$ .

Finally, the **unconstrained finite-horizon discrete-time LQR** has the form

$$\vec{u}^{\text{opt}} = -K[k] \vec{x}[k] \quad \forall k \tag{10.57}$$

where

$$K[k] = \left( G^T P[k] G + R \right)^{-1} \left( G^T P[k] F + S^T \right) \tag{10.58}$$

which notably results .

Finally, if one considers the **unconstrained infinite-horizon discrete-time LQ OCP**, the solution is given by the steady-state of the dynamic Riccati equation, i.e.

$$P = F^T P F + Q - \left( F^T P G + S \right) \left( G^T P G + R \right)^{-1} \left( G^T P F + S^T \right) \tag{10.59}$$

which is known as the **discrete algebraic Riccati equation (DARE)**.

It should be noted that if one has LTV dynamics and/or time-varying cost matrices, the dynamic Riccati equation still applies. However, a steady-state solution will exist if for  $N \gg 1$ , the difference between  $P[N] - P[N-1] \rightarrow 0$  as  $t \rightarrow 0$ . Otherwise, the discrete-time LTV LQR cannot only be implemented for an infinite-horizon. Typically, in order to accomplish this steady-state, one must design  $Q[k]$  and  $R[k]$  to have some structure, but this consideration is beyond the scope of this work.

### 10.3 Unconstrained Linear-Quadratic Regulator Continued

This section discusses some implementation considerations for the unconstrained LQR. In particular, the unconstrained finite-horizon discrete-time LQR formulated as a linear least-squares problem. The four primary methods for cost matrix selection for the LQ OCP. Lastly, an example will be shown for an unconstrained infinite-horizon continuous-time LQR.

#### Discrete-Time LQR as Linear Least-Squares

Consider the unconstrained finite-horizon discrete-time LQ OCP written as

$$\begin{aligned} \vec{u}^{\text{opt}}[k] &= \underset{\vec{u}[k] \text{ for } k=0, \dots, N-1}{\operatorname{argmin}} J = \vec{x}^T[n]E\vec{x}[n] + \sum_{k=0}^{n-1} \vec{x}^T[k]Q\vec{x}[k] + \vec{u}^T[k]R\vec{u}[k] \\ \text{subject to: } \vec{x}[k+1] &= F\vec{x}[k] + G\vec{u}[k] \\ \text{initial condition: } \vec{x}[0] &= \vec{x}_0 \end{aligned} \quad (10.60)$$

where  $S$  has been removed from the cost function for simplification of the following derivations, although it can be included in the LLS solution for the discrete-time LQR as well.

Based on the discrete-time linear dynamics and initial condition, one can write out the general solution of the state as a “large” linear function, i.e.

$$\begin{bmatrix} \vec{x}_0 \\ \vdots \\ \vec{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ G & 0 & 0 & \cdots & 0 \\ FG & G & 0 & \cdots & 0 \\ F^2G & FG & G & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ F^{n-1}G & F^{n-2}G & F^{n-3}G & \cdots & G \end{bmatrix} \begin{bmatrix} \vec{u}_0 \\ \vdots \\ \vec{u}_{n-1} \end{bmatrix} + \begin{bmatrix} I \\ F \\ \vdots \\ F^n \end{bmatrix} \vec{x}_0 \quad (10.61)$$

which one can rewrite succinctly by defining the above vectors and matrices as

$$\vec{x} = \mathbf{G}\vec{u} + \mathbf{F}\vec{x}_0 \quad (10.62)$$

Then, using these new terms and the following block diagonal matrices

$$\mathbf{Q} = \begin{bmatrix} Q & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \cdots \\ 0 & \cdots & Q & 0 \\ 0 & \cdots & 0 & E \end{bmatrix} \quad (10.63)$$

and

$$\mathbf{R} = \begin{bmatrix} R & \cdots & 0 \\ \vdots & \ddots & 0 \\ 0 & \cdots & R \end{bmatrix} \quad (10.64)$$

the cost function for the discrete-time LQ OCP can be rewritten using vector notation as

$$J = \vec{x}^T \mathbf{Q} \vec{x} + \vec{u}^T \mathbf{R} \vec{u} \quad (10.65)$$

and substituting for  $\vec{x}$  from before, one can write

$$J = (\mathbf{G} \vec{u} + \mathbf{F} \vec{x}_0)^T \mathbf{Q} (\mathbf{G} \vec{u} + \mathbf{F} \vec{x}_0) + \vec{u}^T \mathbf{R} \vec{u} \quad (10.66)$$

and by using the definition of the square root of positive semi-definite matrices, i.e.

$$\mathbf{Q} = \mathbf{Q}^{\frac{1}{2}} \mathbf{Q}^{\frac{1}{2}}. \quad (10.67)$$

Next, the cost function can be rewritten as

$$J = (\mathbf{G} \vec{u} + \mathbf{F} \vec{x}_0)^T \mathbf{Q}^{\frac{1}{2}} \mathbf{Q}^{\frac{1}{2}} (\mathbf{G} \vec{u} + \mathbf{F} \vec{x}_0) + \vec{u}^T \mathbf{R}^{\frac{1}{2}} \mathbf{R}^{\frac{1}{2}} \vec{u} \quad (10.68)$$

and using the definition of vector norms, one has

$$J = \left\| \mathbf{Q}^{\frac{1}{2}} \mathbf{G} \vec{u} + \mathbf{Q}^{\frac{1}{2}} \mathbf{F} \vec{x}_0 \right\|_2^2 + \left\| \mathbf{R}^{\frac{1}{2}} \vec{u} \right\|_2^2 \quad (10.69)$$

which is a LLS problem. This form also uses the fact that  $\mathbf{Q}$  and  $\mathbf{R}$  are symmetric since  $E$ ,  $Q$ , and  $R$  are as well.

Finally, taking the gradient for  $L_2$ -norms squared with respect to  $\vec{u}$ ,

$$\nabla J = 2 \left( \mathbf{Q}^{\frac{1}{2}} \mathbf{G} \right)^T \left( \mathbf{Q}^{\frac{1}{2}} \mathbf{G} \vec{u} + \mathbf{Q}^{\frac{1}{2}} \mathbf{F} \vec{x}_0 \right) + 2 \mathbf{R} \vec{u} \quad (10.70)$$

and solving for  $\nabla J = 0$ ,

$$2 \mathbf{G}^T \mathbf{Q}^{\frac{1}{2}} \mathbf{Q}^{\frac{1}{2}} \mathbf{G} \vec{u}^{\text{opt}} + \mathbf{G}^T \mathbf{Q}^{\frac{1}{2}} \mathbf{Q}^{\frac{1}{2}} \mathbf{F} \vec{x}_0 + 2 \mathbf{R} \vec{u}^{\text{opt}} = 0 \quad (10.71)$$

$$\mathbf{G}^T \mathbf{Q} \mathbf{G} \vec{u}^{\text{opt}} + \mathbf{G}^T \mathbf{Q} \mathbf{F} \vec{x}_0 + \mathbf{R} \vec{u}^{\text{opt}} = 0 \quad (10.72)$$

$$(\mathbf{G}^T \mathbf{Q} \mathbf{G} + \mathbf{R}) \vec{u}^{\text{opt}} = -\mathbf{G}^T \mathbf{Q} \mathbf{F} \vec{x}_0 \quad (10.73)$$

$$\vec{u}^{\text{opt}} = -(\mathbf{G}^T \mathbf{Q} \mathbf{G} + \mathbf{R})^{-1} \mathbf{G}^T \mathbf{Q} \mathbf{F} \vec{x}_0 \quad (10.74)$$

which provides a method for solving the entire problem at once instead of recursively through the dynamic Riccati equation.

First, it should be noted that the recursive solution may not be faster computationally due to the large matrix inversions that are necessary in the LLS approach; however the **sparsity** (i.e. the number of 0 entries) in the  $\mathbf{Q}$  and  $\mathbf{R}$  matrices can make this inversion faster. Second, it should also be noted that the LLS solution can be adjusted for time-varying  $F$ ,  $G$ ,  $E$ ,  $Q$ , and  $R$  matrices with  $k$  since this will only change the values within  $\mathbf{F}$ ,  $\mathbf{G}$ ,  $\mathbf{Q}$  and  $\mathbf{R}$ .

## Cost Matrix Selection for the LQR

Control designers using the LQR method must select the  $Q$ ,  $R$ , and  $S$  cost matrices which implicitly affect the optimal controller. Thus, any implementation using the LQR method should use some useful guidelines. First, recall that one requires  $Q$  and  $R$  to be symmetric and positive semi-definite, thus the cost functional or function will be non-negative. Secondly, for an easier design process, one typically sets  $S = 0$  and uses one of the four following methods for selecting  $Q$  and  $R$ .

The first method uses a relative cost,  $\rho$ , between the state and input which sets

$$Q = I \quad R = \rho I \quad (10.75)$$

where the cost functional or function has become

$$J = \int_0^{t_f} \|\vec{x}\|^2 + \rho \|\vec{u}\|^2 \quad (10.76)$$

or

$$J = \sum_{k=0}^{N-1} \|\vec{x}\|^2 + \rho \|\vec{u}\|^2 \quad (10.77)$$

which simply balances the  $L_2$ -norm of the state and input through  $\rho$ . By varying  $\rho$  from  $0 \rightarrow \infty$ , one can sweep through the different values of  $\rho$  to find a satisfactory response. One analysis plot using this sweeping method is  $\rho$  versus  $J$  which is called the **optimal tradeoff curve** to identify the lowest cost overall. Another analysis plot for infinite-horizon LQR would be a root locus as a function of  $\rho$ .

The second method uses a relative cost,  $\rho$ , between output and input which sets

$$Q = C^T C \quad R = \rho I \quad (10.78)$$

where one has incorporated an output equation with no feedthrough term,  $\vec{y} = C\vec{x}$ , in order to balance the output of the system. With this method, the cost functional or function has become

$$J = \int_0^{t_f} \|\vec{y}\|^2 + \rho \|\vec{u}\|^2 \quad (10.79)$$

or

$$J = \sum_{k=0}^{N-1} \|\vec{y}\|^2 + \rho \|\vec{u}\|^2 \quad (10.80)$$

which can be said to balance the  $L_2$ -norm of output and input. Here one can also vary  $\rho$  from  $0 \rightarrow \infty$  to find satisfactory response from the possible options.

The third method uses individual diagonal costs in addition to the relative cost,  $\rho$ , which sets

$$Q = \begin{bmatrix} q_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & q_n \end{bmatrix} \quad R = \rho \begin{bmatrix} r_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & r_p \end{bmatrix} \quad (10.81)$$

where each  $q_i$  and  $r_i$  are selected to normalize the state and input for “equal” levels of error or effort, respectively. This can also be thought of as a weighted  $L_2$ -norm of the state and input, and can also be

extended for a weighted  $L_2$ -norm of the output and input. It should be noted that the normalization is typically based on units. As an example, consider that an error of 5 m/s in  $x_1$  is *as poor as* an error of 3° in  $x_2$ . Then, by setting

$$q_1 = \left(\frac{1}{5}\right)^2 \quad (10.82)$$

and

$$q_2 = \left(\frac{1}{3}\right)^2 \quad (10.83)$$

one has normalized  $q_1 x_1^2 = 1$  and  $q_2 x_2^2 = 1$  for comparable levels of error. Here again, one can vary  $\rho$  from  $0 \rightarrow \infty$  to find a satisfactory response. Here, choosing the diagonal costs may require additional trial and error if no simple method gives a satisfactory controller and is the primary task for the control designer. Note that this method can also be extended to normalize with respect to the output instead of the state using the no feedthrough model,  $\vec{y} = C^T \vec{x}$ .

The third method applies only to the infinite-horizon LQR and uses frequency-shaped cost matrices. This is realized using **Parseval's theorem** which converts scalar quadratic functions in the time domain to the frequency domain using Fourier transforms. This can be used to convert the infinite-horizon quadratic cost functional or function from the time domain to the frequency domain using the following formulas. For continuous-time, one can use

$$\int_0^\infty \vec{x}^T(t) Q \vec{x}(t) + \vec{u}^T(t) R \vec{u}(t) dt = \frac{1}{2} \int_{-\infty}^\infty \vec{X}^T(-j\omega) Q \vec{X}(j\omega) + \vec{U}^T(-j\omega) R \vec{U}(j\omega) d\omega \quad (10.84)$$

where  $\vec{X}(j\omega)$  is the continuous-time Fourier transform of  $\vec{x}(t)$  and for discrete-time, one can use

$$\sum_{k=0}^{\infty} \vec{x}^T[k] Q \vec{x}[k] + \vec{u}^T[k] R \vec{u}[k] = \frac{1}{\pi} \int_{-\pi}^{\pi} \vec{X}^T(-j\omega) Q \vec{X}(j\omega) + \vec{U}^T(-j\omega) R \vec{U}(j\omega) d\omega \quad (10.85)$$

where  $\vec{X}(j\omega)$  is the discrete-time Fourier transform of  $\vec{x}[k]$ . Here, the matrices  $Q$  and  $R$  can be made into functions of frequency,  $\omega$ .

## LQR Example

Consider the following continuous-time LTI state-space system

$$\dot{\vec{x}} = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \vec{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad y = \begin{bmatrix} 1 & 0 \end{bmatrix} \vec{x} \quad (10.86)$$

with an unconstrained infinite-horizon LQ OCP defined as

$$\begin{aligned} u^{\text{opt}} &= \underset{u(t) \forall t \geq 0}{\text{argmin}} \quad J = \int_0^\infty x_1^2 + \rho u^2 dt \\ \text{subject to: } \dot{\vec{x}} &= \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \vec{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\ \text{initial condition: } \vec{x}(0) & \end{aligned} \quad (10.87)$$

which corresponds to output-to-input relative cost weighting

$$Q = C^T C = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad (10.88)$$

and

$$R = \rho \quad (10.89)$$

Before solving the OCP, one should check that the OCP is well-posed by checking the controllability and the observability of the state-space system. First, one can check the controllability matrix

$$\begin{bmatrix} B & AB \end{bmatrix} \quad (10.90)$$

which has full rank. Therefore, the system is controllable. Then, one can check the observability matrix

$$\begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (10.91)$$

which has full rank. Therefore, the system is observable.

Next, one can solve the OCP using the continuous algebraic Riccati equation (CARE) without the  $S$  matrix

$$PA + A^T P - PBR^{-1}B^T P + Q = 0 \quad (10.92)$$

where the Riccati matrix as symmetric and positive definite with elements defined as

$$P = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \quad (10.93)$$

which, by substitution of terms, one has a CARE

$$\begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} - \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \frac{1}{\rho} [0 \ 1] \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = 0 \quad (10.94)$$

or

$$\begin{bmatrix} 0 & p_1 - p_2 \\ 0 & p_2 - p_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ p_1 - p_2 & p_2 - p_3 \end{bmatrix} - \frac{1}{\rho} \begin{bmatrix} p_2^2 & p_2 p_3 \\ p_2 p_3 & p_3^2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix} \quad (10.95)$$

which can be equated to three separate equations

$$\begin{aligned} -\frac{1}{\rho} p_2^2 &= -1 \\ p_1 - p_2 - \frac{1}{\rho} p_2 p_3 &= 0 \\ 2(p_2 - p_3) - \frac{1}{\rho} p_3^2 &= 0 \end{aligned} \quad (10.96)$$

Thus, one has

$$\begin{aligned} p_2 &= \sqrt{\rho} \\ p_3 &= \rho \left( \sqrt{1 + \frac{2}{\sqrt{\rho}}} - 1 \right) \\ p_1 &= \sqrt{\rho \left( 1 + \frac{2}{\sqrt{\rho}} \right)} \end{aligned} \quad (10.97)$$

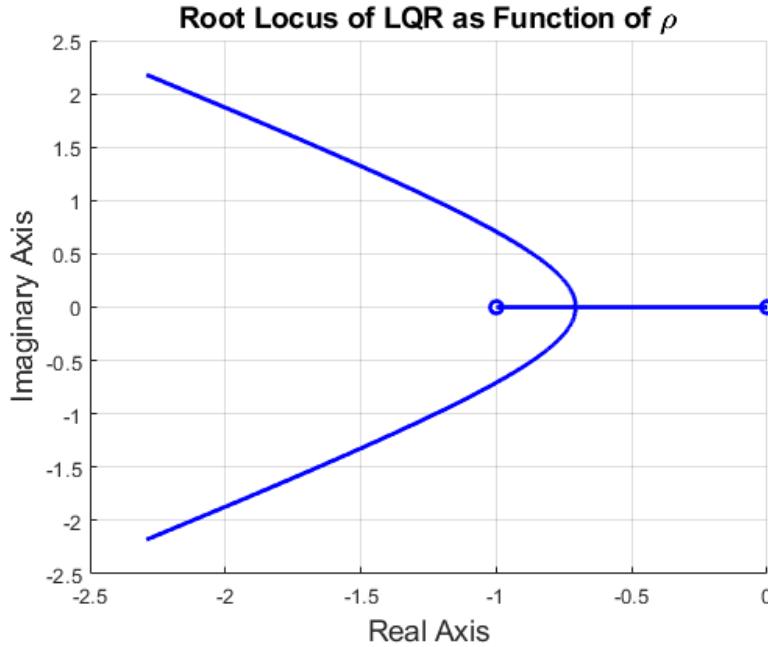
and the fixed-gain control matrix is

$$K = R^{-1}B^T P = \begin{bmatrix} \frac{2}{\rho} & \sqrt{1 + \frac{2}{\sqrt{\rho}}} - 1 \end{bmatrix} \quad (10.98)$$

Furthermore, the closed-loop characteristic equation can be calculated using the eigenvalues of the closed-loop state matrix, i.e.

$$\det(sI - A + BK) = s^2 + s \sqrt{1 + \frac{2}{\sqrt{\rho}}} + \frac{1}{\rho} = 0 \quad (10.99)$$

Analyzing the roots of the characteristic equation on a root locus plot as a function of  $\rho$  from 0.01 to  $\infty$ , results in the following figure.



For large values of  $\rho$ , the closed-loop eigenvalues approach the open-loop eigenvalues of  $A$ , i.e. 0 and  $-1$ . This produces a slow system response with lower control gains, e.g. for  $\rho = 100$ ,  $K = [0.1 \ 0.0954]$ . For small values of  $\rho$ , the closed-loop eigenvalues follow asymptotes further into the left half plane. This produces a faster response at a constant damping, e.g. for  $\rho = 0.01$ ,  $K = [31.623 \ 7.015]$ .

## 10.4 Robust Servomechanism Linear-Quadratic Regulator

In practice, the state-space dynamical system does not exactly represent the real-world system, either because of linearization errors, unmodeled dynamics, and/or changing environmental conditions. Thus, it is often desired to include a control system that drives the system to zero steady-state reference tracking error in the presence of disturbances to the system. From classical control, it is known that integral control action is necessary to achieve zero steady-state error with respect to some reference command,  $r(t)$ , in presence of some disturbance  $w(t)$ . In particular, the number of integrators in the open-loop transfer function, i.e. the **system type**, must be greater than or equal to the reference and disturbance signal orders. Thus, if  $r(t) = w(t) = 0$ , then one needs 0 integrators, i.e. a type 0 control system, if  $\dot{r}(t) = \dot{w}(t) = 0$ , then one needs 1 integrator, i.e. a type 1 control system, and if  $\ddot{r}(t) = \ddot{w}(t) = 0$ , then one needs 2 integrators and one has a type 2 control system. This integrator augmentation for tracking reference commands of certain classes is called the **internal model principle**.

For MIMO systems, the standard LQR acts as a type 0 control system as it will drive the system state to zero, but for any non-zero reference command or disturbance, there will be a steady-state error. Thus, the addition of integral action for LQR is desirable for both tracking constant non-zero reference signals with any potential constant non-zero disturbances. This approach is known as linear-quadratic-integral optimal control. For other types of reference commands and disturbances, one requires additional feedback control considerations for zero-error tracking an array of reference commands in the presence of disturbances. A standard approach for achieving this is the servomechanism state-space augmentation for feedback control. When a servomechanism is used with a LQR, one obtains a robust servomechanism linear-quadratic regulator (RSLQR). This type of controller consists of two components, a servomechanism and a state feedback signal. The “robustness” comes from its ability to reach zero steady-state tracking error in the presence of specified classes of reference commands and disturbances and can be quantified more rigorously using techniques from robust optimal control.

### Servomechanism State-Space Augmentation

Consider the following continuous-time LTI state-space system

$$\begin{aligned}\dot{\vec{x}} &= A\vec{x} + B\vec{u} + M\vec{w} \\ \vec{y} &= C\vec{x} + D\vec{u}\end{aligned}\tag{10.100}$$

with an additive unknown bounded disturbance,  $\vec{w} \in \mathbb{R}^{n_w}$ . In addition, assume that the system is controllable and observable and that the reference command is prescribed as some commanded output,  $y_c(t)$ , which has the following  $p^{\text{th}}$  order ODE form

$$\vec{y}_c^{[p]} = \sum_{i=1}^p c_i \vec{y}_c^{[p-i]}\tag{10.101}$$

where the scalar coefficients,  $c_i$ , are known and the superscript  $[j]$  denotes the  $j^{\text{th}}$  derivative. Likewise, assume that the disturbance inputs have the same  $p^{\text{th}}$  order ODE form

$$\vec{w}^{[p]} = \sum_{i=1}^p c_i \vec{w}^{[p-i]}\tag{10.102}$$

Note that a constant command has the ODE form of  $\dot{\vec{y}}_c = 0$  with  $p = 1$  and  $a_1 = 0$ , a ramp command has the ODE form of  $\ddot{\vec{y}}_c = 0$  with  $p = 2$  and  $a_2 = a_1 = 0$ , and a sinusoidal command at frequency  $\omega_0$  has the ODE form of  $\ddot{\vec{y}}_c = -\omega_0^2 \vec{y}_c$  with  $p = 2$ ,  $a_2 = -\omega_0^2$ , and  $a_1 = 0$ .

Next, define the tracking error signal as

$$\vec{e} = \vec{y}_c - \vec{y} \quad (10.103)$$

which can be differentiated  $p$  times to obtain the error ODE

$$\vec{e}^{[p]} - \sum_{i=1}^p c_i \vec{e}^{[p-i]} = \left( \vec{y}_c^{[p]} - \sum_{i=1}^p c_i \vec{y}_c^{[p-i]} \right) - \left( \vec{y}^{[p]} - \sum_{i=1}^p c_i \vec{y}^{[p-i]} \right) \quad (10.104)$$

where, by definition, the first term on the right hand side will be zero and the second term can be written using the output equation and its derivatives as

$$\vec{e}^{[p]} - \sum_{i=1}^p c_i \vec{e}^{[p-i]} = C \left( \sum_{i=1}^p c_i \vec{x}^{[p-i]} - \vec{x}^{[p]} \right) + D \left( \sum_{i=1}^p c_i \vec{u}^{[p-i]} - \vec{u}^{[p]} \right) \quad (10.105)$$

which represents a set of coupled ODEs. Defining

$$\vec{\eta} = \sum_{i=1}^p c_i \vec{x}^{[p-i]} - \vec{x}^{[p]} \quad (10.106)$$

and

$$\vec{\mu} = \sum_{i=1}^p c_i \vec{u}^{[p-i]} - \vec{u}^{[p]} \quad (10.107)$$

one has

with the control objective is to make  $\vec{e} \rightarrow 0$  as  $t \rightarrow \infty$  in the presence of unmeasurable  $\vec{w}$ .

## Robust Servomechanism Linear-Quadratic Regulator

### Linear-Quadratic-Integral Optimal Control

A special case of RSLQR occurs when one is only concerned with tracking constant reference commands in the presence of constant disturbances. In this case, the RSLQR is also known as **Linear-Quadratic-Integral (LQI) optimal control**.

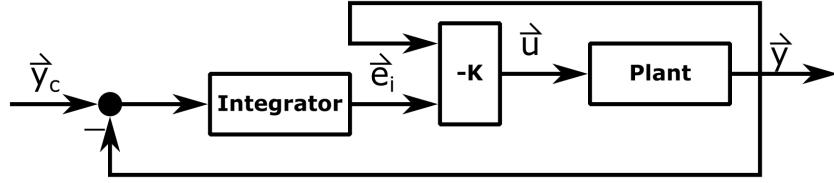
augments the state with an integrated error term,  $\vec{e}_i$ , defined as satisfying

$$\dot{\vec{e}}_i = \vec{x}_{des} - \vec{x} \quad (10.108)$$

where the stabilizing controller requires that

$$\vec{e}_i = 0 \rightarrow \vec{x} = \vec{x}_{des} \quad (10.109)$$

A block diagram of this LQI control is the following



where the controller can be written as

$$\vec{u} = -K \begin{bmatrix} \vec{x} \\ \vec{e}_i \end{bmatrix} \quad (10.110)$$

and  $K$  is found using LQR methods for the augmented state-space dynamics model

$$\begin{bmatrix} \dot{\vec{x}} \\ \vec{e}_i \end{bmatrix} = \begin{bmatrix} A & 0 \\ -I & 0 \end{bmatrix} \begin{bmatrix} \vec{x} \\ \vec{e}_i \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \vec{u} \quad (10.111)$$

It should be noted that this model has been derived with  $\vec{r} = 0$  from the block diagram above. Using discrete-time and either finite- or infinite-horizon, the state-space system augmented with the integral error vector,  $\vec{e}_i$ , i.e.

$$\begin{bmatrix} \vec{x}_k \\ \vec{e}_{i,k} \end{bmatrix} = \begin{bmatrix} F & 0 \\ -I\Delta t & I \end{bmatrix} \begin{bmatrix} \vec{x}_{k-1} \\ \vec{e}_{i,k-1} \end{bmatrix} + \begin{bmatrix} G \\ 0 \end{bmatrix} \vec{u}_{k-1} \quad (10.112)$$

which notably assumes that  $\vec{r} = 0$  as is done when computing the LQR solution to the LQ OCP. When implementing the controller, i.e.  $\vec{x}_{des} \neq 0$ , the integrated error can be computed as

$$\vec{e}_{i,k} = \Delta t (\vec{x}_{k,des} - \vec{x}_{k-1}) + \vec{e}_{i,k-1} \quad (10.113)$$

It is important to note that another numerical integration method besides the Euler integration shown can be used to improve the discretization for the LQI, but will result in a more complicated augmented state-space model. Furthermore, in implementing the controller,  $\vec{x}_{des}$  is differenced from the current state in the feedback control at each time step as

$$u_k = -K_k \left( \begin{bmatrix} \vec{x}_k \\ \vec{e}_{i,k} \end{bmatrix} - \begin{bmatrix} \vec{x}_{des,k} \\ 0 \end{bmatrix} \right) \quad (10.114)$$

## 10.5 Extended and Iterative Linear-Quadratic Regulators

## 10.6 Constrained Linear-Quadratic Regulator

Recall the quadratic programming (QP) problem

$$\begin{aligned} \vec{v}^* = \underset{\vec{v}}{\operatorname{argmin}} \quad & \frac{1}{2} \vec{v}^T \tilde{H} \vec{v} + c^T \vec{v} \\ \text{subject to: } & \vec{b}_L \leq A \vec{v} \leq \vec{b}_U \end{aligned} \quad (10.115)$$

By algebraic manipulation of the LQR OCP, one can convert the problem to a QP problem. This involves incorporating the dynamics constraint into the cost function similar to what was done for the LLS problem. For the QP inequality constraints, one must use some linear algebra to convert the constraints into matrices.

To start, the reference states can be stacked

$$\vec{X}_{ref} = [\vec{x}_{ref,1} \ \cdots \ \vec{x}_{ref,M}]^T \quad (10.116)$$

the reference inputs can be stacked

$$\vec{U}_{ref} = [\vec{u}_{ref,0} \ \cdots \ \vec{u}_{ref,M-1}]^T \quad (10.117)$$

the inputs can be stacked as

$$\vec{U} = [\vec{u}_0 \ \cdots \ \vec{u}_{M-1}]^T \quad (10.118)$$

the states can be stacked as

$$\vec{X} = [\vec{x}_1 \ \cdots \ \vec{x}_M]^T \quad (10.119)$$

the state weight matrices can be stacked as

$$\bar{Q} = \begin{bmatrix} Q & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & Q & 0 \\ 0 & \cdots & 0 & E \end{bmatrix} \quad (10.120)$$

the input weight matrices can be stacked as

$$\bar{R} = \begin{bmatrix} R & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R \end{bmatrix} \quad (10.121)$$

the state transitions can be stacked as

$$\bar{F} = [F \ \cdots \ F^M]^T \quad (10.122)$$

and the input transitions can be stacked as

$$\bar{G} = \begin{bmatrix} G & \cdots & 0 \\ \vdots & \ddots & \vdots \\ F^{M-1}G & \cdots & G \end{bmatrix} \quad (10.123)$$

However, in almost all practical applications, there exists constraints on the state and inputs of a system which can be hard constraints or soft constraints. For example, input constraints may result from actuation constraints while state constraints may result from designated “unsafe” operating points for the system. Therefore, as these constraints create complexities to solving OCPs, many different methods have been developed for handling constraints in an OCP, each with varying numerical accuracy and computational requirements.

The **constrained finite horizon LQR OCP** can be written as

$$\begin{aligned} \vec{u}_0^*, \dots, \vec{u}_{N-1}^* &= \underset{\vec{u}_0, \dots, \vec{u}_{N-1}}{\operatorname{argmin}} \quad J = \vec{x}_N^T E \vec{x}_N + \sum_{k=0}^{N-1} \vec{x}_k^T Q_k \vec{x}_k + \vec{u}_k^T R_k \vec{u}_k \\ &\text{subject to: } \vec{x}_k = F_{k-1} \vec{x}_{k-1} + G_{k-1} \vec{u}_{k-1} \\ &\text{initial condition: } \vec{x}_0 \\ &\text{state constraints: } \vec{x}_k \in \mathcal{X}_k \\ &\text{input constraints: } \vec{u}_k \in \mathcal{U}_k \end{aligned} \quad (10.124)$$

where the generalized set constraints,  $\vec{x}_k \in \mathcal{X}_k$  and  $\vec{u}_k \in \mathcal{U}_k$ , can be a variety of formulations, including inequalities, setpoints, or penalty functions.

For **inequality constraints**, minimum and maximum values limit the state

$$\vec{x}_{min} \leq \vec{x}_k \leq \vec{x}_{max} \quad \forall k \quad (10.125)$$

input

$$\vec{u}_{min} \leq \vec{u}_k \leq \vec{u}_{max} \quad \forall k \quad (10.126)$$

output

$$\vec{y}_{min} \leq H\vec{x}_k \leq \vec{y}_{max} \quad \forall k \quad (10.127)$$

and the control rate from step to step is constrained as

$$\Delta\vec{u}_{min} \leq \vec{u}_k - \vec{u}_{k-1} \leq \Delta\vec{u}_{max} \quad \forall k \quad (10.128)$$

Other hard constraints can be a combination of inequalities. These are typically generalized as either a **linear inequality**, i.e.

$$A\vec{x} \leq \vec{b} \quad (10.129)$$

or as a **linear matrix inequality (LMI)**, i.e.

$$A_0 + x_1A_1 + \dots + x_mA_m \geq 0 \quad (10.130)$$

Another type of “hard” constraint are **setpoints** or prescribed values in the state or input, i.e

$$\vec{x}_k = \vec{a} \quad \text{for some } k \quad (10.131)$$

where  $\vec{a}$  is some constant.

A soft constraint is one that can be added to the cost function instead of being an equality or inequality. These type of constraints commonly take the form of **penalty function**. An example of a generic penalty function being added is

$$\bar{J} = J + \sum_{k=0}^{N-1} p(\vec{x}_k, \vec{u}_k, k) \quad (10.132)$$

where

$$p(\vec{x}_k, \vec{u}_k, k) = \begin{cases} kc(\vec{x}_k, \vec{u}_k, k) & c(\vec{x}_k, \vec{u}_k, k) \geq 0 \\ 0 & c(\vec{x}_k, \vec{u}_k, k) < 0 \end{cases} \quad (10.133)$$

Although each of these are possibilities this course will focus on linear inequality constraints.

# Chapter 11

## Introductory Optimal State Estimation

### 11.1 Optimal Control of Stochastic State-Space Systems

First, recall the stochastic state-space representation which for continuous-time is

$$\begin{aligned} d\vec{x}(t) &= f(\vec{x}(t), \vec{u}(t), d\vec{w}(t), dt) \\ d\vec{y}(t) &= h(d\vec{x}(t), d\vec{v}(t), dt) \end{aligned} \quad (11.1)$$

and for discrete-time is

$$\begin{aligned} \vec{x}_k &= f(\vec{x}_{k-1}, \vec{u}_{k-1}, \vec{w}_k) \\ \vec{y}_k &= h(\vec{x}_k, \vec{v}_k) \end{aligned} \quad (11.2)$$

where  $\vec{w}$  and  $\vec{v}$  are the random processes for continuous-time or random sequences for discrete-time.

Thus, the stochastic state-space model is a function of the random variables and requires an initial *a priori* distribution for  $\vec{x}(0)$  or  $\vec{x}_0$ . Also, it is typical to assume that for continuous time,  $\vec{w}(t)$  and  $\vec{v}(t)$  are Markov processes, thus having independent ‘‘increments’’ for a infinitesimal  $d\vec{w}(t)$  or  $d\vec{v}(t)$ . Thus, in discrete time  $\vec{w}_k$  and  $\vec{v}_k$  are white noise processes, i.e. they are independent and identically distributed (IID) and therefore no autocorrelation for different  $k$ . Lastly, one typically assumes that  $x(0)$ ,  $\vec{w}$ , and  $\vec{v}$  are independent of each other as well.

This general model can be simplified to a **linear stochastic state-space** system which in continuous time is

$$\begin{aligned} d\vec{x}(t) &= A(t)\vec{x}(t)dt + B(t)\vec{u}(t)dt + d\vec{w}(t) \\ d\vec{y}(t) &= C(t)d\vec{x}(t) + d\vec{v}(t) \end{aligned} \quad (11.3)$$

or in discrete time is

$$\begin{aligned} \vec{x}_k &= F_k \vec{x}_{k-1} + G_k \vec{u}_{k-1} + \vec{w}_k \\ \vec{y}_k &= H_k \vec{x}_k + \vec{v}_k \end{aligned} \quad (11.4)$$

where  $\vec{w}(t)$  and  $\vec{v}(t)$  are additive Markov processes and  $\vec{w}_k$  and  $\vec{v}_k$  are additive white noise processes

For the simplest model of the stochastic noise processes, assume that in continuous time,  $\vec{w}$  and  $\vec{v}$  are additive Wiener processes, i.e. these processes have independent Gaussian increments, while in discrete

time,  $\vec{w}$  and  $\vec{v}$  are additive white Gaussian noises (AWGN), i.e. they are the mean-square derivatives of a corresponding Wiener process. This type of model is often referred to as “natural” noise because it arises in several cases including:

- Thermal vibrations of atoms in conductors
- Black-body radiation
- Satellite and deep space signals

Using this model, one can form the **fundamental stochastic state-space OCP** which is described as *Linear* due to the state/process equation, output/measurement equation, and additive noise, *Quadratic* due to the form of the cost function/functional, and having white *Gaussian* noise or independent Gaussian increments. This results in the **Linear-Quadratic-Gaussian (LQG)** OCP which can be defined for both continuous and discrete time.

## Continuous Time LQG

The unconstrained finite horizon continuous time LQG OCP is

$$\begin{aligned} \vec{u}^*(t) = \underset{u(t) \forall t \in [0, t_f]}{\operatorname{argmin}} J &= \mathbb{E} \left[ x^T(t_f) E x(t_f) + \int_0^{t_f} x^T(t) Q(t) x(t) + u^T(t) R(t) u(t) dt \right] \\ \text{subject to: } \dot{\vec{x}}(t) &= A(t) \vec{x}(t) + B(t) \vec{u}(t) + \vec{w}(t) \\ \vec{y}(t) &= C(t) \vec{x}(t) + \vec{v}(t) \\ \text{initial condition: } \vec{x}_0 &\sim \mathcal{N}(\vec{\mu}_0, \Sigma_0) \end{aligned} \quad (11.5)$$

where  $\vec{w}(t)$  and  $\vec{v}(t)$  are AWGN since we've rewritten the state-space equations without differentials.

To solve this OCP, consider a solution using the following forms. First, the initial condition of the observer is

$$\hat{x}(0) = \mathbb{E} [\vec{x}(0)] = \vec{\mu}_0 \quad (11.6)$$

and evolves according to the Luenberger observer equation

$$\dot{\hat{x}}(t) = A(t) \hat{x}(t) + B(t) \vec{u}(t) + L(t) (\vec{y}(t) - C(t) \hat{x}(t)) \quad (11.7)$$

where the optimal Luenberger gain  $L(t)$  is called the **Kalman gain** of **Kalman filter (KF)** equation. Second, we can write the observer-based feedback controller as

$$\vec{u}(t) = -K(t) \hat{x}(t) \quad (11.8)$$

where  $K(t)$  is the optimal feedback gain matrix. Note that one can solve the LQG OCP by the *separation principle*, i.e. one can “design”  $L(t)$  and  $K(t)$  independently.

Thus, to find the Kalman gain  $L(t)$ , one must solve the corresponding Riccati differential equation for the closed-loop dynamics of the observer, i.e.

$$\dot{P}_L(t) = A(t) P_L(t) + P_L(t) A^T(t) - P_L(t) C^T(t) V^{-1}(t) C(t) P_L(t) + W(t) \quad (11.9)$$

with initial condition

$$P_L(0) = \mathbb{E} [\vec{x}(0)\vec{x}^T(0)] \quad (11.10)$$

where  $W(t)$  is the autocorrelation intensity of the  $\vec{w}(t)$  AWGN and  $V(t)$  is the autocorrelation intensity of the  $\vec{v}(t)$  AWGN. Solving for  $P_L$  provides the Kalman gain as

$$L(t) = P_L(t)C^T(t)V^{-1}(t) \quad (11.11)$$

To find the optimal control  $K(t)$ , one must solve the corresponding Riccati differential equation for the closed-loop dynamics of the controller, i.e.

$$-\dot{P}_K(t) = A(t)P_K(t) + P_K(t)A^T(t) - P_K(t)B(t)R^{-1}(t)B^T(t)P_K(t) + Q(t) \quad (11.12)$$

with final condition

$$P_K(t_f) = F \quad (11.13)$$

Solving for  $P_K(t)$  provides the optimal LQR gain as

$$K(t) = R^{-1}(t)B^T(t)P_K(t) \quad (11.14)$$

To summarize this result, the LQG OCP solution is **separable** as it uses the optimal LQR gain and the Kalman gain, also known as the optimal **Linear-Quadratic Estimator (LQE)**. The solution is obtained by solving the **Dual** OCPs, i.e. solving the  $L(t)$  Riccati differential equation forward in time from 0 and the  $K(t)$  Riccati differential equation backward in time from  $t_f$ . Also, note that at each  $t$ , the Kalman filter generates  $\hat{x}(t)$  using past  $\vec{y}(t)$  and  $\vec{u}$  and the feedback controller generates  $\vec{u}(t)$  using  $\hat{x}(t)$ . Lastly, it should be noted that for LTI and infinite horizon one need only solve the continuous algebraic Riccati equation (CARE) which provides a steady-state  $L$  and  $K$ .

## Discrete Time LQG

The unconstrained finite horizon discrete time LQG OCP is

$$\begin{aligned} \vec{u}_0^*, \dots, \vec{u}_{N-1}^* &= \underset{\vec{u}_0, \dots, \vec{u}_{N-1}}{\operatorname{argmin}} J = \mathbb{E} \left[ \vec{x}_N^T E \vec{x}_N + \sum_{k=0}^{N-1} \vec{x}_k^T Q_k \vec{x}_k + \vec{u}_k^T R_k \vec{u}_k \right] \\ \text{subject to: } &\vec{x}_k = F_k \vec{x}_{k-1} + G_k \vec{u}_{k-1} + \vec{w}_k \\ &\vec{y}_k = H_k \vec{x}_{k-1} + \vec{v}_k \\ \text{initial condition: } &\vec{x}_0 \sim \mathcal{N}(\vec{\mu}, \Sigma_0) \end{aligned} \quad (11.15)$$

where  $\vec{w}_k$  and  $\vec{v}_k$  are AWGN.

To solve this OCP, consider a solution using the following forms. First, the initial condition of the observer is

$$\hat{x}_0 = \mathbb{E} [\vec{x}_0] = \vec{\mu}_0 \quad (11.16)$$

and evolves according to the Luenberger observer equation

$$\hat{x}_k = F_k \hat{x}_{k-1} + G_k \vec{u}_{k-1} + L_k (\vec{y}_k - H_k (F_k \hat{x}_{k-1} + G_k \vec{u}_{k-1})) \quad (11.17)$$

where the optimal Luenberger gain  $L_k$  is called the **Kalman gain** of **Kalman filter (KF)** equation. Second, we can write the observer-based feedback controller as

$$\vec{u}_k = -K_k \hat{x}_k \quad (11.18)$$

where  $K_k$  is the optimal feedback gain matrix. Note that one can solve the LQG OCP by the *separation principle*, i.e. one can “design”  $L_k$  and  $K_k$  independently.

Thus, to find the Kalman gain  $L_k$ , one must solve the corresponding Riccati difference equation for the closed-loop dynamics of the observer, i.e.

$$P_{L,k+1} = F_k \left( P_{L,k} - P_{L,k} H_k^T \left( H_k P_{L,k} H_k^T + V_k \right)^{-1} H_k P_{L,k} \right) F_k^T + W_k \quad (11.19)$$

with initial condition

$$P_{L,0} = \mathbb{E} [(\vec{x}_0 - \hat{x}_0)(\vec{x}_0 - \hat{x}_0)^T] = \Sigma_0 \quad (11.20)$$

where  $W_k$  is the covariance of the  $\vec{w}_k$  AWGN and  $V_k$  is the covariance of the  $\vec{v}_k$  AWGN. Solving for  $P_L$  provides the Kalman gain as

$$L_k = P_{L,k} H_k^T \left( H_k P_{L,k} H_k^T + V_k \right)^{-1} \quad (11.21)$$

To find the optimal control  $K(t)$ , one must solve the corresponding Riccati difference equation for the closed-loop dynamics of the controller, i.e.

$$P_{K,k} = F_k^T \left( P_{K,k+1} - P_{K,k+1} G_k \left( G_k^T P_{K,k+1} G_k + V_k \right)^{-1} G_k^T P_{K,k} \right) F_k + Q_k \quad (11.22)$$

with final condition

$$P_{K,N} = F \quad (11.23)$$

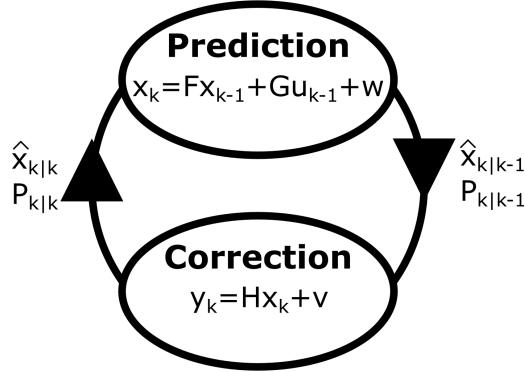
Solving for  $P_{K,k}$  provides the optimal LQR gain as

$$K_k = \left( G_k^T P_{K,k+1} G_k + R_k \right)^{-1} G_k^T P_{K,k+1} F_k \quad (11.24)$$

Note that the unconstrained infinite horizon discrete time LQG OCP would require solving discrete algebraic Riccati equations (DARE) and provide constant steady-state  $L$  and  $K$ .

## The Discrete Time Kalman Filter Revisited

In this section, let's revisit the Riccati difference equation for the Kalman Filter (KF) outside the context of the LQG problem by breaking up it into *two* recursive (i.e. forward-in-time) steps. This will allow us to identify two separate steps, the **Prediction** and **Correction**, that occur in the Kalman filter solution



where we are propagating the observer estimate and Riccati matrices,  $P$ . Note that in this section we'll denote this recursion by the subscript " $i|j$ " which denotes the parameter at step  $i$  given measurements or "corrections" up to step  $j$ .

The prediction step of the Kalman Filter (also known as the Time Update) involves calculating the state estimate and Riccati matrix as

$$\begin{aligned}\hat{x}_{k|k-1} &= F_k \hat{x}_{k-1} + G_k \vec{u}_{k-1} \\ P_{k|k-1} &= F_k P_{k-1|k-1} F_k^T + Q_k\end{aligned}\tag{11.25}$$

where the subscript  $k|k-1$  denotes the *a priori* parameter at  $k$ , i.e. the value prior to the measurement at  $k$ . Also, here we've further changed our notation for the AWGN,  $\vec{w}_k \sim \mathcal{N}(0, Q_k)$ , which in the previous LQG used  $W_k$ , but is typically  $Q_k$  when not considering the LQR.

The correction step of the Kalman Filter (also known as the Measurement Update) involves calculating the following equations

$$\begin{aligned}\tilde{y}_k &= \vec{y}_k - H_k \hat{x}_{k|k-1} \\ K_k &= P_{k|k-1} H_k^T \left( H_k P_{k|k-1} H_k^T + R_k \right)^{-1} \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k \tilde{y}_k \\ P_{k|k} &= (I - K_k H_k) P_{k|k-1}\end{aligned}\tag{11.26}$$

where the subscript  $k|k$  denotes the *a posteriori* parameter at  $k$ , i.e. the value post-measurement at  $k$ . The first equation provides the innovation of the measurement  $\tilde{y}_k$  (also known as the "pre-fit residual").  $K_k$  is now the Kalman gain since we are not considering the LQR problem where it was previously  $L_k$ . Also, the AWGN,  $\vec{v}_k \sim \mathcal{N}(0, R_k)$ , has been altered from the LQG  $V_k$  to the variable  $R_k$ .

Note that for a completely accurate estimator,  $\hat{x}_{0|0}$  and  $P_{0|0}$  must be true, otherwise the Kalman filter will not be optimal. Secondly, the means of the processes are equal to true state, i.e. the Kalman filter is an unbiased estimator.

$$\mathbb{E}[\hat{x}_{k|k-1}] = \vec{x}_k, \quad \mathbb{E}[\hat{x}_{k|k}] = \vec{x}_k, \quad \mathbb{E}[\tilde{y}_k] = 0\tag{11.27}$$

Secondly, it can be shown that the Riccati matrices are the covariances of the state estimates, i.e.

$$\mathbb{E}[(\vec{x}_k - \hat{x}_{k|k-1})(\vec{x}_k - \hat{x}_{k|k-1})^T] = P_{k|k-1}\tag{11.28}$$

and

$$\mathbb{E} [(\vec{x}_k - \hat{x}_{k|k})(\vec{x}_k - \hat{x}_{k|k})^T] = P_{k|k} \quad (11.29)$$

Third, the innovation covariance can be shown to be

$$\text{Cov}(\tilde{y}) = H_k P_{k|k-1} H_k^T + R_k \quad (11.30)$$

which typically is termed  $S_k$ .

The Kalman filter is the *optimal* linear filter if the following three conditions hold

- Model perfectly matches real system
- White noise (uncorrelated)
- Noise covariances are exactly known

However, sub-optimal Kalman filtering methods have also been developed for nonlinear systems which will be discussed later.

An important part of using the Kalman Filter is that if it is optimal, i.e. the parameters match the system, the innovation is a white noise process. Thus, one can often check the innovation for consistency with its expected statistics, i.e. that its covariance is  $S_k$ . This fact can be used for a basic **Adaptive Kalman Filter (AKF)** which alters the unknown  $Q$  or  $R$  to match the observed innovations with the predicted  $S_k$ . The Kalman filter can also be used in the presence of non-Gaussian noise where its performance assessment uses probability inequalities or large-sample theory instead of the true covariance of the state estimate.

Lastly, the Kalman Filter can also be thought of as a Markov chain process built on linear operators that has been perturbed by errors which include Gaussian noise. Namely, the Markov chain state is then a vector of real numbers where the linear operator is applied to the state and input to transition to the new state with noise mixed in. Then, another linear operator mixed with more noise generates observed outputs from true, but “hidden” state. This is analogous to the **hidden Markov model** where the Kalman filter takes state values from a continuous space and a hidden Markov model takes state values from a discrete space.

## 11.2 Introduction to Optimal State Estimation

To this point, we've mainly considered the state equation within our dynamics models, i.e.

$$\vec{x}_k = f(\vec{x}_{k-1}, \vec{u}_{k-1}) \quad (11.31)$$

which is in the nonlinear discrete time form here. With this equation, we've considered how to control the state,  $\vec{x}_k$  using  $\vec{u}_{k-1}$  as a state feedback controller, i.e.

$$\vec{u}_{k-1} = -K \vec{x}_{k-1} \quad (11.32)$$

The next portion of this course will look at the output equation, i.e.

$$\vec{y}_k = h(\vec{x}_k) \quad (11.33)$$

which again is the nonlinear form here. Thus, we have some function of the state itself which can also be a function of the input, but typically it's only of the state. Thus, this equation can be used to perform **state estimation**.

In the form shown, we could simply take the inverse of  $h()$  if it existed. However, in practical applications, we use sensors to *measure* the output which is represented by this output equation. Furthermore, because no sensor is absolutely perfect in measuring the exact and isolated effect of the state,  $\vec{x}$ , we must include an inaccuracy or an uncertainty in our measurement or measurements. This quantity is often also called **measurement noise**, a term from signal processing based on the extra electrical signals in electronic sensors. Another way to think of this inaccuracy is due to the fact that with our physical system representation we are assuming certain things about the physical system which most closely matches the actual sensor system, but the sensor model cannot be exactly quantified at every instant so there is some uncertainty in the measurement. This can be because of calibration or the variations with the individual sensor unit. Thus, since we cannot actually get the exact output, we use a measurement model where we have measurement noise,  $\vec{w}$ , that's been added to the output equation, i.e.

$$\vec{y} = h(\vec{x}, \vec{w}) \quad (11.34)$$

where we've dropped the  $k$  denoting the time step for convenience. In this way our standard state space form has been augmented with this consideration of uncertainty, or measurement noise, in our system. It is this function that we'll investigate in how to estimate the state, given that we receive certain measurements  $\vec{y}$ .

This uncertainty, inaccuracy, or noise is unknown to us, i.e. we cannot predict with absolute certainty what it's exact value will be in the future, otherwise we could just simply combine it into our equations as a parameter. However, though we cannot characterize the precise values of  $\vec{w}$ , we *can* characterize the probabilities or statistics of the observed values of  $\vec{w}$  which leads us to **probability theory** which states  $\vec{w}$  is a **random vector**, the multivariate extension of **random variables**, the content of the rest of this lecture. A random variable a scalar variable whose value depends on the outcomes of some unknown (or “random”) phenomenon which may take values from a discrete or continuous set. A random variable is neither random nor a variable, it's a type of function.

Last lecture we covered the basics of random processes, one of the two foundational principles to state estimation. This lecture will develop the second foundation by looking into the output model using the full linear time-invariant (LTI) models for both continuous time, i.e.

$$\begin{aligned}\dot{\vec{x}}(t) &= A\vec{x}(t) + B\vec{u}(t) \\ \vec{y}(t) &= C\vec{x}(t)\end{aligned}\quad (11.35)$$

and discrete time

$$\begin{aligned}\vec{x}_k &= F\vec{x}_{k-1} + G\vec{u}_{k-1} \\ \vec{y}_k &= H\vec{x}_k\end{aligned}\quad (11.36)$$

where one typically enacts state feedback control in the form of

$$\begin{aligned}\vec{u}(t) &= -K(t)\vec{x}(t) \\ \vec{u}_k &= -K_k\vec{x}_k\end{aligned}\quad (11.37)$$

However, this framework requires that  $\vec{x}$  can be known which is not true for the general state-space model where one **observes** the output vector  $\vec{y}$ . Thus, if  $H \neq I$ , one must compute the state from these observations,

i.e.

$$\hat{x} = \ell(\vec{y}) \quad (11.38)$$

where  $\ell$  is some function of the observations. For our linear model, this can be as simple as taking the inverse (or pseudoinverse of  $H$ ). However, if this is not possible then another simple model would use the state equation model, i.e.

$$\dot{\hat{x}}(t) = A\hat{x}(t) + B\vec{u}(t) \quad (11.39)$$

$$\hat{x}_k = F\hat{x}_{k-1} + B\vec{u}_{k-1} \quad (11.40)$$

which is also known as an **open-loop observer**. Note that this is a deterministic model and has not included any uncertainty as in state estimation.

### Closed-Loop Linear Observer

As we've mentioned earlier, one can combine the state equation and the simple dynamics and observations. Similar to open-loop control, uncertainty in dynamics causes errors in an open-loop observer, thus we desire a closed-loop observer, primarily a linear closed-loop observer such as

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + B\vec{u} + L(\vec{y} - \hat{y}) \\ \hat{y} &= C\hat{x} \end{aligned} \quad (11.41)$$

in continuous time or

$$\begin{aligned} \hat{x}_k &= F\hat{x}_{k-1} + G\vec{u}_{k-1} + L(\vec{y}_{k-1} - \hat{y}_{k-1}) \\ \hat{y}_k &= H\hat{x}_k \end{aligned} \quad (11.42)$$

which is known as the **Luenberger observer**.

For the continuous time LTI system, the observer error can be defined as

$$\vec{e}(t) = \vec{x}(t) - \hat{x}(t) \quad (11.43)$$

Next, by taking the derivative and substituting, we have

$$\dot{\vec{e}}(t) = \vec{x}(t) - \dot{\hat{x}}(t) \quad (11.44)$$

$$\dot{\vec{e}}(t) = A\vec{x} + B\vec{u} - A\hat{x} - B\vec{u} - L(\vec{y} - \hat{y}) \quad (11.45)$$

$$\dot{\vec{e}}(t) = A(\vec{x} - \hat{x}) - L(C\vec{x} - C\hat{x}) \quad (11.46)$$

or finally

$$\dot{\vec{e}}(t) = (A - LC)\vec{e}(t) \quad (11.47)$$

The continuous time Luenberger observer is asymptotically stable, i.e.  $\vec{e} \rightarrow 0$  as  $t/k \rightarrow \infty$  if and only if  $A - LC$  has eigenvalues with positive real part.

Then, one can form the observer-based feedback controller

$$u(t) = -K\hat{x}(t) \quad (11.48)$$

in lieu of the state feedback controller. This can also be rewritten as

$$u(t) = -K(\vec{x}(t) - \vec{e}(t)) \quad (11.49)$$

and the full continuous time closed-loop dynamics become

$$\dot{\vec{x}} = A\vec{x} - BK(\vec{x}(t) - \vec{e}(t)) \quad (11.50)$$

or

$$\dot{\vec{x}} = (A - BK)\vec{x}(t) + BK\vec{e}(t) \quad (11.51)$$

and augmenting with the observer error equation from the Luenberger equations, we have

$$\begin{bmatrix} \dot{\vec{x}} \\ \dot{\vec{e}} \end{bmatrix} = \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} \vec{x} \\ \vec{e} \end{bmatrix} \quad (11.52)$$

which is stable if and only if  $A - BK$  and  $A - LC$  are stable since they are independent in determining the eigenvalues of the augmented system, a property of being an upper triangular matrix. This fact of linear observer-based control having independent criteria for the design of  $L$  and  $K$  is known as the **Separation Principle**, a foundational result in control and estimation theory which justifies the separate use of estimation and control algorithms in modern systems including GNC.

For the discrete time LTI system, the observer error can be defined as

$$\vec{e}_k = \vec{x}_k - \hat{x}_k \quad (11.53)$$

and substituting from the previous time step, we have

$$\vec{e}_k = F\vec{x}_{k-1} + G\vec{u}_{k-1} - F\hat{x}_{k-1} - G\vec{u}_{k-1} - L(\vec{y}_{k-1} - \hat{y}_{k-1}) \quad (11.54)$$

$$\vec{e}_k = F(\vec{x}_{k-1} - \hat{x}_{k-1}) - L(H\vec{x}_{k-1} - H\hat{x}_{k-1}) \quad (11.55)$$

or finally

$$\vec{e}_k = (F - LH)\vec{e}_{k-1} \quad (11.56)$$

and the discrete time Luenberger observer is asymptotically stable, i.e.  $\vec{e} \rightarrow 0$  as  $t/k \rightarrow \infty$ , if and only if  $F - LH$  has eigenvalues of magnitude less than one. Likewise, the discrete time observer-based feedback controller is

$$u_k = -K\hat{x}_k \quad (11.57)$$

which can also be rewritten as

$$u_k = -K(\vec{x}_k - \vec{e}_k) \quad (11.58)$$

and the full discrete time closed-loop dynamics become

$$\vec{x}_{k+1} = F\vec{x}_k - GK(\vec{x}_k - \vec{e}_k) \quad (11.59)$$

$$\vec{x}_{k+1} = (F - GK)\vec{x}_k + GK\vec{e}_k \quad (11.60)$$

and augmenting with observer error equation

$$\begin{bmatrix} \vec{x}_{k+1} \\ \vec{e}_{k+1} \end{bmatrix} = \begin{bmatrix} F - GK & GK \\ 0 & F - LH \end{bmatrix} \begin{bmatrix} \vec{x}_k \\ \vec{e}_k \end{bmatrix} \quad (11.61)$$

which is a stable system if and only if  $F - GK$  and  $F - LH$  are both stable because it is an upper triangular matrix which is also the equivalent **separation principle** for discrete time systems.

## Other Observers

Similar to LTI controllers, one can choose the observer gain arbitrarily high which causes **peaking** in the observer gain, i.e. the initial  $\vec{e}$  is prohibitively large and can create an impractical or unsafe condition in the system.

However, there are nonlinear high gain observer methods available that can converge quickly without peaking such as

- Cubic Observers
- Sliding Mode Observers
- Optimal Stochastic Observers

A **cubic observer** takes the form

$$\dot{\hat{x}} = A\hat{x} + L(y - C\hat{x}) - (y - C\hat{x})^T \theta(y - C\hat{x})N(y - C\hat{x}) \quad (11.62)$$

$$\dot{\vec{e}} = (A - LC)\vec{e} + \vec{e}^T C^T \theta C \vec{e} N C \vec{e} \quad (11.63)$$

and the estimation error dynamics are stable if there exists a symmetric, positive-definite matrix  $P$  satisfying

$$\begin{cases} (A - LC)^T P + P(A - LC) < 0 \\ PNC + C^T N^T P < 0 \end{cases} \quad (11.64)$$

Furthermore, by choosing  $N$  with  $a > 0$ , then

$$N = -aP^{-1}C^T\theta \quad (11.65)$$

A **sliding mode observer** brings the coordinates of the estimator error dynamics to zero in finite time. Intuitively, this uses “infinite” gain to force error dynamics to “slide” along a cross-section of the system’s normal behavior. The main strength of a sliding mode observer is its robustness. This type of observer can be as simple as switching between two states, e.g. “on”/“off” or “forward”/“reverse,” similar to what’s known as **bang-bang control**. At an abstract level, a sliding mode observer uses a finite number of past values and computes an estimate of the state, analogous to the future-looking control horizon in Model Predictive Control (MPC). Likewise, this observer need not be precise and is not sensitive to parameter variations. Furthermore, the estimation error can reach zero in a finite amount of time which is better asymptotic observers. However, the sliding mode observer may not be optimal which is the subject of the next lecture.

## 11.3 Kalman Filter Continued

One can also consider the Kalman filter from a Bayesian inference perspective instead of from an optimal stochastic control perspective through the Linear-Quadratic-Gaussian (LQG) OCP. However, to do so one must know the general framework of the optimal Bayes filter.

## Bayes Filter

The **Bayes filter**, also known as the **recursive Bayesian estimator (RBE)**, is based on Bayes' Rule which for random sequences states with the following notation  $1, 2, 3, \dots k = 1 : k$

$$p(\vec{x}_k | \vec{x}_{1:k-1}) = \frac{p(\vec{x}_{1:k-1} | \vec{x}_k) p(\vec{x}_k)}{p(\vec{x}_{1:k-1})} \quad (11.66)$$

$$p(\vec{x}_k | \vec{x}_{1:k-1}) = \frac{p(\vec{x}_{1:k-1} | \vec{x}_k) p(\vec{x}_k)}{\int_{-\infty}^{\infty} p(\vec{x}_{1:k-1} | \vec{x}_k) p(\vec{x}_k) d\vec{x}_k} \quad (11.67)$$

where Bayes' Rule derives from the joint PDFs, i.e. the probability distribution of all random variables, separated into conditionals and marginals PDFs.

Last lecture we considered the general stochastic state-space model with discrete dynamics of  $\vec{x}$  transitioning at each time step  $k$  while discrete measurements of  $\vec{y}$  are observed at each time step  $k$ . In a Bayesian context, these random sequences can be represented by a conditional PDF for the dynamic state vector at  $k$ , i.e.

$$p(\vec{x}_k | \vec{x}_{1:k-1}, \vec{y}_{1:k}) \quad (11.68)$$

and a conditional PDF for the measurement vector at  $k$

$$p(\vec{y}_k | \vec{x}_{1:k}, \vec{y}_{1:k-1}) \quad (11.69)$$

Likewise, for the general stochastic state-space model, one typically makes the following assumptions of Markovian dynamics, i.e.

$$p(\vec{x}_k | \vec{x}_{1:k-1}, \vec{y}_{1:k}) = p(\vec{x}_k | \vec{x}_{k-1}) \quad (11.70)$$

which represents the Markov property that the current state only depends on the immediately previous state, i.e. a white noise process for the increment between time steps. In addition, one also assumes white measurement noise which implies

$$p(\vec{y}_k | \vec{x}_{1:k}, \vec{y}_{1:k-1}) = p(\vec{y}_k | \vec{x}_k) \quad (11.71)$$

Then, the Bayes Filter can be performed by assuming an initial *a priori* PDF for the state, i.e.  $p(\vec{x}_0)$ . Then, for  $k = 1, 2, \dots$  the **prediction step** computes the *a priori* PDF,  $p(\vec{x}_k | \vec{y}_{1:k-1})$  ("prior" to measurement), and the **correction step** of the *a posteriori* PDF,  $p(\vec{x}_k | \vec{y}_{1:k})$  ("post"-measurement).

For the Bayes filter prediction step, one has knowledge of the *a posteriori* conditional PDF from  $k - 1$

$$p(\vec{x}_{k-1} | \vec{y}_{1:k-1}) \quad (11.72)$$

and the dynamics process conditional PDF for  $k$

$$p(\vec{x}_k | \vec{x}_{k-1}) \quad (11.73)$$

Then, noting that the joint PDF with the current state  $\vec{x}_k$  can be written as

$$\begin{aligned} p(\vec{x}_k, \vec{x}_{k-1} | \vec{y}_{k-1}) &= p(\vec{x}_k | \vec{x}_{k-1}, \vec{y}_{k-1}) p(\vec{x}_{k-1} | \vec{y}_{k-1}) \\ &= p(\vec{x}_k | \vec{x}_{k-1}) p(\vec{x}_{k-1} | \vec{y}_{k-1}) \end{aligned} \quad (11.74)$$

due to the independence between the previous measurement and the current state, the *a priori* conditional PDF at  $k$  can be computed using the **Chapman-Kolmogorov equation**

$$p(\vec{x}_k | \vec{y}_{k-1}) = \int_{-\infty}^{\infty} p(\vec{x}_k | \vec{x}_{k-1}) p(\vec{x}_{k-1} | \vec{y}_{k-1}) d\vec{x}_{k-1} \quad (11.75)$$

For the Bayes filter correction step, one further incorporates knowledge of the conditional measurement PDF at  $k$ , i.e.

$$p(\vec{y}_k | \vec{x}_k) \quad (11.76)$$

and the *a posteriori* conditional PDF can be computed as

$$\begin{aligned} p(\vec{x}_k | \vec{y}_{1:k}) &= \frac{p(\vec{y}_k | \vec{x}_k, \vec{y}_{1:k-1}) p(\vec{x}_k | \vec{y}_{1:k-1})}{\int_{-\infty}^{\infty} p(\vec{y}_k | \vec{x}_k, \vec{y}_{1:k-1}) p(\vec{x}_k | \vec{y}_{1:k-1}) d\vec{x}_k} \\ &= \frac{p(\vec{y}_k | \vec{x}_k) p(\vec{x}_k | \vec{y}_{1:k-1})}{\int_{-\infty}^{\infty} p(\vec{y}_k | \vec{x}_k) p(\vec{x}_k | \vec{y}_{1:k-1}) d\vec{x}_k} \end{aligned} \quad (11.77)$$

which is an application of Bayes' Rule.

Using the final conditional PDF, one can use a variety of different criteria for the optimal state estimate. Three common estimates are the **Maximum A Posteriori (MAP)**, i.e.

$$\hat{x}_k = \underset{\vec{x}}{\operatorname{argmax}} \ p(\vec{x} | \vec{y}_{1:k}) \quad (11.78)$$

the **Maximum Likelihood Estimate (MLE)**, i.e.

$$\hat{x}_k = \underset{\vec{x}}{\operatorname{argmax}} \ p(\vec{y}_{1:k} | \vec{x}) \quad (11.79)$$

and the **Minimum Variance Unbiased Estimate (MVUE)**

$$\begin{aligned} \hat{x}_k &= \underset{\vec{x}}{\operatorname{argmin}} \ \int_{-\infty}^{\infty} (\vec{x} - \mathbb{E}[\hat{x}_k]) (\vec{x} - \mathbb{E}[\hat{x}_k])^T p(\vec{x} | \vec{y}_{1:k}) d\vec{x} \\ \text{subject to: } \mathbb{E}[\hat{x}_k] &= \int_{-\infty}^{\infty} \vec{x} p(\vec{x} | \vec{y}_{1:k}) d\vec{x} \end{aligned} \quad (11.80)$$

### Linear Model with AWGN

Now, let's consider the LTI stochastic discrete time state-space model with Additive White Gaussian Noise (AWGN)

$$\begin{aligned} \vec{x}_k &= F \vec{x}_{k-1} + G \vec{u}_{k-1} + \vec{w}_k \\ \vec{y}_k &= H \vec{x}_k + \vec{v}_k \end{aligned} \quad (11.81)$$

where the process noise is  $\vec{w}_k \sim \mathcal{N}(0, Q)$  and the measurement noise is  $\vec{v}_k \sim \mathcal{N}(0, R)$  which produce the conditional PDFs for the Bayes filter as

$$\begin{aligned} p(\vec{x}_k | \vec{x}_{k-1}) &= \mathcal{N}(\vec{x}_k | F \vec{x}_{k-1} + G \vec{u}_{k-1}, Q) \\ p(\vec{y}_k | \vec{x}_k) &= \mathcal{N}(\vec{y}_k | H \vec{x}_k, R) \end{aligned} \quad (11.82)$$

Furthermore, because multivariate Gaussian random vectors are completely characterized by their mean and covariance

$$p(\vec{x}|\vec{\mu}, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2} (\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu})\right) \quad (11.83)$$

the joint PDF of multivariate Gaussians can be shown to be

$$\begin{bmatrix} \vec{x} \\ \vec{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \vec{\mu}_x \\ \vec{\mu}_y \end{bmatrix}, \begin{bmatrix} \Sigma_x & \Sigma_{xy} \\ \Sigma_{xy}^T & \Sigma_y \end{bmatrix}\right) \quad (11.84)$$

which can be broken into the marginal PDFs

$$\vec{x} \sim \mathcal{N}(\vec{\mu}_x, \Sigma_x) \quad (11.85)$$

$$\vec{y} \sim \mathcal{N}(\vec{\mu}_y, \Sigma_y) \quad (11.86)$$

and conditional PDFs

$$\vec{x}|\vec{y} \sim \mathcal{N}(\vec{\mu}_x + \Sigma_{xy}\Sigma_y^{-1}(\vec{y} - \vec{\mu}_y), \Sigma_x - \Sigma_{xy}\Sigma_y^{-1}\Sigma_{xy}^T) \quad (11.87)$$

$$\vec{y}|\vec{x} \sim \mathcal{N}(\vec{\mu}_y + \Sigma_{xy}^T\Sigma_x^{-1}(\vec{x} - \vec{\mu}_x), \Sigma_y - \Sigma_{xy}^T\Sigma_x^{-1}\Sigma_{xy}) \quad (11.88)$$

This model is the same as the one which produces the Kalman filter which can also be thought of as the Bayes filter for linear dynamics and Gaussian noises. Here, the initial state PDF is modeled as  $\vec{x}_0 \sim \mathcal{N}(\vec{m}_0, P_0)$  and for  $k = 1, 2, \dots$ , the prediction step computes the *a priori* PDF modeled as  $\vec{x}_k|\vec{y}_{1:k-1} = \mathcal{N}(\vec{m}_{k|k-1}, P_{k|k-1})$  while the correction step computes the *a posteriori* PDF modeled as  $\vec{x}_k|\vec{y}_{1:k} = \mathcal{N}(\vec{m}_{k|k}, P_{k|k})$ .

For the Kalman Filter prediction step, one has knowledge of the *a posteriori* PDF from  $k-1$

$$p(\vec{x}_{k-1}|\vec{y}_{1:k-1}) \sim \mathcal{N}(\vec{m}_{k-1|k-1}, P_{k-1|k-1}) \quad (11.89)$$

and the dynamics conditional PDF for  $k$

$$p(\vec{x}_k|\vec{x}_{k-1}) \sim \mathcal{N}(F\vec{m}_{k-1|k-1} + G\vec{u}_{k-1}, Q) \quad (11.90)$$

and then by the **Chapman-Kolmogorov equation** for multivariate Gaussian distributions, one can show that

$$p(\vec{x}_k|\vec{y}_{k-1}) \sim \mathcal{N}\left(F\vec{m}_{k-1|k-1} + G\vec{u}_{k-1}, FP_{k-1|k-1}F^T + Q\right) \quad (11.91)$$

For the Kalman Filter correction step, one also incorporates knowledge of the measurement conditional PDF which can be derived from the joint PDF at  $k$ , i.e.

$$p(\vec{x}_k, \vec{y}_k|\vec{y}_{1:k-1}) \sim \mathcal{N}\left(\begin{bmatrix} \vec{\mu}_{k|k-1} \\ H\vec{\mu}_{k|k-1} \end{bmatrix}, \begin{bmatrix} P_{k|k-1} & P_{k|k-1}H^T \\ HP_{k|k-1} & HP_{k|k-1}H^T + R \end{bmatrix}\right) \quad (11.92)$$

which provides the *a posteriori* PDF

$$p(\vec{x}_k|\vec{y}_{1:k}) = \mathcal{N}\left(\vec{x}_{k|k-1} + K_k\tilde{y}, P_{k|k-1} - K_kS_kK_k^T\right) \quad (11.93)$$

where the innovation is  $\tilde{y} = \vec{y}_k - H\vec{x}_{k|k-1}$  with covariance  $S_k = HP_{k|k-1}H^T + R$ , and the Kalman gain is  $K_k = P_{k|k-1}H^TS_k^{-1}$ .

In this way, one can think of the Kalman gain  $K_k$  as the “transformed” relative weighting between previous  $P$ ,  $Q$ , and  $R$ . To see this consider the combined expression for the Kalman gain for both the prediction and correction steps, i.e.

$$K_k = \left( FP_{k-1|k-1}F^T + Q \right) H^T \left[ H \left( FP_{k-1|k-1}F^T + Q \right) H^T + R \right]^{-1} \quad (11.94)$$

Then, consider the simplified univariate example with

$$P_{k-1|k-1} = \sigma_x^2, \quad F = 1, \quad H = 1, \quad R = \sigma_R^2, \quad Q = \sigma_Q^2 \quad (11.95)$$

$$K_k = \frac{\sigma_x^2 + \sigma_Q^2}{\sigma_x^2 + \sigma_Q^2 + \sigma_R^2} \quad (11.96)$$

which is large when the covariance in the dynamics is relatively larger than the covariance of the measurement, i.e. ( $\sigma_R^2 \ll \sigma_Q^2$ ). One can think of the Kalman filter as optimally balancing the uncertainty one has in the dynamics model vs. the measurement model to estimate the state.

For the Kalman filter, three possible optimality criterion provide the same value because the maximum and minimum variance of multivariate Gaussian PDFs occur at the mean  $\mu_{k|k}$ .

## 11.4 Extended and Iterative Kalman Filters

To begin this section, recall the nonlinear discrete time stochastic state-space equations

$$\begin{aligned} \vec{x}_k &= f(\vec{x}_{k-1}, \vec{u}_{k-1}, \vec{w}_{k-1}) \\ \vec{y}_k &= h(\vec{x}_k, \vec{v}_k) \end{aligned} \quad (11.97)$$

where  $\vec{w}_{k-1}$  is Markovian process noise and  $\vec{v}_k$  is white measurement noise. Without any other assumptions about the statistics of our process, the recursive equations of the Bayes Filter hold, but are only useful if the resulting conditional PDFs have analytical solutions which is very often the case due to the fact that the integration of arbitrary PDFs is not simple.

Thus, nonlinear filtering uses approximation methods. One approximation method is to transform the variables to obtain linear models. However, this is only possible for certain nonlinear models. Another common method is to linearize the state-space models. This can be done in one of two ways. The first is the **Iterative Kalman Filter (IKF)** which linearizes the models about the entire trajectory, solves the Kalman filter equations, and reiterates the entire linearization to find the optimal solution. The second is the **Extended Kalman Filter (EKF)** which recursively linearizes at each time step while simultaneously solving the Kalman Filter. A third approximation method is to approximate the conditional PDFs with sample points (a.k.a. “particles”), if one uses an “optimally” small set of sample points (a.k.a. sigma-points), one can derive the family of **Sigma-Point Kalman Filters (SPKF)** of which the most popular is the **Unscented Kalman Filter (UKF)**. The fourth approximation method is to use a **Particle Filter** which selects *many* particles to form a PMF approximation to the Bayes Filter PDFs. This technique relies heavily on Monte Carlo sampling methods, hence it also being known as **Sequential Monte Carlo**.

This lecture will introduce the last three techniques focusing primarily on the EKF and UKF, the first of which has become the standard algorithm in navigation.

\*

### Extended Kalman Filter

Similar to nonlinear control, linearization techniques can assist in solving some nonlinear problems. These are typically grouped into two different methods. The first is typically distinguished by the term **Iterative** which linearizes the problem over the entire time process and iterates until convergence, e.g. the Iterative Linear-Quadratic Regulator (ILQR), or the Iterative Kalman Filter (IKF). The second is distinguished by the term **Extended** which linearizes recursively with the current “best” value or estimate, e.g. the Extended Linear-Quadratic Regulator (ELQR) or the Extended Kalman Filter (EKF). To note, the “iterative” methods are more computationally expensive than “extended” and one can combine iterative and extended by iterating around the current “best” value or estimate at each step before continuing the recursion. It should also be noted that iteration is used in what is called Kalman Filter “smoothing,” which is similar to model predictive control (MPC), but is backward in time to improve the current estimate.

For the Extended Kalman Filter (EKF), one assumes the following probability models for the nonlinear stochastic state-space

- $\vec{x}_0 \sim \mathcal{N}(\hat{x}_0, P_0)$
- $\vec{w}_k \sim \mathcal{N}(0, Q_k)$
- $\vec{v}_k \sim \mathcal{N}(0, R_k)$

and the linearization of the dynamics and measurement functions is used in the error modeling which is specifically the first order Taylor series expansion.

For the EKF prediction step, one first linearizes the dynamics equation for both the state matrix

$$F_{k-1} = \left[ \frac{\partial f}{\partial \vec{x}} \right]_{\vec{x}=\hat{x}_{k-1|k-1}, \vec{u}=\vec{u}_{k-1}, \vec{w}=0} \quad (11.98)$$

the input matrix

$$G_{k-1} = \left[ \frac{\partial f}{\partial \vec{u}} \right]_{\vec{x}=\hat{x}_{k-1|k-1}, \vec{u}=\vec{u}_{k-1}, \vec{w}=0} \quad (11.99)$$

and the process noise matrix

$$L_{k-1} = \left[ \frac{\partial f}{\partial \vec{w}} \right]_{\vec{x}=\hat{x}_{k-1|k-1}, \vec{u}=\vec{u}_{k-1}, \vec{w}=0} \quad (11.100)$$

Then, the *a priori* mean is computed using the exact nonlinear equation

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, \vec{u}_{k-1}, 0) \quad (11.101)$$

and the *a priori* error covariance is approximately

$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + L_{k-1} Q_{k-1} L_{k-1}^T \quad (11.102)$$

For the EKF correction step, one linearizes the measurement equation for the output matrix

$$H_k = \left[ \frac{\partial h}{\partial \vec{x}} \right]_{\vec{x}=\hat{x}_{k|k-1}, \vec{v}=0} \quad (11.103)$$

and the measurement noise matrix

$$M_k = \left[ \frac{\partial h}{\partial \vec{v}} \right]_{\vec{x}=\hat{x}_{k|k-1}, \vec{v}=0} \quad (11.104)$$

Next, one computes the exact innovation between the measurement and the expected measurement

$$\tilde{y}_k = \vec{y}_k - h(\vec{x}_{k|k-1}, 0) \quad (11.105)$$

Then, one computes the approximate innovation error covariance

$$S_k = H_k P_{k|k-1} H_h^T + M_k R_k M_k^T \quad (11.106)$$

which leads to the computation of the approximately optimal Kalman gain as

$$K_k = P_{k|k-1} H_k^T S_k^{-1} \quad (11.107)$$

The approximate optimal fusion of the *a priori* estimate with the innovation provides the *a posteriori* mean

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \quad (11.108)$$

and the approximate *A posteriori* error covariance

$$P_{k|k} = P_{k|k-1} - K_k S_k K_k^T \quad (11.109)$$

To reiterate, the EKF is *not* an optimal estimator. Furthermore, if the linearization is too poor, the estimator may diverge from the true state. This can occur from a poor initial estimate or the process/measurement models being too poor. Lastly, it should be noted that the EKF tends to underestimate the true covariance. However, regardless of these shortcomings, the EKF is the *de facto standard* in navigation.

# Chapter 12

## Advanced Optimal State Estimation

### 12.1 Sigma-Point Kalman Filters

The next most common nonlinear filtering technique views the nonlinearity approximation in a different manner. To explain, one can think of the EKF as using the linearized function with partial distribution information, i.e. mean and covariance statistics, which applies the probability distribution statistics through the approximate nonlinear function. Alternatively, the Sigma-Point Kalman Filters (SPKF) apply an approximate probability distribution to the exact nonlinear functions. This is done by encoding the exact mean and covariance as an optimal set of points **sigma points** and is analogous to a PMF approximation with a consistent mean and covariance to the PDF. Then, the SPKF functions propagate the PMF approximation exactly through the nonlinear functions using the **Unscented Transform (UT)**, which guarantees that the mean and covariance of the propagated sigma-points optimally approximate the true mean and covariance. This technique has two distinct advantages by fully exploiting the nonlinear functions and by eliminating the linearization computation, this can improve the estimate quality, especially of the covariances.

To select a suitable set of sigma-points, assume one is given an  $n \times 1$  mean,  $\vec{m}$ , and an  $n \times n$  covariance matrix,  $P$ . One then subtracts off the mean and computes a symmetric set of  $2n + 1$  sigma-points ( $\mathbb{R}^{n \times 1}$  vectors) which uses the zero vector, and the  $n$  columns of  $\pm\sqrt{n}P$ . Note: for the sigma-points to represent a proper PMF, one must assign weights  $W$  to each sigma-point such that

$$\sum_{i=1}^{2n+1} W_i = 1 \quad (12.1)$$

and these weights must be assigned such that the mean and covariance of the PMF of sigma-points match the true mean and covariance.

The family of Sigma-Point Kalman Filters includes the

- Unscented Kalman Filter (UKF)
- Central Difference Kalman Filter (CDKF)
- Divided Difference Kalman Filter (DDKF)
- Square-Root alternatives of UKF and CDKF

In the UKF prediction step, for the indexes  $i = 1, 2, \dots, n$  one forms the  $2n$  sigma-points by first using the equations

$$\tilde{x}_{i,k-1} = \left( \sqrt{n P_{k-1|k-1}} \right)_i \quad (12.2)$$

$$\tilde{x}_{i+n,k-1} = -\left( \sqrt{n P_{k-1|k-1}} \right)_i \quad (12.3)$$

where  $(A)_i$  represents the  $i^{\text{th}}$  column of  $A$ . Then, the  $2n + 1$  sigma-points are generated by

$$\hat{x}_{i,k-1|k-1} = \hat{x}_{k-1|k-1} + \tilde{x}_{i,k-1}, \quad i = 1, 2, \dots, 2n \quad (12.4)$$

and

$$\hat{x}_{2n+1,k-1|k-1} = \hat{x}_{k-1|k-1} \quad (12.5)$$

Next, one propagates these sigma points through the exact dynamics equation

$$\hat{x}_{i,k|k-1} = f(\hat{x}_{i,k-1|k-1}, \vec{u}_{k-1}) \quad (12.6)$$

and from the resulting *a priori* sigma-points, one can compute the *a priori* mean as

$$\hat{x}_{k|k-1} = \frac{1}{2n+1} \sum_{i=1}^{2n+1} \hat{x}_{i,k} \quad (12.7)$$

and the *a priori* covariance as

$$P_{k|k-1} = Q_{k-1} + \frac{1}{2n+1} \sum_{i=1}^{2n+1} (\hat{x}_{i,k|k-1} - \hat{x}_{k|k-1}) (\hat{x}_{i,k|k-1} - \hat{x}_{k|k-1})^T \quad (12.8)$$

In the correction step, it is optional to resample the *a priori* sigma points from  $\hat{x}_{k|k-1}$  and  $P_{k|k-1}$  before continuing to propagate sigma points through measurement equation

$$\tilde{y}_{i,k} = h(\hat{x}_{i,k|k-1}) \quad (12.9)$$

to get the approximate innovation

$$\tilde{y}_k = \frac{1}{2n+1} \sum_{i=1}^{2n+1} \tilde{y}_{i,k} \quad (12.10)$$

and the measurement covariance

$$P_y = R_k + \frac{1}{2n+1} \sum_{i=1}^{2n+1} (\tilde{y}_{i,k} - \tilde{y}_k) (\tilde{y}_{i,k} - \tilde{y}_k)^T \quad (12.11)$$

Next, one can compute the measurement-state cross-covariance as

$$P_{xy} = \frac{1}{2n+1} \sum_{i=1}^{2n+1} (\hat{x}_{i,k|k-1} - \hat{x}_{k|k-1}) (\tilde{y}_{i,k} - \tilde{y}_k)^T \quad (12.12)$$

Then, the Kalman gain is given by the definition

$$K_k = P_{xy}P_y^{-1} \quad (12.13)$$

which allows one to compute the *a posteriori* mean

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (\vec{y}_k - \tilde{y}_k) \quad (12.14)$$

and the *a posteriori* covariance

$$P_{k|k} = P_{k|k-1} - K_k S_y K_k^T \quad (12.15)$$

In closing, it can be shown that the UKF is equivalent to a third order Taylor Series expansion of the nonlinear stochastic functions, thus one can simply use the EKF when the nonlinearities are small, e.g. GNSS multilateration with satellites (i.e. radio beacons) that are far away from the receivers. In addition, second order EKFs have also been studied which only provide benefits for small  $R$  and are not as common as the EKF or UKF. Lastly, it should be noted that for *very* large state dimensions, the UKF can be computationally faster than the EKF since no Jacobian matrices must be calculated and no matrix inversion of the state dimension is necessary.

## 12.2 Sigma-Point Kalman Filters Continued

### 12.3 Particle Filter

The particle filter is also known as Sequential Monte Carlo (SMC) which is a general description of assessing probabilities by performing many “games” to find the probabilities of certain processes, thus making it a sort of brute force method. In contrast to SPKFs, PFs use *many* sample points which in this context are called particles. By many, this can be on the order of 100 to 10,000. PFs also heavily uses resampling similar to SPKF, but typically more complex in order to resample intelligently, a requirement that can be difficult. One of the primary advantages for the PF is that it can be used for *any* probability distribution. One of the primary disadvantages, other than computational cost, is the phenomenon of **particle depletion** which describes the fact that after many time steps, particles tend to combine into less and less unique particles due to the resampling methods. However, modern techniques have been developed to overcome these resampling issues, e.g. Markov Chain Monte Carlo (MCMC).

A basic Particle Filter outline is as follows

- Sample  $N$  particles by the dynamics probability distribution for  $i = 1, \dots, N$

$$\hat{x}_{i,k|k-1} \sim p(\vec{x}_k | \hat{x}_{i,k-1}) \quad (12.16)$$

- Compute likelihood weights for  $i = 1, \dots, N$

$$q_i = \frac{p(\vec{y}_k | \vec{x}_{i,k|k-1})}{\sum_{j=1}^N p(\vec{y}_k | \vec{x}_{j,k|k-1})} \quad (12.17)$$

- Resample  $\hat{x}_{i,k|k-1}$  using the  $q_i$  for  $j = 1, \dots, N$  new particles

$$\begin{aligned}
 & \text{sample } p_j \text{ from } \mathcal{U}(0, 1) \\
 & \underset{i}{\operatorname{argmax}} q_i < p_j \\
 & \hat{x}_{j,k|k} = \hat{x}_{i,k|k-1}
 \end{aligned} \tag{12.18}$$

- Form MAP/MLE/MVUE/MMSE from  $\hat{x}_{j,k|k}$ 's

## 12.4 Particle Filter Continued

# Chapter 13

## Advanced Optimal Control

### 13.1 Advanced Methods of Optimization

With the advent of modern computers, **direct methods** have become more prevalent than indirect methods for solving the OCP because the approximations necessary for such methods are often more reliable than indirect approximations for complex problems. For direct methods, three approximations must be made: the integration in cost functional must be approximated, the differential equation of the nonlinear dynamics, and the constraints in the state and/or control input. An ideal approximation direct method would be efficient for all three types, but because of the variety of problems, different methods have been developed, each with their own strengths and weaknesses. The basic approach here is to model the cost functional and dynamics as *functions*, typically piecewise functions, polynomials, or piecewise polynomials. Then, the parameters of these functions become the optimization variables  $\vec{z}$  of a new OCP of the form

$$\begin{aligned}\vec{z}^* &= \operatorname{argmin}_{\vec{z}} J(\vec{z}) \\ \text{subject to: } &\quad g(\vec{z}) = 0 \\ &\quad h(\vec{z}) \leq 0\end{aligned}\tag{13.1}$$

where  $g()$  and  $h()$  represent equality and inequality constraints that can be enforced for the function parameters. Thus, these direct methods can be thought of as a type of optimal parameter problem.

Most methods fall into one of three types, each with varying computational requirements: **direct shooting** which can be single or multiple, **pseudospectral** which uses particular functions, and **direct collocation** which solves the entire problem simultaneously. Another reason for using direct methods rather than indirect is their improved ability to handle inequality and multi-point constraints.

### 13.2 Convex Optimization in Control

#### Convex Optimization

Convex versus non-convex optimization and definitions.

## Semidefinite Programming

### 13.3 Optimal Control for LPV Systems

### 13.4 Receding Horizon Control

Before modern computers and sophisticated algorithms were available that could reasonably solve the variety of constrained OCPs, a sub-optimal approach was adopted by many control engineers that has seen wide use in the control community. In this method, instead of attempting to solve the OCP for the full control sequence over the entire finite time horizon, receding horizon control (RHC) or, as it's better known as, model predictive control (MPC), optimizes the control input over a much shorter **control horizon** of length  $M$  and re-optimizes at *every* time step of the control process. It also enforces that state and input constraints remain true over the entire finite horizon, a.k.a. the **prediction horizon**. The name RHC derives from the control and prediction horizons receding with each time step while the name MPC derives from the optimization method using a dynamics model to predict the state forward in time for the OCP solved at every time step.

The general MPC OCP can be constructed as

$$\begin{aligned} \vec{u}_0^*, \dots, \vec{u}_{M-1}^* &= \underset{\vec{u}}{\operatorname{argmin}} \quad J = E(\vec{x}_M) + \sum_{k=0}^{M-1} L(\vec{x}_k, \vec{u}_k) \\ \text{subject to: } \vec{x}_k &= f(\vec{x}_{k-1}, \vec{u}_{k-1}) \\ \text{initial condition: } \vec{x}_0 & \\ \text{state constraints: } \vec{x}_k &\in X_k \quad \forall 0 \leq k \leq N \\ \text{input constraints: } \vec{u}_k &\in \mathcal{U}_k \quad \forall 0 \leq k \leq M \end{aligned} \tag{13.2}$$

where  $L$  is the Lagrangian and must be nonnegative. It is important to note that the state constraints are enforced over the prediction horizon and the input constraints are enforced over the control horizon. In addition, for  $M \leq k \leq N$ , the control input  $u$ , must be specified. Remember, that this OCP is solved every time step in an MPC formulation.

#### Linear-Quadratic MPC

In this course, we will consider linear-quadratic MPC, though nonlinear MPC (NMPC) is of growing interest in aircraft systems.

The linear-quadratic MPC OCP can be written as

$$\begin{aligned}
 \vec{u}_0^*, \dots, \vec{u}_{M-1}^* = \underset{\vec{u}}{\operatorname{argmin}} \quad J &= (\vec{x}_M - \vec{x}_{des,M})^T E (\vec{x}_M - \vec{x}_{des,M}) \\
 &\quad + \sum_{k=0}^{M-1} (\vec{x}_k - \vec{x}_{des,k})^T Q (\vec{x}_k - \vec{x}_{des,k}) \\
 &\quad + (\vec{u}_k - \vec{u}_{des,k})^T R (\vec{u}_k - \vec{u}_{des,k}) \tag{13.3} \\
 \text{subject to: } \vec{x}_k &= F \vec{x}_{k-1} + G \vec{u}_{k-1} \\
 \text{initial condition: } \vec{x}_0 & \\
 \text{state constraints: } \vec{x}_{min} &\leq \vec{x}_k \leq \vec{x}_{max} \quad \forall 0 \leq k \leq N \\
 \text{input constraints: } \vec{u}_{min} &\leq \vec{u}_k \leq \vec{u}_{max} \quad \forall 0 \leq k \leq M
 \end{aligned}$$

where  $\vec{x}_{des,k}$  and  $\vec{u}_{des,k}$  provide the reference trajectory for the system. These may be constant or vary with  $k$ , but they need to be consistent with the plant model

$$\vec{x}_{des,k} = F \vec{x}_{des,k-1} + G \vec{u}_{des,k} \tag{13.4}$$

Furthermore, it is should be assumed that

$$\vec{u}_k = \vec{u}_{des,k} \quad \forall M \leq k \leq N-1 \tag{13.5}$$

To incorporate integral action for the MPC, one typically uses the input *changes*, i.e.

$$\Delta \vec{u}_k = \vec{u}_k - \vec{u}_{k-1} \tag{13.6}$$

where  $\Delta \vec{u}_k$  are now the free variables to optimize. This produces the following derived state-space model

$$\begin{bmatrix} \vec{x}_{k+1} \\ \vec{u}_k \end{bmatrix} = \begin{bmatrix} F & G \\ 0 & I \end{bmatrix} \begin{bmatrix} \vec{x}_k \\ \vec{u}_{k-1} \end{bmatrix} + \begin{bmatrix} G \\ I \end{bmatrix} \Delta \vec{u}_k \tag{13.7}$$

$$y_k = [H \quad 0] \begin{bmatrix} \vec{x}_k \\ \vec{u}_k \end{bmatrix} \tag{13.8}$$

Remember that  $E, P, R$  are assumed to be symmetric and positive semi-definite. Notice that we've dropped the  $S$  cross-weight matrix in this formulation.

MPC must be computed *on-line* as the system runs and cannot be precomputed. Thus, MPC serves as an open-loop control scheme since it recomputes its action at every time step and does not explicitly provide a control law for the feedback loop, though it does reuse the “new” state as the “new” initial in its re-optimization of the problem every time step. This feature of MPC may be desirable if the system dynamics are not exactly represented in the model (either through model inaccuracy, especially nonlinearity). Thus, MPC continues to be used in many circumstances.

## Rewriting the MCP Cost Function

To formulate the MPC solution for an LTI state-space system with a quadratic cost function, one typically uses quadratic programming of which the LLS problem is one type.

Then, the cost function can be written as

$$J = (\vec{X} - \vec{X}_{ref})^T \bar{Q} (\vec{X} - \vec{X}_{ref}) + (\vec{U} - \vec{U}_{ref})^T \bar{R} (\vec{U} - \vec{U}_{ref}) \quad (13.9)$$

the state trajectory as

$$\vec{X} = \bar{F} \vec{x}_0 + \bar{G} \vec{U} \quad (13.10)$$

and the stacked nominal state trajectory as

$$\vec{X}_0 = \bar{F} \vec{x}_0 + \bar{G} \vec{U}_{ref} - \vec{X}_{ref} \quad (13.11)$$

which assumes the reference input is used at every time step.

Next, the deviation in the stacked states from the references can be rewritten as

$$\vec{X} - \vec{X}_{ref} = \bar{F} \vec{x}_0 + \bar{G} \vec{U} - \vec{X}_{ref} \quad (13.12)$$

$$\vec{X} - \vec{X}_{ref} = \vec{X}_0 - \bar{G} \vec{U}_{ref} + \bar{G} \vec{U} \quad (13.13)$$

and by assigning the **QP free vector** as the deviation between free inputs and the reference input

$$\vec{v} = \vec{U} - \vec{U}_{ref} \quad (13.14)$$

we have

$$\vec{X} - \vec{X}_{ref} = \vec{X}_0 + \bar{G} \vec{v} \quad (13.15)$$

and the cost function can be rewritten as

$$J = (\vec{x}_0 - \vec{x}_{ref,0})^T Q (\vec{x} - \vec{x}_{ref,0}) + (\vec{X}_0 + \bar{G} \vec{v})^T \bar{Q} (\vec{X}_0 + \bar{G} \vec{v}) + \vec{v}^T \bar{R} \vec{v} \quad (13.16)$$

$$J = (\vec{x}_0 - \vec{x}_{ref,0})^T Q (\vec{x} - \vec{x}_{ref,0}) + \vec{X}_0^T \bar{Q} \vec{X}_0 + 2 \vec{X}_0^T \bar{Q} \bar{G} \vec{v} + \vec{v}^T \bar{G}^T \bar{Q} \bar{G} \vec{v} + \vec{v}^T \bar{R} \vec{v} \quad (13.17)$$

Finally, noting that the first two terms can be ignored since they are not affected by the choice of  $\vec{v}$  we can assign

$$\tilde{H} = \bar{G}^T \bar{Q} \bar{G} + \bar{R} \quad (13.18)$$

and

$$c^T = \vec{X}_0^T \bar{Q} \bar{G} \quad (13.19)$$

to rewrite the cost function in the form of the QP cost function. By inspection, we see that the only parameter that changes in this optimization from one time step to the next is the term  $\vec{X}_0$  which change with  $\vec{x}_0$ . This state corresponds to the  $\vec{x}_1$  we predicted by implementing the first control input  $u_0$  from the previous time step's optimization.

### Rewriting the MPC Constraints

In order to put the inequality constraints into the QP form above, stacking techniques can be used. For the input constraints,

$$\begin{bmatrix} \vec{u}_{min} \\ \vdots \\ \vec{u}_{min} \end{bmatrix} - \vec{U}_{ref} \leq I\vec{v} \leq \begin{bmatrix} \vec{u}_{max} \\ \vdots \\ \vec{u}_{max} \end{bmatrix} - \vec{U}_{ref} \quad (13.20)$$

for the state constraints for  $1 \leq k \leq M$ ,

$$\begin{bmatrix} \vec{x}_{min} \\ \vdots \\ \vec{x}_{min} \end{bmatrix} - (\vec{X}_0 + \vec{X}_{ref}) \leq \vec{G}\vec{v} \leq \begin{bmatrix} \vec{x}_{max} \\ \vdots \\ \vec{x}_{max} \end{bmatrix} - (\vec{X}_0 + \vec{X}_{ref}) \quad (13.21)$$

For the state constraints for  $M+1 \leq k \leq N$ ,

$$\begin{bmatrix} \vec{x}_{min} \\ \vdots \\ \vec{x}_{min} \end{bmatrix} \leq \begin{bmatrix} \vec{x}_{M+1} \\ \vdots \\ \vec{x}_N \end{bmatrix} \leq \begin{bmatrix} \vec{x}_{max} \\ \vdots \\ \vec{x}_{max} \end{bmatrix} \quad (13.22)$$

where the stacked states can be written as

$$\begin{bmatrix} \vec{x}_{M+1} \\ \vdots \\ \vec{x}_N \end{bmatrix} = \begin{bmatrix} F \\ \vdots \\ F^{N-M} \end{bmatrix} \vec{x}_M + \begin{bmatrix} G & \cdots & 0 \\ \vdots & \ddots & \vdots \\ F^{N-M-1}G & \cdots & G \end{bmatrix} \begin{bmatrix} \vec{u}_{ref} \\ \vdots \\ \vec{u}_{ref} \end{bmatrix} \quad (13.23)$$

and using the solution for the state at time step  $M$

$$\vec{x}_M = F^M \vec{x}_0 + [F^{M-1}G \ \cdots \ G] \vec{U} \quad (13.24)$$

we can write

$$\begin{bmatrix} \vec{x}_{M+1} \\ \vdots \\ \vec{x}_N \end{bmatrix} = \begin{bmatrix} F \\ \vdots \\ F^{N-M} \end{bmatrix} \left( F^M \vec{x}_0 + [F^{M-1}G \ \cdots \ G] \vec{U} \right) + \begin{bmatrix} G & \cdots & 0 \\ \vdots & \ddots & \vdots \\ F^{N-M-1}G & \cdots & G \end{bmatrix} \begin{bmatrix} \vec{u}_{ref} \\ \vdots \\ \vec{u}_{ref} \end{bmatrix} \quad (13.25)$$

Then, after substitution and rearrangement, the state constraints for  $M+1 \leq k \leq N$ , can finally be put into the QP form as

$$\begin{aligned} & \begin{bmatrix} \vec{x}_{min} \\ \vdots \\ \vec{x}_{min} \end{bmatrix} - \begin{bmatrix} F \\ \vdots \\ F^{N-M} \end{bmatrix} F^M \vec{x}_0 - \begin{bmatrix} G & \cdots & 0 \\ \vdots & \ddots & \vdots \\ F^{N-M-1}G & \cdots & G \end{bmatrix} \begin{bmatrix} \vec{u}_{ref} \\ \vdots \\ \vec{u}_{ref} \end{bmatrix} - \begin{bmatrix} F \\ \vdots \\ F^{N-M} \end{bmatrix} [F^{M-1}G \ \cdots \ G] \vec{U}_{ref} \\ & \leq \begin{bmatrix} F \\ \vdots \\ F^{N-M} \end{bmatrix} [F^{M-1}G \ \cdots \ G] \vec{v} \leq \\ & \begin{bmatrix} \vec{x}_{max} \\ \vdots \\ \vec{x}_{max} \end{bmatrix} - \begin{bmatrix} F \\ \vdots \\ F^{N-M} \end{bmatrix} F^M \vec{x}_0 - \begin{bmatrix} G & \cdots & 0 \\ \vdots & \ddots & \vdots \\ F^{N-M-1}G & \cdots & G \end{bmatrix} \begin{bmatrix} \vec{u}_{ref} \\ \vdots \\ \vec{u}_{ref} \end{bmatrix} - \begin{bmatrix} F \\ \vdots \\ F^{N-M} \end{bmatrix} [F^{M-1}G \ \cdots \ G] \vec{U}_{ref} \end{aligned} \quad (13.26)$$

It should also be noted that rate constraints on the inputs can also be incorporated into the QP problem form in a similar fashion, the minimum and maximum values for  $\vec{x}$  and  $\vec{u}$  can be dependent on  $k$ , and equality constraints can be represented by choosing the element of  $\vec{b}_L$  equal to the same element in  $\vec{b}_U$  for some  $k$ .

## Python

Though SciPy does not offer an explicit QP solver, for these types of algorithms, one can use the generic `scipy.optimize.minimize()` function which has many available methods. For constrained optimization, it offers Constrained Optimization BY Linear Approximation, (COBYLA), Sequential Least SQuares Programming (SLSQP) and an interior point trust-region algorithm for constrained optimization (`trust-constr`). Each of these methods work for nonlinear systems as well. For the linear-quadratic problem, the `trust-constr` method is recommended since it allows the user to precompute the gradient and Hessians are as functions which can be passed to `jac` and `hess` keyworded arguments, respectively.

## **Part III**

# **Advanced Flight Dynamics and Control**

# Chapter 14

## Advanced Rigid Flight Vehicle Dynamics

### 14.1 Mass Effects on Flight Dynamics

This section will discuss two mass effects for flight vehicle dynamics, namely the presence of rotating masses, e.g. propellers, and variable mass, e.g. rocket engines. Both of these will effects will result in *additive* terms for the nonlinear equations of motion for flight vehicles. Thus, including these mass effects (and additional relevant states) can easily be added or neglected in the state-space model for flight dynamics where the linearized dynamics will need to expand the dimension of the LTI state-space model. Note that this section will develop these effects as additions to the rigid flight vehicle EOMs as expressed in the navigation frame using a flat-Earth model.

#### Rotating Mass Effect

The presence of a mass rotating about its center of mass can be shown to have no effect on the translation equations of motion of a vehicle. However, the presence of a mass rotating about its center of mass does have an effect on the rotation equations of motion of a vehicle as this rotation directly contributes to the total angular momentum of the vehicle. To show this, consider the angular momentum of the vehicle as

$$\begin{bmatrix} I_{xx}L \\ I_{yy}M \\ I_{zz}N \end{bmatrix} = \dot{\vec{H}}_N = \dot{\vec{H}}_B + \vec{\omega}_{B \leftarrow N} \times \vec{H}_B \quad (14.1)$$

where  $\vec{H}_B$  consists of two components, the rigid vehicle's angular momentum (with the mass of the propeller disk) and the propeller's angular momentum, i.e.

$$\vec{H}_B = \vec{H}_{rig,B} + \vec{H}_{prop,B} \quad (14.2)$$

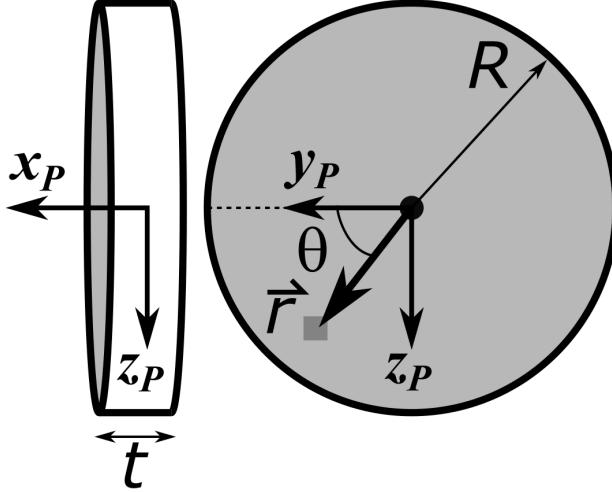
where

$$\vec{H}_{rig,B} = I_G \times \vec{\omega}_{B \leftarrow N} \quad (14.3)$$

and the general form for the propeller angular momentum is

$$\vec{H}_{prop,B} = \int_{Vol} \vec{x}_B \times \rho_V \vec{x}_B dV \quad (14.4)$$

where  $\vec{x}_B$  is the radial position of a mass element  $\rho_V dV$  with respect to the center of mass. Next, one can idealize the propeller as a rotating disk with radius  $R$  and constant thickness  $t$  as



where the  $x_P - y_P - z_P$  axes denotes the propeller-fixed frame centered at the disk's center (subscript  $P$ ), the mass element volume at radius  $r$  is

$$dV = rtd\theta dr , \quad (14.5)$$

and the disk has a radial mass distribution matching that of the propeller, i.e. setting the disk density,  $\rho_{disk}$ , to cover the entire disk volume, but with equivalent mass. Then, one can write the mass element velocity as

$$\dot{\vec{x}}_{B'} = \dot{\vec{x}}_P + \vec{\omega}_{B' \leftarrow P} \times \vec{x}_P \quad (14.6)$$

where  $B'$  here is a body-fixed frame centered at the propeller. Assuming the propeller disk is rigid, this can be rewritten as

$$\begin{aligned} \dot{\vec{x}}_{B'} &= \vec{\omega}_{I \leftarrow P} \times \vec{x}_P = \begin{bmatrix} \omega_{prop} \\ 0 \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ r \cos \theta \\ r \sin \theta \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ \omega_{prop} r \sin \theta \\ -\omega_{prop} r \cos \theta \end{bmatrix} \end{aligned} \quad (14.7)$$

Thus, one has

$$\begin{aligned} \vec{x}_{B'} \times \dot{\vec{x}}_{B'} &= \rho_V \begin{bmatrix} 0 \\ r \cos \theta \\ r \sin \theta \end{bmatrix} \times \begin{bmatrix} 0 \\ \omega_{prop} r \sin \theta \\ -\omega_{prop} r \cos \theta \end{bmatrix} \\ &= \begin{bmatrix} \omega_{prop} r^2 \\ 0 \\ 0 \end{bmatrix} \end{aligned} \quad (14.8)$$

and the angular momentum of the propeller disk in body frame coordinates is

$$\begin{aligned}\vec{H}_{prop,B'} &= \begin{bmatrix} \int_0^R \int_0^{2\pi} \omega_{prop} r^2 \rho_{disk} (r t d\theta dr) \\ 0 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \omega_{prop} 2\pi t \int_0^R r^3 \rho_{disk} dr \\ 0 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \omega_{prop} I_{prop} \\ 0 \\ 0 \end{bmatrix}\end{aligned}\quad (14.9)$$

where  $I_{prop}$  is the moment of inertia of the propeller about its center of mass.

Next, returning to the propeller angular momentum in the vehicle body frame centered at the center of mass, one has

$$\vec{H}_{prop,B} = \begin{bmatrix} h_{x,prop} \\ h_{y,prop} \\ h_{z,prop} \end{bmatrix} = C_{B \leftarrow B'} \begin{bmatrix} \omega_{prop} I_{prop} \\ 0 \\ 0 \end{bmatrix} \quad (14.10)$$

where the DCM,  $C_{B \leftarrow B'}$ , will depend on the orientation of the propeller relative to the body frame. For example, if the  $B'$  at some positive rotation angle,  $\tau_P$ , about the  $y_B$  axis, then

$$\vec{H}_{prop,B} = \begin{bmatrix} \omega_{prop} I_{prop} \cos \tau_P \\ 0 \\ -\omega_{prop} I_{prop} \sin \tau_P \end{bmatrix} \quad (14.11)$$

Then, by substituting the additive angular momentum terms into the angular momentum differential equation, one has

$$\begin{bmatrix} I_{xx}L \\ I_{yy}M \\ I_{zz}N \end{bmatrix} = (\dot{\vec{H}}_{rig,B} + \dot{\vec{H}}_{prop,B}) + \vec{\omega}_{B \leftarrow N} \times (\vec{H}_{rig,B} + \vec{H}_{prop,B}) \quad (14.12)$$

$$\begin{bmatrix} I_{xx}L \\ I_{yy}M \\ I_{zz}N \end{bmatrix} = (\dot{\vec{H}}_{rig,B} + \vec{\omega}_{B \leftarrow N} \times \vec{H}_{rig,B}) + (\dot{\vec{H}}_{prop,B} + \vec{\omega}_{B \leftarrow N} \times \vec{H}_{prop,B}) \quad (14.13)$$

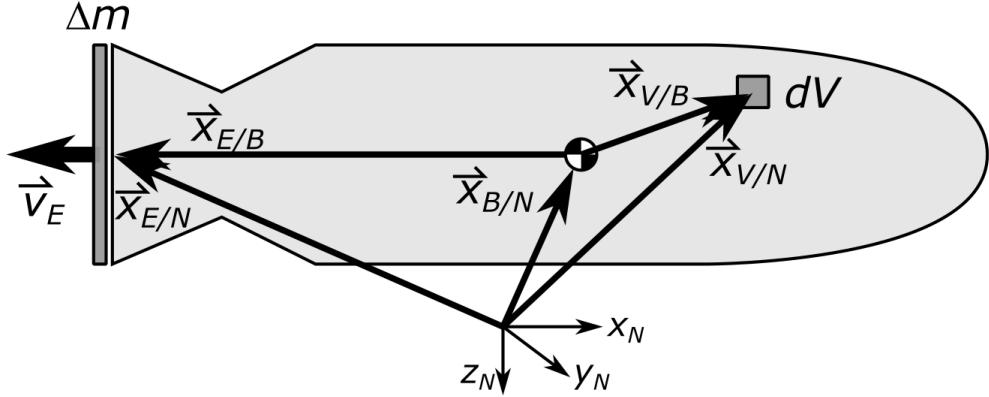
$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} \dot{p} + \frac{I_{zz} - I_{yy}}{I_{xx}} qr - \frac{I_{xz}}{I_{xx}} (\dot{r} + pq) \\ \dot{q} + \frac{I_{xx} - I_{zz}}{I_{yy}} pr - \frac{I_{xy}}{I_{yy}} (r^2 - p^2) \\ \dot{r} + \frac{I_{yy} - I_{xx}}{I_{zz}} pq - \frac{I_{xz}}{I_{zz}} (\dot{p} - qr) \end{bmatrix} + \begin{bmatrix} \frac{1}{I_{xx}} (\dot{h}_{x,prop} - rh_{y,prop} + qh_{z,prop}) \\ \frac{1}{I_{yy}} (\dot{h}_{y,prop} + rh_{x,prop} - ph_{z,prop}) \\ \frac{1}{I_{zz}} (\dot{h}_{z,prop} - qh_{x,prop} + ph_{y,prop}) \end{bmatrix} \quad (14.14)$$

Thus, the effect of adding additional rotating masses simply results in more additive terms which are a function of additional states, e.g. the propeller angular rate would be a control input that would affect the  $\dot{p}$ ,  $\dot{q}$  and  $\dot{r}$  equations whether in the nonlinear state-space EOMs or in the LTI state-space EOMs. Note that additional rotating masses can easily be added to this model, though for an even number of rotating masses, often each is spun in opposite directions to counteract this contribution.

### Variable Mass Effect

Another mass effect is associated with the production of propulsive thrust through expelling mass. While this includes combustion jet engines that combust expelled fuel in air-breathing vehicles, the variable mass effects are typically only significant with rocket engines where the expelled mass includes the fuel *and* oxidizer for the combustion.

Thus, to this end, consider the position vectors of the various elements of a rocket as shown below



where the rocket's *instantaneous* body frame center, i.e. the center of mass, is  $\vec{x}_{B/N}$ , the mass element of the vehicle,  $\rho_V dV$ , with position vectors relative to the body and navigation frames as  $\vec{x}_{V/B}$  and  $\vec{x}_{V/N}$ , and the expelled mass  $\Delta m$  has position vectors relative to the body and navigation frames as  $\vec{x}_{E/B}$  and  $\vec{x}_{E/N}$ . Note that this derivation will consider the each of these position vectors as represented in inertial navigation frame  $N$  coordinates unless otherwise noted.

At time  $t$ , one has for the translational momentum of the variable mass vehicle

$$\vec{P}(t) = \int_{Vol} \rho_V \dot{\vec{x}}_{V/N} dV \quad (14.15)$$

while at time  $t + \Delta t$ , the translational momentum is

$$\vec{P}(t + \Delta t) = \int_{Vol} \rho_V \dot{\vec{x}}_{V/N} + \Delta \dot{\vec{x}}_{V/N} \rho_V dV + \Delta m \left( \dot{\vec{x}}_{E/N} + \Delta \dot{\vec{x}}_{E/N} \right) \quad (14.16)$$

where the last term represents the change in the system's translational momentum due to the change in mass, i.e.  $\Delta m < 0$  for expelled mass. Then taking the limit as  $\Delta t \rightarrow \infty$ , one has

$$\dot{\vec{P}} = \int_{Vol} \frac{d}{dt} \left( \rho_V \dot{\vec{x}}_{V/N} \right) dV + \dot{m} \dot{\vec{x}}_{E/N} \quad (14.17)$$

Comparing this expression with Newton's second law, one can see that the total rate of change in the translational momentum may be rewritten as

$$\dot{\vec{P}} = \int_{Vol} \rho_V \vec{g} dV + \int_{Area} d\vec{F}_{ext} + \dot{m} \dot{\vec{x}}_{E/N} \quad (14.18)$$

where  $dF_{ext}$  is the external force acting at some infinitesimal surface area. Furthermore, from the definition of the center of mass at time  $t$ , one has

$$m \vec{x}_{B/N} = \int_{Vol} \rho_V \vec{x}_{V/N} \quad (14.19)$$

and at time  $t + \Delta t$ , one has

$$m \vec{x}_{B/N} + \Delta m \vec{x}_{B/N} = \int_{Vol} \rho_V (\vec{x}_{V/N} + \Delta \vec{x}_{V/N}) dV + \Delta m (\vec{x}_{E/N} + \Delta \vec{x}_{E/N}) \quad (14.20)$$

then as  $\Delta t \rightarrow \infty$ , one has

$$\frac{d}{dt} (m \vec{x}_{B/N}) = \vec{P}(t) + \dot{m} \vec{p}_{E/N} \quad (14.21)$$

where notably  $\dot{m} < 0$ . Then, by the chain rule and realizing

$$\vec{x}_{E/N} = \vec{x}_{B/N} + \vec{x}_{E/B} \quad (14.22)$$

is the position of the expelled mass  $dm$ , the translational momentum can be written as

$$\vec{P}_N(t) = m \dot{\vec{x}}_{B/N} - \dot{m} \vec{x}_{E/B} \quad (14.23)$$

and differentiating results in

$$\dot{\vec{P}}_N = m \ddot{\vec{x}}_{B/N} + \dot{m} (\dot{\vec{x}}_{B/N} - \dot{\vec{x}}_{E/B}) - \ddot{m} \vec{x}_{E/B} \quad (14.24)$$

Note that the inertial velocity of the expelled mass  $dm$  is

$$\dot{\vec{x}}_{E/N} = \vec{v}_{B/N} + \vec{\omega}_{B/N} \times \vec{x}_{E/B} + \vec{v}_E \quad (14.25)$$

where  $\vec{v}_E$  is the inertial **exit velocity** of  $dm$  relative to the vehicle, and  $\vec{v}_{B/N}$  is the inertial velocity of the vehicle's center of mass.

Using this relation and equating Equations 14.18 and 14.24, one has

$$\begin{aligned} & m \ddot{\vec{x}}_{B/N} + \dot{m} (\dot{\vec{x}}_{B/N} - \dot{\vec{x}}_{E/B}) - \ddot{m} \vec{x}_{E/B} \\ &= \int_{Vol} \rho_V \vec{g} dV + \int_{Area} d\vec{F}_{ext} + \dot{m} (\vec{v}_{B/N} + \dot{\vec{x}}_{E/B} + \vec{v}_E) \end{aligned} \quad (14.26)$$

Then, taking gravity to be constant over the volume, defining the total aerodynamic force as

$$\vec{F}_{aero} = \int_{Body Area} d\vec{F}_{ext} \quad (14.27)$$

and defining the propulsive thrust force to be acting forward on the vehicle as

$$\vec{T} = \dot{m} \vec{v}_E + \int_{Exit Area} d\vec{F}_{ext} \quad (14.28)$$

where  $\int_{Exit\ Area} d\vec{F}_{ext}$  is known as the **pressure thrust**, one has the translational equation of motion for a variable mass vehicle as

$$m \dot{\vec{v}}_{B/N} = m \vec{g} + \vec{T} + \vec{F}_{aero} + 2\dot{m} \dot{\vec{x}}_{E/B} + \ddot{m} \vec{x}_E \quad (14.29)$$

or in terms of body frame coordinates

$$m \begin{bmatrix} u - qw + rv \\ v - ru + pw \\ w - pv + qu \end{bmatrix} = m \vec{g} + \vec{T} + \vec{F}_{aero} + 2\dot{m} \left( \dot{\vec{x}}_{E/B,B} + \vec{\omega}_{B/N} \times \vec{x}_{E/B,B} \right) + \ddot{m} \vec{x}_E \quad (14.30)$$

Thus, accounting for the variable mass results in two additive terms for the translational equation of motion. Note that  $\dot{\vec{x}}_{E/B,B}$  changes with time as the expelled mass will alter the location of the center of mass, i.e. the origin of the body frame. These terms are often neglected due to their small magnitude relative to the propellant exit velocity and the mass rate is roughly constant.

Continuing the rigid body EOM analysis, at time  $t$ , one has for the inertial rotational momentum of the variable mass vehicle

$$\vec{H}_N(t) = \int_{Vol} \vec{x}_{V/N} \times \rho_V \vec{x}_{V/N} dV \quad (14.31)$$

while at time  $t + \Delta t$ , the rotational momentum is

$$\begin{aligned} \vec{H}_N(t + \Delta t) &= \int_{Vol} (\vec{x}_{V/N} + \Delta \vec{x}_{V/N}) \times \rho_V (\dot{\vec{x}}_{V/N} + \Delta \dot{\vec{x}}_{V/N}) dV \\ &\quad + (\vec{x}_{E/N} + \Delta \vec{x}_{E/N}) \times \Delta m (\dot{\vec{x}}_{E/N} + \Delta \dot{\vec{x}}_{E/N}) \end{aligned} \quad (14.32)$$

represents the change in the vehicle's rotational momentum due to the change in mass, i.e.  $\Delta m < 0$  for expelled mass. Then taking the limit as  $\Delta t \rightarrow \infty$  (and neglecting higher order  $\Delta$  terms), one has

$$\dot{\vec{H}}_N = \int_{Vol} \frac{d}{dt} (\vec{x}_{V/N} \times \rho_V \vec{x}_{V/N}) dV + \vec{x}_{E/N} \times \dot{m} \dot{\vec{x}}_{E/N} \quad (14.33)$$

Comparing this expression with Newton's second law, one can see that the total rate of change in the rotational momentum may be rewritten as

$$\dot{\vec{H}}_N = \int_{Vol} \vec{x}_{V/N} \times \rho_V \vec{g} dV + \int_{Area} \vec{x}_{V/N} \times d\vec{F}_{ext} + \vec{x}_{E/N} \dot{m} \dot{\vec{x}}_{E/N} \quad (14.34)$$

Next, noting  $\vec{x}_{V/N} = \vec{x}_{B/N} + \vec{x}_{V/B}$ , one has

$$\begin{aligned} \dot{\vec{H}}_N &= \vec{x}_{B/N} \times \int_{Vol} \rho_V \dot{\vec{x}}_{V/N} dV + \int_{Vol} \vec{x}_{V/B} \times \rho_V \dot{\vec{x}}_{V/B} \\ &= \vec{x}_{B/N} \times \int_{Vol} \rho_V \dot{\vec{x}}_{V/N} dV + \vec{H}_{B,N} \end{aligned} \quad (14.35)$$

where  $\vec{H}_{B,N}$  is the angular momentum of the vehicle in navigation frame coordinates, and differentiating with respect to the navigation frame, one has

$$\dot{\vec{H}}_N = \vec{x}_{B/N} \times \vec{P} + \vec{x}_{B/N} \times \dot{\vec{P}} + \dot{\vec{H}}_{B,N} \quad (14.36)$$

Then, substituting for  $\vec{P}$  and  $\dot{\vec{P}}$  from before, one has

$$\dot{\vec{H}}_N = -\dot{\vec{x}}_{B/N} \times \dot{m} \vec{x}_{E/B} + \vec{x}_{B/N} \times \left( \int_{Vol} \rho_V \vec{g} dV + \int_{Area} d\vec{F}_{ext} + \dot{m} \dot{\vec{x}}_{E/N} \right) + \dot{\vec{H}}_{B,N} \quad (14.37)$$

Finally, equating Equations 14.34 and 14.37, one has

$$\begin{aligned} \dot{\vec{H}}_{B,N} &= \dot{\vec{x}}_{B/N} \times \dot{m} \vec{x}_{E/B} + \vec{x}_{B/N} \times \left( \int_{Vol} \rho_V \vec{g} dV + \int_{Area} d\vec{F}_{ext} + \dot{m} \dot{\vec{x}}_{E/N} \right) \\ &= \int_{Vol} \vec{x}_{V/N} \times \rho_V \vec{g} dV + \int_{Area} \vec{x}_{V/N} \times d\vec{F}_{ext} + \vec{x}_{E/N} \dot{m} \dot{\vec{x}}_{E/N} \end{aligned} \quad (14.38)$$

and noting that the first mass moment about the center of mass is zero and

$$\begin{aligned} \vec{x}_{V/N} &= \vec{x}_{B/N} + \vec{x}_{V/B} \\ \vec{x}_{E/N} &= \vec{x}_{B/N} + \vec{x}_{E/B} \\ \dot{\vec{x}}_{E/N} &= \dot{\vec{x}}_{B/N} + \dot{\vec{x}}_{E/B} + \vec{v}_E \end{aligned} \quad (14.39)$$

one can rewrite the angular momentum of the variable mass vehicle about its center of mass as

$$\dot{\vec{H}}_{B,N} = \int_{Area} \vec{x}_{V/B} \times d\vec{F}_{ext} + \vec{x}_{E/B} \times \dot{m} \left( \dot{\vec{x}}_{E/B} + \vec{v}_E \right) \quad (14.40)$$

which after separating the external moments into aerodynamic and propulsive contributions, one has

$$\dot{\vec{H}}_{B,N} = C_{N \leftarrow B} \begin{bmatrix} I_{xx} L \\ I_{yy} M \\ I_{zz} N \end{bmatrix} = \vec{M}_{aero} + \vec{M}_{prop} + \vec{x}_{E/B} \times \dot{m} \left( \dot{\vec{x}}_{E/B,B} + \vec{\omega}_{B/N} \times \vec{x}_{E/B} \right) \quad (14.41)$$

where  $\vec{M}_{prop} = \vec{x}_{E/B} \times \vec{T}$  and the additive triple product is known as the **jet-damping effect** because it tends to act against the vehicle's angular motion while being proportional to the vehicle's angular velocity. This damping is more significant for long vehicles with large mass flow rates, e.g. missiles.

## 14.2 Atmospheric and Gravity Effects on Flight Dynamics

For introductory FDC, one typically assumes simple atmospheric conditions, namely constant density and no wind. Furthermore, one typically simulates the flight motion of an aircraft at a particular reference altitude and assumes that the acceleration due to gravity is constant. These assumptions simplify the equations of motion as altitude does not need to be considered as state (affects density and gravity), the stability and control derivatives are constant (changes with density), the body frame velocity is equal to the airspeed velocity. This lecture will discuss how to alter the equations of motion to account for these atmospheric and variable gravity effects.

## Air Density Effect

In FDC, one typically includes the assumption that the aerodynamic forces and moments can be developed with a constant atmospheric density. However, if atmospheric density change is important (e.g. hypersonic flight vehicles), then one may include aerodynamic forces due to density variations as

$$\begin{aligned} X_h &= \frac{Q_\infty S_W}{\rho_\infty m} \bar{C}_X \left( \frac{\partial \rho_\infty}{\partial h} \right) \\ Y_h &= \frac{Q_\infty S_W}{\rho_\infty m} \bar{C}_Y \left( \frac{\partial \rho_\infty}{\partial h} \right) \\ Z_h &= \frac{Q_\infty S_W}{\rho_\infty m} \bar{C}_Z \left( \frac{\partial \rho_\infty}{\partial h} \right) \end{aligned} \quad (14.42)$$

while typically disregarding the effect on the moments as they are typically quite small. Here the density gradient  $\partial \rho_\infty / \partial h$  is obtained from an atmospheric model. A common atmospheric model is the 1962 Standard Atmosphere uses a linear temperature vs. altitude model for different layers of the atmosphere, i.e.

$$T = T_0 + \ell(h - h_0) \quad (14.43)$$

where the linear variation,  $\ell$ , is called **temperature lapse rate** as shown in the following table.

Atmospheric Layer	Lapse Rate $\ell$ ( $^{\circ}\text{R}/\text{ft}$ )	Lower Altitude $h_0$ (ft)	Temperature $T_0$ ( $^{\circ}\text{R}$ )	Pressure $p_0$ (psf)	Density $\rho_0$ ( $\text{sl}/\text{ft}^3$ )
Troposphere	$-3.5662 \times 10^{-3}$	0	518.69	2,116.2	$2.3769 \times 10^{-3}$
Stratosphere I	0	36,089	389.99	472.68	$7.0613 \times 10^{-4}$
Stratosphere II	$5.4864 \times 10^{-4}$	65,617	389.99	114.35	$1.7083 \times 10^{-4}$
Mesosphere		104,990	411.59	18.13	$2.5661 \times 10^{-5}$

Using this information, the aerostatic equation

$$\frac{dp}{dh} = -\rho g_0 \quad (14.44)$$

where  $g_0$  is the acceleration due to gravity at mean sea level (MSL), and the perfect-gas equation

$$p = \rho RT \quad (14.45)$$

where  $R$  is the universal gas constant for air,  $1716.5 \text{ ft}^2/(\text{s}^2 - ^{\circ}\text{R})$ , one can form the following models for the atmosphere for the different layers. For the Troposphere, one can use the mathematical model:

$$\begin{aligned} T &= 518.69 - (3.5662 \times 10^{-3})h \quad ^{\circ}\text{R} \\ p &= (1.1376 \times 10^{-11})T^{5.256} \quad \text{psf} \\ \rho &= (6.6277 \times 10^{-15})T^{4.256} \quad \text{sl}/\text{ft}^3 \end{aligned} \quad (14.46)$$

for the Stratosphere I, one can use the mathematical model:

$$\begin{aligned} T &= 389.99 \quad ^{\circ}\text{R} \\ p &= (2678.4)\exp\left((-4.8063 \times 10^{-5})h\right) \quad \text{psf} \\ \rho &= (1.4939 \times 10^{-6})p \quad \text{sl}/\text{ft}^3 \end{aligned} \quad (14.47)$$

and for the Stratosphere II, one can use the mathematical model:

$$\begin{aligned} T &= 389.99 + (5.4864 \times 10^{-4})(h - 65617) \text{ } ^\circ\text{R} \\ p &= (3.7930 \times 10^{90})T^{-34.164} \text{ psf} \\ \rho &= (2.2099 \times 10^{87})T^{-35.164} \text{ sl/ft}^3 \end{aligned} \quad (14.48)$$

An approximation for the density specifically that is often used for these models are:

$$\begin{aligned} \text{Troposphere: } \rho &= (2.3769 \times 10^{-3}) \exp\left(\frac{-h}{29,730}\right) \text{ sl/ft}^3 \\ \text{Stratosphere I: } \rho &= (7.0613 \times 10^{-4}) \exp\left(\frac{h - 36,089}{20,806}\right) \text{ sl/ft}^3 \\ \text{Stratosphere II: } \rho &= (1.7083 \times 10^{-4}) \exp\left(\frac{-(h - 65,617)}{29,730}\right) \text{ sl/ft}^3 \end{aligned} \quad (14.49)$$

for which the density-altitude gradient can be easily calculated as a function of  $h$ .

## Wind Effect

In deriving the equations of motion in introductory FDC, one typically uses the stability frame for coordinated flight where the velocity vector is colinear with the body frame  $x_B$ -axis, i.e.  $\bar{u} = v_a$ , and one can use approximations for  $\Delta v$  and  $\Delta w$  by  $\beta$  and  $\alpha$ . However, in the presence of wind, this relationship is more complex due to the wind triangle which can be expressed as

$$\vec{v}_{B/N} = \vec{v}_a + \vec{v}_w \quad (14.50)$$

where  $\vec{v}_{B/N}$  is the ground speed vector, i.e. the velocity of the body frame relative to the navigation frame,  $\vec{v}_a$  is the airspeed vector, i.e. the velocity of the body frame relative to the air mass, and  $\vec{v}_w$  is the wind speed vector, i.e. the velocity of the air mass relative to the navigation frame. In FDC, as the aerodynamic forces and moments are typically a function of the airspeed velocity instead of the ground speed vector, it is often more useful to use the airspeed velocity vector as part of the state vector in the equations of motion. Differentiating the wind triangle equation for body frame coordinates, one has

$$\dot{\vec{v}}_{B/N,B} + \vec{\omega}_{B/N} \times \vec{v}_{B/N,B} = \left( \dot{\vec{v}}_{a,B} + \vec{v}_{w,B} \right) + \vec{\omega}_{B/N} \times (\vec{v}_{a,B} + \vec{v}_{w,B}) \quad (14.51)$$

However, keeping  $\vec{v}_w$  in navigation frame coordinates, one has

$$\dot{\vec{v}}_{B/N,B} + \vec{\omega}_{B/N} \times \vec{v}_{B/N,B} = \dot{\vec{v}}_{a,B} + \vec{\omega}_{B/N} \times \vec{v}_{a,B} + \vec{v}_{w,N} \quad (14.52)$$

Then, substituting on the left hand side for the sum of forces in the body frame, one has

$$\vec{F}_{aero} + C_{B \leftarrow N} \vec{g}_N = \dot{\vec{v}}_{a,B} + \vec{\omega}_{B/N} \times \vec{v}_{a,B} + \dot{\vec{v}}_{w,N} \quad (14.53)$$

or in terms of coordinates

$$\begin{bmatrix} X - g \sin \theta \\ Y + g \cos \theta \sin \phi \\ Z + g \cos \theta \cos \phi \end{bmatrix} = \begin{bmatrix} \dot{u}_a + q w_a - r v_a \\ \dot{v}_a + r u_a - p w_a \\ \dot{w}_a + p v_a - q u_a \end{bmatrix} + \begin{bmatrix} \dot{u}_w \\ \dot{v}_w \\ \dot{w}_w \end{bmatrix} \quad (14.54)$$

Note that if  $\vec{v}_w = 0$ , i.e. **steady wind**, then these equations have the same mathematical form as the no-wind translation EOMs. Furthermore, one can integrate these velocity equations of motion to find the vehicle's inertial position, i.e.

$$\vec{x}_{B/N,N} = \int \vec{v}_{B/N,N} dt = \int C_{N \leftarrow B}(t) \vec{v}_{a,B}(t) + \vec{v}_{w,N} dt \quad (14.55)$$

Thus, one can simply use the airspeed vector components in the EOMs which would add a positional offset that grows linearly with time due to the steady wind. Often, this is preferred as the airspeed vector is what affects the aerodynamic forces and moments.

For assessing the effects of **unsteady wind** on the velocity EOMs (as opposed to steady winds), one typically assumes that the unsteady wind, also known as **stochastic gusts**, take on some stochastic properties of varying magnitude denoted as  $\vec{v}_g$ . Here, one can redefine the body frame velocity as

$$\vec{v}_{B/N} = \vec{v}_a + \vec{v}_g \quad (14.56)$$

which uses both  $\vec{v}_a = [u \ v \ w]^T$  and  $\vec{v}_g = [u_g \ v_g \ w_g]^T$  in the state vector for the EOMs. Thus, by component one has

$$\begin{bmatrix} u_{tot} \\ v_{tot} \\ w_{tot} \end{bmatrix} = \begin{bmatrix} u + u_g \\ v + v_g \\ w + w_g \end{bmatrix} \quad (14.57)$$

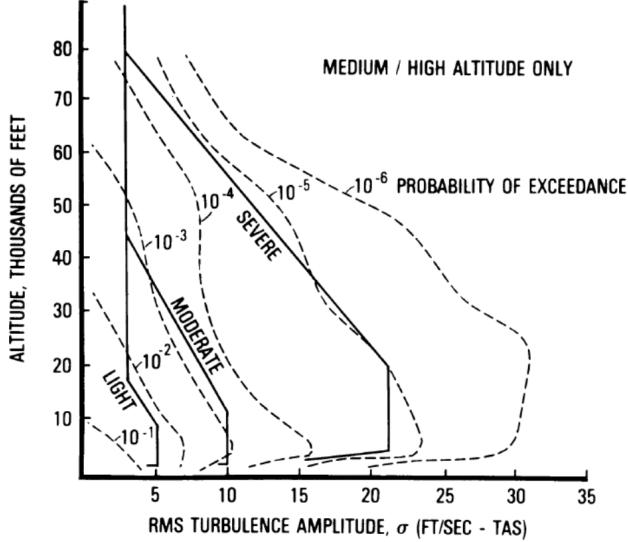
where  $u$ ,  $v$ , and  $w$  implicitly model the velocity relative to a *still* air mass. Two of the most widely used stochastic gust models are the **Dryden gust model** and the **von Kármán gust model**. The Dryden gust state equations are provided here for addition to the stability frame coordinates as

$$\begin{bmatrix} \dot{u}_g(t) \\ \dot{v}_g(t) \\ \dot{v}_{g1}(t) \\ \dot{w}_g(t) \\ \dot{w}_{g1}(t) \end{bmatrix} = \begin{bmatrix} -\frac{\bar{u}}{L_u} & 0 & 0 & 0 & 0 \\ 0 & -\frac{\bar{u}}{L_u} & \sigma_v(1 - \sqrt{3}) \left(\frac{\bar{u}}{L_v}\right)^{3/2} & 0 & 0 \\ 0 & 0 & -\frac{\bar{u}}{L_v} & 0 & 0 \\ 0 & 0 & 0 & -\frac{\bar{u}}{L_w} & \sigma_w(1 - \sqrt{3}) \left(\frac{\bar{u}}{L_w}\right)^{3/2} \\ 0 & 0 & 0 & 0 & -\frac{\bar{u}_a}{L_w} \end{bmatrix} \begin{bmatrix} u_g(t) \\ v_g(t) \\ v_{g1}(t) \\ w_g(t) \\ w_{g1}(t) \end{bmatrix} + \begin{bmatrix} \sigma_u \left(\frac{2\bar{u}}{\pi L_u}\right)^{1/2} \\ \sigma_v \left(\frac{3\bar{u}}{L_v}\right)^{1/2} \\ 1 \\ \sigma_w \left(\frac{3\bar{u}}{L_w}\right)^{1/2} \\ 1 \end{bmatrix} \vec{n}(t) \quad (14.58)$$

$$\dot{\vec{x}}_g = A_g \vec{x}_g + B_g n$$

where the driving function,  $n(t)$ , is a zero-mean, additive white Gaussian noise (AWGN) of unit intensity across all five channels,  $[L_u \ L_v \ L_w]$  are  $[h \ 145h^{1/3} \ 145h^{1/3}]$  for  $h < 1750$  ft and  $[1750 \ 1750 \ 1750]$  for  $h \geq 1750$  ft, and  $\sigma_u$ ,  $\sigma_v$ , and  $\sigma_w$  are the standard deviations of the gusts, or **RMS gust intensities** which can be obtained from data such as "MIL-F-8785C Military Specification: Flying Qualities of Piloted Airplanes"

which provides the plot of three levels of RMS gust intensities for different altitudes: *light*, *moderate*, and *severe*.



Note that here the **probability of exceedance** is the probability that the RMS gust intensity would exceed the value shown on the curves at that altitude. Also note that for numerical simulations,  $n(t)$  can be approximated as continuous over a short time step of size  $\Delta t$ , thus approximating  $n(t)$  by a random sequence  $n_i$  which are zero-mean Gaussian with variance  $1/\Delta t$ .

Lastly, using the wind frame Euler angles, i.e. the angle of attack and sideslip angles, to transform the instantaneous velocity magnitude,  $|v_a|$ , to body frame coordinates, one has

$$\begin{bmatrix} |v_a| \cos \alpha_{tot} \cos \beta_{tot} \\ |v_a| \sin \beta_{tot} \\ |v_a| \sin \alpha_{tot} \cos \beta_{tot} \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \begin{bmatrix} u_g \\ v_g \\ w_g \end{bmatrix} \quad (14.59)$$

From this equation, one can form the following relationships. First, the magnitude equation can be written as

$$|v_a|^2 = (u + u_g)^2 + (v + v_g)^2 + (w + w_g)^2 \quad (14.60)$$

Second, taking the third row divided by the first row, one has

$$\tan \alpha_{tot} = \frac{w + w_g}{u + u_g} \quad (14.61)$$

Third, the second row can be rewritten as

$$\sin \beta_{tot} = \frac{v + v_g}{|v_a|} \quad (14.62)$$

For small angles and  $u_w \ll u$ , one can write

$$\alpha_{tot} \approx \frac{w}{u} + \frac{w_g}{u} = \alpha + \alpha_g \quad (14.63)$$

$$\beta_{tot} \approx \frac{v}{|v_a|} + \frac{v_g}{v_a} = \beta + \beta_g \quad (14.64)$$

and similarly

$$\dot{\alpha}_{tot} \approx \dot{\alpha} + \dot{\alpha}_g \quad (14.65)$$

which are often used in linear flight dynamics instead of  $v_{tot}$  and  $w_{tot}$ .

With these relationships and a simplifying assumption, it is easy to model the aerodynamic forces and moments using the total velocity components. The assumption is that the axial and lateral gusts velocities have a negligible variation across the flight vehicle compared to the global variations of the gusts relative to the surface of the Earth. Note that this does not assume anything about the vertical gusts due to the presence of  $\dot{\alpha}$ . Then, one may simply use the addition formulas for  $u + u_g$ ,  $v + v_g$  (or  $\approx \beta + \beta_g$ ),  $w + w_g$  (or  $\approx \alpha + \alpha_g$ ), and  $\dot{\alpha} + \dot{\alpha}_g$  for computing the stability derivative contributions with respect to  $u$ ,  $v$  (or  $\beta$ ),  $w$  (or  $\alpha$ ), and  $\dot{w}$  (or  $\dot{\alpha}$ ) for  $X$ ,  $Y$ ,  $Z$ ,  $L$ ,  $M$ , and  $N$ .

Furthermore, for the linearized flight dynamics, if one can uses the linear Dryden wind model above combined with the nominal LTI state-space model, i.e.

$$\begin{aligned} \Delta \dot{\vec{x}} &= A\Delta \vec{x} + B\Delta \vec{u} \\ \Delta \vec{y} &= C\Delta \vec{x} + D\Delta \vec{u} \end{aligned} \quad (14.66)$$

where  $\Delta \vec{x} = [\Delta u \ \Delta \beta \ \Delta \alpha \ \Delta p \ \Delta q \ \Delta r \ \Delta \phi \ \Delta \theta \ \Delta \psi]^T$  and  $\Delta \vec{u} = [\Delta \delta_a \ \Delta \delta_e \ \Delta \delta_r \ \Delta \delta_t]^T$ , then one can form the following augmented LTI state-space system

$$\begin{aligned} \begin{bmatrix} \Delta \dot{\vec{x}} \\ \dot{\vec{x}}_g \end{bmatrix} &= \begin{bmatrix} A & G_g C_g \\ 0 & A_g \end{bmatrix} \begin{bmatrix} \Delta \vec{x} \\ \vec{x}_g \end{bmatrix} + \begin{bmatrix} B & G_g D_g \\ 0 & B_g \end{bmatrix} \begin{bmatrix} \Delta \vec{u} \\ n \end{bmatrix} \\ \Delta \vec{y} &= [C \ 0] \begin{bmatrix} \Delta \vec{x} \\ \vec{x}_g \end{bmatrix} + [D \ 0] \begin{bmatrix} \Delta \vec{u} \\ n \end{bmatrix} \end{aligned} \quad (14.67)$$

where the gust output matrix and feedthrough matrix can be constructed as

$$C_g = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\bar{u}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\bar{u}} & 0 \\ 0 & 0 & 0 & -\frac{1}{L_w} & \sigma_w(1 - \sqrt{3}) \left(\frac{\bar{u}}{L_w^3}\right)^{1/2} \end{bmatrix} \quad (14.68)$$

and

$$D_g = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \sigma_w \left(\frac{3}{L_w \bar{u}}\right)^{1/2} \end{bmatrix} \quad (14.69)$$

to output  $u_g, \beta_g, \alpha_g$ , and  $\dot{\alpha}_g$  from the gust state vector  $\vec{x}_g = [u_g \ v_g \ v_{g1} \ w_g \ w_{g1}]^T$ , and the mapping matrix from these gust outputs to the stability derivatives

$$G_g = \begin{bmatrix} X_u + \frac{X_\alpha Z_u}{\bar{u} - Z_\alpha} & 0 & X_\alpha + \frac{X_{\dot{\alpha}} Z_\alpha}{\bar{u} - Z_{\dot{\alpha}}} & X_{\dot{\alpha}} + \frac{X_{\ddot{\alpha}} Z_{\dot{\alpha}}}{\bar{u} - Z_{\ddot{\alpha}}} \\ 0 & \frac{Y_\beta}{\bar{u}} & 0 & 0 \\ \frac{Z_u}{\bar{u} - Z_\alpha} & 0 & \frac{Z_\alpha}{\bar{u} - Z_{\dot{\alpha}}} & \frac{Z_{\dot{\alpha}}}{\bar{u} - Z_{\ddot{\alpha}}} \\ 0 & L_\beta^* & 0 & 0 \\ M_u + \frac{M_\alpha Z_u}{\bar{u} - Z_\alpha} & 0 & M_\alpha + \frac{M_{\dot{\alpha}} Z_\alpha}{\bar{u} - Z_{\dot{\alpha}}} & M_{\dot{\alpha}} + \frac{M_{\ddot{\alpha}} Z_{\dot{\alpha}}}{\bar{u} - Z_{\ddot{\alpha}}} \\ 0 & N_\beta^* & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (14.70)$$

### Variable Gravity with Altitude

Though the Earth is truly a geoid with a variable gravitational field due to local features such as mountains, oceans, lakes, and even ore deposits, it is often useful to use a model for the acceleration due to gravity  $g$  for an ellipsoidal Earth model which will be a function of latitude  $\ell$  and altitude  $h$ . One such model for the **latitude correction** at MSL is the **WGS 84 Ellipsoidal Gravity Formula**

$$g(\ell) = g_e \left( \frac{1 + k \sin^2 \ell}{\sqrt{1 - e^2 \sin^2 \ell}} \right) \quad (14.71)$$

where  $e^2 = 1 - (b/a)^2$  is the ellipsoid's eccentricity squared, 0.00669437999013,  $k = \frac{bg_p - ag_e}{ag_e}$  is a formula constant, 0.00193185138639,  $a$  is the equatorial semi-axis, 6378137.0 m,  $b$  is the polar semi-axis, 6356752.3 m,  $g_e$  is the acceleration due to gravity at the equator, 9.7803267714 m/s<sup>2</sup>, and  $g_p$  is the acceleration due to gravity at the poles, 9.8321849378 m/s<sup>2</sup>. Next, one can use the **free air correction (FAC)** to account for the altitude *above* MSL. Using the reference value for a specific latitude, one has

$$g(h) = g_0 \left( \frac{R_e}{R_e + h} \right)^2 \quad (14.72)$$

where  $g_0$  is the **standard acceleration due to gravity**, 9.80665 m/s<sup>2</sup> and  $R_e$  is Earth's *mean* radius, 6,371,000 m. So the FAC for an altitude,  $h$ , in this case becomes

$$\Delta g(h) = g_0 \left( \frac{R_e}{R_e + h} \right)^2 - g_0 = g_0 \left( \left( \frac{R_e}{R_e + h} \right)^2 - 1 \right) \quad (14.73)$$

Thus, combining the two corrections one has

$$g(\ell, h) = g(\ell) + g_0 \left( \left( \frac{R_e}{R_e + h} \right)^2 - 1 \right) \quad (14.74)$$

However, an approximation for  $h \ll R_e$  is

$$\Delta g(h) \approx \frac{-2g_0}{R_e^2} h \quad (14.75)$$

Thus, one common model in FDC is the linear equation

$$g(\ell, h) \approx g(\ell) - 3.086 \times 10^{-6}h \quad (14.76)$$

Lastly, note that for a flat or perfectly spherical Earth, one would only use the Earth's gravity variation with altitude,  $g(h)$ , i.e.

$$g(h) = g_0 \left( \frac{R_e}{R_e + h} \right)^2 \quad (14.77)$$

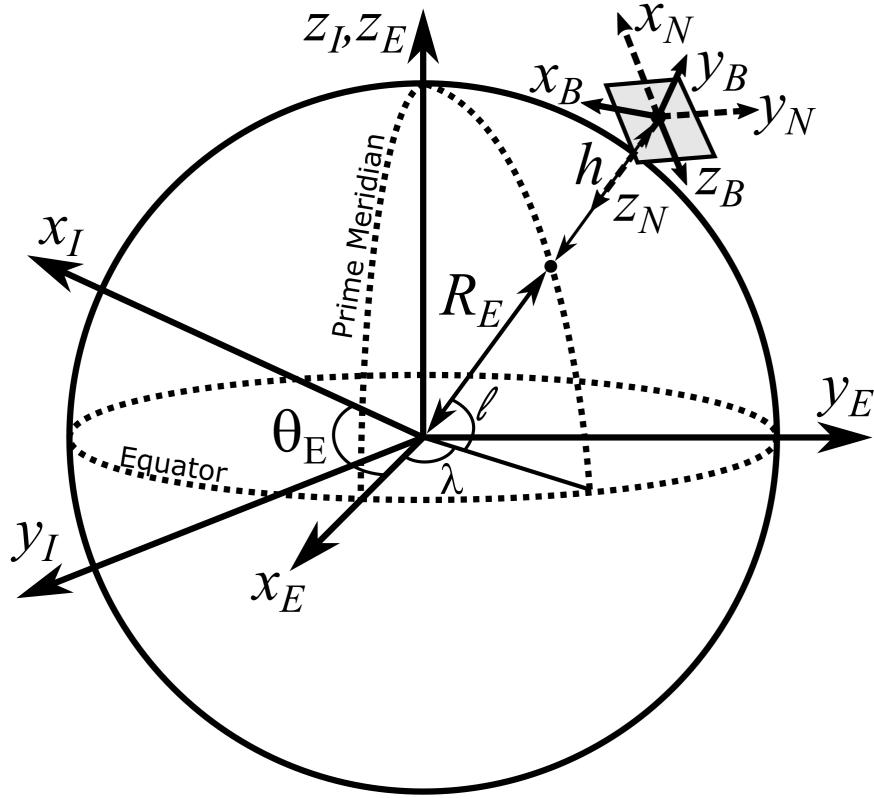
or as a linear approximation

$$g(h) \approx g_0 - 3.086 \times 10^{-6}h \quad (14.78)$$

### 14.3 Rotating Spherical Earth Effects on Flight Dynamics

In introductory FDC, one typically assumes that the navigation frame axes (e.g. North, East, and Down) can be treated as inertial, i.e. a “flat-Earth” model. However, for vehicles that operate at high supersonic velocities and/or fly long distances, this assumption may not have a negligible effect. Thus, this lecture will discuss the effects of modeling the Earth frame as rotating about a constant axis for a perfectly spherical shape. As discussed previously, the Earth is truly non-spherical and has a very slight wobble in its rotation axis. Thus, for precision guidance and navigation applications, one typically requires the use of an oblate spheroid model, i.e. the **reference ellipsoid** model which will be discussed as an appended section to this course material. Note that this lecture will look at the changes to the EOMs for a rigid flight vehicle with constant mass, no rotating mass, and no wind.

Assuming this model of the Earth, consider the relationship between the ECI frame (subscript  $I$ ), the ECEF frame (subscript  $E$ ), the navigation frame (subscript  $N$ ), and the body frame (subscript  $B$ ).



Note that here the navigation frame origin is at the center of mass of the flight vehicle with its axes orientated in the instantaneous North-East-Down (NED) frame which will now change as a function of time as the curvature of the Earth changes the sense of North, East, and Down. This was not the case previously for a flat-Earth, these directions never changed.

### Rotation Equation of Motion Effects

The angular velocity effect of the rotating Earth will be as two additional angular velocity terms for the ECI and ECEF frames, i.e.

$$\vec{\omega}_{B/I} = \vec{\omega}_{B/N} + \vec{\omega}_{N/I} = \vec{\omega}_{B/N} + \vec{\omega}_{N/E} + \vec{\omega}_{E/I} \quad (14.79)$$

Thus, for the “flat-Earth” model, one considers both the angular velocities of the ECEF frame relative to the ECI frame,  $\vec{\omega}_{E/I}$ , and the navigation frame relative to the ECEF frame,  $\vec{\omega}_{N/E}$ , to be zero.

For  $\vec{\omega}_{E/I}$ , by definition of the ECI frame, one has

$$\vec{\omega}_{E/I,E} = \begin{bmatrix} 0 \\ 0 \\ \omega_{Earth} \end{bmatrix} \quad (14.80)$$

where  $\omega_{Earth}$  is defined as  $72.92115 \times 10^{-6}$  rad/s by the WGS. This can be written in navigation frame coordinates as

$$\vec{\omega}_{E/I,N} = C_{N \leftarrow E} \begin{bmatrix} 0 \\ 0 \\ \omega_{Earth} \end{bmatrix} \quad (14.81)$$

where recall that

$$C_{N \leftarrow E} = \begin{bmatrix} -\sin \ell \cos \lambda & -\sin \ell \sin \lambda & \cos \ell \\ -\sin \lambda & \cos \lambda & 0 \\ -\cos \ell \cos \lambda & -\cos \ell \sin \lambda & -\sin \ell \end{bmatrix} \quad (14.82)$$

Thus,

$$\vec{\omega}_{E/I,N} = \begin{bmatrix} \omega_{Earth} \cos \ell \\ 0 \\ -\omega_{Earth} \sin \ell \end{bmatrix} \quad (14.83)$$

and in body frame coordinates as

$$\vec{\omega}_{E/I,B} = C_{B \leftarrow N}(\phi, \theta, \psi) \begin{bmatrix} \omega_{Earth} \cos \ell \\ 0 \\ -\omega_{Earth} \sin \ell \end{bmatrix} \quad (14.84)$$

For  $\vec{\omega}_{N/E}$ , one can use the definition of the derivative of a rotation matrix, i.e.

$$\dot{C}_{E \leftarrow N} = C_{E \leftarrow N} [\vec{\omega}_{N/E,N}]_\times \quad (14.85)$$

where the **skew-symmetric matrix operator** is defined as

$$[\vec{a}]_\times = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \quad (14.86)$$

and can be used to convert the cross-product to matrix multiplication. Thus, one has

$$\begin{aligned} & \begin{bmatrix} \dot{\lambda} \sin \ell \sin \lambda - \dot{\ell} \cos \ell \cos \lambda & -\dot{\lambda} \sin \ell \cos \lambda - \dot{\ell} \cos \ell \sin \lambda & -\dot{\ell} \sin \ell \\ -\dot{\lambda} \cos \lambda & -\dot{\lambda} \sin \lambda & 0 \\ \dot{\lambda} \cos \ell \sin \lambda + \dot{\ell} \sin \ell \cos \lambda & -\dot{\lambda} \cos \ell \cos \lambda + \dot{\ell} \sin \ell \sin \lambda & -\dot{\ell} \cos \ell \end{bmatrix} \\ &= \begin{bmatrix} -\sin \ell \cos \lambda & -\sin \ell \sin \lambda & \cos \ell \\ -\sin \lambda & \cos \lambda & 0 \\ -\cos \ell \cos \lambda & -\cos \ell \sin \lambda & -\sin \ell \end{bmatrix} [\vec{\omega}_{N/E,N}]_\times \end{aligned} \quad (14.87)$$

from which it can be shown that

$$\vec{\omega}_{N/E,N} = \begin{bmatrix} \dot{\lambda} \cos \ell \\ -\dot{\ell} \\ -\dot{\lambda} \sin \ell \end{bmatrix} \quad (14.88)$$

Thus,

$$\vec{\omega}_{N/I,N} = \begin{bmatrix} (\omega_{Earth} + \dot{\lambda}) \cos \ell \\ -\dot{\ell} \\ -(\omega_{Earth} + \dot{\lambda}) \sin \ell \end{bmatrix} \quad (14.89)$$

Finally, recall that

$$\vec{\omega}_{B/N,B} = \begin{bmatrix} p_{B/N} \\ q_{B/N} \\ r_{B/N} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & -\sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (14.90)$$

which is the angular velocity for a flat-Earth model. Thus, to account for the rotation of a spherical Earth, one must instead use the angular velocity as a *supplemental equation* given by

$$\begin{aligned} \vec{\omega}_{B/I,B} &= \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & -\sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + C_{B \leftarrow N}(\phi, \theta, \psi) \begin{bmatrix} (\omega_{Earth} + \dot{\lambda}) \cos \ell \\ -\ell \\ -(\omega_{Earth} + \dot{\lambda}) \sin \ell \end{bmatrix} \\ \begin{bmatrix} p \\ q \\ r \end{bmatrix} &= \begin{bmatrix} p_{B/N} \\ q_{B/N} \\ r_{B/N} \end{bmatrix} + \begin{bmatrix} p_{N/I} \\ q_{N/I} \\ r_{N/I} \end{bmatrix} \end{aligned} \quad (14.91)$$

Furthermore, with this new inertial angular velocity, the *rotation equation of motion* remains the same as before, i.e.

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} \dot{p} + \frac{I_{zz}-I_{yy}}{I_{xx}} qr - \frac{I_{xz}}{I_{xx}} (\dot{r} + pq) \\ \dot{q} + \frac{I_{xx}-I_{zz}}{I_{yy}} pr - \frac{I_{yz}}{I_{yy}} (r^2 - p^2) \\ \dot{r} + \frac{I_{yy}-I_{xx}}{I_{zz}} pq - \frac{I_{xy}}{I_{zz}} (\dot{p} - qr) \end{bmatrix} \quad (14.92)$$

However, the relationship between the different rotation matrices must be used to find the navigation-to-body frame Euler angle rates as a *supplemental equation*, i.e.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \left( \begin{bmatrix} p \\ q \\ r \end{bmatrix} - \begin{bmatrix} p_{N/I} \\ q_{N/I} \\ r_{N/I} \end{bmatrix} \right) \quad (14.93)$$

### Translation Equation of Motion Effects

By inspection of the spherical Earth diagram, note that the position of the origin of the body/navigation frame with respect to the ECI/ECEF frame in navigation frame coordinates is simply in the (negative) down direction, i.e.

$$\vec{x}_{B/I,N} = \begin{bmatrix} 0 \\ 0 \\ -(R_E + h) \end{bmatrix} \quad (14.94)$$

Rewriting this in ECEF frame coordinates, one has

$$\vec{x}_{B/I,E} = C_{E \leftarrow N} \begin{bmatrix} 0 \\ 0 \\ -(R_E + h) \end{bmatrix} = \begin{bmatrix} (R_E + h) \cos \lambda \cos \ell \\ (R_E + h) \sin \lambda \cos \ell \\ (R_E + h) \sin \ell \end{bmatrix} \quad (14.95)$$

and in ECI frame coordinates, one has

$$\vec{x}_{B/I,I} = C_{I \leftarrow E} C_{E \leftarrow N} \begin{bmatrix} 0 \\ 0 \\ -(R_E + h) \end{bmatrix} \quad (14.96)$$

Differentiating this and relating this to body frame coordinates, one has

$$\dot{\vec{x}}_{B/I,I} = C_{I \leftarrow E} \left( C_{E \leftarrow N} \begin{bmatrix} 0 \\ 0 \\ -\dot{h} \end{bmatrix} + \dot{C}_{E \leftarrow N} \begin{bmatrix} 0 \\ 0 \\ -(R_E + h) \end{bmatrix} \right) + \dot{C}_{I \leftarrow E} C_{E \leftarrow N} \begin{bmatrix} 0 \\ 0 \\ -(R_E + h) \end{bmatrix} \quad (14.97)$$

or

$$\dot{\vec{x}}_{B/I,I} = C_{I \leftarrow E} \left( C_{E \leftarrow N} \begin{bmatrix} 0 \\ 0 \\ -\dot{h} \end{bmatrix} + C_{E \leftarrow N} [\vec{\omega}_{N/E,N}]_x \begin{bmatrix} 0 \\ 0 \\ -(R_E + h) \end{bmatrix} \right) + C_{I \leftarrow E} [\vec{\omega}_{E/I,E}]_x C_{E \leftarrow N} \begin{bmatrix} 0 \\ 0 \\ -(R_E + h) \end{bmatrix} \quad (14.98)$$

which can be written in ECEF frame coordinates as

$$\dot{\vec{x}}_{B/I,E} = C_{E \leftarrow N} \begin{bmatrix} 0 \\ 0 \\ -\dot{h} \end{bmatrix} + [\vec{\omega}_{N/E,E}]_x C_{E \leftarrow N} \begin{bmatrix} 0 \\ 0 \\ -(R_E + h) \end{bmatrix} + [\vec{\omega}_{E/I,E}]_x C_{E \leftarrow N} \begin{bmatrix} 0 \\ 0 \\ -(R_E + h) \end{bmatrix} \quad (14.99)$$

$$\dot{\vec{x}}_{B/I,E} = C_{E \leftarrow N} \begin{bmatrix} 0 \\ 0 \\ -\dot{h} \end{bmatrix} + ([\vec{\omega}_{N/E,E}]_x + [\vec{\omega}_{E/I,E}]_x) C_{E \leftarrow N} \begin{bmatrix} 0 \\ 0 \\ -(R_E + h) \end{bmatrix} \quad (14.100)$$

$$\dot{\vec{x}}_{B/I,E} = C_{E \leftarrow N} \begin{bmatrix} 0 \\ 0 \\ -\dot{h} \end{bmatrix} + C_{E \leftarrow N} ([\vec{\omega}_{N/E,N}]_x + [\vec{\omega}_{E/I,N}]_x) \begin{bmatrix} 0 \\ 0 \\ -(R_E + h) \end{bmatrix} \quad (14.101)$$

which can be transformed to navigation frame coordinates as

$$\dot{\vec{x}}_{B/I,N} = \begin{bmatrix} 0 \\ 0 \\ -\dot{h} \end{bmatrix} + [\vec{\omega}_{N/E,N}]_x \begin{bmatrix} 0 \\ 0 \\ -(R_E + h) \end{bmatrix} + [\vec{\omega}_{E/I,N}]_x \begin{bmatrix} 0 \\ 0 \\ -(R_E + h) \end{bmatrix} \quad (14.102)$$

and substituting for the angular velocities as previously derived, one has

$$\dot{\vec{x}}_{B/I,N} = \begin{bmatrix} 0 \\ 0 \\ -\dot{h} \end{bmatrix} + \begin{bmatrix} \dot{\lambda} \cos \ell \\ -\dot{\ell} \\ -\dot{\lambda} \sin \ell \end{bmatrix}_x \begin{bmatrix} 0 \\ 0 \\ -(R_E + h) \end{bmatrix} + \begin{bmatrix} \omega_{Earth} \cos \ell \\ 0 \\ -\omega_{Earth} \sin \ell \end{bmatrix}_x \begin{bmatrix} 0 \\ 0 \\ -(R_E + h) \end{bmatrix} \quad (14.103)$$

$$\dot{\vec{x}}_{B/I,N} = \begin{bmatrix} \dot{\ell}(R_E + h) \\ \dot{\lambda}(R_E + h) \cos \ell \\ -\dot{h} \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_{Earth}(R_E + h) \cos \ell \\ 0 \end{bmatrix} \quad (14.104)$$

where for the flat-Earth assumption,  $\omega_{Earth} = 0$ , and one had the definition

$$\dot{\vec{x}}_{B/I,N} = C_{N \leftarrow B} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (14.105)$$

Thus, one can define for the rotating, spherical Earth that

$$\vec{v}_{B/E,N} = \begin{bmatrix} \dot{\ell}(R_E + h) \\ \dot{\lambda}(R_E + h) \cos \ell \\ -\dot{h} \end{bmatrix} = C_{N \leftarrow B} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (14.106)$$

which will be required as a *supplemental equation* for the rotating, spherical Earth equations. Returning to the derivations, note that Equation 14.104 can be rewritten as

$$\dot{\vec{x}}_{B/I,N} = C_{N \leftarrow B} \begin{bmatrix} u \\ v \\ w \end{bmatrix} + [\vec{\omega}_{E/I,N}]_x \begin{bmatrix} 0 \\ 0 \\ -(R_E + h) \end{bmatrix} \quad (14.107)$$

or returning to inertial coordinates, one has

$$\dot{\vec{x}}_{B/I,I} = C_{I \leftarrow N} C_{N \leftarrow B} \begin{bmatrix} u \\ v \\ w \end{bmatrix} + [\vec{\omega}_{E/I,I}]_x C_{I \leftarrow N} \begin{bmatrix} 0 \\ 0 \\ -(R_E + h) \end{bmatrix} \quad (14.108)$$

and taking the derivative for the inertial acceleration, one has

$$\begin{aligned} \ddot{\vec{x}}_{B/I,I} &= \dot{C}_{I \leftarrow N} C_{N \leftarrow B} \begin{bmatrix} u \\ v \\ w \end{bmatrix} + C_{I \leftarrow N} \left( \dot{C}_{N \leftarrow B} \begin{bmatrix} u \\ v \\ w \end{bmatrix} + C_{N \leftarrow B} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} \right) \\ &\quad + [\vec{\omega}_{E/I,I}]_x \left( \dot{C}_{I \leftarrow N} \begin{bmatrix} 0 \\ 0 \\ -(R_E + h) \end{bmatrix} + C_{I \leftarrow N} \begin{bmatrix} 0 \\ 0 \\ -\dot{h} \end{bmatrix} \right) \end{aligned} \quad (14.109)$$

or by definition of the derivatives of rotation matrices, one has

$$\begin{aligned} \ddot{\vec{x}}_{B/I,I} &= C_{I \leftarrow N,I} C_{N \leftarrow B} [\vec{\omega}_{N/I,B}]_x \begin{bmatrix} u \\ v \\ w \end{bmatrix} + C_{I \leftarrow N} \left( C_{N \leftarrow B} [\vec{\omega}_{B/N,B}]_x \begin{bmatrix} u \\ v \\ w \end{bmatrix} + C_{N \leftarrow B} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} \right) \\ &\quad + C_{I \leftarrow N} [\vec{\omega}_{E/I,N}]_x [\vec{\omega}_{E/I,N}]_x \begin{bmatrix} 0 \\ 0 \\ -(R_E + h) \end{bmatrix} + C_{I \leftarrow N} [\vec{\omega}_{E/I,N}]_x \begin{bmatrix} 0 \\ 0 \\ -\dot{h} \end{bmatrix} \end{aligned} \quad (14.110)$$

which can be transformed to navigation frame coordinates as

$$\begin{aligned} \ddot{\vec{x}}_{B/I,N} &= C_{N \leftarrow B} ([\vec{\omega}_{N/I,B}]_x + [\vec{\omega}_{B/N,B}]_x) \begin{bmatrix} u \\ v \\ w \end{bmatrix} + C_{N \leftarrow B} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} \\ &\quad + \begin{bmatrix} \omega_{Earth} \cos \ell \\ 0 \\ -\omega_{Earth} \sin \ell \end{bmatrix}_x \begin{bmatrix} \omega_{Earth} \cos \ell \\ 0 \\ -\omega_{Earth} \sin \ell \end{bmatrix}_x \begin{bmatrix} 0 \\ 0 \\ -(R_E + h) \end{bmatrix} + \begin{bmatrix} \omega_{Earth} \cos \ell \\ 0 \\ -\omega_{Earth} \sin \ell \end{bmatrix}_x \begin{bmatrix} 0 \\ 0 \\ -\dot{h} \end{bmatrix} \end{aligned} \quad (14.111)$$

and finally to body frame coordinates as

$$\ddot{\vec{x}}_{B/I,B} = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + [\vec{\omega}_{B/I,B}]_x \begin{bmatrix} u \\ v \\ w \end{bmatrix} + C_{B \leftarrow N} \begin{bmatrix} \omega_{Earth}^2 (R_E + h) \cos \ell \sin \ell \\ h \omega_{Earth} \cos \ell \\ \omega_{Earth}^2 (R_E + h) \cos^2 \ell \end{bmatrix} \quad (14.112)$$

With this redefinition of the components of the translational inertial acceleration, one may then write the translation equation of motion as

$$\begin{bmatrix} X - g \sin \theta \\ Y + g \cos \theta \sin \phi \\ Z + g \cos \theta \cos \phi \end{bmatrix} = \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} + C_{B \leftarrow N} \begin{bmatrix} \omega_{Earth}^2 (R_E + h) \cos \ell \sin \ell \\ h \omega_{Earth} \cos \ell \\ \omega_{Earth}^2 (R_E + h) \cos^2 \ell \end{bmatrix} \quad (14.113)$$

## 14.4 Advanced Rigid Airplane Dynamics Simulation

Recall that the nonlinear rigid vehicle equations of motion can be written as

$$\begin{bmatrix} X - g \sin \theta \\ Y + g \cos \theta \sin \phi \\ Z + g \cos \theta \cos \phi \\ L \\ M \\ N \end{bmatrix} = \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \\ \dot{p} + \frac{I_{zz} - I_{yy}}{I_{xx}} qr - \frac{I_{xz}}{I_{xx}} (\dot{r} + pq) \\ \dot{q} + \frac{I_{xx} - I_{zz}}{I_{yy}} pr - \frac{I_{xz}}{I_{yy}} (r^2 - p^2) \\ \dot{r} + \frac{I_{yy} - I_{xx}}{I_{zz}} pq - \frac{I_{xz}}{I_{zz}} (\dot{p} - qr) \end{bmatrix} \quad (14.114)$$

Then, defining a reference flight condition (not necessarily a trim condition) as

$$\bar{u}, \bar{v}, \bar{w}, \bar{p}, \bar{q}, \bar{r}, \bar{\phi}, \bar{\theta} \quad (14.115)$$

which also corresponds to the reference flight condition aerodynamic forces and moments

$$\bar{X}, \bar{Y}, \bar{Z}, \bar{L}, \bar{M}, \bar{N} \quad (14.116)$$

where the reference forces may alternatively be defined in the wind frame coordinates as the reference lift,  $\bar{L}$ , drag,  $\bar{D}$ , side force  $\bar{S}$ , and thrust vector  $\bar{T}$

$$\begin{bmatrix} m\bar{X} \\ m\bar{Y} \\ m\bar{Z} \end{bmatrix} = \begin{bmatrix} \bar{T}_x \\ \bar{T}_y \\ \bar{T}_z \end{bmatrix} + \begin{bmatrix} \cos \alpha \cos \beta & -\cos \alpha \sin \beta & -\sin \alpha \\ \sin \beta & \cos \beta & 0 \\ \sin \alpha \cos \beta & -\sin \alpha \sin \beta & \cos \alpha \end{bmatrix} \begin{bmatrix} -\bar{D} \\ \bar{S} \\ -\bar{L} \end{bmatrix} \quad (14.117)$$

These can be linearized as

$$\begin{bmatrix} \Delta X - g \cos \bar{\theta} \Delta \theta \\ \Delta Y + g(\cos \bar{\theta} \cos \bar{\phi} \Delta \phi - \sin \bar{\theta} \sin \bar{\phi} \Delta \theta) \\ \Delta Z - g(\cos \bar{\theta} \sin \bar{\phi} \Delta \phi + \sin \bar{\theta} \cos \bar{\phi} \Delta \theta) \\ \Delta L \\ \Delta M \\ \Delta N \end{bmatrix} = \begin{bmatrix} \Delta \dot{u} + \bar{q} \Delta w + \bar{w} \Delta q - \bar{v} \Delta r - \bar{r} \Delta v \\ \Delta \dot{v} + \bar{r} \Delta u + \bar{w} \Delta p - \bar{r} \Delta u - \bar{u} \Delta r \\ \Delta \dot{w} + \bar{p} \Delta v + \bar{v} \Delta p - \bar{q} \Delta u - \bar{u} \Delta q \\ \Delta \dot{p} + \frac{I_{zz} - I_{yy}}{I_{xx}} (\bar{q} \Delta r + \bar{r} \Delta q) - \frac{I_{xz}}{I_{xx}} (\Delta \dot{r} + \bar{p} \Delta q + \bar{q} \Delta p) \\ \Delta \dot{q} + \frac{I_{xx} - I_{zz}}{I_{yy}} (\bar{p} \Delta r + \bar{r} \Delta p) - 2 \frac{I_{xz}}{I_{yy}} (\bar{r} \Delta r - \bar{p} \Delta p) \\ \Delta \dot{r} + \frac{I_{yy} - I_{xx}}{I_{zz}} (\bar{p} \Delta q + \bar{q} \Delta p) - \frac{I_{xz}}{I_{zz}} (\Delta \dot{p} - \bar{q} \Delta r - \bar{r} \Delta q) \end{bmatrix} \quad (14.118)$$

and which can be written two sets of linear matrix equations, i.e.

$$\begin{bmatrix} \Delta u \\ \Delta v \\ \Delta w \end{bmatrix} = \begin{bmatrix} 0 & \bar{r} & -\bar{q} \\ -\bar{r} & 0 & \bar{p} \\ \bar{q} & -\bar{p} & 0 \end{bmatrix} \begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{v} \\ \Delta \dot{w} \end{bmatrix} + \begin{bmatrix} \Delta u \\ \Delta v \\ \Delta w \\ \Delta p \\ \Delta q \\ \Delta r \\ \Delta \phi \\ \Delta \theta \end{bmatrix} \quad (14.119)$$

$$\begin{bmatrix} 1 & 0 & -\frac{I_{xz}}{I_{xx}} \\ 0 & 1 & 0 \\ -\frac{I_{xz}}{I_{xx}} & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta \dot{p} \\ \Delta \dot{q} \\ \Delta \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{I_{xz}}{I_{xx}} \bar{q} & \frac{I_{yy}-I_{zz}}{I_{xx}} \bar{r} + \frac{I_{xz}}{I_{xx}} \bar{p} & \frac{I_{yy}-I_{zz}}{I_{xx}} \bar{q} \\ \frac{I_{zz}-I_{xx}}{I_{yy}} \bar{r} - 2 \frac{I_{zz}-I_{xx}}{I_{yy}} \bar{p} & 0 & \frac{I_{zz}-I_{xx}}{I_{yy}} \bar{p} + 2 \frac{I_{zz}-I_{xx}}{I_{yy}} \bar{r} \\ \frac{I_{xx}-I_{yy}}{I_{zz}} \bar{q} & -\frac{I_{xz}}{I_{zz}} \bar{r} + \frac{I_{xx}-I_{yy}}{I_{zz}} \bar{p} & -\frac{I_{xz}}{I_{zz}} \bar{q} \end{bmatrix} \begin{bmatrix} \Delta p \\ \Delta q \\ \Delta r \end{bmatrix} + \begin{bmatrix} \Delta L \\ \Delta M \\ \Delta N \end{bmatrix} \quad (14.120)$$

To complete these equations, recall that one can model the normalized aerodynamic and propulsive forces and moments by a constant terms and the following linear relationships to the states and control surface inputs captured by the stability and control derivatives (which may vary as the flight conditions vary), i.e.

$$\begin{bmatrix} X \\ Z \\ M \end{bmatrix} = \begin{bmatrix} X_0 \\ Z_0 \\ M_0 \end{bmatrix} + \begin{bmatrix} 0 & X_{\dot{\alpha}} & 0 \\ 0 & Z_{\dot{\alpha}} & 0 \\ 0 & M_{\dot{\alpha}} & 0 \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{q} \end{bmatrix} + \begin{bmatrix} X_u & X_{\alpha} & X_q \\ Z_u & Z_{\alpha} & Z_q \\ M_u & M_{\alpha} & M_q \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ q \end{bmatrix} + \begin{bmatrix} X_{\delta_e} & X_{\delta_t} \\ Z_{\delta_e} & Z_{\delta_t} \\ M_{\delta_e} & M_{\delta_t} \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_t \end{bmatrix} \quad (14.121)$$

and

$$\begin{bmatrix} Y \\ L \\ N \end{bmatrix} = \begin{bmatrix} Y_0 \\ L_0 \\ N_0 \end{bmatrix} + \begin{bmatrix} Y_{\beta} & Y_p & Y_r \\ L_{\beta} & L_p & L_r \\ N_{\beta} & N_p & N_r \end{bmatrix} \begin{bmatrix} \beta \\ p \\ r \end{bmatrix} + \begin{bmatrix} Y_{\delta_a} & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad (14.122)$$

and for the perturbations about the trimmed forces and moments, the stability and control derivative values should be taken *at the reference condition* as

$$\begin{bmatrix} \Delta X \\ \Delta Z \\ \Delta M \end{bmatrix} = \begin{bmatrix} 0 & X_{\dot{\alpha}} & 0 \\ 0 & Z_{\dot{\alpha}} & 0 \\ 0 & M_{\dot{\alpha}} & 0 \end{bmatrix} \begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{\alpha} \\ \Delta \dot{q} \end{bmatrix} + \begin{bmatrix} X_u & X_{\alpha} & X_q \\ Z_u & Z_{\alpha} & Z_q \\ M_u & M_{\alpha} & M_q \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta q \end{bmatrix} + \begin{bmatrix} X_{\delta_e} & X_{\delta_t} \\ Z_{\delta_e} & Z_{\delta_t} \\ M_{\delta_e} & M_{\delta_t} \end{bmatrix} \begin{bmatrix} \Delta \delta_e \\ \Delta \delta_t \end{bmatrix} \quad (14.123)$$

and

$$\begin{bmatrix} \Delta Y \\ \Delta L \\ \Delta N \end{bmatrix} = \begin{bmatrix} Y_{\beta} & Y_p & Y_r \\ L_{\beta} & L_p & L_r \\ N_{\beta} & N_p & N_r \end{bmatrix} \begin{bmatrix} \Delta \beta \\ \Delta p \\ \Delta r \end{bmatrix} + \begin{bmatrix} Y_{\delta_a} & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{bmatrix} \begin{bmatrix} \Delta \delta_a \\ \Delta \delta_r \end{bmatrix} \quad (14.124)$$

Furthermore, recall the supplemental Euler angle rate equation

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (14.125)$$

which can be linearized as

$$\begin{bmatrix} \Delta\dot{\phi} \\ \Delta\dot{\theta} \\ \Delta\dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \tan\bar{\theta}\sin\bar{\phi} & \tan\bar{\theta} \\ 0 & \cos\bar{\phi} & -\sin\bar{\phi} \\ 0 & \sec\bar{\theta}\sin\bar{\phi} & \sec\bar{\theta}\cos\bar{\phi} \end{bmatrix} \begin{bmatrix} \tan\bar{\theta}(\bar{q}\cos\bar{\phi} - \bar{r}\sin\bar{\phi}) & \bar{q}\sin\bar{\phi} + \bar{r}\cos\bar{\phi} + \dot{\psi}\sin\bar{\theta}\tan\bar{\theta} \\ -\bar{q}\sin\bar{\phi} - \bar{r}\cos\bar{\phi} & 0 \\ -\sec\bar{\theta}(\bar{r}\sin\bar{\phi} + \bar{q}\cos\bar{\phi}) & \dot{\psi}\tan\bar{\theta} \end{bmatrix} \begin{bmatrix} \Delta p \\ \Delta q \\ \Delta r \\ \Delta\phi \\ \Delta\theta \end{bmatrix} \quad (14.126)$$

Lastly, for the ground speed velocity for a flat-Earth model and recalling  $\dot{z}_N = -\dot{h}$ , one has

$$\begin{bmatrix} \dot{x}_N \\ \dot{y}_N \\ -\dot{h} \end{bmatrix} = \left( \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \right) \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (14.127)$$

which can be linearized as

$$\begin{bmatrix} \Delta\dot{x}_N \\ \Delta\dot{y}_N \\ -\Delta\dot{h} \end{bmatrix} = \left( \begin{bmatrix} \cos\bar{\psi}\Delta\psi & -\sin\bar{\psi}\Delta\psi & 0 \\ \sin\bar{\psi}\Delta\psi & \cos\bar{\psi}\Delta\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\bar{\theta} & 0 & \sin\bar{\theta} \\ 0 & 1 & 0 \\ -\sin\bar{\theta} & 0 & \cos\bar{\theta} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\bar{\phi} & -\sin\bar{\phi} \\ 0 & \sin\bar{\phi} & \cos\bar{\phi} \end{bmatrix} \right. \begin{aligned} &+ \begin{bmatrix} \cos\bar{\psi} & -\sin\bar{\psi} & 0 \\ \sin\bar{\psi} & \cos\bar{\psi} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\bar{\theta}\Delta\theta & 0 & \sin\bar{\theta}\Delta\theta \\ 0 & 1 & 0 \\ -\sin\bar{\theta}\Delta\theta & 0 & \cos\bar{\theta}\Delta\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\bar{\phi} & -\sin\bar{\phi} \\ 0 & \sin\bar{\phi} & \cos\bar{\phi} \end{bmatrix} \\ &+ \begin{bmatrix} \cos\bar{\psi} & -\sin\bar{\psi} & 0 \\ \sin\bar{\psi} & \cos\bar{\psi} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\bar{\theta} & 0 & \sin\bar{\theta} \\ 0 & 1 & 0 \\ -\sin\bar{\theta} & 0 & \cos\bar{\theta} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\bar{\phi}\Delta\phi & -\sin\bar{\phi}\Delta\phi \\ 0 & \sin\bar{\phi}\Delta\phi & \cos\bar{\phi}\Delta\phi \end{bmatrix} \Big) \begin{bmatrix} \bar{u} \\ \bar{v} \\ \bar{w} \end{bmatrix} \\ &+ \begin{bmatrix} \cos\bar{\psi} & -\sin\bar{\psi} & 0 \\ \sin\bar{\psi} & \cos\bar{\psi} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\bar{\theta} & 0 & \sin\bar{\theta} \\ 0 & 1 & 0 \\ -\sin\bar{\theta} & 0 & \cos\bar{\theta} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\bar{\phi} & -\sin\bar{\phi} \\ 0 & \sin\bar{\phi} & \cos\bar{\phi} \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta v \\ \Delta w \end{bmatrix} \end{aligned} \quad (14.128)$$

Secondly, one is typically interested in using the free-stream airflow properties along with the aircraft equations of motion. For the nonlinear case, one may use the relationships

$$\begin{bmatrix} v_\infty \\ \beta \\ \alpha \end{bmatrix} = \begin{bmatrix} \sqrt{u^2 + v^2 + w^2} \\ \sin^{-1} \left( \frac{v}{\sqrt{u^2 + v^2 + w^2}} \right) \\ \tan^{-1} \frac{w}{u} \end{bmatrix} \quad (14.129)$$

alternatively, one may use the linearized relationships for small angles of attack and sideslip as

$$\begin{bmatrix} \Delta v_\infty \\ \Delta\beta \\ \Delta\alpha \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{\bar{u}} & 0 \\ 0 & 0 & \frac{1}{\bar{u}} \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta v \\ \Delta w \end{bmatrix} \quad (14.130)$$

where it should be noted that often the **flank angle** is measured by the airplane's air data system (ADS) instead of the sideslip angle, i.e.

$$\beta_f = \tan^{-1} \frac{v}{u} \quad (14.131)$$

which is approximately equal to the linearized sideslip angle, i.e.  $\Delta\beta_f = \Delta\beta$ . However, for the nonlinear case, one strictly has

$$\beta = \tan^{-1} (\tan \beta_f \cos \alpha) \quad (14.132)$$

In addition, one may also be interested in the acceleration of a particular point,  $\vec{P}$ , on the rigid flight vehicle, e.g. an accelerometer is placed there. Expressing this point with coordinates *relative to the center of mass* as  $\vec{P} = [x_P \ y_P \ z_P]^T$ , it can be shown to be

$$\ddot{\vec{P}} = \vec{v}_{B/I} + \omega_{B/I} \times \vec{v}_{B/I} + \omega_{B/I} \times \omega_{B/I} \times \vec{P} + \dot{\omega}_{B/I} \times \vec{P} \quad (14.133)$$

or written out by component for a *flat-Earth* model, one has

$$\ddot{\vec{P}} = \begin{bmatrix} \dot{u} + q(w + (py_P - qx_P)) - r(v + (rx_P - pz_P)) + \dot{q}z_P - \dot{r}y_P \\ \dot{v} + r(u + (qz_P - ry_P)) - p(w + (py_P - qx_P)) + \dot{r}x_P - \dot{p}z_P \\ \dot{w} + p(v + (rx_P - pz_P)) - q(u + (qz_P - ry_P)) + \dot{p}y_P - \dot{q}x_P \end{bmatrix} \quad (14.134)$$

Lastly, it should be noted that often one must work with both the stability frame (subscript  $S$ ) which is defined as the body frame for which  $\bar{\alpha}_S = 0$  at a *selected* flight condition and the fuselage-fixed body frame (subscript  $F$ ) which is defined as fixed to the physical structure of the aircraft (or mean-axes). Here, the rotation matrix based on the fuselage frame reference condition,  $\bar{\alpha}_F$ , is

$$C_{F \leftarrow S} = \begin{bmatrix} \cos \bar{\alpha}_F & 0 & -\sin \bar{\alpha}_F \\ 0 & 1 & 0 \\ \sin \bar{\alpha}_F & 0 & \cos \bar{\alpha}_F \end{bmatrix} \quad (14.135)$$

which can be used to rotate any vector defined in either frame. In particular, for the inertia matrix,  $I_G$ , note that one has that

$$I_{G,F} = C_{F \leftarrow S} I_{G,S} C_{F \leftarrow S}^T \quad (14.136)$$

Thus, the previous nonlinear ordinary differential equations can be used for a full nonlinear simulation of the flight vehicle dynamics, or the 12 linear ordinary differential equations for the perturbations about a reference flight condition can be rearranged into an LTI state-space system simulation, though in many cases the matrices here will greatly simplify for many important reference flight conditions, e.g. steady flight. Furthermore, one may include the additional effects as derived additions to the various equations in previous lectures.

## Numerical Solver for Steady Flight

By the definition of steady (i.e. equilibrium) flight as stated above, one requires that

$$\dot{u} = \dot{v} = \dot{w} = \dot{p} = \dot{q} = \dot{r} = \dot{\phi} = \dot{\theta} = \dot{\delta}_a = \dot{\delta}_e = \dot{\delta}_r = \dot{\delta}_t = 0 \quad (14.137)$$

which is often studied for two particular conditions. Also, note that as the acceleration due to gravity and dynamic pressure also change with altitude, one would also require that  $\dot{h} = 0$  which is typically ignored in

introductory steady flight analysis. Thus, the primary steady flight conditions studied in FDC are *straight-and-level flight* which is defined by the *additional* constraints

$$\begin{aligned}\bar{v}_\infty^2 &= \bar{u}^2 + \bar{w}^2 \\ \bar{p} &= \bar{q} = \bar{r} = \bar{v} = 0 \\ \bar{\theta} &= \bar{\alpha} = \tan^{-1} \frac{\bar{w}}{\bar{u}}\end{aligned}\tag{14.138}$$

and *level, coordinated turning flight* which is defined by the *additional* constraints

$$\begin{aligned}\bar{v}_\infty^2 &= \bar{u}^2 + \bar{v}^2 + \bar{w}^2 \\ \bar{p} &= -\dot{\bar{\psi}} \sin \bar{\theta} \\ \bar{q} &= \dot{\bar{\psi}} \cos \bar{\theta} \sin \bar{\phi} \\ \bar{r} &= \dot{\bar{\psi}} \cos \bar{\theta} \cos \bar{\phi} \\ g \tan \bar{\phi} &= \dot{\bar{\psi}} \bar{v}_\infty\end{aligned}\tag{14.139}$$

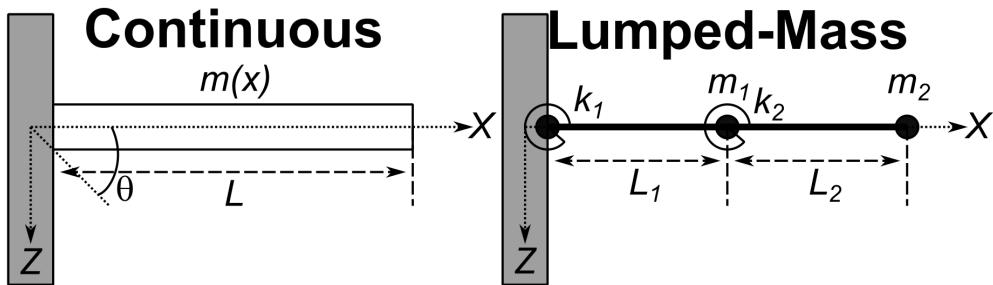
With these constraints in mind, one must solve Equations 15.42 and 14.125 as well as the last element of Equation 14.127 for these constraints. In general, this requires a thorough model of the aerodynamic stability and control derivatives, either discretely tabulated or modeled analytically. To solve such a problem, one requires the use of numerical solvers, e.g. MATLAB's `fminsearch()` or `trim()` functions, which can be used to find the trim/equilibrium points of differential equations using optimization search methods.

# Chapter 15

## Elastic Flight Vehicle Dynamics

### 15.1 Lumped-Mass Vibrations

As noted previously, introductory FDC assumes that rigid body dynamics can be used to model the equations of motion for flight vehicles. However, in reality, flight vehicle structures are not rigid, but have some elasticity or flexibility in the structure. Thus, this course will look at the additional modeling of structural vibrations in the airplane equations of motion. To contextualize this topic, consider the following two figures for a beam-vibration problem



In the left figure of the beam-vibration example, one has a continuous deformable body with some mass distribution,  $m(x)$ , as a function of the horizontal coordinate,  $x$ . It can be shown that the partial differential equation (PDE) governing the vertical deformation of the beam,  $Z$ , is given by

$$\frac{\partial^2}{\partial x^2} \left( EI(x) \frac{\partial^2 Z(x, t)}{\partial x^2} \right) + m(x) \frac{\partial^2 Z(x, t)}{\partial t^2} = 0 \quad (15.1)$$

where  $E$  is the elastic modulus of the beam material and  $I$  is the area moment of inertia of the beam cross-section about its neutral axis. Using the separation of variables technique, one can write the solution to this PDE as

$$Z(x, t) = \sum_{i=1}^{\infty} v_i(x) \eta_i(t) \quad (15.2)$$

which is an infinite sum of terms, each consisting of a purely space-dependent function  $v_i(x)$  and a purely time-dependent function,  $\eta_i(t)$ . The functions  $v_i(x)$  are called the **mode shapes**, also known as the **eigenfunctions**, and the functions  $\eta_i(t)$  are called the **modal coordinates**. As the solution is an infinite sum, the beam-vibration problem is called an **infinite-dimensional problem**.

To obtain a finite-dimensional approximation to this infinite-dimensional problem, one may simplify the continuous beam model as a discrete mass model with  $i = 1, \dots, n$  particles, a.k.a. **lumped-mass approximation**, where each mass particle,  $m_i$ , has an associated spring stiffness,  $k_i$ , and different particles are connected by massless rigid rods of lengths  $L_i$ . It should be noted that the solution to a vibration problem using a finite-element analysis (FEA) will result in a lumped-mass approximation. Thus, lumped-mass approximations are always used for real, complex, flight vehicle structures in FDC. As such, in the right figure of the beam-vibration example, the continuous beam has been approximated by two masses, two springs, and two rods. Note that if one desired to have a better approximation, one would simply add more lumped-masses to the beam.

To solve for the equations of motion for these lumped-mass systems, one can use Lagrange's equation, which states that with no external forces acting on the system, one has

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{q}_i} \right) - \frac{\partial T}{\partial q_i} + \frac{\partial U}{\partial q_i} = 0 \quad (15.3)$$

where  $T$  is the kinetic energy of the system,  $U$  is the potential (or strain) energy of the system, and  $q_i$  is the  $i$ -th generalized coordinate used to describe the system. These **generalized coordinates** can include both physical and nonphysical coordinates and will be discussed next lecture in detail. Note that this course will make continually use of Lagrange's equation in the treatment of elastic body dynamics and will be invoked without proof as it's derivation and properties are beyond the scope of this course.

For the simple beam-vibration example previously, possible coordinates to describe the beam's motion could be the transverse displacements of the masses  $Z_i(t)$ , or the angular displacements of the masses,  $\theta_i(t)$  which are related by

$$\begin{bmatrix} \dot{Z}_1 \\ \dot{Z}_2 \end{bmatrix} = \begin{bmatrix} L_1 & 0 \\ L_1 + L_2 & L_2 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \quad (15.4)$$

Recalling systems-of-particles dynamics, the kinetic energy of the beam can be written as

$$T = \frac{1}{2} (m_1 \dot{Z}_1^2 + m_2 \dot{Z}_2^2) = \frac{1}{2} \begin{bmatrix} \dot{Z}_1 \\ \dot{Z}_2 \end{bmatrix}^T \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} \dot{Z}_1 \\ \dot{Z}_2 \end{bmatrix} \quad (15.5)$$

and the potential energy of the beam can be written as

$$U = \frac{1}{2} (k_1 \theta_1^2 + k_2 \theta_2^2) = \frac{1}{2} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}^T \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \quad (15.6)$$

Then, using Lagrange's equation, one has

$$\begin{bmatrix} m_1^2 L_1^2 + m_2 (L_1 + L_2)^2 & m_1 L_2 (L_1 + L_2) \\ m_1 L_2 (L_1 + L_2) & m_2 L_2^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (15.7)$$

which demonstrates the general matrix-vector differential form for *all* lumped-mass vibration problems

$$M \ddot{\vec{q}} + K \vec{q} = \vec{0} \quad (15.8)$$

where  $M$  is the **mass matrix** and  $K$  is the **stiffness matrix**, both of which will be real, symmetric matrices and  $M$  will always be positive-definite, i.e. have strictly positive eigenvalues. Thus, a vibration problem can be completely described by selecting the generalized coordinates, setting up the initial conditions, and finding the mass and stiffness matrices.

## Modal Analysis

Furthermore, this general lumped-mass vibration problem can be rewritten as

$$\ddot{\vec{q}} + D\vec{q} = \vec{0} \quad (15.9)$$

where  $D = M^{-1}K$  is the **dynamic matrix** and always exists since  $M$  is positive-definite. Then, using the standard transformation

$$\vec{q} = \Psi\vec{\eta} \quad (15.10)$$

where  $\Psi$  is the **modal matrix** of  $D$  consisting of its  $n$  eigenvectors,  $\vec{v}_i$  where  $i = 1, \dots, n$ . Furthermore, one can write

$$\Psi^{-1}D\Psi = \Lambda \quad (15.11)$$

where  $\Lambda$  is a diagonal matrix of the  $n$  eigenvalues of  $D$ ,  $\lambda_i$  where  $i = 1, \dots, n$ .

Recall that the eigenvectors satisfy the equation

$$(\lambda_i I - D) \vec{v}_i = \vec{0} \quad (15.12)$$

and

$$\Psi = [\vec{v}_1 | \vec{v}_2 | \cdots | \vec{v}_n] \quad (15.13)$$

Thus, the general lumped-mass vibration problem becomes

$$\ddot{\vec{\eta}} + \Lambda\vec{\eta} = \vec{0} \quad (15.14)$$

whose  $n$  differential equations are now independent, i.e.

$$\ddot{\eta}_i + \lambda_i \eta_i = 0 \quad (15.15)$$

for  $i = 1, \dots, n$  where each has a solution

$$\eta_i(t) = A_i \cos(\sqrt{\lambda_i}t + \Gamma_i) \quad (15.16)$$

where the constants of integration,  $A_i$  and  $\Gamma_i$ , depend on initial conditions. Thus, for general lumped-mass vibration problems, one can find  $n$  natural modes each oscillating at natural frequencies  $\omega_i = \sqrt{\lambda_i}$ . Furthermore, from the definition of  $\Psi$ , one has

$$\vec{q}(t) = [\vec{v}_1 | \vec{v}_2 | \cdots | \vec{v}_n] \vec{\eta}(t) = \sum_{i=1}^n \vec{v}_i \eta_i(t) \quad (15.17)$$

where each modal response  $\eta_i$  contributes to the system response through the eigenvectors or mode shapes. As eigenvectors have arbitrary magnitude, they are typically normalized to a unit length, unity displacement of a selected element, or unity generalized mass as will be shown.

Recall that by definition of  $D$  and  $\vec{v}_i$ , one has

$$\left( \lambda_i I - M^{-1} K \right) \vec{v}_i = 0 \quad (15.18)$$

or

$$\lambda_i M \vec{v}_i = K \vec{v}_i \quad (15.19)$$

Thus, if one multiplies by another eigenvector such that

$$\lambda_i \vec{v}_j^T M \vec{v}_i = v_j^T K \vec{v}_i \quad (15.20)$$

and by the same process, one also has

$$\lambda_j \vec{v}_i^T M \vec{v}_j = v_i^T K \vec{v}_j \quad (15.21)$$

Finally by noting that  $M$  and  $K$  are symmetric, one can further write that

$$(\lambda_i - \lambda_j) \vec{v}_j^T M \vec{v}_i = 0 \quad (15.22)$$

and

$$(\lambda_i - \lambda_j) \vec{v}_j^T K \vec{v}_i = 0 \quad (15.23)$$

which means that if  $\lambda_i \neq \lambda_j \forall i \neq j$ , then the **orthogonality property** holds for the restrained lumped-mass modes, i.e.

$$\vec{v}_j^T M \vec{v}_i = 0, i \neq j \quad (15.24)$$

and

$$\vec{v}_j^T K \vec{v}_i = 0, i \neq j \quad (15.25)$$

Furthermore if  $i = j$ , one then can define the **i-th generalized mass**

$$\mathcal{M}_i = \vec{v}_i^T M \vec{v}_i \quad (15.26)$$

and **i-th generalized stiffness**

$$\mathcal{K}_i = \vec{v}_i^T K \vec{v}_i \quad (15.27)$$

Finally, by this orthogonality property, the definition of  $\Psi$ , and defining  $\mathcal{M} = \text{diag}[\mathcal{M}_1, \dots, \mathcal{M}_n]$  and  $\mathcal{K} = \text{diag}[\mathcal{K}_1, \dots, \mathcal{K}_n]$ , one can alternatively write the lumped-mass vibration equations of motion as

$$\Psi^{-1} M \Psi \ddot{\vec{\eta}} + \Psi^{-1} K \Psi \vec{\eta} = \vec{0} \quad (15.28)$$

$$\mathcal{M} \ddot{\vec{\eta}} + \mathcal{K} \vec{\eta} = \vec{0} \quad (15.29)$$

or

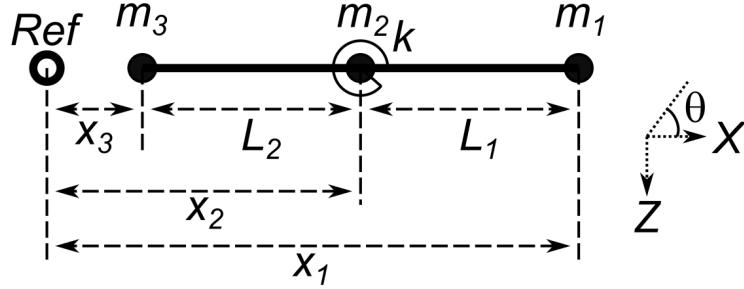
$$\ddot{\vec{\eta}} + \mathcal{M}^{-1} \mathcal{K} \vec{\eta} = \vec{0} \quad (15.30)$$

which demonstrates that

$$\lambda_i = \frac{\mathcal{M}_i}{\mathcal{K}_i} \quad (15.31)$$

## Rigid-Body Degrees of Freedom

This section will consider elastic bodies that are free to translate and/or rotate. Unlike the restrained beam earlier, consider the following unrestrained three-lumped-mass system



where first only the vertical displacement  $Z$  will be analyzed. Note that the bending displacement of the beam occurs by the relative deflection angle  $\theta$  between the lines for rods 1 and 2. This deflection angle can be shown to be (for small angle approximation)

$$\theta = \frac{Z_1 - Z_2}{x_1 - x_2} - \frac{Z_2 - Z_3}{x_2 - x_3} = \left[ \frac{1}{x_1 - x_2} - \frac{1}{x_1 - x_2} - \frac{1}{x_2 - x_3} \quad \frac{1}{x_2 - x_3} \right] \vec{Z} = C \vec{Z} \quad (15.32)$$

where  $C$  is a constraint matrix that relates the beam-displacement coordinates.

Similar to the restrained beam, the kinetic energy of the beam can be written as

$$T = \frac{1}{2} \left( m_1 \dot{Z}_1^2 + m_2 \dot{Z}_2^2 + m_3 \dot{Z}_3^2 \right) = \frac{1}{2} \begin{bmatrix} \dot{Z}_1 \\ \dot{Z}_2 \\ \dot{Z}_3 \end{bmatrix}^T \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{bmatrix} \dot{Z}_1 \\ \dot{Z}_2 \\ \dot{Z}_3 \end{bmatrix} = \frac{1}{2} \dot{\vec{Z}}^T M \dot{\vec{Z}} \quad (15.33)$$

and the potential energy of the beam can be written as

$$U = \frac{1}{2} k \theta^2 = \frac{1}{2} \theta^T k \theta = \frac{1}{2} \vec{Z}^T C^T k C \vec{Z} = \frac{1}{2} \vec{Z}^T K_c \vec{Z} \quad (15.34)$$

where  $K_c$  is the **constrained stiffness matrix**. Then, again using Lagrange's equation, one has

$$M \ddot{\vec{Z}} + K_c \dot{\vec{Z}} = \vec{0} \quad (15.35)$$

or defining  $D_c = M^{-1} K_c$  as the **constrained dynamic matrix**, one has

$$\ddot{\vec{Z}} + D_c \dot{\vec{Z}} = \vec{0} \quad (15.36)$$

Continuing with the modal analysis one has

$$\lambda_i \vec{v}_i = D_c \vec{v}_i \quad (15.37)$$

or for two eigenvalues/eigenvectors

$$\lambda_i M \vec{v}_i = K_c \vec{v}_i \quad (15.38)$$

$$\lambda_j M \vec{v}_j = K_c \vec{v}_j \quad (15.39)$$

and repeating the previous equations as  $M$  and  $K_c$  are still symmetric, one has

$$(\lambda_i - \lambda_j) \vec{v}_j^T M \vec{v}_i = 0 \quad (15.40)$$

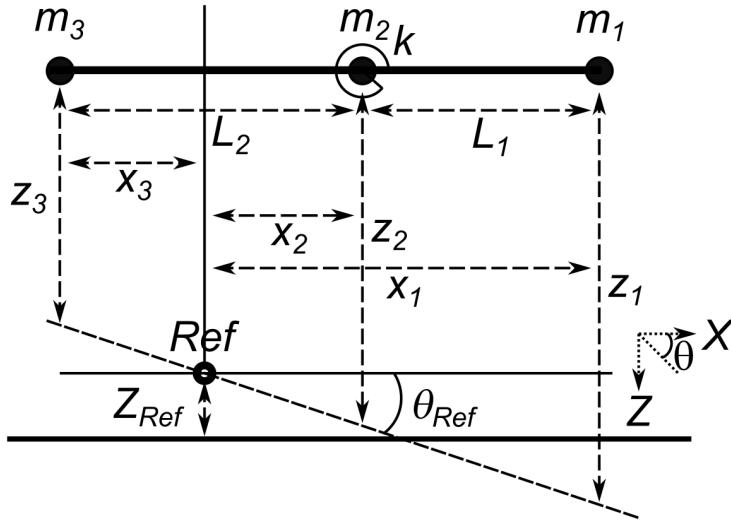
and

$$(\lambda_i - \lambda_j) \vec{v}_j^T K_c \vec{v}_i = 0 \quad (15.41)$$

However, for the unrestrained beam, two of the eigenvalues of  $D$  can be shown to be 0 and hence equal due to the existence of two rigid-body degrees-of-freedom (DOF), vertical translation and rotation of the *entire* beam. Thus, this system will have two rigid-body modes and a single vibration mode corresponding to the non-zero eigenvalue and associated eigenvector.

Thus, further work must be done to describe the entire elastic body motion in terms of mutually orthogonal or **normal** modes. From linear algebra, if a matrix has repeated eigenvalues, *any* linear combination of the eigenvectors associated with the repeated eigenvalues are also eigenvectors of the given matrix. This fact can be used to obtain mutually orthogonal modes for unrestrained bodies.

Before this is derived further, consider an alternate approach that will consider the rigid-body degrees-of-freedom more directly, in particular consider the total (i.e. inertial) vertical position of the masses are referenced to some reference axis, as demonstrated in the following three-lumped-mass example



Here the total vertical displacements of the lumped-masses (with the small angle approximation) are

$$\begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \end{bmatrix} = \begin{bmatrix} Z_{Ref} \\ Z_{Ref} \\ Z_{Ref} \end{bmatrix} + \begin{bmatrix} x_1 \\ x_2 \\ -x_3 \end{bmatrix} \theta_{Ref} + \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \quad \vec{Z} = \vec{Z}_{Ref} + \vec{x} \theta_{Ref} + \vec{z} \quad (15.42)$$

Using Lagrange's equation for the equations of motion, the kinetic energy can be written as

$$T = \frac{1}{2} \vec{Z}^T M \vec{Z} \quad (15.43)$$

and the potential (or strain) energy can be written as

$$U = \frac{1}{2} \theta^T k \theta \quad (15.44)$$

As before, one can model the angular displacement as

$$\theta = \frac{z_1 - z_2}{x_1 - x_2} - \frac{z_2 - z_3}{x_2 + x_3} = \begin{bmatrix} \frac{1}{x_1 - x_2} & \left( -\frac{1}{x_1 - x_2} - \frac{1}{x_2 + x_3} \right) & \frac{1}{x_2 + x_3} \end{bmatrix} \vec{Z} = C \vec{Z} \quad (15.45)$$

Thus,

$$U = \frac{1}{2} \vec{z}^T C^T k C \vec{z} = \frac{1}{2} \vec{z}^T K_c \vec{z} \quad (15.46)$$

where it should be noted that  $T$  is a function of the inertial velocities, while  $U$  is a function of the relative displacements. This will need to be resolved before using Lagrange's equation on the generalized coordinates.

Next, two additional constraints must be imposed. First, in the absence of external forces and moments, both translational and rotational momenta must be constant. Taking this arbitrary constant to 0 results in

$$m_1 \dot{Z}_1 + m_2 \dot{Z}_2 + m_3 \dot{Z}_3 = \vec{1}^T M \vec{Z} = 0 \quad (15.47)$$

and

$$m_1 x_1 \dot{Z}_1 + m_2 x_2 \dot{Z}_2 - m_3 x_3 \dot{Z}_3 = \vec{x}^T M \vec{Z} = 0 \quad (15.48)$$

These two constraints imply that  $\dot{\vec{Z}}$  must be orthogonal (with respect to  $M$ ) to the vectors  $\vec{1}$  and  $\vec{x}$ . Therefore, one can define  $\vec{Z}_c$  and  $\vec{z}$  as the absolute and relative vertical displacement velocity which satisfy these constraints, respectively. Furthermore, from this analysis one can infer that  $\vec{1}$  and  $\vec{x}$  may be appropriate rigid-body mode shapes. If so, then these two shapes must be mutually orthogonal (with respect to  $M$ ), i.e.

$$m_1 x_1 + m_2 x_2 - m_3 x_3 = \vec{1}^T M \vec{x} = 0 \quad (15.49)$$

which is equivalent to setting the reference point at the center of mass of the beam, point  $G$ . Lastly, recall that the total mass of the beam can be written as

$$M_{tot} = \vec{1}^T M \vec{1} \quad (15.50)$$

and the moment of inertia of the beam about its center of mass,  $G$ , can be written as

$$I_G = \sum_{i=1}^3 m_i x_i^2 = \vec{x}^T M \vec{x} \quad (15.51)$$

Then, rewriting Equation 15.42 in terms of constrained displacements, one has

$$\vec{Z}_c = \vec{1} Z_{Ref} + \vec{x} \theta_{Ref} + \vec{z}_c \quad (15.52)$$

from which if one can invoke the constraints and relative motion  $\vec{z}_c$  in terms of mutually orthogonal modal responses, then the desired solution to the vibration problem will be derived.

To that end, differentiating Equation 15.52 with respect to time and using the momenta constraints, one has

$$\vec{1}^T M \left[ \vec{1} \dot{Z}_{Ref} + \vec{x} \dot{\theta}_{Ref} + \vec{z}_c \right] = 0 \quad (15.53)$$

and

$$\vec{x}^T M \left[ \vec{1} \dot{Z}_{Ref} + \vec{x} \dot{\theta}_{Ref} + \vec{z}_c \right] = 0 \quad (15.54)$$

Then, noting the total mass and moment of inertia equations and center of mass constraint, one has

$$\dot{Z}_{Ref} = -\frac{1}{M_{tot}} \vec{1}^T M \vec{z}_c \quad (15.55)$$

and

$$\dot{\theta}_{Ref} = \frac{1}{I_G} \vec{x}^T M \vec{z}_c \quad (15.56)$$

Finally, applying these two constraints, one can write that constrained total velocities as functions of the constrained relative velocities as

$$\vec{Z}_c = \left[ I_{3 \times 3} - \frac{1}{M_{tot}} \vec{1} \vec{1}^T M - \frac{1}{I_G} \vec{x} \vec{x}^T M \right] \vec{z}_c = \Xi \vec{z}_c \quad (15.57)$$

Then, the kinetic energy in terms of the constrained relative velocities can be written as

$$T = \frac{1}{2} \vec{z}_c^T \Xi^T M \Xi \vec{z}_c = \frac{1}{2} \vec{z}_c^T M_c \vec{z}_c \quad (15.58)$$

and the potential (or strain) energy in terms of the constrained relative displacements can be written as

$$U = \frac{1}{2} \vec{z}_c^T K_c \vec{z}_c \quad (15.59)$$

Then utilizing  $\vec{z}_c$  as the generalized coordinates  $\vec{q}$  in Lagrange's equation in vector form, i.e.

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{\vec{q}}} \right) - \frac{\partial T}{\partial \vec{q}} + \frac{\partial U}{\partial \vec{q}} = \vec{0}^T \quad (15.60)$$

one has for the unrestrained beam-vibration equations of motion

$$M_c \ddot{\vec{z}}_c + K_c \vec{z}_c = \vec{0} \quad (15.61)$$

where the constrained mass matrix  $M_c$  is now singular and its inverse does not exist. Thus, to solve for the mode shapes and vibration frequencies, one must solve the **generalized eigenvalue problem**, i.e.

$$(\lambda_i M_c - K_c) \vec{v}_i = 0 \quad (15.62)$$

## 15.2 Elastic Body Dynamics

### Modal Analysis of Generalized Eigenvalue Problem

With the unrestrained beam, the modal analysis of the generalized eigenvalue problem will provide a model for incorporating vibration modes into the rigid body dynamics to produce elastic body dynamics. Recall the generalized eigenvalue problem from the previous section

$$\lambda_i M_c \vec{v}_i = K_c \vec{v}_i \quad (15.63)$$

So, as before, consider two  $i, j$  pairs of generalized eigenvalues and eigenvectors

$$(\lambda_i - \lambda_j) \vec{v}_i^T M_c \vec{v}_j = 0 \quad (15.64)$$

However, when  $i \neq j$  and the eigenvalues are distinct, e.g. not both zero, this equation assures that the two associated eigenvectors are orthogonal with respect to the *constrained* mass matrix, not the mass matrix  $M$ , as required. From the definition of  $M_c$ , one can write

$$(\lambda_i - \lambda_j) \vec{v}_i^T \Xi^T M \Xi \vec{v}_j = 0 \quad (15.65)$$

or defining the **constrained eigenvectors** as

$$\vec{v}_{c,i} = \Xi \vec{v}_i \quad (15.66)$$

Thus, when  $i \neq j$  and the two eigenvalues are distinct,  $\vec{v}_{c,i}$  are mutually orthogonal with respect to the mass matrix  $M$ . Note that for the unrestrained beam example, two of the transformed generalized eigenvectors will be equal to  $\vec{0}$  due to the two rigid body modes and one will satisfy the three orthogonal constraints, the single vibration mode shape  $\vec{v}_{vib}$ . For an  $n$ -lumped-mass beam, one can extend this to the relative displacement equation for  $n - 2$  vibration modes

$$\vec{z}_c(t) = \sum_{i=1}^{n-2} \vec{v}_{vib,i} A_i \cos(\omega_{vib,i} t + \Gamma_i) \quad (15.67)$$

Note that all *vibration* mode shapes will be mutually orthogonal with respect to the mass matrix. To prove that these are also orthogonal to the rigid body mode shapes,  $\vec{1}$  and  $\vec{x}$ , first consider the relative motion  $\vec{z}_c$  as a function of the original eigenvectors.

$$\vec{z}_c(t) = \sum_{i=1}^n \vec{v}_i \eta_i(t) \quad (15.68)$$

Next, recall the orthogonality constraints for the  $\dot{\vec{Z}}_c$  relative to  $\vec{1}$  and  $\vec{x}$ , i.e.

$$\vec{1}^T M \dot{\vec{Z}} = 0 \quad (15.69)$$

and

$$\vec{x}^T M \dot{\vec{Z}} = 0 \quad (15.70)$$

as well as the relation

$$\dot{\vec{Z}} = \Xi \vec{z}_c \quad (15.71)$$

Thus, one has

$$\vec{1}^T M \dot{\vec{Z}} = \vec{1}^T M \Xi \vec{z}_c = \vec{1}^T M \Xi \sum_{i=1}^n \vec{v}_i \dot{\eta}_i(t) = 0 \quad (15.72)$$

and

$$\vec{x}^T M \dot{\vec{Z}} = \vec{x}^T M \Xi \vec{z}_c = \vec{x}^T M \Xi \sum_{i=1}^n \vec{v}_i \dot{\eta}_i(t) = 0 \quad (15.73)$$

Finally, by inspection, this requires

$$\vec{1}^T M \Xi \vec{v}_i = \vec{1}^T M \vec{v}_{c,i} = 0 \quad \forall i \quad (15.74)$$

and

$$\vec{x}^T M \Xi \vec{v}_i = \vec{x}^T M \vec{v}_{c,i} = 0 \quad \forall i \quad (15.75)$$

These equations signify that all the constrained eigenvectors are orthogonal to  $\vec{x}$  with respect to  $M$ . Thus, these constrained eigenvectors must be the desired orthogonal vibration mode shapes. Furthermore, all these eigenvectors (including  $\vec{1}$  and  $\vec{x}$ ) are mutually orthogonal with respect to  $M$ .

The key idea for orthogonality among the modal decomposition is that the physical responses of the unrestrained beam can be expressed in terms of the linear combination of *mutually orthogonal* modes, i.e.

$$\sum_{i=1}^n \vec{v}_{c,i} \eta_i = \vec{1} \eta_n + \vec{x} \eta_{n-1} + \sum_{i=1}^{n-2} \vec{v}_{vib,i} \eta_i \quad (15.76)$$

where the rigid body and vibration modes derived previously are also known as the **normal modes**. Then, the unrestrained three-lumped mass model had the following model for the vertical displacements (with the small angle approximation)

$$\begin{aligned} \vec{Z}(t) &= \vec{1} Z_{Ref}(t) + \vec{x} \theta_{Ref}(t) + \vec{z}_{vib}(t) \\ \vec{Z}(t) &= \vec{1} Z_{Ref}(t) + \vec{x} \theta_{Ref}(t) + \sum_{i=1}^{n-2} \vec{v}_{vib,i} \eta_i(t) \end{aligned} \quad (15.77)$$

Thus, with this consideration, one can form generalized coordinates for deriving the equation of motion for the vertical displacement of an elastic body as

$$\begin{aligned} \vec{Z} &= [\vec{1} \quad \vec{x} \quad \vec{v}_{vib,1} \quad \cdots \quad \vec{v}_{vib,n-2}] \begin{bmatrix} Z_{Ref} \\ \theta_{Ref} \\ \eta_1 \\ \vdots \\ \eta_{n-2} \end{bmatrix} \\ &= \Psi \vec{q} \end{aligned} \quad (15.78)$$

Then, the kinetic energy of the beam can be rewritten

$$\begin{aligned} T &= \frac{1}{2} \dot{\vec{Z}} M \dot{\vec{Z}} = \frac{1}{2} \dot{\vec{q}} \Psi^T M \Psi \dot{\vec{q}} \\ &= \frac{1}{2} M_{tot} \dot{Z}_{Ref}^2 + \frac{1}{2} I_G \dot{\theta}_{Ref}^2 + \frac{1}{2} \dot{\vec{\eta}}_{vib}^T \mathcal{M}_{vib} \dot{\vec{\eta}}_{vib} \\ &= \frac{1}{2} \dot{\vec{q}} \mathcal{M} \dot{\vec{q}} \end{aligned} \quad (15.79)$$

where the rigid body kinetic energy terms and elastic kinetic energy terms notably linearly combine, the **generalized mass matrix** is

$$\mathcal{M} = \begin{bmatrix} M_{tot} & 0 & 0 \\ 0 & I_G & 0 \\ 0 & 0 & \mathcal{M}_{vib} \end{bmatrix}, \quad (15.80)$$

and

$$\vec{\eta}_{vib} = [\eta_1 \quad \cdots \quad \eta_{n-2}]^T. \quad (15.81)$$

The potential (or strain) energy can be rewritten as

$$U = \frac{1}{2} \vec{z}^T K_c \vec{z} = \frac{1}{2} \vec{\eta}_{vib}^T \Psi_{vib}^T K_c \Psi_{vib} \vec{\eta}_{vib} = \frac{1}{2} \vec{\eta}_{vib}^T \mathcal{K}_{vib} \vec{\eta}_{vib} \quad (15.82)$$

where  $\mathcal{K}_{vib}$  is the generalized stiffness matrix and  $\Psi_{vib}$  is the vibration modal matrix, i.e.

$$\Psi_{vib} = [\vec{v}_{vib,1} \quad \cdots \quad \vec{v}_{vib,n-2}] \quad (15.83)$$

Finally, using Lagrange's equation for these energy expressions, one has

$$\mathcal{M} \ddot{\vec{q}} + \mathcal{K} \vec{q} = \begin{bmatrix} M_{tot} & 0 & 0 \\ 0 & I_G & 0 \\ 0 & 0 & \mathcal{M}_{vib} \end{bmatrix} \ddot{\vec{q}} + \begin{bmatrix} 0 & 0 \\ 0 & \mathcal{K}_{vib} \end{bmatrix} \vec{q} = 0 \quad (15.84)$$

which can alternatively be written as

$$\begin{aligned} M_{tot} \ddot{Z}_{Ref} &= 0 \\ I_G \ddot{\theta}_{Ref} &= 0 \\ \mathcal{M}_{vib} \ddot{\vec{\eta}}_{vib} + \mathcal{K}_{vib} \vec{\eta}_{vib} &= 0 \end{aligned} \quad (15.85)$$

for the free response of the unrestrained beam's EOMs which will be fundamental for the elastic body flight dynamics.

## Forced Motion and Virtual Work

Next consider the addition of external forces on the lumped-mass systems which will result in how one can incorporate the aerodynamic forces into the elastic body flight dynamics. To do so, consider Lagrange's equation with external forces applied

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{\vec{q}}} \right) - \frac{\partial T}{\partial \vec{q}} + \frac{\partial U}{\partial \vec{q}} = \vec{Q}^T \quad (15.86)$$

where  $Q$  is the generalized force, i.e.

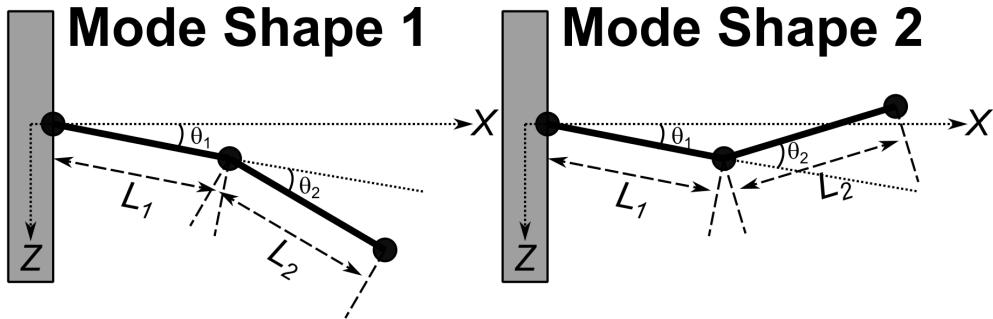
$$\vec{Q}^T = \frac{\partial \delta W}{\partial \delta \vec{q}} \quad (15.87)$$

where  $\delta \vec{q}$  is a virtual displacement of the generalized coordinates  $\vec{q}$  and the **virtual work** is defined as

$$\begin{aligned} \delta W &= \sum_{i=1}^m \vec{F}_i \cdot \delta \vec{d}_i \\ \delta W &= [\vec{F}_1^T \quad \dots \quad \vec{F}_m^T]^T \begin{bmatrix} \delta \vec{d}_1 \\ \vdots \\ \delta \vec{d}_m \end{bmatrix} \end{aligned} \quad (15.88)$$

where  $\delta \vec{d}_i$  is the virtual displacement of the point of application of force  $F_i$ .

As an example, consider the vertical displacement of two-lumped-mass restrained beam shown here with its two mode shapes.



Defining  $\hat{k}$  as the unit vector for the  $Z$  direction, one can define the vertical forces as

$$\vec{F}_1 = F_1 \hat{k} \quad \vec{F}_2 = F_2 \hat{k} \quad (15.89)$$

and the virtual physical displacements at the points of application as

$$\begin{aligned} \delta \vec{d}_1 &= \delta Z_1 \hat{k} \\ \delta \vec{d}_2 &= \delta Z_2 \hat{k} \end{aligned} \quad (15.90)$$

Thus,

$$\begin{aligned} \delta W &= F_1 \delta Z_1 + F_2 \delta Z_2 \\ \delta W &= [F_1 \quad F_2] \begin{bmatrix} \delta Z_1 \\ \delta Z_2 \end{bmatrix} \end{aligned} \quad (15.91)$$

where

$$\begin{bmatrix} \delta Z_1 \\ \delta Z_2 \end{bmatrix} = \begin{bmatrix} L_1 & 0 \\ L_1 + L_2 & L_2 \end{bmatrix} \begin{bmatrix} \delta \theta_1 \\ \delta \theta_2 \end{bmatrix} \quad (15.92)$$

Now, assuming that the unforced (i.e. “free”) vibration problem has been solved in terms of  $\theta_1$  and  $\theta_2$ , then the elements of the free-vibration mode shapes would correspond to angular displacements and the virtual displacements  $\delta\theta_1$  and  $\delta\theta_2$  can be expressed in terms of these mode shapes and two vibration modal coordinates,  $\eta_1$  and  $\eta_2$  as

$$\begin{bmatrix} \delta\theta_1 \\ \delta\theta_2 \end{bmatrix} = [\vec{v}_1 \quad \vec{v}_2] \begin{bmatrix} \delta\eta_1 \\ \delta\eta_2 \end{bmatrix}$$

$$\delta\vec{\theta} = \Psi\delta\vec{\eta}$$
(15.93)

Then, by substitution the virtual work can be written as

$$\delta W = [F_1 \quad F_2] \begin{bmatrix} L_1 & 0 \\ L_1 + L_2 & L_2 \end{bmatrix} \Psi\delta\vec{\eta}$$

$$\delta W = \vec{\mathcal{F}}^T \Psi\delta\vec{\eta}$$
(15.94)

where  $\vec{\mathcal{F}}$  is the vector of applied forces relative to the virtual displacements. Finally, recall one can write the kinetic energy in terms of the modal coordinates as

$$T = \frac{1}{2} \dot{\vec{\theta}}^T M \dot{\vec{\theta}} = \frac{1}{2} \dot{\vec{\eta}}^T \Psi^T M \Psi \dot{\vec{\eta}} = \frac{1}{2} \dot{\vec{\eta}}^T \mathcal{M} \dot{\vec{\eta}}$$
(15.95)

and the potential (or strain) energy in terms of the modal coordinates as

$$U = \frac{1}{2} \vec{\theta}^T K \vec{\theta} = \frac{1}{2} \vec{\eta}^T \Psi^T K \Psi \vec{\eta} = \frac{1}{2} \vec{\eta}^T \mathcal{K} \vec{\eta}$$
(15.96)

Then, using Lagrange’s equation with external forces applied results in

$$\mathcal{M}\ddot{\vec{\eta}} + \mathcal{K}\vec{\eta} = Q = \Psi^T \vec{\mathcal{F}}$$
(15.97)

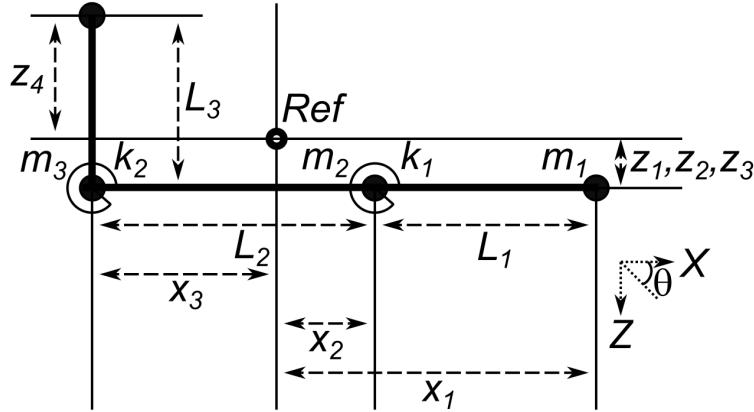
Furthermore, for the unrestrained three-lumped-mass beam as described earlier, one can derive a similar expression which results in the form

$$\begin{aligned} M_{tot} \ddot{\vec{Z}}_{Ref} &= F_1 + F_2 + F_3 \\ I_G \ddot{\theta}_{Ref} &= F_1 x_1 + F_2 x_2 - F_3 x_3 \\ \mathcal{M}_{vib} \ddot{\vec{\eta}}_{vib} + \mathcal{K}_{vib} \vec{\eta}_{vib} &= Q = \Psi_{vib}^T \vec{\mathcal{F}} \end{aligned}$$
(15.98)

for the forced response of the unrestrained beam’s EOMs which will be fundamental for the elastic body flight dynamics. Recall that the *Ref* here is the center of mass of the lumped-mass system. This can easily be extended to  $n$ -lumped-mass systems.

### Multi-Directional Elastic Motion

Lastly, to extend these results to multi-directional motion requires more generalized vectors and matrices as each element of a mode shape/eigenvector can only correspond to direction of motion. To demonstrate this, consider the following bi-directional example of a unforced 2D truss.



where the directions of motion are  $X$  and  $Z$  and the reference point,  $Ref$  is the center of mass of the truss. The kinetic energy of the truss can be written as

$$\begin{aligned} T &= \frac{1}{2} [m_1(\dot{X}_1^2 + \dot{Z}_1^2) + m_2(\dot{X}_2^2 + \dot{Z}_2^2) + m_3(\dot{X}_3^2 + \dot{Z}_3^2) + m_4(\dot{X}_4^2 + \dot{Z}_4^2)] \\ &= \frac{1}{2} [\dot{\vec{X}}^T \quad \dot{\vec{Z}}^T] \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} \dot{\vec{X}} \\ \dot{\vec{Z}} \end{bmatrix} \end{aligned} \quad (15.99)$$

The potential (or strain) energy of the truss can be written as

$$U = \frac{1}{2} k_1 \theta_1^2 + \frac{1}{2} k_2 \theta_2^2 = \frac{1}{2} [\theta_1 \quad \theta_2] \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \quad (15.100)$$

where these two deflections  $\theta_1$  &  $\theta_2$  are the relative angular displacements between rods 1 and 2 & 2 and 3, respectively. From the geometry of the truss structure (and small angle approximations), one can form the following geometric constraints for these relative angular displacements as

$$\begin{aligned} \theta_1 &= \frac{Z_1 - Z_2}{x_1 - x_2} - \frac{Z_2 - Z_3}{x_2 + x_3} \\ &= \left[ \frac{1}{x_1 - x_2} \quad \left( \frac{-1}{x_1 - x_2} - \frac{1}{x_2 + x_3} \right) \quad \frac{1}{x_2 + x_3} \quad 0 \right] \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \end{bmatrix} \\ &= C_1 \vec{Z} \end{aligned} \quad (15.101)$$

and

$$\begin{aligned}\theta_2 &= \frac{Z_3 - Z_2}{x_2 + x_3} - \frac{X_3 - X_4}{z_3 + z_4} \\ &= \begin{bmatrix} 0 & 0 & \frac{-1}{z_3+z_4} & \frac{1}{z_3+z_4} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} + \begin{bmatrix} 0 & \left(\frac{-1}{x_2+x_3} - \frac{1}{x_2+x_3}\right) & \frac{1}{x_2+x_3} & 0 \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \end{bmatrix} \\ &= [C_2 \quad C_3] \begin{bmatrix} \vec{X} \\ \vec{Z} \end{bmatrix}\end{aligned}\quad (15.102)$$

Furthermore, assuming equal vibration displacements for the colinear masses, one has

$$\begin{aligned}X_1 &= X_2 = X_3 \\ \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_3 \\ X_4 \end{bmatrix} \\ \vec{X} &= C'_x \vec{X}'\end{aligned}\quad (15.103)$$

$$\begin{aligned}Z_3 &= Z_4 \\ \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \end{bmatrix} \\ \vec{Z} &= C'_z \vec{Z}'\end{aligned}\quad (15.104)$$

where the degrees of freedom of the truss has been reduced using these constraint matrices, resulting in the modified coordinate vectors,  $\vec{X}'$  and  $\vec{Z}'$ .

With these constraints, the kinetic energy of the truss can be rewritten as

$$\begin{aligned}T &= \frac{1}{2} \begin{bmatrix} \dot{\vec{X}}'^T & \dot{\vec{Z}}'^T \end{bmatrix} \begin{bmatrix} C'_x & 0 \\ 0 & C'_z \end{bmatrix}^T \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} C'_x & 0 \\ 0 & C'_z \end{bmatrix} \begin{bmatrix} \dot{\vec{X}}' \\ \dot{\vec{Z}}' \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} \dot{\vec{X}}'^T & \dot{\vec{Z}}'^T \end{bmatrix} [MM]' \begin{bmatrix} \dot{\vec{X}}' \\ \dot{\vec{Z}}' \end{bmatrix}\end{aligned}\quad (15.105)$$

and the potential (or strain) energy of the truss can be rewritten as

$$\begin{aligned}
 U &= \frac{1}{2} [\vec{X}^T \quad \vec{Z}^T] \begin{bmatrix} 0 & C_1 \\ C_2 & C_3 \end{bmatrix}^T \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \begin{bmatrix} 0 & C_1 \\ C_2 & C_3 \end{bmatrix} [\vec{X}] [\vec{Z}] \\
 &= \frac{1}{2} [\vec{X}^T \quad \vec{Z}^T] K_c \begin{bmatrix} \vec{X} \\ \vec{Z} \end{bmatrix} \\
 &= \frac{1}{2} [\vec{X}'^T \quad \vec{Z}'^T \begin{bmatrix} C'_x & 0 \\ 0 & C'_z \end{bmatrix}^T] K_c \begin{bmatrix} C'_x & 0 \\ 0 & C'_z \end{bmatrix} \begin{bmatrix} \vec{X}' \\ \vec{Z}' \end{bmatrix} \\
 &= \frac{1}{2} [\vec{X}'^T \quad \vec{Z}'^T] K'_c \begin{bmatrix} \vec{X}' \\ \vec{Z}' \end{bmatrix}
 \end{aligned} \tag{15.106}$$

Finally, solving Lagrange's equation using these newly defined modified constrained mass matrix  $[MM]'$  and modified constrained stiffness matrix  $K'_c$ , one has

$$[MM]' \begin{bmatrix} \ddot{\vec{X}} \\ \ddot{\vec{Z}} \end{bmatrix} + K'_c \begin{bmatrix} \vec{X} \\ \vec{Z} \end{bmatrix} = 0 \tag{15.107}$$

which has a modified constrained dynamic matrix  $D'_c = [MM]^{-1} K'_c$ . Thus, the same modal analysis for mutually orthogonal modes and coordinates previously performed can also be done for multi-directional motion as long as the “proper” mass and stiffness matrices are used.

In the end, this lecture and the previous lecture have derived generalized coordinates to model vibration problems where the vibration modes are orthogonal to the rigid body modes and demonstrated this for some simple lumped-mass models to assist in visualizing the construction of these elastic body dynamic models. As such, these types of coordinate frames satisfy the **mean-axis constraints** which will be used for the mean axes derived in the equations of motion for describing elastic body flight dynamics in the subsequent lectures.

### 15.3 Elastic Vehicle Mean Axes

The perspective in this course on elastic body dynamics is the ability to study the effects of elastic deformation on the flight dynamics of a vehicle, not to study purely structural/aeroelastic phenomena such as flutter or divergence. Thus, only the lower frequency modes are typically of interest for addition to the rigid body 6-DOF developed in introductory flight dynamics. Though all real vehicles are elastic, in many cases, this type of vibration analysis may only be necessary to check if the rigid body assumption can be made for the modeling of the flight vehicle. In general, larger flight vehicles typically require some elastic modeling as the natural frequencies will be lower and more likely to interact with the rigid body or flight controller modes of the vehicle.

The treatment for elastic vehicles will continue to make use of Lagrangian mechanics and energy concepts which will be stated here without proof, as typically these are developed in an *intermediate dynamics* graduate course. These concepts will allow the derivation of important results for the reference frame requirements for elastic vehicles, namely the mean axes, which will be developed in this lecture. Before developing the concept of mean axes, first assume that the navigation frame  $N$  is inertial, i.e. “flat-Earth” approximation. Note that this course will later account for the ECI and ECEF frames on the vehicle dynamics.

## Lagrangian Energies for Vehicles

Let the kinetic energy of the entire vehicle be

$$T = \frac{1}{2} \int_{Vol} \vec{v}_N \cdot \vec{v}_N \rho_V dV \quad (15.108)$$

where  $\vec{v}_N$  is the inertial velocity of a mass element of the vehicle, i.e.

$$\vec{v}_N = \vec{v}_{B/N,N} + \vec{v}_B + \vec{\omega}_{B/N} \times \vec{x}_B \quad (15.109)$$

where  $\vec{v}_{B/N}$  is the velocity of the body frame relative to the navigation frame,  $\vec{v}_B$  is the velocity of the mass element in the body frame,  $\vec{\omega}_{B/N}$  is the angular velocity of the body frame relative to the navigation frame, and  $\vec{x}_B$  is the position of the mass element in the body frame, i.e. relative to the center of mass. In addition, one has

$$\vec{x}_N = \vec{x}_B + \vec{x}_{B/N} \quad (15.110)$$

where  $\vec{x}_{B/N}$  is the position of the origin of the body frame relative to the navigation frame. Then, by substitution, one has

$$\begin{aligned} T = \frac{1}{2} \int_{Vol} & [ \vec{v}_{B/N,N} \cdot \vec{v}_{B/N,N} \\ & + 2 \vec{v}_{B/N,N} \cdot \vec{v}_B \\ & + 2 (\vec{v}_{B/N,N} + \vec{v}_B) \cdot (\vec{\omega}_{B/N} \times \vec{x}) \\ & + \vec{v}_B \cdot \vec{v}_B \\ & + (\vec{\omega}_{B/N} \times \vec{x}) \cdot (\vec{\omega}_{B/N} \times \vec{x}) ] \rho_V dV \end{aligned} \quad (15.111)$$

Furthermore, the potential energy of the vehicle includes gravitational potential energy  $U_g$  and elastic strain energy  $U_e$ . The gravitational potential energy for the vehicle can be modeled as

$$\begin{aligned} U_g &= - \int_{Vol} \vec{g} \cdot \vec{x}_N \rho_V dV \\ U_g &= - \int_{Vol} \vec{g} \cdot (\vec{x}_B + \vec{x}_{B/N}) \rho_V dV \end{aligned} \quad (15.112)$$

where  $\vec{g}$  is the acceleration due to gravity. The elastic strain energy is the energy stored in an elastic structure due to its deformation resulting from some applied force. The strain energy is the negative of the work done on the structure by the applied force, and the work is the force acting over a displacement  $\vec{d}_e$ . Thus, the position of a mass element of the vehicle can be represented as

$$\vec{x} = \vec{x}_{rb} + \vec{d}_e \quad (15.113)$$

where  $\vec{x}_{rb}$  is the position of the mass element in terms of its undeformed or rigid body position. Furthermore, since  $\vec{x}_{rb,B}$  is invariant with respect to the body frame, one has

$$\vec{v}_B = \dot{\vec{d}}_{e,B} \quad (15.114)$$

Thus, using D'Alembert's principle to express the force on a mass element in terms of the mass of the element and its acceleration, one has for the elastic strain energy

$$U_e = -\frac{1}{2} \int_{Vol} \ddot{\vec{d}}_{e,B} \cdot \vec{d}_{e,B} \rho_V dV \quad (15.115)$$

### Mean-Axes Body Frame

For rigid vehicle dynamics, the selection of body frame axes was arbitrary with the only requirement being the body frame origin as the vehicle's center of mass. Such a construction decoupled the rotation mode and the translation mode of the dynamics. For elastic vehicle dynamics, one must further consider the existence of any number of vibration modes which results in *additional* requirements for a body frame to exhibit decoupled dynamic modes, namely the **mean-axes constraints** which define coordinate axes about which the relative translational and angular momenta (about the center of mass) due to elastic vibrations are zero, i.e.

$$\int_{Vol} \vec{v}_B \rho_V dV = \int_{Vol} \vec{x}_B \times \vec{v}_B \rho_V dV = \vec{0} \quad (15.116)$$

This “special” **mean-axes body frame** can be shown to always exist for an elastic body which will be assumed here and not proven. By substitution for these quantities from the relations in the previous section, one has

$$\int_{Vol} \frac{d}{dt} (\vec{x}_{rb} + \vec{d}_{e,B}) \rho_V dV = \vec{0} \quad (15.117)$$

and

$$\int_{Vol} \vec{x}_{rb} \times \dot{\vec{d}}_{e,B} \rho_V dV + \int_{Vol} \vec{d}_{e,B} \times \vec{v}_B \rho_V dV = \vec{0} \quad (15.118)$$

which if the elastic displacement is sufficiently small that only linear effects are considered, then one can neglect the moment from the elastic displacements, and one has the *practical mean-axes constraints* written as

$$\int_{Vol} \dot{\vec{d}}_{e,B} \rho_V dV = \int_{Vol} \vec{x}_{rb} \times \dot{\vec{d}}_{e,B} \rho_V dV = \vec{0} \quad (15.119)$$

which are analogous to the modal orthogonality constraints for the lumped-mass systems considered earlier. Thus, while the mean-axes will be used for theoretical development, in practice, one uses the practical mean-axes constraints to confirm the selected axes are mean-axes through mutual orthogonality among all modes.

To demonstrate this, assume a free-vibration analysis has been performed as previously described yielding the  $n$  free-vibration mutually orthogonal mode shapes,  $\vec{v}_i$ , and frequencies,  $\omega_i = \sqrt{\lambda_i}$ , (i.e. eigenvectors and eigenvalues) including both the rigid body and elastic modes. Then, the elastic vibrations can be expressed as

$$\vec{d}_{e,B} = \sum_{i=1}^n \vec{v}_i(\vec{x}) \eta_i(t) \quad (15.120)$$

where  $\eta_i(t)$  is the generalized coordinate associated with the  $i$ -th vibration mode. In general, each mode shape,  $\vec{v}_i(\vec{x})$ , is a vector with components defined in the body frame with each component a function of the  $\vec{x}$  location on the *undeformed* structure. With this analysis, the *practical mean-axes constraints* can be rewritten as

$$\int_{Vol} \dot{\vec{d}}_{e,B} \rho_V dV = \sum_{i=1}^n \dot{\eta}_i(t) \left( \int_{Vol} \vec{v}_i(\vec{x}) \rho_V dV \right) = 0 \quad (15.121)$$

and

$$\int_{Vol} \vec{x}_{rb} \times \dot{\vec{d}}_{e,B} \rho_V dV = \sum_{i=1}^n \left( \int_{Vol} \vec{x}_{rb} \times \vec{v}_i(\vec{x}) \rho_V dV \right) = 0 \quad (15.122)$$

which are satisfied as the integrals inside the parentheses above correspond to the momenta conservation requirements and the selected vibration modes are, by design, orthogonal to the rigid body translation and rotation modes (with respect to the mass distribution here).

### Mean-Axis Constraints on Energies

Now, one can apply the *mean-axis constraints* to the energies to greatly simplify these equations. For the first term of the kinetic energy, as the velocity of the center of mass is independent of the volume, one has

$$\begin{aligned} \int_{Vol} \vec{v}_{B/N,N} \cdot \vec{v}_{B/N,N} \rho_V dV &= \vec{v}_{B/N,N} \cdot \vec{v}_{B/N,N} \int_{Vol} \rho_V dV \\ &= m \vec{v}_{B/N,N} \cdot \vec{v}_{B/N,N} \end{aligned} \quad (15.123)$$

where  $m$  is the total mass of the vehicle. The second term of the kinetic energy becomes zero as

$$\int_{Vol} \vec{v}_{B/N,N} \cdot \vec{v}_B \rho_V dV = \vec{v}_{B/N,N} \cdot \int_{Vol} \vec{v}_B \rho_V dV = 0 \quad (15.124)$$

For the third term of the kinetic energy, as the requirement for the origin of the mean-axis body frame be the instantaneous center of mass states that

$$\int_{Vol} \vec{x}_B \rho_V dV = 0 \quad (15.125)$$

and

$$\int_{Vol} \vec{v}_B \cdot (\vec{\omega}_{B/N} \times \vec{x}_B) \rho_V dV = \left( \int_{Vol} \vec{x}_B \times \vec{v}_B \rho_V dV \right) \cdot \vec{\omega}_{B/N} = 0 \quad (15.126)$$

one has

$$\int_{Vol} (\vec{v}_{B/N,N} + \vec{v}_B) \cdot (\vec{\omega}_{B/N} \times \vec{x}) \rho_V dV = \vec{v}_{B/N,N} \cdot \left( \vec{\omega}_{B/N} \times \int_{Vol} \vec{x} \rho_V dV \right) = 0 \quad (15.127)$$

The fourth term of the kinetic energy can be rewritten using the mode shapes and generalized coordinate summations for the displacement *velocity*, i.e.

$$\begin{aligned} \int_{Vol} \vec{v}_B \cdot \vec{v}_B \rho_V dV &= \frac{1}{2} \int_{Vol} \left( \sum_{i=1}^n \vec{v}_i(\vec{x}) \dot{\eta}_i \cdot \sum_{i=1}^n \vec{v}_i(\vec{x}) \dot{\eta}_i \right) \rho_V dV \\ &= \int_{Vol} \left( \sum_{i=1}^n \vec{v}_i(\vec{x}) \cdot \vec{v}_i(\vec{x}) \dot{\eta}_i^2 \right) \rho_V dV \\ &= \sum_{i=1}^n \mathcal{M}_i \dot{\eta}_i^2 \end{aligned} \quad (15.128)$$

where  $\mathcal{M}_i$  is the  $i$ -th generalized mass of the  $i$ -th vibration mode. This simplification is due to the mutual orthogonality of the vibration modes, i.e.

$$\int_{Vol} \vec{v}_i \cdot \vec{v}_j \rho_V dV = \begin{cases} 0 & i \neq j \\ \mathcal{M}_i & i = j \end{cases} \quad (15.129)$$

The fifth term of the kinetic energy can be rewritten in terms of the inertia matrix as

$$\int_{Vol} (\vec{\omega}_{B/N} \times \vec{x}) \cdot (\vec{\omega}_{B/N} \times \vec{x}) [\rho_V dV] = \vec{\omega}_{B/N}^T I_G \vec{\omega}_{B/N} \quad (15.130)$$

Thus, the kinetic energy can be rewritten as

$$T = \frac{1}{2} m \vec{v}_{B/N,N} \cdot \vec{v}_{B/N,N} + \frac{1}{2} \vec{\omega}_{B/N}^T I_G \vec{\omega}_{B/N} + \frac{1}{2} \sum_{i=1}^n M_i \dot{\eta}_i^2 \quad (15.131)$$

Next, applying the *mean-axis constraints* to the gravitational potential energy, one has

$$\begin{aligned} U_g &= - \int_{Vol} \vec{g} \cdot (\vec{x}_B + \vec{x}_{B/N}) \rho_V dV \\ U_g &= - \vec{g} \cdot \int_{Vol} \vec{x}_B \rho_V dV - \vec{g} \cdot \vec{x}_{B/N} \int_{Vol} \rho_V dV \\ U_g &= - \vec{g} \cdot \vec{x}_{B/N} m \end{aligned} \quad (15.132)$$

Finally, applying the *mean-axis constraints* to the elastic strain energy, one has

$$U_e = - \frac{1}{2} \int_{Vol} \sum_{i=1}^n \vec{v}_i(\vec{x}) \ddot{\eta}_i(t) \cdot \vec{v}_i(\vec{x}) \eta_i(t) \rho_V dV \quad (15.133)$$

and due to the orthogonality of the modes, one has

$$U_e = - \frac{1}{2} \int_{Vol} \sum_{i=1}^n \vec{v}_i(\vec{x}) \cdot \vec{v}_i(\vec{x}) \ddot{\eta}_i(t) \eta_i(t) \rho_V dV \quad (15.134)$$

Then, recalling the sinusoidal solution for the modal coordinates, i.e.

$$\eta_i(t) = A_i \cos(\omega_i t + \Gamma_i) \quad (15.135)$$

one has

$$U_e = - \frac{1}{2} \sum_{i=1}^n \omega_i^2 \eta_i^2(t) M_i \quad (15.136)$$

## Generalized Coordinate Selection

Before applying Lagrange's equation to the energies to obtain the elastic vehicle equations of motion, one must select suitable generalized coordinates. First, one can select the inertial position of the body frame's origin (i.e. the center of mass) as the coordinates

$$\vec{x}_{B/N} = \begin{bmatrix} x_{B/N} \\ y_{B/N} \\ z_{B/N} \end{bmatrix} \quad (15.137)$$

which was represented in rigid vehicle dynamics without the referencing slash which is appropriate for rigid bodies as the relative position of any point on the body can be represented with the location of the center of

mass and the attitude of the vehicle. Though for elastic vehicles this is no longer the case, the  $B/N$  subscript will be dropped to easily compare the elastic vehicle dynamics with the rigid body dynamics, i.e.

$$\vec{x}_{B/N,N} = \begin{bmatrix} x, N \\ y, N \\ z, N \end{bmatrix} \quad (15.138)$$

will be the selected generalized coordinates. Also recall that  $z_N = -h$  for the flat-Earth approximation. Thus, the velocity of the body frame's origin expressed in the navigation frame is

$$\vec{v}_{B/N,N} = \begin{bmatrix} \dot{x}_N \\ \dot{y}_N \\ \dot{z}_N \end{bmatrix} \quad (15.139)$$

and the body frame axes as

$$\vec{v}_{B/N,B} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (15.140)$$

Secondly, one can select the angular velocity of the body frame relative to the navigation frame as

$$\vec{\omega}_{B/N} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (15.141)$$

which can also be expressed in either the body frame or the navigation frame (with an appropriate subscript). The generalized coordinates for this derivation will use the body frame axes. Thirdly, one can select the 3-2-1 Euler angles  $\psi, \theta, \phi$  to define the orientation of the mean-axes body frame with respect to the navigation frame. Recall that the Euler angles are related to the angular velocity by the equations

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & -\sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (15.142)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (15.143)$$

Thus, one can select the generalized coordinates as

$$\vec{q} = [x \ y \ z \ \psi \ \theta \ \phi \ \eta_i, i = 1, 2, \dots]^T \quad (15.144)$$

which is the same as for rigid vehicle dynamics with the addition of the modal coordinates corresponding to the mutually orthogonal mode shapes.

Using these definitions, one can write the kinetic energy of the elastic vehicle as

$$T = \frac{1}{2}m [\dot{x}_N \ \dot{y}_N \ \dot{z}_N] \begin{bmatrix} \dot{x}_N \\ \dot{y}_N \\ \dot{z}_N \end{bmatrix} + \frac{1}{2} [p \ q \ r] I_G \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \frac{1}{2} \sum_{i=1}^n M_i \dot{\eta}_i^2 \quad (15.145)$$

the gravitational potential energy as

$$U_g = -mgz_N = mgh \quad (15.146)$$

and the elastic strain energy remains the same.

$$U_e = -\frac{1}{2} \sum_{i=1}^n \omega_i^2 \eta_i^2(t) \mathcal{M}_i \quad (15.147)$$

## 15.4 Introduction to Elastic Flight Vehicle Dynamics

Recall the following definitions. The generalized coordinates are

$$\vec{q} = [x \ y \ z \ \phi \ \theta \ \psi \ \eta_i, i = 1, \dots, n]^T \quad (15.148)$$

The kinetic energy of an elastic vehicle is

$$T = \frac{1}{2} m \begin{bmatrix} \dot{x}_N & \dot{y}_N & \dot{z}_N \end{bmatrix} \begin{bmatrix} \dot{x}_N \\ \dot{y}_N \\ \dot{z}_N \end{bmatrix} + \frac{1}{2} [p \ q \ r] I_G \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \frac{1}{2} \sum_{i=1}^n \mathcal{M}_i \dot{\eta}_i^2 \quad (15.149)$$

the gravitational potential energy of an elastic vehicle is

$$U_g = -mgz_N = mgh \quad (15.150)$$

and the elastic strain energy of an elastic vehicle is

$$U_e = -\frac{1}{2} \sum_{i=1}^n \omega_i^2 \eta_i^2(t) \mathcal{M}_i \quad (15.151)$$

which can be used with Lagrange's equation in vector form

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \vec{q}} \right) - \frac{\partial T}{\partial \vec{q}} + \frac{\partial U}{\partial \vec{q}} = \vec{Q}^T = \frac{\partial \delta W}{\partial \delta \vec{q}} \quad (15.152)$$

Using these, one can derive the equations of motion for elastic flight vehicles in three coupled equations of motion: the rigid body translation, the rigid body rotation, and the elastic vibrations.

### Translation Equations of Motion

For a constant-mass vehicle, consider the inertial center of mass coordinates,  $\vec{x}_N = [x_N \ y_N \ z_N]^T$ , chosen as the translation generalized coordinates, and applying Lagrange's equation for the translational kinetic energy, i.e.

$$T_{tran} = \frac{1}{2} m \begin{bmatrix} \dot{x}_N & \dot{y}_N & \dot{z}_N \end{bmatrix} \begin{bmatrix} \dot{x}_N \\ \dot{y}_N \\ \dot{z}_N \end{bmatrix} \quad (15.153)$$

one has

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{\vec{x}}_N} \right) - \frac{\partial T}{\partial \vec{x}_N} = m \begin{bmatrix} \ddot{x}_N \\ \ddot{y}_N \\ \ddot{z}_N \end{bmatrix} = m \ddot{\vec{x}}_N \quad (15.154)$$

and for the gravitational potential energy, one has

$$\frac{\partial U_g}{\partial \vec{x}_N} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \quad (15.155)$$

Including the generalized forces in the navigation frame  $\vec{Q}_N$ , one has the following translation EOMs

$$\ddot{\vec{x}}_N = \begin{bmatrix} \ddot{x}_N \\ \ddot{y}_N \\ \ddot{z}_N \end{bmatrix} = \begin{bmatrix} \frac{Q_{x,N}}{m} \\ \frac{Q_{y,N}}{m} \\ \frac{Q_{z,N}}{m} + g \end{bmatrix} \quad (15.156)$$

However, for flight dynamics, one typically desires to convert this to the body frame accelerations and velocities, i.e. recalling the conversion of the velocity by

$$\ddot{\vec{x}}_N = \ddot{\vec{x}}_B + \vec{\omega}_{B/N} \times \dot{\vec{x}}_B \quad (15.157)$$

one has using the definitions for the body frame linear and angular velocity components

$$\begin{aligned} \ddot{\vec{x}}_N &= \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} \\ &= \begin{bmatrix} \dot{u} - rv + qw \\ \dot{v} + ru - wp \\ \dot{w} - qu + pv \end{bmatrix} \end{aligned} \quad (15.158)$$

For the generalized forces for flight vehicle, recall that the net force (besides gravitational) are the net aerodynamic and propulsive which are defined in the body frame as

$$\vec{F}_{aero,B} + \vec{F}_{prop,B} = \begin{bmatrix} mX \\ mY \\ mZ \end{bmatrix} \quad (15.159)$$

The virtual work,  $\delta W_F$ , is done by these forces as virtual displacements in the body frame, i.e.  $\delta \vec{x}_B = [\delta x_B \ \delta y_B \ \delta z_B]^T$ , which can be written as

$$\delta W_F = [mX \ mY \ mZ] \begin{bmatrix} \delta x_B \\ \delta y_B \\ \delta z_B \end{bmatrix} \quad (15.160)$$

In terms of the selected generalized coordinates, one can use the DCM from  $N \rightarrow B$  which is a function of the Euler angles as follows:

$$\delta W_F = [mX \ mY \ mZ] C_{B \leftarrow N}(\phi, \theta, \psi) \begin{bmatrix} \delta x_N \\ \delta y_N \\ \delta z_N \end{bmatrix} \quad (15.161)$$

Thus the generalized forces become

$$\begin{aligned} \begin{bmatrix} Q_{x,N} \\ Q_{y,N} \\ Q_{z,N} \end{bmatrix} &= \frac{\partial \delta W_F}{\partial \delta \vec{x}_N} = [mX \ mY \ mZ] C_{B \leftarrow N}(\phi, \theta, \psi) \\ &= C_{B \leftarrow N}^T(\phi, \theta, \psi) \begin{bmatrix} mX \\ mY \\ mZ \end{bmatrix} \end{aligned} \quad (15.162)$$

which can be rewritten in body frame coordinates as simply

$$\begin{bmatrix} Q_{x,B} \\ Q_{y,B} \\ Q_{z,B} \end{bmatrix} = C_{B \leftarrow N}(\phi, \theta, \psi) C_{B \leftarrow N}^T(\phi, \theta, \psi) \begin{bmatrix} mX \\ mY \\ mZ \end{bmatrix} = \begin{bmatrix} mX \\ mY \\ mZ \end{bmatrix} \quad (15.163)$$

as expected. Lastly, for the gravitational force, one has in the body frame coordinates

$$\begin{bmatrix} -mg \sin \theta \\ mg \cos \theta \sin \phi \\ mg \cos \theta \cos \phi \end{bmatrix} = C_{B \leftarrow N}(\phi, \theta, \psi) \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (15.164)$$

Thus, one has the same rigid vehicle translation equations of motion for elastic vehicle translation equations of motion, i.e.

$$\begin{bmatrix} \dot{u} - rv + qw \\ \dot{v} + ru - wp \\ \dot{w} - qu + pv \end{bmatrix} = \begin{bmatrix} X - g \sin \theta \\ Y + g \cos \theta \sin \phi \\ Z + g \cos \theta \cos \phi \end{bmatrix} \quad (15.165)$$

and any elastic deformation effects will enter by the aerodynamic and propulsive forces.

## Rotation Equations of Motion

For a constant-mass vehicle, consider the Euler angles,  $\vec{q}_\perp = [\phi \ \theta \ \psi]^T$ , chosen as the rotation generalized coordinates. However as the Euler angles represent three sequential rotations, the relationship between the body frame angular velocity (which appears in the kinetic energy) and the Euler angles is

$$\vec{\omega}_{B/N} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = C_\omega(\vec{q}_\perp) \dot{\vec{q}}_\perp = \begin{bmatrix} 1 & 0 & -\sin \phi \\ 0 & \cos \phi & \cos \theta \sin \phi \\ 0 & -\sin \phi & \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (15.166)$$

Thus, one has

$$\frac{\partial \vec{\omega}_{B/N}}{\partial \dot{\vec{q}}_\perp} = C_\omega \quad (15.167)$$

Next, note that the rotational kinetic energy becomes

$$T_{rot} = \frac{1}{2} \vec{\omega}_{B/N}^T I_G \vec{\omega}_{B/N} \quad (15.168)$$

Thus, one has

$$\frac{\partial T}{\partial \vec{\omega}_{B/N}} = \vec{\omega}_{B/N}^T I_G \quad (15.169)$$

Furthermore, one can show

$$\frac{\partial \omega_{B/N}}{\partial \vec{q}_\perp} = \begin{bmatrix} 0 & -\dot{\psi} \cos \theta & 0 \\ \dot{\psi} \cos \theta \cos \phi - \dot{\theta} \sin \phi & -\dot{\psi} \sin \theta \sin \phi & 0 \\ -\dot{\psi} \cos \theta \sin \phi - \dot{\theta} \cos \phi & -\dot{\psi} \sin \theta \cos \phi & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\dot{\psi} \cos \theta & 0 \\ r & -\dot{\psi} \sin \theta \sin \phi & 0 \\ -q & -\dot{\psi} \sin \theta \cos \phi & 0 \end{bmatrix} \quad (15.170)$$

Then, applying Lagrange's equation for the kinetic energy, one has

$$\frac{d}{dt} \frac{\partial T}{\partial \dot{\vec{q}}_\perp} - \frac{\partial T}{\partial \vec{q}_\perp} = \frac{d}{dt} \left( \frac{\partial T}{\partial \omega_{B/N}} \frac{\partial \omega_{B/N}}{\partial \vec{q}_\perp} \right) - \frac{\partial T}{\partial \omega_{B/N}} \frac{\partial \vec{\omega}_{B/N}}{\partial \vec{q}_\perp} \quad (15.171)$$

which after the matrix multiplications and some algebra, it can be shown

$$\frac{d}{dt} \frac{\partial T}{\partial \dot{\vec{q}}_\perp} - \frac{\partial T}{\partial \vec{q}_\perp} = C_\omega^T (I_G \dot{\omega}_{B/N} + \vec{\omega}_{B/N} \times I_G \vec{\omega}_{B/N}) = \vec{Q}_\perp \quad (15.172)$$

where the virtual work associated with the net moment on the flight vehicle in the body frame can be related to the virtual angular displacements simply by

$$\delta W_M = [I_{xx}L \quad I_{yy}M \quad I_{zz}N] \begin{bmatrix} \delta\phi \\ \delta\theta \\ \delta\psi \end{bmatrix} \quad (15.173)$$

where the infinitesimal rotations due to the angular displacements are related to the Euler angles themselves, by

$$\begin{bmatrix} \delta\phi \\ \delta\theta \\ \delta\psi \end{bmatrix} = C_\omega \vec{q}_\perp \quad (15.174)$$

Thus, in terms of the generalized coordinates

$$\vec{Q}_\perp^T = \frac{\partial \delta W_M}{\partial \vec{q}_\perp} = [I_{xx}L \quad I_{yy}M \quad I_{zz}N] C_\omega \quad (15.175)$$

$$\vec{Q}_\perp = C_\omega^T \begin{bmatrix} I_{xx}L \\ I_{yy}M \\ I_{zz}N \end{bmatrix} \quad (15.176)$$

Thus, one has the same rigid vehicle rotation equations of motion for elastic vehicle rotation equations of motion, i.e.

$$I_G \dot{\omega}_{B/N} + \vec{\omega}_{B/N} \times I_G \vec{\omega}_{B/N} = \begin{bmatrix} I_{xx}L \\ I_{yy}M \\ I_{zz}N \end{bmatrix} \quad (15.177)$$

which for  $I_{xy} = I_{yz} = 0$  can be written out as

$$\begin{bmatrix} \dot{p} + \frac{I_{zz}-I_{yy}}{I_{xx}I_{zz}} qr - \frac{I_{xz}}{I_{xx}}(\dot{r} + pq) \\ \dot{q} + \frac{I_{xx}-I_{zz}}{I_{yy}} pr - \frac{I_{xz}}{I_{yy}}(r^2 - p^2) \\ \dot{r} + \frac{I_{yy}-I_{xx}}{I_{zz}} pq - \frac{I_{xz}}{I_{zz}}(\dot{p} - qr) \end{bmatrix} = \begin{bmatrix} L \\ M \\ N \end{bmatrix} \quad (15.178)$$

and any elastic deformation effects will enter by the aerodynamic and propulsive moments.

### Vibration Equations of Motion

For a constant-mass vehicle, consider the  $n$  vibration coordinates,  $\vec{\eta} = [\eta_1 \cdots \eta_n]^T$ , chosen as the vibration generalized coordinates. Applying Lagrange's equation for the vibration kinetic energy and elastic strain energy for each individual modal coordinate, one has

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{\eta}_i} \right) - \frac{\partial T}{\partial \eta_i} + \frac{\partial U_e}{\partial \eta_i} = Q_i = \frac{\partial \delta W}{\partial \delta \eta_i} \quad (15.179)$$

one has  $n$  equations of motion for the vibration coordinates of the form

$$\ddot{\eta}_i + \omega_i^2 \eta_i = \frac{Q_i}{M_i} \quad i = 1, \dots, n \quad (15.180)$$

where  $M$  is the generalized mass and  $Q_i$  is the generalized force, each associated with the  $i$ -th vibration mode.

For aerodynamics modeling for the generalized mass, let the external pressure distribution acting on the surface of the vehicle's structure at point  $\vec{x}_B$  be defined as  $\vec{P}(\vec{x}_B)$ . By construction, the integral of this pressure distribution will result in the aerodynamic and propulsive forces and moments  $X, Y, Z, L, M, N$ . Then, the local elastic deformation of the structure can be written in terms of the modes as

$$\delta d_e(\vec{x}_B) = \sum_{i=1}^n \vec{v}_i \delta \eta_i(t) \quad (15.181)$$

Thus, the virtual work done due to the pressure  $\vec{P}$  located at  $\vec{x}_B$  on the structure is

$$d\delta W_P = \vec{P}(\vec{x}_B) \cdot \sum_{i=1}^n \vec{v}_i(\vec{x}_B) \delta \eta_i(t) dS \quad (15.182)$$

where  $dS$  is the infinitesimal surface area over which the pressure applies. Thus, the total virtual work is

$$\begin{aligned} \delta W_P &= \int_{Area} \vec{P}(\vec{x}_B) \cdot \sum_{i=1}^n \vec{v}_i(\vec{x}_B) \delta \eta_i(t) dS \\ &= \sum_{i=1}^n \int_{Area} \vec{P}(\vec{x}_B) \cdot \vec{v}_i(\vec{x}_B) dS \delta \eta_i(t) \end{aligned} \quad (15.183)$$

Finally, the  $n$  vibration equations of motion become

$$\ddot{\eta}_i + \omega_i^2 \eta_i = \frac{1}{M_i} \int_{Area} \vec{P}(\vec{x}_B) \cdot \vec{v}_i(\vec{x}_B) dS \quad i = 1, \dots, n \quad (15.184)$$

which connects the aerodynamic pressure with the structural deformations.

### Point Motion on Elastic Vehicle

With these three sets of equations of motion, one is now able to describe the motion of any point on the elastic flight vehicle which is a combination of the rigid body motion *and* the vibration motion. Specifically, the position of any point on the elastic vehicle as defined in the navigation frame is a linear combination of the body frame's origin (i.e. the center of mass), the rigid body position of the point,  $\vec{x}_{rb}$ , and the elastic displacement,  $d_e$ , i.e.

$$\vec{x}_N = \vec{x}_{B/N} + \vec{x}_{rb} + d_e(\vec{x}_B, t) \quad (15.185)$$

or

$$\vec{x}_N = \vec{x}_{B/N} + \vec{x}_{rb} + \sum_{i=1}^n \vec{v}_i(\vec{x}_B) \eta_i(t) \quad (15.186)$$

where each of these terms can be determined by the elastic vehicle equations of motion (assuming one has a solution for the free-vibration problem for the mode shapes). Note that here the position of the instantaneous center of mass is governed by the rigid body translation equations of motion which typically uses the body frame coordinates for the velocity of the body frame, thus, one should recall

$$\dot{\vec{x}}_{B/N} = \begin{bmatrix} \dot{x}_N \\ \dot{y}_N \\ \dot{z}_N \end{bmatrix} = C_{N \leftarrow B} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (15.187)$$

based on the final equations given previously.

### Reference and Perturbation for Vibration

Lastly, as linearized flight dynamics models are typically used in analysis and control design, note that one can also use the reference and perturbation form for the modal coordinates as

$$\eta_i(t) = \bar{\eta}_i + \Delta\eta_i(t) \quad (15.188)$$

and for the pressure distribution as

$$\vec{P}(\vec{x}) = \bar{P}(\vec{x}) + \Delta\vec{P}(\vec{x}) \quad (15.189)$$

Thus, for the perturbation or linearized vibration equation of motion, one has

$$\Delta\ddot{\eta}_i + \omega_i^2 \Delta\eta_i = \frac{1}{M} \int_{Area} \Delta\vec{P}(\vec{x}_B) \cdot \vec{v}(\vec{x}_B) dS, \quad i = 1, \dots, n \quad (15.190)$$

## 15.5 Dynamic-Elastic Effects on Flight Vehicles

Previous sections have provided the background for developing the elastic vibration equations of motion (EOMs) alongside the rigid body EOMs. These showed that the vibration modes have their own set of equations of motion and enter the rigid body EOMs only through elastic effects on the aerodynamic and propulsive forces and moments. These considerations can be studied as **dynamic-elastic effects**, which models the elastic effects on these forces and moments in the rigid body EOMs and the vibration dynamics, or the **static-elastic effects** which *only* consider the elastic deformation effects on the forces and moments in the rigid body EOMs and not do include the vibration dynamics. This section will consider the dynamic-elastic effects modeling for both the nonlinear and linearized forms of airplane dynamics while the subsequent section will discuss the static-elastic effects modeling for airplane dynamics.

## Nonlinear Elastic Flight Vehicle Dynamics

To this end, recall the rigid airplane equations of motion

$$\begin{bmatrix} X - g \sin \theta \\ Y + g \cos \theta \sin \phi \\ Z + g \cos \theta \cos \phi \\ L \\ M \\ N \end{bmatrix} = \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \\ \dot{p} + \frac{I_{zz} - I_{yy}}{I_{xx}} qr - \frac{I_{xz}}{I_{xx}} (\dot{r} + pq) \\ \dot{q} + \frac{I_{xx} - I_{zz}}{I_{yy}} pr - \frac{I_{xz}}{I_{yy}} (r^2 - p^2) \\ \dot{r} + \frac{I_{yy} - I_{xx}}{I_{zz}} pq - \frac{I_{xz}}{I_{zz}} (\dot{p} - qr) \end{bmatrix} \quad (15.191)$$

where the body frame forces can alternatively be written using the thrust,  $\vec{T}$ , and the wind frame aerodynamic forces: lift  $L$ , side  $S$ , and drag  $D$ , as

$$\begin{aligned} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} &= \vec{T} + C_{B \leftarrow W}(\alpha, \beta) \begin{bmatrix} -D \\ S \\ -L \end{bmatrix} \\ \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} &= \vec{T} + \begin{bmatrix} \cos \alpha \cos \beta & -\cos \alpha \sin \beta & -\sin \alpha \\ \sin \beta & \cos \beta & 0 \\ \sin \alpha \cos \beta & -\sin \alpha \sin \beta & \cos \alpha \end{bmatrix} \begin{bmatrix} -D \\ S \\ -L \end{bmatrix} \end{aligned} \quad (15.192)$$

where the dynamic-elastic effects can be alternatively applied to  $L$ ,  $S$ , and  $D$  instead of  $X$ ,  $Y$ ,  $Z$ . In addition, recall that one can model these aerodynamic and propulsive forces using stability and control derivatives/coefficients (where  $\delta_t = T$ ). The same approach can be used to add coefficients for each modal coordinate and modal coordinate rate, i.e.

$$\begin{aligned} \begin{bmatrix} X \\ Z \\ M \end{bmatrix} &= \begin{bmatrix} X_0 \\ Z_0 \\ M_0 \end{bmatrix} + \begin{bmatrix} 0 & X_{\dot{\alpha}} & 0 \\ 0 & Z_{\dot{\alpha}} & 0 \\ 0 & M_{\dot{\alpha}} & 0 \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{a} \\ \dot{q} \end{bmatrix} + \begin{bmatrix} X_u & X_\alpha & X_q \\ Z_u & Z_\alpha & Z_q \\ M_u & M_\alpha & M_q \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ q \end{bmatrix} + \begin{bmatrix} X_{\delta_e} & X_{\delta_t} \\ Z_{\delta_e} & Z_{\delta_t} \\ M_{\delta_e} & M_{\delta_t} \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_t \end{bmatrix} \\ &\quad + \begin{bmatrix} X_{\dot{\eta}_1} & \cdots & X_{\dot{\eta}_1} \\ Z_{\dot{\eta}_1} & \cdots & Z_{\dot{\eta}_1} \\ M_{\dot{\eta}_1} & \cdots & M_{\dot{\eta}_1} \end{bmatrix} \begin{bmatrix} \dot{\eta}_1 \\ \vdots \\ \dot{\eta}_n \end{bmatrix} + \begin{bmatrix} X_{\eta_1} & \cdots & X_{\eta_1} \\ Z_{\eta_1} & \cdots & Z_{\eta_1} \\ M_{\eta_1} & \cdots & M_{\eta_1} \end{bmatrix} \begin{bmatrix} \eta_1 \\ \vdots \\ \eta_n \end{bmatrix} \end{aligned} \quad (15.193)$$

and

$$\begin{aligned} \begin{bmatrix} Y \\ L \\ N \end{bmatrix} &= \begin{bmatrix} Y_0 \\ L_0 \\ N_0 \end{bmatrix} + \begin{bmatrix} Y_\beta & Y_p & Y_r \\ L_\beta & L_p & L_r \\ N_\beta & N_p & N_r \end{bmatrix} \begin{bmatrix} \beta \\ p \\ r \end{bmatrix} + \begin{bmatrix} Y_{\delta_a} & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \\ &\quad + \begin{bmatrix} Y_{\dot{\eta}_1} & \cdots & Y_{\dot{\eta}_1} \\ L_{\dot{\eta}_1} & \cdots & L_{\dot{\eta}_1} \\ N_{\dot{\eta}_1} & \cdots & N_{\dot{\eta}_1} \end{bmatrix} \begin{bmatrix} \dot{\eta}_1 \\ \vdots \\ \dot{\eta}_n \end{bmatrix} + \begin{bmatrix} Y_{\eta_1} & \cdots & Y_{\eta_1} \\ L_{\eta_1} & \cdots & L_{\eta_1} \\ N_{\eta_1} & \cdots & N_{\eta_1} \end{bmatrix} \begin{bmatrix} \eta_1 \\ \vdots \\ \eta_n \end{bmatrix} \end{aligned} \quad (15.194)$$

Furthermore, the conversions from coefficient to derivative can be shown to be as follows. Recall that  $Q_\infty = \frac{1}{2}\rho_\infty \bar{v}_a^2$  is the dynamic pressure with trimmed airspeed  $\bar{v}_a = \sqrt{u^2 + v^2 + w^2}$ .

$\bullet$	$X_\bullet$	$Z_\bullet$	$M_\bullet$
$u$	$\frac{Q_\infty S_w}{m \bar{v}_a} C_{X_u}$	$\frac{Q_\infty S_w}{m \bar{v}_a} C_{Z_u}$	$\frac{Q_\infty S_w \bar{c}_w}{I_{yy} \bar{v}_a} C_{m_u}$
$\alpha$	$\frac{Q_\infty S_w}{m} C_{X_\alpha}$	$\frac{Q_\infty S_w}{m} C_{Z_\alpha}$	$\frac{Q_\infty S_w \bar{c}_w}{I_{yy}} C_{m_\alpha}$
$q$	$\frac{Q_\infty S_w \bar{c}_w}{2m \bar{v}_a} C_{Z_q}$	$\frac{Q_\infty S_w \bar{c}_w}{2m \bar{v}_a} C_{Z_q}$	$\frac{Q_\infty S_w \bar{c}_w^2}{2I_{yy} \bar{v}_a} C_{m_q}$
$\dot{\alpha}$	$\frac{Q_\infty S_w \bar{c}_w}{2m \bar{v}_a} C_{X_{\dot{\alpha}}}$	$\frac{Q_\infty S_w \bar{c}_w}{2m \bar{v}_a} C_{Z_{\dot{\alpha}}}$	$\frac{Q_\infty S_w \bar{c}_w^2}{2I_{yy} \bar{v}_a} C_{m_{\dot{\alpha}}}$
$\delta_e$	$\frac{Q_\infty S_w}{m} C_{X_{\delta_e}}$	$\frac{Q_\infty S_w}{m} C_{Z_{\delta_e}}$	$\frac{Q_\infty S_w \bar{c}_w}{I_{yy}} C_{m_{\delta_e}}$
$\delta_t$	$\frac{Q_\infty S_w}{m} C_{X_{\delta_t}}$	$\frac{Q_\infty S_w}{m} C_{Z_{\delta_t}}$	$\frac{Q_\infty S_w \bar{c}_w}{I_{yy}} C_{m_{\delta_t}}$
$\eta_i$	$\frac{Q_\infty S_w}{m} C_{X_{\eta_i}}$	$\frac{Q_\infty S_w}{m} C_{Z_{\eta_i}}$	$\frac{Q_\infty S_w \bar{c}_w}{I_{yy}} C_{m_{\eta_i}}$
$\dot{\eta}_i$	$\frac{Q_\infty S_w}{m \bar{v}_a} C_{X_{\dot{\eta}_i}}$	$\frac{Q_\infty S_w}{m \bar{v}_a} C_{Z_{\dot{\eta}_i}}$	$\frac{Q_\infty S_w \bar{c}_w}{I_{yy} \bar{v}_a} C_{m_{\dot{\eta}_i}}$

$\bullet$	$Y_\bullet$	$L_\bullet$	$N_\bullet$
$\beta$	$\frac{Q_\infty S_w}{m} C_{Y_\beta}$	$\frac{Q_\infty S_w b_w}{I_{xx}} C_{l_\beta}$	$\frac{Q_\infty S_w b_w}{I_{zz}} C_{n_\beta}$
$p$	$\frac{Q_\infty S_w b_w}{2m \bar{v}_a} C_{Y_p}$	$\frac{Q_\infty S_w b_w^2}{2I_{xx} \bar{v}_a} C_{l_p}$	$\frac{Q_\infty S_w b_w^2}{2I_{zz} \bar{v}_a} C_{n_p}$
$r$	$\frac{Q_\infty S_w b_w}{2m \bar{v}_a} C_{Y_r}$	$\frac{Q_\infty S_w b_w^2}{2I_{xx} \bar{v}_a} C_{l_r}$	$\frac{Q_\infty S_w b_w^2}{2I_{zz} \bar{v}_a} C_{n_r}$
$\delta_a$	$\frac{Q_\infty S_w}{m} C_{Y_{\delta_a}}$	$\frac{Q_\infty S_w b_w}{I_{xx}} C_{l_{\delta_a}}$	$\frac{Q_\infty S_w b_w}{I_{zz}} C_{n_{\delta_a}}$
$\delta_r$	$\frac{Q_\infty S_w}{m} C_{Y_{\delta_r}}$	$\frac{Q_\infty S_w b_w}{I_{xx}} C_{l_{\delta_r}}$	$\frac{Q_\infty S_w b_w}{I_{zz}} C_{n_{\delta_r}}$
$\eta_i$	$\frac{Q_\infty S_w}{m} C_{Y_{\eta_i}}$	$\frac{Q_\infty S_w b_w}{I_{xx}} C_{l_{\eta_i}}$	$\frac{Q_\infty S_w b_w}{I_{zz}} C_{n_{\eta_i}}$
$\dot{\eta}_i$	$\frac{Q_\infty S_w}{m \bar{v}_a} C_{Y_{\dot{\eta}_i}}$	$\frac{Q_\infty S_w b_w}{I_{xx} \bar{v}_a} C_{n_{\dot{\eta}_i}}$	$\frac{Q_\infty S_w b_w}{I_{zz} \bar{v}_a} C_{n_{\dot{\eta}_i}}$

Lastly, one must also include the  $n$  vibration LTI ODEs where typically one includes some level of damping for each mode,  $\zeta_i$ , whose value is typically assessed from matching analytical modeling with experimental data (usually quite low, e.g. 0.02). Thus, one has

$$\ddot{\eta}_i + 2\zeta_i \omega_i \dot{\eta}_i + \omega_i^2 \eta_i = \frac{Q_i}{M_i}, \quad i = 1, \dots, n \quad (15.195)$$

where the generalized forces can be modeled as linear equations of the states, i.e.

$$Q_i = Q_{i_0} + [Q_{i_u} \quad Q_{i_\beta} \quad Q_{i_\alpha} \quad Q_{i_p} \quad Q_{i_q} \quad Q_{i_r}] \begin{bmatrix} u \\ \beta \\ \alpha \\ p \\ q \\ r \end{bmatrix} + [Q_{i_{\delta_a}} \quad Q_{i_{\delta_e}} \quad Q_{i_{\delta_r}} \quad Q_{i_{\delta_t}}] \begin{bmatrix} \delta_a \\ \delta_e \\ \delta_r \\ \delta_t \end{bmatrix} + [Q_{i_{\dot{\eta}_1}} \quad \cdots \quad Q_{i_{\dot{\eta}_n}}] \begin{bmatrix} \dot{\eta}_1 \\ \vdots \\ \dot{\eta}_n \end{bmatrix} + [Q_{i_{\eta_1}} \quad \cdots \quad Q_{i_{\eta_n}}] \begin{bmatrix} \eta_1 \\ \vdots \\ \eta_n \end{bmatrix} \quad (15.196)$$

where

$$Q_{i_\bullet} = Q_\infty S_w \bar{c}_w C_{Q_{i_\bullet}} \quad (15.197)$$

for  $\bullet = \alpha, \beta, \delta_a, \delta_e, \delta_r, \delta_t, \eta_j$  for  $j = 1, \dots, n$ , and

$$Q_{i_\bullet} = \frac{Q_\infty S_w \bar{c}_w}{\bar{v}_a} C_{Q_{i_\bullet}} \quad (15.198)$$

for  $\bullet = u, p, q, r, \dot{\eta}_j$  for  $j = 1, \dots, n$ .

### Linearized Elastic Flight Vehicle Dynamics

As opposed to rigid body modeling, the linearized equations of motion for elastic flight vehicles typically use the fuselage body frame (subscript  $F$ ) instead of the stability body frame (subscript  $S$ ) for developing the vibration and dynamic-elastic coefficients. Thus, if one has developed the linearized rigid flight vehicle in the stability frame, one must first transform the perturbed rigid body aerodynamic and propulsive forces and moments from the stability frame to the fuselage frame as

$$\begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix}_F = \begin{bmatrix} \cos \bar{\alpha} & 0 & -\sin \bar{\alpha} \\ 0 & 1 & 0 \\ \sin \bar{\alpha} & 0 & \cos \bar{\alpha} \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix}_S \quad (15.199)$$

and

$$\begin{bmatrix} \Delta L \\ \Delta M \\ \Delta N \end{bmatrix}_F = \begin{bmatrix} \cos \bar{\alpha} & 0 & -\sin \bar{\alpha} \\ 0 & 1 & 0 \\ \sin \bar{\alpha} & 0 & \cos \bar{\alpha} \end{bmatrix} \begin{bmatrix} \Delta L \\ \Delta M \\ \Delta N \end{bmatrix}_S \quad (15.200)$$

Having redefined these terms, one may use the linearized equations of motion in the fuselage frame as opposed to the stability frame. However, in this case  $\bar{\alpha}$  may not equal 0, thus, the linearized equations become

$$\begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix} + g \begin{bmatrix} -\cos \bar{\theta} & 0 \\ -\sin \bar{\theta} \sin \bar{\phi} & \cos \bar{\theta} \cos \bar{\phi} \\ \sin \bar{\theta} \cos \bar{\phi} & \cos \bar{\theta} \sin \bar{\phi} \end{bmatrix} \begin{bmatrix} \theta \\ \phi \\ \psi \end{bmatrix} = \begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{v} \\ \Delta \dot{w} \end{bmatrix} \\ + \begin{bmatrix} 0 & -\bar{r} & \bar{q} \\ \bar{r} & 0 & -\bar{p} \\ -\bar{q} & \bar{p} & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta v \\ \Delta w \end{bmatrix} + \begin{bmatrix} 0 & \bar{w} & -\bar{v} \\ -\bar{w} & 0 & \bar{u} \\ \bar{v} & -\bar{u} & 0 \end{bmatrix} \begin{bmatrix} \Delta p \\ \Delta q \\ \Delta r \end{bmatrix} \quad (15.201)$$

and

$$\begin{bmatrix} \Delta L \\ \Delta M \\ \Delta N \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\frac{I_{xz}}{I_{xx}} \\ 0 & 1 & 0 \\ -\frac{I_{xz}}{I_{zz}} & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta \dot{p} \\ \Delta \dot{q} \\ \Delta \dot{r} \end{bmatrix} \\ + \begin{bmatrix} -\frac{I_{xz}}{I_{xx}} \bar{q} & -\frac{I_{xz}}{I_{xx}} \bar{p} + \frac{I_{zz}-I_{yy}}{I_{xx}} \bar{r} & \frac{I_{zz}-I_{yy}}{I_{xx}} \bar{q} \\ \frac{I_{xx}-I_{zz}}{I_{yy}} \bar{r} + 2 \frac{I_{xz}}{I_{yy}} \bar{p} & 1 & \frac{I_{xx}-I_{zz}}{I_{yy}} \bar{p} - 2 \frac{I_{xz}}{I_{yy}} \bar{r} \\ \frac{I_{yy}-I_{xx}}{I_{zz}} \bar{q} & \frac{I_{xz}}{I_{zz}} \bar{r} + \frac{I_{yy}-I_{xx}}{I_{zz}} \bar{p} & \frac{I_{xz}}{I_{zz}} \bar{q} \end{bmatrix} \begin{bmatrix} \Delta p \\ \Delta q \\ \Delta r \end{bmatrix} \quad (15.202)$$

where the perturbed forces and moments can be modeled as

$$\begin{bmatrix} \Delta X \\ \Delta Z \\ \Delta M \end{bmatrix} = \begin{bmatrix} 0 & X_{\dot{\alpha}} & 0 \\ 0 & Z_{\dot{\alpha}} & 0 \\ 0 & M_{\dot{\alpha}} & 0 \end{bmatrix} \begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{\alpha} \\ \Delta \dot{q} \end{bmatrix} + \begin{bmatrix} X_u & X_{\alpha} & X_q \\ Z_u & Z_{\alpha} & Z_q \\ M_u & M_{\alpha} & M_q \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta q \end{bmatrix} + \begin{bmatrix} X_{\delta_e} & X_{\delta_t} \\ Z_{\delta_e} & Z_{\delta_t} \\ M_{\delta_e} & M_{\delta_t} \end{bmatrix} \begin{bmatrix} \Delta \delta_e \\ \Delta \delta_t \end{bmatrix} \\ + \begin{bmatrix} X_{\dot{\eta}_1} & \dots & X_{\dot{\eta}_1} \\ Z_{\dot{\eta}_1} & \dots & Z_{\dot{\eta}_1} \\ M_{\dot{\eta}_1} & \dots & M_{\dot{\eta}_1} \end{bmatrix} \begin{bmatrix} \Delta \dot{\eta}_1 \\ \vdots \\ \Delta \dot{\eta}_n \end{bmatrix} + \begin{bmatrix} X_{\eta_1} & \dots & X_{\eta_1} \\ Z_{\eta_1} & \dots & Z_{\eta_1} \\ M_{\eta_1} & \dots & M_{\eta_1} \end{bmatrix} \begin{bmatrix} \Delta \eta_1 \\ \vdots \\ \Delta \eta_n \end{bmatrix} \quad (15.203)$$

and

$$\begin{bmatrix} \Delta Y \\ \Delta L \\ \Delta N \end{bmatrix} = \begin{bmatrix} Y_{\beta} & Y_p & Y_r \\ L_{\beta} & L_p & L_r \\ N_{\beta} & N_p & N_r \end{bmatrix} \begin{bmatrix} \Delta \beta \\ \Delta p \\ \Delta r \end{bmatrix} + \begin{bmatrix} Y_{\delta_a} & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{bmatrix} \begin{bmatrix} \Delta \delta_a \\ \Delta \delta_r \end{bmatrix} \\ + \begin{bmatrix} Y_{\dot{\eta}_1} & \dots & Y_{\dot{\eta}_1} \\ L_{\dot{\eta}_1} & \dots & L_{\dot{\eta}_1} \\ N_{\dot{\eta}_1} & \dots & N_{\dot{\eta}_1} \end{bmatrix} \begin{bmatrix} \Delta \dot{\eta}_1 \\ \vdots \\ \Delta \dot{\eta}_n \end{bmatrix} + \begin{bmatrix} Y_{\eta_1} & \dots & Y_{\eta_1} \\ L_{\eta_1} & \dots & L_{\eta_1} \\ N_{\eta_1} & \dots & N_{\eta_1} \end{bmatrix} \begin{bmatrix} \Delta \eta_1 \\ \vdots \\ \Delta \eta_n \end{bmatrix} \quad (15.204)$$

Note that alternatively one may also use the angle of attack and sideslip angle to make the substitutions

$$\Delta w = \bar{u} \Delta \alpha \quad (15.205)$$

and

$$\Delta v = \bar{v}_a \Delta \beta \quad (15.206)$$

assuming no wind speed. If wind is modeled, then one would require using the wind triangle to compute these relationships which would not provide simple substitutions, but require that one also knows or can estimate the wind speed. This will be addressed later in linear and nonlinear simulations. Furthermore, if  $\bar{v} = \bar{\phi} = \bar{p} = \bar{q} = \bar{r} = 0$ , then one can decouple the dynamics into the longitudinal and lateral-directional.

Lastly, one must also include the linearized vibration equations. However, as these are already linearly modeled, one can simply write

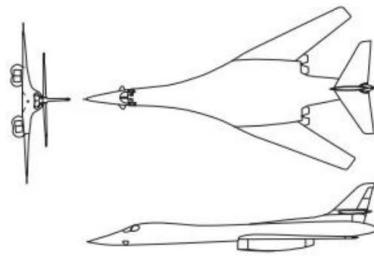
$$\Delta\ddot{\eta}_i + 2\zeta_i\omega_i\Delta\dot{\eta}_i + \omega_i^2\Delta\eta_i = \frac{\Delta Q_i}{M_i}, \quad i = 1, \dots, n \quad (15.207)$$

where the generalized forces can be modeled as linear equations of the states, i.e.

$$\begin{aligned} \Delta Q_i = & [Q_{i_u} \quad Q_{i_\beta} \quad Q_{i_\alpha} \quad Q_{i_p} \quad Q_{i_q} \quad Q_{i_r}] \begin{bmatrix} \Delta u \\ \Delta \beta \\ \Delta \alpha \\ \Delta p \\ \Delta q \\ \Delta r \end{bmatrix} + [Q_{i_{\delta_a}} \quad Q_{i_{\delta_e}} \quad Q_{i_{\delta_r}} \quad Q_{i_{\delta_t}}] \begin{bmatrix} \Delta \delta_a \\ \Delta \delta_e \\ \Delta \delta_r \\ \Delta \delta_t \end{bmatrix} \\ & + [Q_{i_{\dot{\eta}_1}} \quad \cdots \quad Q_{i_{\dot{\eta}_n}}] \begin{bmatrix} \Delta \dot{\eta}_1 \\ \vdots \\ \Delta \dot{\eta}_n \end{bmatrix} + [Q_{i_{\eta_1}} \quad \cdots \quad Q_{i_{\eta_n}}] \begin{bmatrix} \Delta \eta_1 \\ \vdots \\ \Delta \eta_n \end{bmatrix} \end{aligned} \quad (15.208)$$

### Case Study of Large High-Speed Airplane

As an example of an elastic flight vehicle model, consider the following large, high-speed airplane

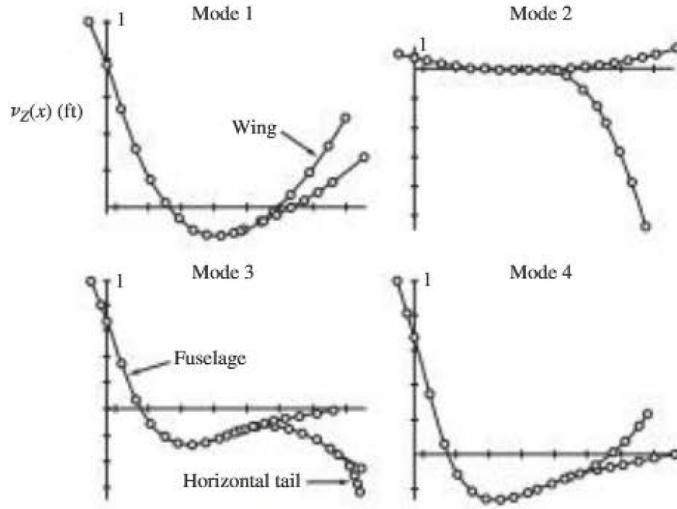


which has the following geometric, mass, and vibration data

<b>Vehicle Length</b>	$l_f$	=	143 ft	<b>Weight</b>	$W$	=	288,000 lb at MSL
<b>Wing Geometry</b>	$S_w$	=	1,950 ft <sup>2</sup>	<b>Inertias</b>	$I_{xx}$	=	$9.5 \times 10^5$ sl-ft <sup>2</sup>
	$\bar{c}_w$	=	15.3 ft		$I_{yy}$	=	$6.4 \times 10^6$ sl-ft <sup>2</sup>
	$b_w$	=	70 ft		$I_{zz}$	=	$7.1 \times 10^6$ sl-ft <sup>2</sup>
	$\Lambda_w$	=	65°		$I_{xz}$	=	$-5.27 \times 10^4$ sl-ft <sup>2</sup>
<b>Modal Generalized</b>	$M_1$	=	184 sl-ft <sup>3</sup>	<b>Modal Frequencies</b>	$\omega_1$	=	12.6 rad/sec
	$M_2$	=	9,587 sl-ft <sup>3</sup>		$\omega_2$	=	14.1 rad/sec

<b>Masses</b>	$M_3 = 1,334 \text{ sl-ft}^3$	$\omega_3 = 21.2 \text{ rad/sec}$
	$M_4 = 436,000 \text{ sl-ft}^3$	$\omega_4 = 22.1 \text{ rad/sec}$

and the following mode shapes have been identified (normalized to 1 ft at the nose of the airplane)



where mode 1 can be understood as the *first fuselage bending mode*, while mode 2 can be understood as the *first wing bending mode*.

Furthermore, the following elastic coefficients on the lift force, pitching moment, and generalized forces has been either analytically derived, obtained through CFD and/or flight testing.

Coefficient	Mode 1	Mode 2	Mode 3	Mode 4
$C_{L\eta_i}$	0.029	-0.306	-0.015	0.014
$C_{L\dot{\eta}_i}$	0.658	-7.896	-0.461	0.132
$C_{m\eta_i}$	-0.032	-0.025	-0.041	-0.018
$C_{m\dot{\eta}_i}$	-1.184	9.409	1.316	-0.395
$C_{Q_{i0}}$	0	0	0	0
$C_{Q_{i\alpha}}$	$-1.49 \times 10^{-2}$	$2.58 \times 10^{-2}$	$1.49 \times 10^{-2}$	$3.35 \times 10^{-5}$
$C_{Q_{iq}}$	-0.726	0.089	0.304	$\approx 0$
$C_{Q_{i\delta_e}}$	$-1.28 \times 10^{-2}$	$-6.42 \times 10^{-2}$	$2.56 \times 10^{-4}$	$1.50 \times 10^{-4}$
$C_{Q_{in_1}}$	$5.85 \times 10^{-5}$	$4.21 \times 10^{-3}$	$2.91 \times 10^{-3}$	$2.21 \times 10^{-5}$
$C_{Q_{in_2}}$	$-9.0 \times 10^{-5}$	$-9.22 \times 10^{-2}$	$1.44 \times 10^{-4}$	$-1.32 \times 10^{-4}$
$C_{Q_{in_3}}$	$3.55 \times 10^{-4}$	$1.97 \times 10^{-3}$	$-3.46 \times 10^{-4}$	$9.68 \times 10^{-6}$
$C_{Q_{in_4}}$	$1.20 \times 10^{-4}$	$3.37 \times 10^{-3}$	$1.44 \times 10^{-4}$	$1.77 \times 10^{-3}$
$C_{Q_{i\dot{\eta}_1}}$	-0.0032	0.0665	-0.0048	-0.0004
$C_{Q_{i\dot{\eta}_2}}$	-0.0015	-2.277	0.1494	0.0031
$C_{Q_{i\dot{\eta}_3}}$	0.0050	0.0320	-0.0001	-0.0004

$C_{Q_{i\dot{\eta}_4}}$	-0.0011	0.0317	-0.0100	0.6112
-------------------------	---------	--------	---------	--------

## 15.6 Advanced Elastic Flight Vehicle Dynamics

Recall the elastic flight vehicle equations of motion

$$\begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \\ \dot{p} + \frac{I_{zz}-I_{yy}}{I_{xx}}qr - \frac{I_{xz}}{I_{xx}}(\dot{r} + pq) \\ \dot{q} + \frac{I_{xx}-I_{zz}}{I_{yy}}pr - \frac{I_{xz}}{I_{yy}}(r^2 - p^2) \\ \dot{r} + \frac{I_{yy}-I_{xx}}{I_{zz}}pq - \frac{I_{xz}}{I_{zz}}(\dot{p} - qr) \end{bmatrix} = \begin{bmatrix} X - g \sin \theta \\ Y + g \cos \theta \sin \phi \\ Z + g \cos \theta \cos \phi \\ L \\ M \\ N \end{bmatrix} \quad (15.209)$$

$$\ddot{\eta}_i + 2\zeta_i \omega_i \dot{\eta}_i + \omega_i^2 \eta_i = \frac{Q_i}{M_i}, \quad i = 1, \dots, n$$

And defining the linear and angular velocity of the body frame as the rigid state vector

$$\vec{x}_{rig} = [u \ v \ w \ p \ q \ r]^T \quad (15.210)$$

the modal coordinates and modal coordinate rates as the vibration state vector

$$\vec{x}_{vib} = [\eta_1 \ \dots \ \eta_n \ \dot{\eta}_1 \ \dots \ \dot{\eta}_n]^T \quad (15.211)$$

and control surface deflections and thrust input as the control input vector

$$\vec{u} = \begin{bmatrix} \delta_a \\ \delta_e \\ \delta_r \\ \delta_t \end{bmatrix} \quad (15.212)$$

With these definitions, the nonlinear equations of motion for an elastic flight vehicle can be rewritten as

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -\frac{I_{xz}}{I_{zz}} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\frac{I_{xz}}{I_{zz}} & 0 & 1 \end{bmatrix} \dot{\vec{x}}_{rig} = \begin{bmatrix} rv - qw - g \sin \theta \\ pw - ru + g \cos \theta \sin \phi \\ qu - pv + g \cos \theta \cos \phi \\ \frac{I_{yy}-I_{zz}}{I_{xx}}qr + \frac{I_{xz}}{I_{xx}}pq \\ \frac{I_{zz}-I_{xx}}{I_{yy}}pr + \frac{I_{xz}}{I_{yy}}(r^2 - p^2) \\ \frac{I_{yy}-I_{yy}}{I_{zz}}pq - \frac{I_{xz}}{I_{zz}}qr \end{bmatrix} + \begin{bmatrix} X \\ Y \\ Z \\ L \\ M \\ N \end{bmatrix} \quad (15.213)$$

$$\dot{\vec{x}}_{vib} = \begin{bmatrix} 0_{n \times n} & I_{n \times n} \\ -\Omega^2 & -2\Omega\zeta \end{bmatrix} \vec{x}_{vib} + \begin{bmatrix} \vec{0}_n \\ \frac{Q_1}{M_1} \\ \vdots \\ \frac{Q_n}{M_n} \end{bmatrix}$$

where

$$\Omega^2 = \begin{bmatrix} \omega_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \omega_n^2 \end{bmatrix} \quad (15.214)$$

and

$$\Omega_\zeta = \begin{bmatrix} \zeta_1 \omega_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \zeta_n \omega_n \end{bmatrix} \quad (15.215)$$

Then, recalling that the aerodynamic forces and moments and the generalized forces can both be modeled as linear functions of the rigid states, vibration states, and control inputs, one can define the following terms:

$$\mathcal{I} = \begin{bmatrix} 1 & 0 & -X_{\dot{w}} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 - Z_{\dot{w}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -\frac{I_{xz}}{I_{zz}} \\ 0 & 0 & -M_{\dot{w}} & 0 & 1 & 0 \\ 0 & 0 & 0 & -\frac{I_{xz}}{I_{zz}} & 0 & 1 \end{bmatrix} \quad (15.216)$$

(note if  $I_{xz} = 0$  and one ignores  $\dot{w}$  effects,  $\mathcal{I}$  will be a  $6 \times 6$  identity matrix),

$$\mathcal{M} = \begin{bmatrix} \mathcal{M}_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathcal{M}_n \end{bmatrix} \quad (15.217)$$

$$f_{rig}(\vec{x}_{rig}, \theta, \phi) = \mathcal{I}^{-1} \begin{bmatrix} rv - qw - g \sin \theta + X_0 \\ pw - ru + g \cos \theta \sin \phi + Y_0 \\ qu - pv + g \cos \theta \cos \phi + Z_0 \\ \frac{I_{xx} - I_{zz}}{I_{xx}} qr + \frac{I_{xz}}{I_{xx}} pq + L_0 \\ \frac{I_{zz} - I_{xx}}{I_{yy}} pr + \frac{I_{xz}}{I_{yy}} (r^2 - p^2) + M_0 \\ \frac{I_{xx} - I_{yy}}{I_{zz}} pq - \frac{I_{xz}}{I_{zz}} qr + N_0 \end{bmatrix} \quad (15.218)$$

$$\mathcal{A}_{rig \leftarrow rig} = \mathcal{I}^{-1} \begin{bmatrix} X_u & 0 & X_w & 0 & X_q & 0 \\ 0 & Y_v & 0 & Y_p & 0 & Y_r \\ Z_u & 0 & Z_w & 0 & Z_q & 0 \\ 0 & L_v & 0 & L_p & 0 & L_r \\ M_u & 0 & M_w & 0 & M_q & 0 \\ 0 & N_v & 0 & N_p & 0 & N_r \end{bmatrix} \quad (15.219)$$

$$\mathcal{A}_{rig \leftarrow \eta} = \mathcal{I}^{-1} \begin{bmatrix} X_{\eta_1} & \cdots & X_{\eta_n} \\ Y_{\eta_1} & \cdots & Y_{\eta_n} \\ Z_{\eta_1} & \cdots & Z_{\eta_n} \\ L_{\eta_1} & \cdots & L_{\eta_n} \\ M_{\eta_1} & \cdots & M_{\eta_n} \\ N_{\eta_1} & \cdots & N_{\eta_n} \end{bmatrix} \quad (15.220)$$

$$\mathcal{A}_{rig \leftarrow \dot{\eta}} = \mathcal{I}^{-1} \begin{bmatrix} X_{\dot{\eta}_1} & \cdots & X_{\dot{\eta}_n} \\ Y_{\dot{\eta}_1} & \cdots & Y_{\dot{\eta}_n} \\ Z_{\dot{\eta}_1} & \cdots & Z_{\dot{\eta}_n} \\ L_{\dot{\eta}_1} & \cdots & L_{\dot{\eta}_n} \\ M_{\dot{\eta}_1} & \cdots & M_{\dot{\eta}_n} \\ N_{\dot{\eta}_1} & \cdots & N_{\dot{\eta}_n} \end{bmatrix} \quad (15.221)$$

$$B_{rig} = \mathcal{I}^{-1} \begin{bmatrix} 0 & X_{\delta_e} & 0 & X_{\delta_t} \\ 0 & 0 & Y_{\delta_r} & 0 \\ 0 & Z_{\delta_e} & 0 & Z_{\delta_t} \\ L_{\delta_a} & 0 & L_{\delta_r} & 0 \\ 0 & M_{\delta_e} & 0 & M_{\delta_t} \\ N_{\delta_a} & 0 & N_{\delta_r} & 0 \end{bmatrix} \quad (15.222)$$

$$A_{vib \leftarrow rig} = \mathcal{M}^{-1} \begin{bmatrix} Q_{1_u} & Q_{1_v} & Q_{1_w} & Q_{1_p} & Q_{1_q} & Q_{1_r} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ Q_{n_u} & Q_{n_v} & Q_{n_w} & Q_{n_p} & Q_{n_q} & Q_{n_r} \end{bmatrix} \quad (15.223)$$

$$A_{vib \leftarrow \eta} = \mathcal{M}^{-1} \begin{bmatrix} Q_{1_{\eta_1}} & \cdots & Q_{1_{\eta_n}} \\ \vdots & \ddots & \vdots \\ Q_{n_{\eta_1}} & \cdots & Q_{n_{\eta_n}} \end{bmatrix} - \Omega^2 \quad (15.224)$$

$$A_{vib \leftarrow \dot{\eta}} = \mathcal{M}^{-1} \begin{bmatrix} Q_{1_{\dot{\eta}_1}} & \cdots & Q_{1_{\dot{\eta}_n}} \\ \vdots & \ddots & \vdots \\ Q_{n_{\dot{\eta}_1}} & \cdots & Q_{n_{\dot{\eta}_n}} \end{bmatrix} - 2\Omega_\zeta \quad (15.225)$$

and

$$B_{vib} = \mathcal{M}^{-1} \begin{bmatrix} Q_{1_{\delta_a}} & Q_{1_{\delta_e}} & Q_{1_{\delta_r}} & Q_{1_{\delta_t}} \\ \vdots & \vdots & \vdots & \vdots \\ Q_{n_{\delta_a}} & Q_{n_{\delta_e}} & Q_{n_{\delta_r}} & Q_{n_{\delta_t}} \end{bmatrix} \quad (15.226)$$

With these definitions, one may finally rewrite the elastic flight vehicle equations of motion in state-space form as

$$\begin{aligned} \dot{\vec{x}}_{rig} &= f_{rig}(\vec{x}_{rig}, \phi, \theta) + \mathcal{A}_{rig \leftarrow rig} \vec{x}_{rig} + [\mathcal{A}_{rig \leftarrow \eta} \quad \mathcal{A}_{rig \leftarrow \dot{\eta}}] \vec{x}_{vib} + B_{rig} \vec{u} \\ \dot{\vec{x}}_{vib} &= \begin{bmatrix} 0_{n \times 6} \\ A_{vib \leftarrow rig} \end{bmatrix} \vec{x}_{rig} + \begin{bmatrix} 0_{n \times n} & I_{n \times n} \\ A_{vib \leftarrow \eta} & A_{vib \leftarrow \dot{\eta}} \end{bmatrix} \vec{x}_{vib} + \begin{bmatrix} 0_{n \times 4} \\ B_{vib} \end{bmatrix} \vec{u} \end{aligned} \quad (15.227)$$

and it should be noted that here,  $v$ ,  $w$ , and  $\dot{w}$  have been used in place of  $\beta$ ,  $\alpha$ , and  $\dot{\alpha}$ , respectively, though these could be replaced by the linear approximations and coefficient conversions if required. Note also that one would also require the supplemental Euler angle equation to relate  $p$ ,  $q$ , and  $r$  to  $\dot{\phi}$  and  $\dot{\theta}$  to complete the state-space formulation, i.e.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (15.228)$$

### Static-Elastic Effects on Aerodynamics

To consider only the static-elastic effects, one can set all  $\dot{\eta}_i = \ddot{\eta}_i = 0 \forall i$ . Then, one can solve for the **static-elastic modal coordinates**,

$$\bar{\eta} = [\bar{\eta}_1 \quad \dots \quad \bar{\eta}_n] \quad (15.229)$$

in terms of the rigid vehicle state and control inputs, i.e. the **static-elastic constraint**

$$\bar{\eta} = A_{vib \leftarrow \eta}^{-1} (A_{vib \leftarrow rig} \vec{x}_{rig} + B_{vib} \vec{u}) \quad (15.230)$$

Finally, by back-substitution, one has for the **static-elastic rigid vehicle EOMs**

$$\begin{aligned} \dot{\vec{x}}_{rig} &= f_{rig}(\vec{x}_{rig}, \phi, \theta) \\ &+ \left( \mathcal{A}_{rig \leftarrow rig} - \mathcal{A}_{rig \leftarrow \eta} A_{vib \leftarrow \eta}^{-1} A_{vib \leftarrow rig} \right) \vec{x}_{rig} \\ &+ \left( B_{rig} - \mathcal{A}_{rig \leftarrow \eta} \mathcal{A}_{vib \leftarrow \eta}^{-1} B_{vib} \right) \vec{u} \end{aligned} \quad (15.231)$$

which is a process known as **residualization** of the vibration degrees-of-freedom into the new matrices of static-elastic stability and control derivatives/coefficients, i.e. elements of

$(\mathcal{A}_{rig \leftarrow rig} - \mathcal{A}_{rig \leftarrow \eta} A_{vib \leftarrow \eta}^{-1} A_{vib \leftarrow rig})$  and  $(B_{rig} - \mathcal{A}_{rig \leftarrow \eta} A_{vib \leftarrow \eta}^{-1} B_{vib})$ . It is important to note that, in general, these residualized static-elastic derivatives/coefficients depend on the flight conditions as these directly affect the loads on the vehicle's structure, especially the dynamic pressure. Furthermore, if the aerodynamic forces and moments are not truly linear, then one must use numerical techniques to find the static-elastic modal coordinates.

### Linearized Elastic Flight Vehicle Dynamics

Furthermore, as shown previously for the rigid flight vehicle dynamics, one can linearize the elastic flight vehicle EOMs for easier analysis, simulation, and design. For elastic vehicles as  $f_{rig}(\vec{x}_{rig}, \theta, \phi)$  and the Euler angle equations are the only nonlinear term of the EOMs, but as these terms are rigid vehicle specific, one can simply use a rigid flight vehicle linearization method to form the LTI state-space system as

$$\begin{bmatrix} \Delta \dot{\vec{x}}_{rig} \\ \Delta \dot{\vec{x}}_{eul} \\ \Delta \dot{\vec{x}}_{vib} \end{bmatrix} = \begin{bmatrix} A_{rig \leftarrow rig} & A_{rig \leftarrow eul} & A_{rig \leftarrow vib} \\ A_{eul \leftarrow rig} & A_{eul \leftarrow eul} & 0_{3 \times 2n} \\ A_{vib \leftarrow rig} & 0_{2n \times 3} & A_{vib \leftarrow vib} \end{bmatrix} \begin{bmatrix} \Delta \vec{x}_{rig} \\ \Delta \vec{x}_{eul} \\ \Delta \vec{x}_{vib} \end{bmatrix} + \begin{bmatrix} B_{rig} \\ 0_{3 \times 4} \\ B_{vib} \end{bmatrix} \Delta \vec{u} \quad (15.232)$$

where

$$\Delta \vec{x}_{eul} = \begin{bmatrix} \Delta \phi \\ \Delta \theta \\ \Delta \psi \end{bmatrix} \quad (15.233)$$

$$A_{vib \leftarrow vib} = \begin{bmatrix} 0_{n \times n} & I_{n \times n} \\ A_{vib \leftarrow \eta} & A_{vib \leftarrow \dot{\eta}} \end{bmatrix} \quad (15.234)$$

$A_{rig \leftarrow rig}$  has been altered from  $\mathcal{A}_{rig \leftarrow rig}$  due to the linearization of  $f_{rig}(\vec{x}_{rig}, \theta, \phi)$  with respect to  $\vec{x}_{rig}$ ,  $A_{rig \leftarrow eul}$  results from the linearization of  $f_{rig}(\vec{x}_{rig}, \theta, \phi)$  with respect to  $\vec{x}_{eul}$ , and  $A_{eul \leftarrow rig}$  and  $A_{eul \leftarrow eul}$  result from the linearization of the supplemental Euler angle equation. The explicit computation of this

linearized state matrix for the rigid vehicle about a general trim/reference flight condition will be addressed in more detail later.

For an explicit example of a linearized elastic flight vehicle equation of motion, consider the simpler case where the trim flight condition is straight-and-level flight and the longitudinal and lateral-directional EOMs can be decoupled. Furthermore, assume that the vibration modes can also be decoupled between the longitudinal and lateral-directional and  $\dot{\alpha}$  derivatives are zero. Then, the longitudinal rigid airplane LTI state-space model

$$\begin{bmatrix} \Delta\dot{u} \\ \Delta\dot{\alpha} \\ \Delta\dot{q} \\ \Delta\dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_\alpha & X_q & -g \\ \frac{Z_u}{\bar{u}} & \frac{Z_\alpha}{\bar{u}} & 1 + \frac{Z_q}{\bar{u}} & 0 \\ M_u & M_\alpha & M_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta\alpha \\ \Delta q \\ \Delta\theta \end{bmatrix} + \begin{bmatrix} X_{\delta_e} & X_{\delta_t} \\ \frac{Z_{\delta_e}}{\bar{u}} & \frac{Z_{\delta_t}}{\bar{u}} \\ M_{\delta_e} & M_{\delta_t} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\delta_e \\ \Delta\delta_t \end{bmatrix} \quad (15.235)$$

which models the following portions of Equation 15.232 as

$$\begin{bmatrix} \Delta\dot{\vec{x}}_{rig} \\ \Delta\dot{\vec{x}}_{eul} \end{bmatrix} = \begin{bmatrix} A_{rig \leftarrow rig} & A_{rig \leftarrow eul} \\ A_{eul \leftarrow rig} & A_{eul \leftarrow eul} \end{bmatrix} \begin{bmatrix} \Delta\vec{x}_{rig} \\ \Delta\vec{x}_{eul} \end{bmatrix} + \begin{bmatrix} B_{rig} \\ 0 \end{bmatrix} \Delta\vec{u} \quad (15.236)$$

Note that here  $\Delta\alpha$  has been used in place of  $\Delta w$ . Then, one may form the other matrices as

$$A_{rig \leftarrow vib} = \begin{bmatrix} X_{\eta_1} & \cdots & X_{\eta_n} & X_{\dot{\eta}_1} & \cdots & X_{\dot{\eta}_n} \\ \frac{Z_{\eta_1}}{\bar{u}} & \cdots & \frac{Z_{\eta_n}}{\bar{u}} & \frac{Z_{\dot{\eta}_1}}{\bar{u}} & \cdots & \frac{Z_{\dot{\eta}_n}}{\bar{u}} \\ M_{\eta_1} & \cdots & M_{\eta_n} & M_{\dot{\eta}_1} & \cdots & M_{\dot{\eta}_n} \end{bmatrix} \quad (15.237)$$

and defining the **aeroelastic stability and control derivative** for the state or input  $\bullet$  as

$$\Xi_{i\bullet} = \frac{Q_{i\bullet}}{\mathcal{M}_i} \quad (15.238)$$

one can write the vibration state and input sub-matrices as

$$A_{vib \leftarrow rig} = \begin{bmatrix} \Xi_{1_u} & \Xi_{1_\alpha} & \Xi_{1_q} \\ \vdots & \vdots & \vdots \\ \Xi_{n_u} & \Xi_{n_\alpha} & \Xi_{n_q} \end{bmatrix} \quad (15.239)$$

$$A_{vib \leftarrow \eta} = \begin{bmatrix} \Xi_{1_{\eta_1}} & \cdots & \Xi_{1_{\eta_n}} \\ \vdots & \ddots & \vdots \\ \Xi_{n_{\eta_1}} & \cdots & \Xi_{n_{\eta_n}} \end{bmatrix} - \Omega^2 \quad (15.240)$$

$$A_{vib \leftarrow \dot{\eta}} = \begin{bmatrix} \Xi_{1_{\dot{\eta}_1}} & \cdots & \Xi_{1_{\dot{\eta}_n}} \\ \vdots & \ddots & \vdots \\ \Xi_{n_{\dot{\eta}_1}} & \cdots & \Xi_{n_{\dot{\eta}_n}} \end{bmatrix} - 2\Omega_\zeta \quad (15.241)$$

and

$$B_{vib} = \begin{bmatrix} \Xi_{1\delta_e} & \Xi_{1\delta_t} \\ \vdots & \vdots \\ \Xi_{n\delta_e} & \Xi_{n\delta_t} \end{bmatrix} \quad (15.242)$$

Thus, in the end, one has

$$\begin{bmatrix} \Delta\dot{u} \\ \Delta\dot{\alpha} \\ \Delta\dot{q} \\ \Delta\dot{\theta} \\ \Delta\dot{\eta}_1 \\ \vdots \\ \Delta\dot{\eta}_n \\ \Delta\ddot{\eta}_1 \\ \vdots \\ \Delta\ddot{\eta}_n \end{bmatrix} = \begin{bmatrix} X_u & X_\alpha & X_q & -g & X_{\eta_1} & \cdots & X_{\eta_n} & X_{\dot{\eta}_1} & \cdots & X_{\dot{\eta}_n} \\ \frac{Z_u}{\bar{u}} & \frac{Z_\alpha}{\bar{u}} & 1 + \frac{Z_q}{\bar{u}} & 0 & \frac{Z_{\eta_1}}{\bar{u}} & \cdots & \frac{Z_{\eta_n}}{\bar{u}} & \frac{Z_{\dot{\eta}_1}}{\bar{u}} & \cdots & \frac{Z_{\dot{\eta}_n}}{\bar{u}} \\ M_u & M_\alpha & M_q & 0 & M_{\eta_1} & \cdots & M_{\eta_n} & M_{\dot{\eta}_1} & \cdots & M_{\dot{\eta}_n} \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 1 \\ \Xi_{1u} & \Xi_{1\alpha} & \Xi_{1q} & 0 & \Xi_{1\eta_1} & \cdots & \Xi_{1\eta_n} & \Xi_{1\dot{\eta}_1} & \cdots & \Xi_{1\dot{\eta}_n} \\ \vdots & \vdots & \vdots & 0 & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \Xi_{nu} & \Xi_{n\alpha} & \Xi_{nq} & 0 & \Xi_{n\eta_1} & \cdots & \Xi_{n\eta_n} & \Xi_{n\dot{\eta}_1} & \cdots & \Xi_{n\dot{\eta}_n} \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta q \\ \Delta \theta \\ \Delta \eta_1 \\ \vdots \\ \Delta \eta_n \\ \Delta \dot{\eta}_1 \\ \vdots \\ \Delta \dot{\eta}_n \end{bmatrix} \quad (15.243)$$

$$+ \begin{bmatrix} X_{\delta_e} & X_{\delta_t} \\ \frac{Z_{\delta_e}}{\bar{u}} & \frac{Z_{\delta_t}}{\bar{u}} \\ M_{\delta_e} & M_{\delta_t} \\ 0 & 0 \\ \Xi_{1\delta_e} & \Xi_{1\delta_t} \\ \vdots & \vdots \\ \Xi_{n\delta_e} & \Xi_{n\delta_t} \end{bmatrix} \begin{bmatrix} \Delta \delta_e \\ \Delta \delta_t \end{bmatrix}$$

# Chapter 16

## MIMO LTI Control System Robustness

### 16.1 MIMO LTI Control System Analysis

#### Norms for Signals and Systems

Recall that one can define multiple signals within the system, e.g. commands, errors, states, outputs, or internal variables within the dynamics. Furthermore, in control system design, one is typically concerned with the “size” of certain signals within the system. Consider piecewise continuous scalar signals,  $u(t)$ , which map  $\mathbb{R} \rightarrow \mathbb{R}$ . Then, one can define the signal  $p$ -norm as

$$\|u\|_p = \left( \int_{-\infty}^{\infty} |u(t)|^p dt \right)^{\frac{1}{p}} \quad (16.1)$$

which for the  $\infty$ -norm is notably

$$\|u\|_{\infty} = \sup_t |u(t)| \quad (16.2)$$

In addition, one is often interested in the power of signals which can be the instantaneous power or an average power. The average power of  $u$  is

$$\lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T u^2(t) dt \quad (16.3)$$

where if this limit exists, then the signal is called a **power signal** and one can define

$$\text{pow}(u) = \left( \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T u^2(t) dt \right)^{\frac{1}{2}} \quad (16.4)$$

which is notably not a norm as nonzero signals can have zero average power. For example, note that

$$\frac{1}{2T} \int_{-T}^T u^2(t) dt \leq \frac{1}{2T} \int_{-T}^T |u(t)|^2 dt = \frac{1}{2T} \|u\|_2^2 \quad (16.5)$$

which if  $\|u\|_2 < \infty$ , then as  $T \rightarrow \infty$ , one has  $\text{pow}(u) = 0$ .

In addition, note that

$$\frac{1}{2T} \int_{-T}^T u^2(t) dt \leq \frac{1}{2T} \int_{-T}^T \|u\|_\infty^2 dt = \|u\|_\infty^2 \frac{1}{2T} \int_{-T}^T dt = \|u\|_\infty^2 \quad (16.6)$$

$\|u\|_\infty \leq \infty$ , then  $\text{pow}(u) \leq \|u\|_\infty$ . For input signals that exhibit this second characteristic, consider the output response of a stable SISO LTI system, i.e.

$$y(s) = G(s)u(s) \rightarrow y(t) = \int_{-\infty}^{\infty} g(t-\tau)u(\tau)d\tau \quad (16.7)$$

where  $G$  can be described as

- **stable:**  $G$  has all poles in the closed RHP, i.e.  $\text{Re}(s) \geq 0$
- **proper:**  $G(j\infty)$  is finite (denominator order  $\geq$  numerator order)
- **strictly proper:**  $G(j\infty) = 0$  (denominator order  $>$  numerator order)
- **biproper:**  $G(j\infty) = 0$  and  $G$  and  $G^{-1}$  are both proper

For a stable  $G$ ,  $\|G\|_1$  is given by

$$\|G\|_1 = \frac{1}{2\pi} \int_{-\infty}^{\infty} |G(j\omega)| d\omega = \int_{-\infty}^{\infty} |g(t)| dt = \|g\|_1 \quad (16.8)$$

Then, recall from Parseval's theorem

$$\|G\|_2 = \left( \frac{1}{2\pi} \int_{-\infty}^{\infty} |G(j\omega)|^2 d\omega \right)^{\frac{1}{2}} = \left( \int_{-\infty}^{\infty} g^2(t) dt \right)^{\frac{1}{2}} = \|g\|_2 \quad (16.9)$$

For a stable  $G$ ,  $\|G\|_2$  is finite if and only if  $G$  is strictly proper with no poles on the  $j\omega$  axis and explicitly is given by

$$\|G\|_2^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} |G(j\omega)|^2 d\omega = \frac{1}{2\pi j} \int_{-j\infty}^{j\infty} G(-s)G(s) ds = \frac{1}{2\pi j} \oint G(-s)G(s) ds \quad (16.10)$$

For a stable  $G$ ,  $\|G\|_\infty$  is finite if and only if  $G$  is proper with no poles on the  $j\omega$  axis, and explicitly is

$$\|G\|_\infty = \max |G(j\omega)| \quad (16.11)$$

i.e. the peak of the Bode plot of  $G(j\omega)$ . Note also that there is a sub-multiplicative property of the  $\infty$ -norm, i.e.

$$\|GH\|_\infty \leq \|G\|_\infty \|H\|_\infty \quad (16.12)$$

which allows the bounding the combined system norm via norms on its elements. Thus, the relationships between the input-to-output signal norms and the defined system norms, one has

	$\ u\ _2$	$\ u\ _\infty$	$\text{pow}(y)$
$\ y\ _2$	$\ G\ _\infty$	$\infty$	$\infty$
$\ y\ _\infty$	$\ G\ _2$	$\ G\ _1$	$\infty$
$\text{pow}(y)$	0	$\leq \ G\ _\infty$	$\ G\ _\infty$

For a SISO LTI system in state-space form can be written generally as

$$\begin{aligned}\dot{\vec{x}} &= A\vec{x} + \vec{b}u \\ y &= \vec{c}^T\vec{x}\end{aligned}\quad (16.13)$$

whose transfer function is

$$G(s) = \vec{c}^T(sI - A)^{-1}\vec{b} \quad (16.14)$$

and the time solution is

$$G(t) = \vec{c}^T e^{At} \vec{b} \quad (16.15)$$

Then, letting

$$P = \int_0^\infty e^{A\tau} \vec{b} \vec{b}^T e^{A\tau} d\tau \quad (16.16)$$

and if the system is stable, the matrix exponential

$$e^{At} = I + tA + \frac{t^2}{2!}A^2 + \dots \quad (16.17)$$

converges in time. Thus,

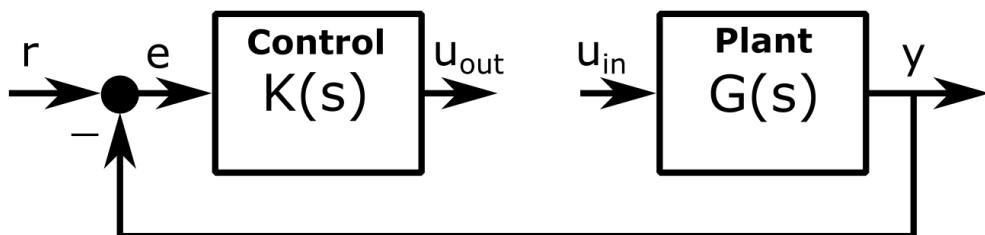
$$AP + PA^T + \vec{b} \vec{b}^T = 0 \quad (16.18)$$

and

$$\|G\|_2^2 = \int_0^\infty \vec{c}^T e^{A\tau} \vec{b} \vec{b}^T e^{A\tau} \vec{c} d\tau = \vec{c}^T P \vec{c} \quad (16.19)$$

### System Transfer Functions

Many of the frequency domain analysis models for MIMO systems are natural extensions of transfer functions used to analyze SISO systems. However, because of the non-commutativity of matrices, the analysis of these systems can depend on where the closed-loop is broken for analysis. To demonstrate this, first, consider the following SISO block diagram



The loop gain for this system can be calculated by breaking the loop at the control generation point and injecting a signal  $u_{in}$ . Then, the returned signal is

$$u_{out} = -K(s)G(s)u_{in} = -L(s)u_{in} \quad (16.20)$$

where  $L(s)$  is the **loop gain transfer function**, also known as the **open-loop transfer function**. Differencing the two signals, then results in the **return difference**

$$u_{in} - u_{out} = u_{in} + K(s)G(s)u_{in} = (1 + K(s)G(s))u_{in} = (1 + L(s))u_{in} \quad (16.21)$$

The inverse of the return difference is the **error transfer function**, also known as the **sensitivity transfer function**, defined as

$$S(s) = \frac{E(s)}{R(s)} = \frac{1}{1 + L(s)} \quad (16.22)$$

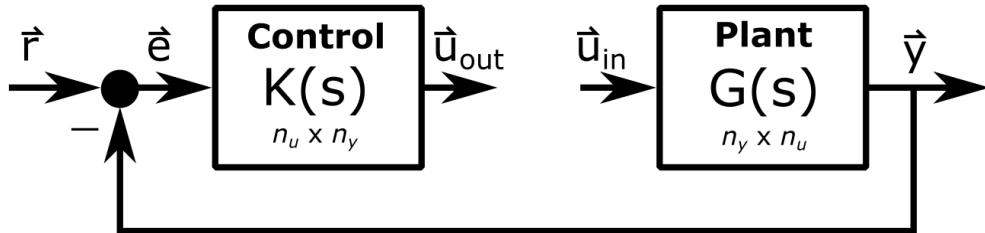
while the **closed-loop transfer function** is defined as

$$T(s) = \frac{Y(s)}{R(s)} = \frac{L(s)}{1 + L(s)} \quad (16.23)$$

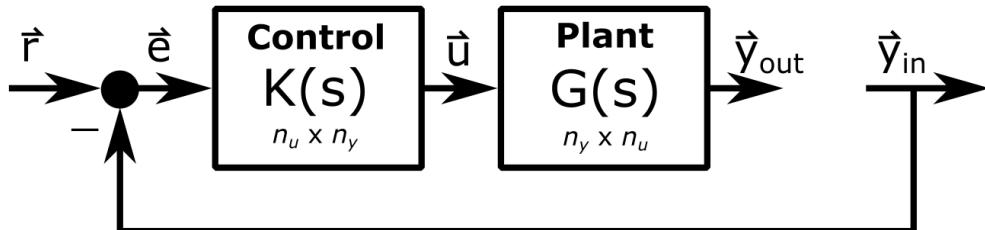
which is also known as the **complementary sensitivity transfer function** as

$$S(s) + T(s) = 1 \quad (16.24)$$

Each of these definitions can also be extended to the MIMO feedback control system for breaking the loop at both the *plant input* as the block diagram



and the *plant output* as the block diagram



where the signals are now vector-valued and the transfer functions are matrices of transfer functions. The **MIMO loop gain at the plant input** is thus the  $n_u \times n_u$  matrix

$$L_i(s) = K(s)G(s) \quad (16.25)$$

while the **loop gain at the plant output**

$$L_o(s) = G(s)K(s) \quad (16.26)$$

The **return difference at the plant input** is

$$u_{in} - u_{out} = I_{n_u \times n_u} + L_i(s) \quad (16.27)$$

while the **return difference at the plant output**

$$y_{in} - y_{out} = I_{n_y \times n_y} + L_o(s) \quad (16.28)$$

Furthermore the **sensitivity at the plant input**

$$S_i(s) = (I_{n_u \times n_u} + L_i(s))^{-1} \quad (16.29)$$

while the **sensitivity at the plant output**

$$S_o(s) = (I_{n_y \times n_y} + L_o(s))^{-1} \quad (16.30)$$

and the **complementary sensitivity at the plant input**

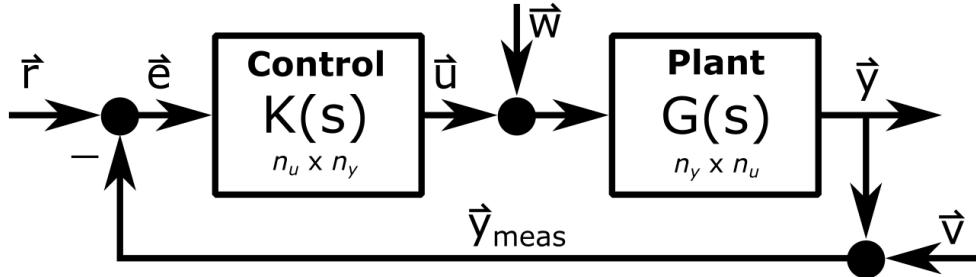
$$T_i(s) = (I_{n_u \times n_u} + L_i(s))^{-1} L_i(s) \quad (16.31)$$

while the **complementary sensitivity at the plant output**

$$T_o(s) = (I_{n_y \times n_y} + L_o(s))^{-1} L_o(s) \quad (16.32)$$

Note that typically the subscripts on these may be dropped at times and context should provide the dimension of the identity matrices and which of the blocks' transfer function definitions is being used. Note that when the plant and controller matrices are non-square, one should conduct the stability analysis at the loop break point of minimum dimension.

Finally, in most MIMO LTI systems, one can generally write the additional input signals to the feedback control system as a disturbance  $\vec{w}(t) \in \mathbb{R}^{n_u}$  and sensor noise  $v(t) \in \mathbb{R}^{n_y}$



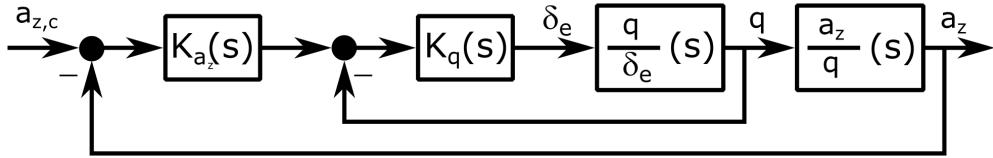
Thus, one can write that the output response can be written as

$$Y(s) = T(s)R(s) + S(s)G(s)W(s) + T(s)V(s) \quad (16.33)$$

which demonstrates that the output response depends on both  $S(s)$  and  $T(s)$ . At frequencies  $s = j\omega$  where the commands are to be followed, one desires  $T(s) = I$ , or  $S(s) \rightarrow 0$  as  $S(s) + T(s) = I$ . However, this also passes any noise signals through the system to the output. Thus, it is not possible to track commands and reject sensor noise *at the same frequencies*. Thus, one typically desires  $T(s) \rightarrow 0$  at high frequencies to reject noise (and high frequency unmodeled dynamics). At frequencies  $s = j\omega$  where the plant disturbances are to be rejected, one also desires  $S(s) \rightarrow 0$  which occurs at low frequencies for good reference tracking. Thus, because  $S(s) + T(s) = 1$  at all frequencies, control engineers design  $K(s)$  such that one optimally balances these tradeoffs.

### Airplane Cascade Loop Example

As an example, consider a proportional-integral (PI) cascade loop control architecture for the longitudinal dynamics of an airplane. This type of controller can be represented as a block diagram



or as a single-input, multiple-output system (SIMO) LTI state-space system

$$\begin{aligned} \begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} &= \begin{bmatrix} \frac{Z_\alpha}{v_\infty} & 1 \\ M_\alpha & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} \frac{Z_\delta}{v_\infty} \\ M_\delta \end{bmatrix} \delta_e \\ \begin{bmatrix} a_z \\ q \end{bmatrix} &= \begin{bmatrix} Z_\alpha & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} Z_\delta \\ 0 \end{bmatrix} \delta_e \end{aligned} \quad (16.34)$$

where  $\vec{x} = [\alpha \ q]^T$ ,  $\vec{u} = \delta_e$ ,  $\vec{y} = [a_z \ q]^T$ ,

$$A = \begin{bmatrix} \frac{Z_\alpha}{v_\infty} & 1 \\ M_\alpha & 0 \end{bmatrix} \quad (16.35)$$

$$B = \begin{bmatrix} \frac{Z_\delta}{v_\infty} \\ M_\delta \end{bmatrix} \quad (16.36)$$

$$C = \begin{bmatrix} Z_\alpha & 0 \\ 0 & 1 \end{bmatrix} \quad (16.37)$$

and

$$D = \begin{bmatrix} Z_\delta \\ 0 \end{bmatrix} \quad (16.38)$$

The  $2 \times 1$  transfer function matrix for the plant dynamics is

$$G(s) = C(sI - A)^{-1}B + D = \begin{bmatrix} \frac{a_z}{\delta_e} \\ \frac{q}{\delta_e} \end{bmatrix} \quad (16.39)$$

The controller for this plant contains two PI controllers designed as

$$K_{a_z}(s) = \frac{\beta_{a_z}(s + \bar{\omega}_{a_z})}{s} \quad (16.40)$$

and

$$K_q(s) = \frac{\beta_q(s + \bar{\omega}_q)}{s} \quad (16.41)$$

with the  $2 \times 1$  controller transfer function matrix given as

$$K(s) = [K_{a_z}(s)K_q(s) \quad K_q(s)] \quad (16.42)$$

A LTI state-space model for this controller is

$$\begin{aligned} \vec{x}_c &= A_c \vec{x}_c + B_{c,1} \vec{y} + B_{c,2} \vec{r} \\ \vec{u} &= C_c \vec{x}_c + D_{c,1} \vec{y} + D_{c,2} \vec{r} \end{aligned} \quad (16.43)$$

where

$$A_c = \begin{bmatrix} 0 & 0 \\ K_q \bar{\omega}_q & 0 \end{bmatrix} \quad (16.44)$$

$$B_{c,1} = \begin{bmatrix} -\beta_{a_z} \bar{\omega}_{a_z} & 0 \\ -\beta_{a_z} \beta_q \bar{\omega}_q & -\beta_q \bar{\omega}_q \end{bmatrix} \quad (16.45)$$

$$B_{c,2} = \begin{bmatrix} \beta_{a_z} \bar{\omega}_{a_z} \\ \beta_{a_z} \beta_q \bar{\omega}_q \end{bmatrix} \quad (16.46)$$

$$C_c = [\beta_q \quad 1] \quad (16.47)$$

$$D_{c,1} = [-\beta_{a_z} \beta_q \quad -\beta_q] \quad (16.48)$$

$$D_{c,2} = [\beta_{a_z} \beta_q] \quad (16.49)$$

The loop gain at the plant input is

$$L_i(s) = K(s)G(s) = K_{a_z}(s)K_q(s)\frac{a_z}{\delta_e}(s) + K_q(s)\frac{q}{\delta_e}(s) \quad (16.50)$$

which is a scalar transfer function. Thus, the stability of this system can be analyzed using SISO analysis techniques. The loop gain at the plant output is

$$L_o(s) = G(s)K(s) = \begin{bmatrix} \frac{a_z}{\delta_e}(s)K_{a_z}(s)K_q(s) & \frac{a_z}{\delta_e}(s)K_q(s) \\ \frac{q}{\delta_e}(s)K_{a_z}(s)K_q(s) & \frac{q}{\delta_e}(s)K_q(s) \end{bmatrix} \quad (16.51)$$

which is a  $2 \times 2$  singular matrix.

## 16.2 Multivariate Frequency Response

### Singular Values

The **singular value decomposition (SVD)** of a matrix  $A \in \mathbb{C}^{n \times m}$  is given by

$$A = U\Sigma V^* \quad (16.52)$$

where  $\bullet^*$  denotes complex conjugate transpose, and where  $\Sigma \in \mathbb{R}^{n \times m}$ ,  $U \in \mathbb{C}^{n \times n}$ , and  $V \in \mathbb{C}^{m \times m}$  are unitary matrices, i.e.  $U^* = U^{-1}$  and  $V^* = V^{-1}$ . The columns of  $U$  denote the **left singular vectors** of  $A$ , while the columns of  $V$  denote the **right singular values** of  $A$ . Assuming  $A$  is of rank  $k$ , the singular value matrix is

$$\Sigma = \begin{bmatrix} \Sigma_1 & \tilde{0} \\ \tilde{0} & \tilde{0} \end{bmatrix}, \quad \Sigma_1 = \text{diag}(\sigma_1, \dots, \sigma_k) \quad (16.53)$$

with the singular values ordered in size with  $\bar{\sigma} = \sigma_1 \geq \dots \geq \sigma_k = \underline{\sigma}$ . The term “singular values” derives from their use for analysis of the near singularity of matrices, e.g. if  $A$  is a square singular matrix, then  $\underline{\sigma} = 0$ , and  $A$  is not invertible.

The **maximum singular value** of  $A$  can be shown to be

$$\bar{\sigma}(A) = \max_{\vec{x} \neq 0} \frac{\|A\vec{x}\|_2}{\|\vec{x}\|_2} = \|A\|_2 \quad (16.54)$$

while the **minimum singular value** of  $A$  can be shown to be

$$\underline{\sigma}(A) = \min_{\vec{x} \neq 0} \frac{\|A\vec{x}\|_2}{\|\vec{x}\|_2} \quad (16.55)$$

The  $\bar{\sigma}(A)$ , i.e.  $\|A\|_2$ , represents how “big”  $A$  is or how large the “gain” of  $A$  is. The  $\underline{\sigma}(A)$  represents how nearly singular  $A$  is. The **condition number** for  $A$  is defined as

$$\kappa(A) = \frac{\bar{\sigma}(A)}{\underline{\sigma}(A)} \quad (16.56)$$

and can be used to determine how invertible  $A$  is. Each singular value has an input and output “direction” determined by the right and left singular vectors associated with the SVD, i.e.

$$V = [\vec{v}_1 \quad \dots \quad \vec{v}_m] \quad (16.57)$$

where  $\vec{v}_i$  are the right singular vectors and

$$U = [\vec{u}_1 \quad \dots \quad \vec{u}_n] \quad (16.58)$$

where  $\vec{u}_i$  are the left singular vectors. All vectors are orthonormal. These singular vectors satisfy the equations

$$\begin{aligned} A\vec{v}_i &= \sigma_i \vec{u}_i \\ A^*\vec{u}_i &= \sigma_i \vec{v}_i \end{aligned} \quad (16.59)$$

Furthermore, recall  $\text{rank}(A) = k = \min(n, m)$  and the  $k$  nonzero singular values of  $A$ ,  $\sigma_i(A)$ , can be shown to be related to the eigenvalue decomposition by

$$\sigma_i(A) = \sqrt{\lambda_i(A^*A)} = \sqrt{\lambda_i(AA^*)} > 0 \quad (16.60)$$

and

$$\begin{aligned} A^*A\vec{v}_i &= \sigma_i^2\vec{v}_i \\ AA^*\vec{u}_i &= \sigma_i^2\vec{u}_i \end{aligned} \quad (16.61)$$

Thus, all  $\sigma_i^2$  are eigenvalues of  $AA^*$  and  $A^*A$ , all  $\vec{v}_i$  are eigenvectors of  $A^*A$ , and all  $\vec{u}_i$  are eigenvectors of  $AA^*$ . With these definitions, the SVD of  $A$  can also be written as

$$A = \sum_{i=1}^k \sigma_i \vec{u}_i \vec{v}_i^* \quad (16.62)$$

and the following properties also hold for invertible  $A$ :

$$\bar{\sigma}(A^{-1}) = \frac{1}{\sigma(A)} \quad \text{and} \quad \underline{\sigma}(A^{-1}) = \frac{1}{(\bar{A})} \quad (16.63)$$

$$\|A\|_F^2 = \sum_{i=1}^k \sigma_i^2(A) \quad (16.64)$$

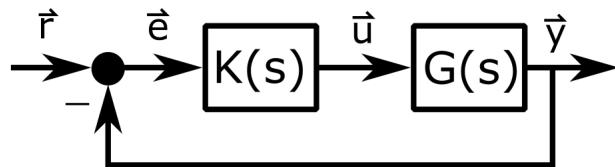
and

$$\sigma_i(UA) = \sigma_i(A) \quad \sigma_i(AV) = \sigma_i(A) \quad (16.65)$$

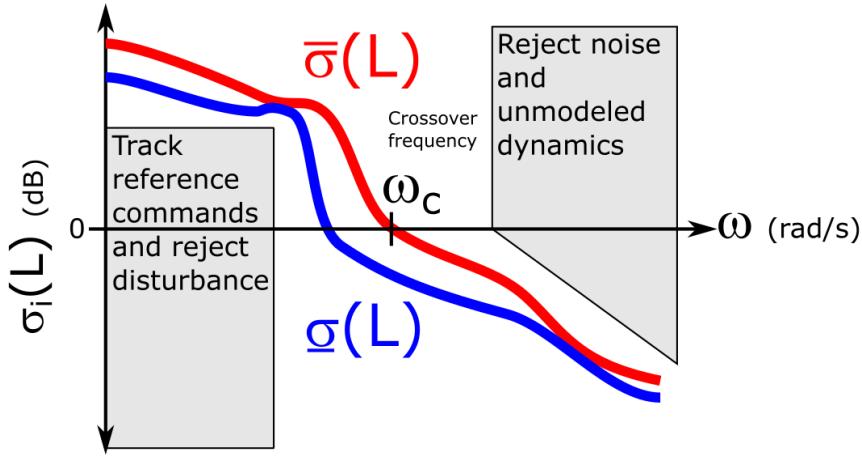
Thus, for complex-valued transfer function matrices,  $G(j\omega) \in \mathbb{C}^{n \times m}$ , one can assess the frequency response input-output behavior through the SVD, in particular the relative gain by the singular values.

## MIMO Frequency Domain Characteristics

Consider the following MIMO feedback control system diagram



where  $\vec{r}$  is the reference command,  $\vec{e}$  is the error,  $\vec{u}$  is the control input,  $\vec{y}$  is the output, and  $L(s) = K(s)G(s)$  is the loop gain transfer function matrix. For MIMO LTI systems, the following diagram can be used to analyze the desired frequency response for the system.



In order to track reference commands at low frequency, the loop gain must have sufficiently large magnitude. In order to reject high frequency sensor noise and unmodeled dynamics, the loop gain must have sufficiently small magnitude. The frequency band between these two conflicting requirements is where the loop gain crosses 0 dB at the **loop gain crossover frequency**,  $\omega_c$ . For MIMO systems, the loop gain range of magnitudes at any given frequency are given by the maximum and minimum singular values, i.e.  $\bar{\sigma}(L)$  and  $\underline{\sigma}(L)$ , respectively, and the loop gain crossover frequency is defined for  $\bar{\sigma}(L)$ .

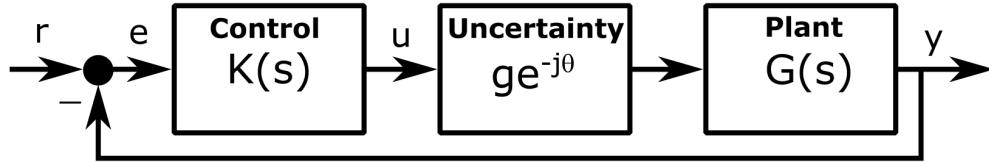
### 16.3 MIMO LTI System Uncertainty Modeling

Feedback control system robustness to uncertainties in dynamics is a crucial design criterion for the stability of flight vehicles. Historically, the most widely used measure of stability robustness in flight vehicles has been single-loop gain and phase margins derived from classical frequency response analysis, typically designated as at least  $\pm 6$  dB gain margin and  $45^\circ$  phase margin. It has been proven in many real-world applications that poor stability margins lead to poor system performance. Whether these stability margins be classical or singular value based, having adequate gain and phase margins is an important aspect of control system design.

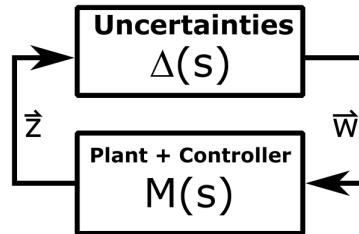
However, additional methods exist for robustness analysis of control systems due to neglected and mismodeled dynamics and parameter uncertainties. These methods try to determine bounds on how large the uncertainties can be before the system becomes unstable. The amount of conservatism in computing these bounds varies between methods due to different uncertainty modeling and connections to the nominal control system which can be generalized through the  $\Delta(s)M(s)$  uncertainty model. This lecture will also discuss two common methods for robustness analysis, namely, the small-gain theorem (SGT) and the structured singular value (SSV), both of which assume that  $\Delta(s)$  is complex valued. Thus, if the uncertainty can be modeled by real and complex parameters, one could reduce the conservatism by additional structure in  $\Delta$ , e.g. DeGaston-Safanov's Real Stability Margin for only real-parameter uncertainty. However, before introducing these methods, the  $\Delta M$  model for uncertain control systems will be discussed.

### $\Delta M$ Uncertainty Model

Recall that SISO system uncertainty typically uses a gain and phase uncertainty model as

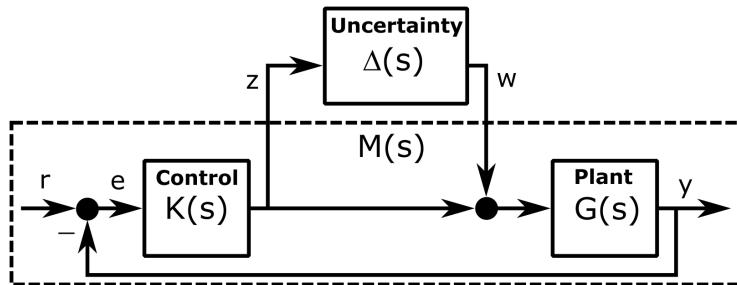


where the gain margin computes the “smallest”  $g$  (holding  $\theta = 0$ ) and the phase margin (holding  $g = 1$ ) for which the system goes unstable. The MIMO extension of this uncertainty model can be constructed using different methods such as block diagram and/or control loop equation algebra. The resulting uncertainty models,  $\Delta(s)$  may have a “structure” associated with them depending on the specific problem and the analysis will depend on the structure. With these models, the uncertainties in the system are isolated from the nominal plant and controller model  $M(s)$  as shown in the  $\Delta M$  analysis model



where if  $\Delta(s)$  is considered as a full complex-valued matrix, it would be considered **unstructured**.

As an example of how this framework can be used generally to form a structured uncertainty model, consider the following construction for the SISO uncertainty model



where, by inspection,

$$ge^{-j\theta} = 1 + \Delta(s) \quad (16.66)$$

in a similar way, one can add any number  $n$  of additional uncertainty blocks along any signal in the nominal control system,  $M(s)$ , even vector-valued signals. These uncertainty blocks,  $\Delta_1(s), \dots, \Delta_n(s)$  could be added

to model actuator/sensor uncertainties, unmodeled dynamics, and/or signal time delays. Then, by block diagram manipulation, one can construct a block matrix  $\Delta(s)$  of all included uncertainty blocks, i.e.

$$\Delta(s) = \begin{bmatrix} \Delta_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \Delta_n \end{bmatrix} \quad (16.67)$$

where

$$\vec{w} = \Delta(s) \vec{z} \quad (16.68)$$

and

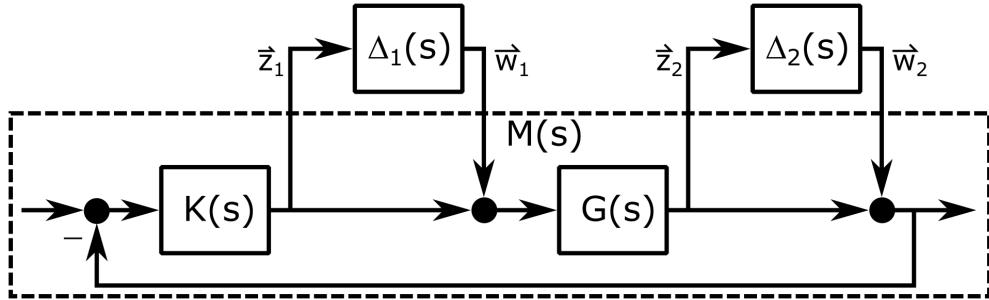
$$\vec{z} = M(s) \vec{w} \quad (16.69)$$

which must be solved using block diagram or control loop equation algebra. For the simple SISO system above, this can be shown to be

$$z = \left( \frac{G(s)K(s)}{1 - G(s)K(s)} \right) w \quad (16.70)$$

### $\Delta M$ Example Problem

Given: the following feedback control system



Determine: the block components of  $M(s)$  and  $\Delta(s)$ , i.e.

$$\vec{z} = M(s) \vec{w} \quad (16.71)$$

and

$$\vec{w} = \Delta(s) \vec{z} \quad (16.72)$$

where

$$\vec{z} = \begin{bmatrix} \vec{z}_1 \\ \vec{z}_2 \end{bmatrix} \quad (16.73)$$

$$\vec{w} = \begin{bmatrix} \vec{w}_1 \\ \vec{w}_2 \end{bmatrix} \quad (16.74)$$

Solution:

The block diagram above infers the control loop equations

$$\vec{z}_1 = K(s)(\vec{z}_2 + \vec{w}_2) \quad (16.75)$$

and

$$\vec{z}_2 = G(s)(\vec{z}_1 + \vec{w}_1) \quad (16.76)$$

Substituting the second equation into the first results in

$$\vec{z}_1 = K(s)(G(s)(\vec{z}_1 + \vec{w}_1) + \vec{w}_2) \quad (16.77)$$

$$\vec{z}_1 = K(s)G(s)\vec{z}_1 + K(s)G(s)\vec{w}_1 + K(s)\vec{w}_2 \quad (16.78)$$

$$(I - K(s)G(s))\vec{z}_1 = K(s)G(s)\vec{w}_1 + K(s)\vec{w}_2 \quad (16.79)$$

$$\vec{z}_1 = (I - K(s)G(s))^{-1}K(s)G(s)\vec{w}_1 + (I - K(s)G(s))^{-1}K(s)\vec{w}_2 \quad (16.80)$$

Substituting the first equation into the second results in

$$\vec{z}_2 = G(s)(K(s)(\vec{z}_2 + \vec{w}_2) + \vec{w}_1) \quad (16.81)$$

$$\vec{z}_2 = G(s)K(s)\vec{z}_2 + G(s)K(s)\vec{w}_2 + G(s)\vec{w}_1 \quad (16.82)$$

$$(I - G(s)K(s))\vec{z}_2 = G(s)K(s)\vec{w}_2 + G(s)\vec{w}_1 \quad (16.83)$$

$$\vec{z}_2 = (I - G(s)K(s))^{-1}G(s)\vec{w}_1 + (I - G(s)K(s))^{-1}G(s)K(s)\vec{w}_2 \quad (16.84)$$

Combining these two expressions and writing in matrix form yields

$$\begin{bmatrix} \vec{z}_1 \\ \vec{z}_2 \end{bmatrix} = \begin{bmatrix} (I - K(s)G(s))^{-1}K(s)G(s) & (I - K(s)G(s))^{-1}K(s) \\ (I - G(s)K(s))^{-1}G(s) & (I - G(s)K(s))^{-1}G(s)K(s) \end{bmatrix} \begin{bmatrix} \vec{w}_1 \\ \vec{w}_2 \end{bmatrix} \quad (16.85)$$

and

$$\underline{M(s)} = \begin{bmatrix} (I - K(s)G(s))^{-1}K(s)G(s) & (I - K(s)G(s))^{-1}K(s) \\ (I - G(s)K(s))^{-1}G(s) & (I - G(s)K(s))^{-1}G(s)K(s) \end{bmatrix} \quad (16.86)$$

and by inspection

$$\underline{\Delta} = \begin{bmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{bmatrix} \quad (16.87)$$

## 16.4 Multivariate Nyquist Stability Criterion

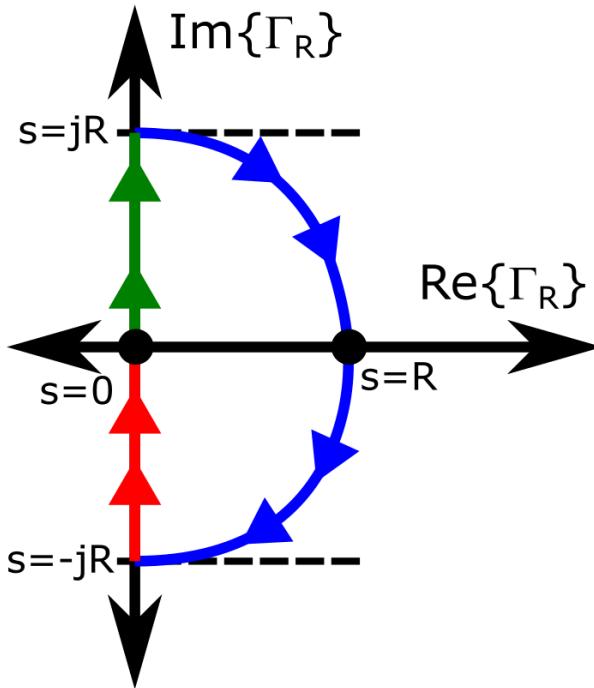
Classical control uses frequency response methods, e.g. Bode and Nyquist, to determine the stability margins of SISO systems. These methods manipulate the open-loop transfer function of the system to derive gain and phase margins, typical measures of relative stability. In MIMO systems, the open-loop transfer function is a complex-valued matrix, making it more difficult to apply the same gain and phase margin arguments to determine relative stability. Thus, the notion of gain or magnitude of the loop transfer function matrix requires the use of a “magnitude” of a matrix versus frequency. To do this, one can use the singular values of a matrix which will be described in this lecture. In addition, one can extend the Multivariable Nyquist Stability Criterion to derive stability margins for MIMO systems and define singular value stability margins.

These are important metrics in robust control design. Though the multivariable Nyquist Stability Criterion only provides a yes/no answer to the stability question, but it also leads to an important understanding of robustness tests when analyzing model uncertainties. This criterion is derived from the **Cauchy's argument principle** from complex analysis.

Let  $\Gamma$  be closed clockwise contour in the  $s$ -plane. Let  $f(s)$  be a complex-valued function. Suppose that

1.  $f(s)$  is analytic on  $\Gamma$
2.  $f(s)$  has  $Z$  zeros inside  $\Gamma$
3.  $f(s)$  has  $P$  poles inside  $\Gamma$

Then,  $f(s)$  will encircle the origin  $O$ ,  $Z - P$  in a clockwise sense as  $s$  transverses  $\Gamma$ . Let  $N_{cw}(P, f(s), \Gamma)$  denote the number of encirclements of the point  $P$  made by the function  $f(s)$  as  $s$  transverses the closed clockwise contour  $\Gamma$ . If  $\Gamma$  equals the **standard Nyquist contour** ( $\Gamma_R$ ), encircling the right half plane (RHP), i.e.



and  $f(s)$  is a rational function in  $s$ , then

$$N_{cw}(0, f(s), \Gamma_R) = Z - P \quad (16.88)$$

Note that if  $f(s) = f_1(s)f_2(s)$ , then

$$N_{cw}(0, f_1(s)f_2(s), \Gamma_R) = N(0, f_1(s), \Gamma_R) + N(0, f_2(s), \Gamma_R) = (Z_1 - P_1) + (Z_2 - P_2) = Z - P \quad (16.89)$$

Thus, because it can be shown that

$$\det(I + L(s)) = \frac{\phi_{c-l}(s)}{\phi_{o-l}(s)} \quad (16.90)$$

where  $\phi_{c-l}$  is the closed-loop system's characteristic polynomial and  $\phi_{o-l}$  is the open-loop system's characteristic polynomial. If  $\phi_{c-l}(s)$  is stable, then  $N(0, \phi_{c-l}(s), \Gamma_R) = 0$ , and by Cauchy's argument principle requires that  $N(0, \phi_{o-l}(s), \Gamma_R) + N(0, \det(I + L(s)), \Gamma_R) = 0$ .

With this in mind, one can state the **Multivariable Nyquist Theorem (MNT)**: The feedback control system will be closed-loop stable in the sense that  $\phi_{c-l}(s)$  has no closed RHP zeros if and only if for all  $R$  sufficiently large (radius of the  $\Gamma_R$ -contour)

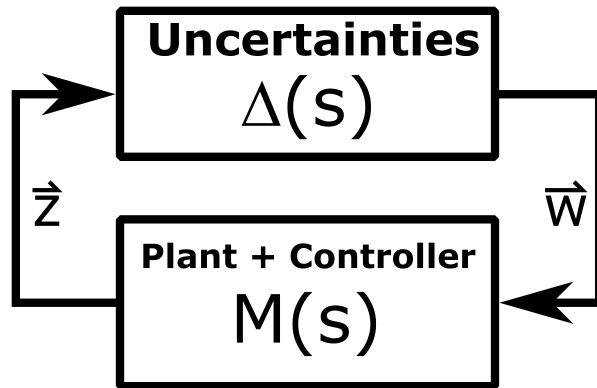
$$N_{cw}(0, \det(I + L(s)), \Gamma_R) = -P_{o-l} \quad (16.91)$$

or

$$N_{cw}(-1, -1 + \det(I + L(s)), \Gamma_R) = -P_{o-l} \quad (16.92)$$

where  $P_{o-l} = N_{cw}(0, \phi_{o-l}(s), \Gamma_R)$  equals the number of open-loop RHP poles. In words, this means the number of encirclements made by the determinant of the return difference matrix's Nyquist plot to be equal to the number of unstable open-loop poles. Encirclements can be counted relative to the origin (0, j0) or as in classical Nyquist diagrams about (-1, j0).

The MIMO system stability can be derived using the MNT by assuming  $K(s)$  stabilizes the nominal  $G(s)$ , and that gain and phase uncertainties are large enough to change the number of encirclements made by the determinant of the return difference matrix's Nyquist plot. The assumption that the nominal plant is stabilized by the controller requires that the determinant of the return difference matrix encircles the origin  $P_{o-l}$  times clockwise. Gain and phase margins for SISO systems can be computed by inserting a gain and phase variation,  $ke^{j\theta}$  in between  $K(s)$  and  $G(s)$  and solving for  $k$  (with  $\theta = 0$ ) and phase  $\theta$  (with  $k = 1$ ) that destabilizes the system. For MIMO systems, consider the stability analysis model where the uncertainties in the system are represented in a block matrix  $\Delta(s)$  and the nominal plant and controller are represented in a matrix  $M(s)$ .



which notably has no negative sum blocks. The stability analysis question here is “how large can the uncertainties  $\Delta(s)$  become before the system becomes unstable?” The loop transfer function  $L(s)$  for this

system is  $L(s) = \Delta(s)M(s)$ , with the return difference matrix being  $I - \Delta(s)M(s)$ . Using MNT, the uncertainties must change the number of encirclements made by  $\det(I - \Delta(s)M(s))$ .

### Stability Margins for MIMO Systems

Uncertainty models used for stability analysis may be categorized as unstructured or structured. If the uncertainty is modeled as a block diagonal matrix, the uncertainty is structured. Both unstructured and structured uncertainty analysis procedures use singular value analysis to measure the size of complex-valued matrices. The stability of a MIMO system can be observed by the near singularity of its return difference matrix,  $I + L(s)$ , at some frequency  $s = j\omega_0$ . If  $I + L(s)$  is nearly singular, then a small change in  $L(s)$  could make  $I + L(s)$  singular. From a SISO viewpoint, this is the distance from the  $(-1, j0)$  point in the complex plane made by the Nyquist plot of  $L(j\omega)$  which will be unstable if it encircles  $(-1, j0)$ . The robustness theory discussed here provides an analogous distance measure for MIMO systems.

Direct application of the MNT does not provide a robustness indication as  $\det(I + L(s))$  does not indicate the near singularity of  $I + L(s)$  as the MNT only determines absolute stability. Thus, to determine the degree of robustness, one requires an analysis of the singular values of the return difference matrix as a function of frequency. In particular, examining the magnitude of the singular values of the return difference matrix will indicate how close the matrix is to being singular, analogous to a gain margin. However, for this singular value analysis recall there is a restriction that the nominal loop transfer function,  $L(s)$ , is closed-loop stable. Then, letting  $L'(s)$  denote the **perturbed loop transfer function**, i.e. the true system due to uncertainties in  $L(s)$ , which has open and closed-loop polynomials,  $\phi'_{o-l}(s)$  and  $\phi'_{c-l}(s)$ , respectively. Finally, defining  $\bar{L}(s, \epsilon)$  as a matrix of rational transfer function with real coefficients which are continuous in  $\epsilon \forall \epsilon$  such that  $0 \leq \epsilon \leq 1$  and  $\forall s \in \Gamma_R$ , which satisfies  $\bar{L}(s, 0) = L(s)$  and  $\bar{L}(s, 1) = L'(s)$ , the **Fundamental Robustness Theorem**: states that the polynomial  $\phi'_{c-l}(s)$  has no zeros in the closed RHP and the perturbed feedback system is stable if

1.  $\phi_{o-l}(s)$  and  $\phi_{o-l}(s)'$  have the same number of zeros in the closed RHP.
2.  $\phi_{c-l}(s)$  has no zeros in the closed RHP.
3.  $\det(I + \bar{L}(s, \epsilon)) = 0 \forall s \in \Gamma_R, \epsilon \in [0, 1]$ , and  $R$  sufficiently large.

In other words, the closed-loop perturbed system will be stable, if, by continuously deforming the Nyquist plot for  $I + \bar{L}(s, \epsilon)$  and the number of encirclements of the critical point is the same for  $L'(s)$  and  $L(s)$ , then, no closed RHP zeros were introduced into  $\phi'_{c-l}(s)$ , resulting in a stable closed-loop system.

This theorem can be used to derive simple tests for different uncertainty models. The most common are additive (or absolute) and multiplication (or relative) errors. The additive error model is defined as

$$\Delta_a(s) = L'(s) - L(s) \quad (16.93)$$

and the multiplicative error model is defined as

$$\Delta_m(s) = [L'(s) - L(s)]L^{-1}(s) \quad (16.94)$$

The perturbed  $L(s)$  can be written as

$$\bar{L}(s, \epsilon) = L(s) + \epsilon\Delta_a(s) \quad (16.95)$$

for the additive error model and

$$\bar{L}(s, \epsilon) = (I + \epsilon \Delta_m(s))L(s) \quad (16.96)$$

for the multiplicative error model. Both equations imply the same  $\bar{L}(s, \epsilon)$  for both model error characterizations, i.e.

$$\bar{L}(s, \epsilon) = (1 - \epsilon)L(s) + \epsilon L'(s) \quad (16.97)$$

showing  $\bar{L}(s, \epsilon)$  is continuous for  $\epsilon \in [0, 1]$  and all  $s \in \Gamma_R$ . The return difference matrix is defined as

$$I + \bar{L}(s, \epsilon) = (I + L(s)) + (\epsilon \Delta_a(s)) \quad (16.98)$$

for the additive error model and

$$I + \bar{L}(s, \epsilon) = (I + L(s)) + (\epsilon \Delta_m(s)L(s)) \quad (16.99)$$

for the multiplicative error model. In both cases, one can write these in the form

$$I + \bar{L}(s, \epsilon) = A + B \quad (16.100)$$

where  $A = I + L(s)$  and  $B = \epsilon \Delta_a(s)$  or  $= \epsilon \Delta_m(s)L(s)$ . For the perturbed system to be unstable, one requires that  $A + B$  be singular for some  $\epsilon \in [0, 1]$  and  $s \in \Gamma_R$ . As  $A$  in both cases is assumed non-singular,  $B$  when added to  $A$  must make  $A + B$  singular. If  $A + B$  is singular, then  $A + B$  is rank deficient and there exists a vector  $\vec{x} \neq 0$  such that  $(A + B)\vec{x} = 0$ , i.e.  $\vec{x}$  is in the null space of  $A + B$ . This leads to  $A\vec{x} = -B\vec{x}$  with  $\|A\vec{x}\|_2 = \|B\vec{x}\|_2$ . Assuming unit magnitude  $\|\vec{x}\|_2 = 1$ , one has

$$\underline{\sigma}(A) \leq \|A\vec{x}\|_2 = \|B\vec{x}\|_2 \leq \|B\|_2 = \bar{\sigma}(B) \quad (16.101)$$

Thus, for  $A + B$  to be non-singular,  $\underline{\sigma}(A) > \bar{\sigma}(B)$ . This is precisely how the stability robustness theorems are derived.

The **Stability Robustness Theorem**: states that the polynomial  $\phi'_{c-l}(s)$  has no closed RHP zeros and the perturbed feedback system is stable if

1.  $\phi_{c-l}(s)$  has no zeros in the closed RHP
2. AND

- for additive uncertainty:  $\underline{\sigma}(I + L(s)) > \bar{\sigma}(\Delta_a(s)) \forall s \in \Gamma_R$  and  $R$  sufficiently large.
- for multiplicative uncertainty:  $\underline{\sigma}(I + L^{-1}(s)) > \bar{\sigma}(\Delta_m(s)) \forall s \in \Gamma_R$  and  $R$  sufficiently large.

Thus, as long as the singular value frequency responses do not overlap, stability is guaranteed.

For singular value stability margins, one can define for the return difference matrix

$$\min_{\omega} \underline{\sigma}(I + L(j\omega)) = \alpha_{\sigma} \quad (16.102)$$

and the **stability robustness matrix** as

$$\min_{\omega} \underline{\sigma}(I + L^{-1}(j\omega)) = \beta_{\sigma} \quad (16.103)$$

then the gain margin (GM) and phase margin (PM) for each matrix corresponding to the additive and multiplicative error model tests, respectively, can be written as

$$GM_{I+L} = \left[ \frac{1}{1 + \alpha_\sigma}, \frac{1}{1 - \alpha_\sigma} \right], \quad PM_{I+L} = \pm 2 \sin^{-1} \frac{\alpha_\sigma}{2} \quad (16.104)$$

$$GM_{I+L^{-1}} = [1 - \beta_\sigma, 1 + \beta_\sigma], \quad PM_{I+L^{-1}} = \pm 2 \sin^{-1} \frac{\beta_\sigma}{2} \quad (16.105)$$

which can be combined as

$$GM = GM_{I+L} \cup GM_{I+L^{-1}}, \quad PM = PM_{I+L} \cup PM_{I+L^{-1}} \quad (16.106)$$

Note that the best minimum singular values from the return difference matrix is  $\alpha_\sigma = 1$  as at high frequencies  $L \rightarrow 0$ , thus  $GM_{I+L} = [\frac{1}{2}, +\infty]$  or  $GM_{I+L} = [-6, +\infty]$  dB. Also note that the best minimum singular values from the stability robustness matrix is  $\beta_\sigma = 1$  as at low frequencies  $L^{-1} \rightarrow 0$ , thus  $GM_{I+L^{-1}} = [0, 2]$  or  $GM_{I+L^{-1}} = [-\infty, +6]$  dB.

## 16.5 Structured Singular Value Analysis

For unstructured  $\Delta(s)$  and as a quick stability robustness test, one may apply the small-gain theorem, though this method can be quite conservative if  $\Delta(s)$  has any structure. Recall that the return difference for the  $\Delta M$  model is given by  $I - \Delta M$  which can be directly analyzed using the  $A + B$  argument, i.e. if  $\det(I - \Delta M(s)) = 0$ , then

$$\underline{\sigma}(I) > \bar{\sigma}(\Delta M) > \bar{\sigma}(\Delta) \bar{\sigma}(M) \quad (16.107)$$

or using  $\underline{\sigma}(I) = 1$ , one has the **small-gain theorem (SGT)**:

$$\bar{\sigma}(\Delta) < \frac{1}{\bar{\sigma}(M)} \quad (16.108)$$

which is a sufficient test for stability, but may be quite conservative due to the use of the bound  $\bar{\sigma}(\Delta M) > \bar{\sigma}(\Delta) \bar{\sigma}(M)$ , which assumes the worst-case  $\Delta(s)$ .

For a less conservative stability robustness test, define the **structured singular value**  $\mu$  for some  $M \in \mathbb{C}^{n \times n}$  as

$$\mu_\Delta(M) = \begin{cases} \frac{1}{\min_{\Delta}(\bar{\sigma}(\Delta) \text{ s.t. } \det(I - \Delta M) = 0)} \\ 0, \text{ if no } \Delta \text{ makes } I - \Delta M \text{ singular} \end{cases} \quad (16.109)$$

where if  $\Delta$  is a full complex matrix, the SGT produces an accurate bound on the uncertainty with the **structured singular value (SSV)**  $\mu$  then given as

$$\mu_\Delta(M) = \frac{1}{\bar{\sigma}(\Delta)} = \bar{\sigma}(M) \quad (16.110)$$

Next, consider the simplest structure for  $\Delta(s)$  that is a diagonal matrix whose diagonal is a complex scalar  $\delta \in \mathbb{C}$ , i.e.

$$\Delta = \delta I_{n \times n} \quad (16.111)$$

Substituting this into the return difference for the  $\Delta M$  model, one has

$$I - \Delta M = I - \delta I_{n \times n} M = \delta \left( \frac{1}{\delta} I_{n \times n} - M \right) \quad (16.112)$$

which will be singular if  $\Delta$  destabilizes the system, i.e.

$$\left( \frac{1}{\delta} I_{n \times n} - M \right) \vec{w} = 0 \quad (16.113)$$

for some  $\vec{w}$  which defines an eigenvalue problem. Thus, for the simplest structure, one has

$$\mu_\Delta(M) = \bar{\rho}(M) \quad (16.114)$$

where  $\bar{\rho}(M)$  is the maximum **spectral radius** of the matrix,  $M$ , i.e. the largest absolute value of the eigenvalues of  $M$ . Thus, for problems of arbitrary structure, i.e. a block diagonal  $\Delta$ , the SSV  $\mu$  will be bounded above and below by

$$\bar{\rho}(M) \leq \mu_\Delta(M) \leq \bar{\sigma}(M) \quad (16.115)$$

Commercial software (e.g. MATLAB `mussv`) can be used to numerically compute the SSV  $\mu$  by

$$\max_Q \lambda(QM) \leq \mu_\Delta(M) \leq \inf_D \bar{\sigma}(DMD^{-1}) \quad (16.116)$$

which uses numerical methods for computing the SSV  $\mu$  bounds.

### Stability Robustness Example Problem

Given: the following plant LTI state-space model

$$A_p = \begin{bmatrix} -0.0251 & 0.10453 & -0.99452 \\ 574.70 & 0 & 0 \\ 16.2 & 0 & 0 \end{bmatrix}$$

$$B_p = \begin{bmatrix} 0.1228 & -0.27630 \\ -53.610 & 33.25 \\ 195.5 & -529.40 \end{bmatrix}$$

$$C_p = I_{3 \times 3}$$

$$D_p = 0_{2 \times 2} \quad (16.117)$$

and the following controller LTI state-space model

$$A_c = 0$$

$$B_{c,1} = [0 \ 0.9945 \ 0.1045]$$

$$B_{c,2} = -1$$

$$C_c = \begin{bmatrix} 0.7852 \\ -1.0409 \end{bmatrix} \quad (16.118)$$

$$D_{c,1} = \begin{bmatrix} 2.0536 & 0.0797 & -0.0458 \\ -3.8238 & -0.1280 & 0.1020 \end{bmatrix}$$

$$D_{c,2} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Determine: in MATLAB, the SSV  $\mu$  versus frequency for both

$$\Delta = \begin{bmatrix} \delta_1 & 0 \\ 0 & \delta_2 \end{bmatrix} \quad (16.119)$$

and unstructured  $\Delta$  (i.e. the SGT) as well as the elements of  $M$  versus frequency. Do this for

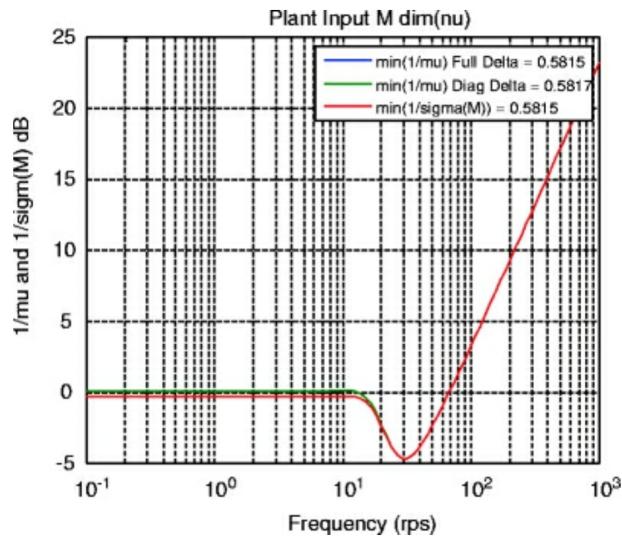
- a)  $\Delta$  enters at plant input
- b)  $\Delta$  enters at plant output
- c)  $\Delta$  enters at plant input and output

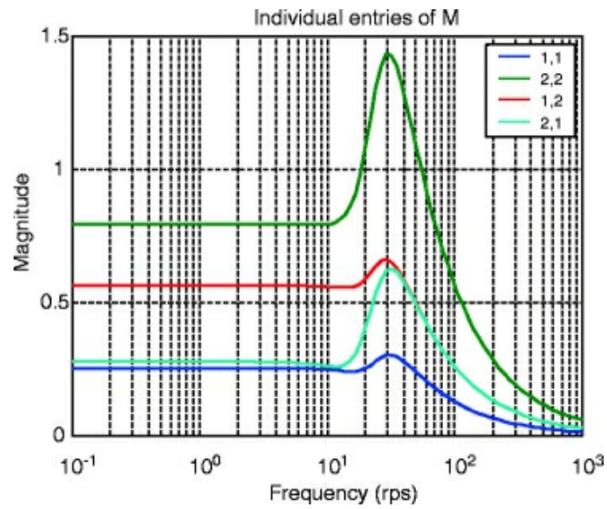
Solution:

For each

1. form the plant  $G(s)$  and controller  $K(s)$  using MATLAB's `ss` function;
2. use  $K(s)$  and  $G(s)$  to form the closed-loop system  $M(s)$  using MATLAB's `lft` function; and
3. for a range of frequencies, compute  $M(j\omega)$  using MATLAB's `freqresp`.

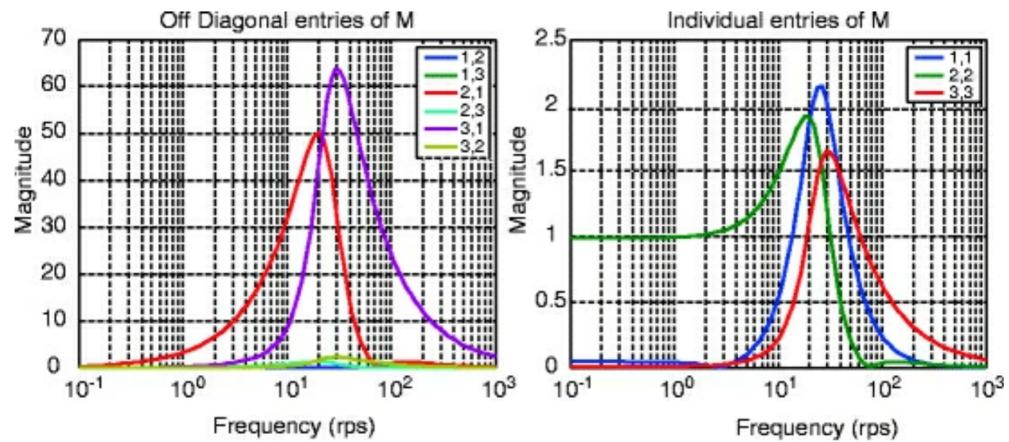
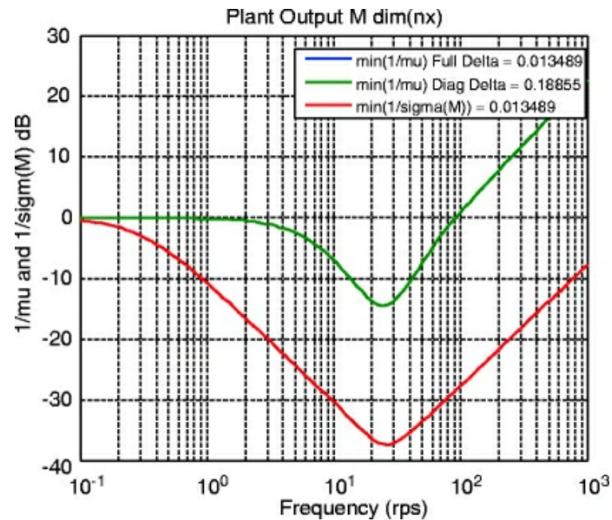
(a)





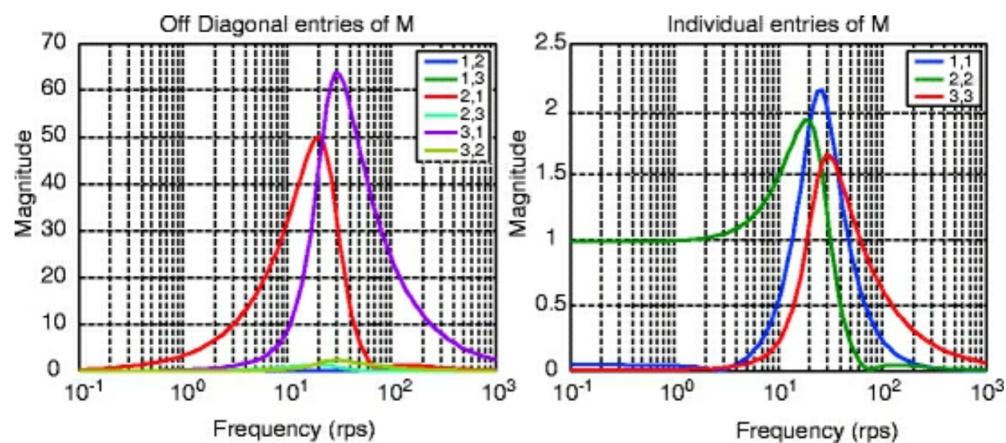
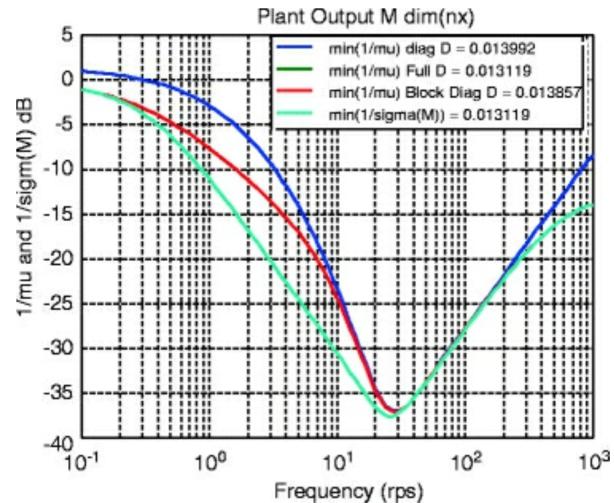
which shows that the  $2 \times 2$   $M$  matrix is dominated by the  $(2,2)$  element, i.e. the second element of  $\vec{z}$  drives second element of  $\vec{w}$ , thus diagonalizing  $\Delta$  doesn't alter the SSV  $\mu$  bound.

(b)



which shows that the  $3 \times 3 M$  matrix is dominated by an off-diagonal element, thus diagonalizing  $\Delta$  does greatly improve the SSV  $\mu$  bound.

(c)



which shows that the matrix is dominated by the (3,3) element, i.e. the third element of  $\vec{z}$  drives third element of  $\vec{w}$ , thus diagonalizing  $\Delta$  doesn't alter the SSV  $\mu$  bound.

# Chapter 17

## MIMO Loop-Shaping Robust Control

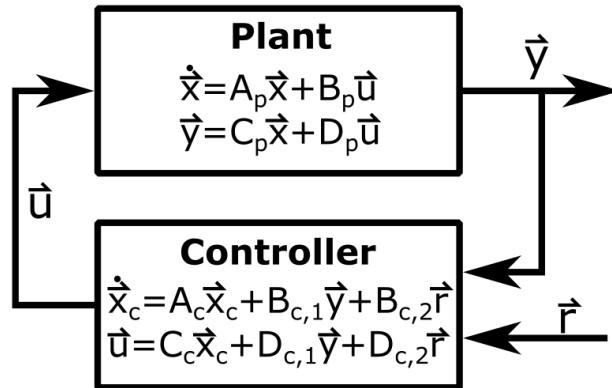
### 17.1 Introduction to Robust Optimal Control

Recall the general model for LTI state-space systems as

$$\begin{aligned}\dot{\vec{x}} &= A \vec{x} + B \vec{u} \\ \vec{y} &= C \vec{x} + D \vec{u}\end{aligned}\tag{17.1}$$

where  $\vec{u}$ ,  $\vec{y}$ , and  $\vec{x}$  are the input, output, and state vectors, respectively, and  $A$ ,  $B$ ,  $C$ , and  $D$  are the constant state, input, output, and feedthrough matrices, respectively.

With this definition, one may define two different LTI state-space models for state-space control design as shown in the following figure



The top system is the **plant model** written as

$$\begin{aligned}\dot{\vec{x}} &= A_p \vec{x} + B_p \vec{u} \\ \vec{y} &= C_p \vec{x} + D_p \vec{u}\end{aligned}\tag{17.2}$$

where  $\vec{x} \in \mathbb{R}^{n_x}$  is the plant state,  $\vec{u} \in \mathbb{R}^{n_u}$  is the control, and  $\vec{y} \in \mathbb{R}^{n_y}$  is the output. The bottom system is the **controller model** written as

$$\begin{aligned}\dot{\vec{x}}_c &= A_c \vec{x}_c + B_{c,1} \vec{y} + B_{c,2} \vec{r} \\ \vec{u} &= C_c \vec{x}_c + D_{c,1} \vec{y} + D_{c,2} \vec{r}\end{aligned}\quad (17.3)$$

where  $\vec{x}_c \in \mathbb{R}^{n_{xc}}$  is the controller state and  $\vec{r} \in \mathbb{R}^{n_r}$  is the possibly time-varying reference command.

Note that with this more general formulation, the static proportional state feedback controller, i.e.

$$\vec{u} = -K \vec{x} \quad (17.4)$$

would be constructed with  $C_p = I_{n_x \times n_x}$ ,  $D_p = 0_{n_x \times n_u}$ ,  $D_{c,1} = -K$ , and the rest of the controller matrices set to zero.

To analyze the closed-loop dynamics, one can connect the generic controller to the plant model, and then derive state-space models for the closed-loop system. To do so, first substitute the plant output equation into the control law

$$\vec{u} = C_c \vec{x}_c + D_{c,1} (C_p \vec{x} + D_p \vec{u}) + D_{c,2} \vec{r} \quad (17.5)$$

$$(I - D_{c,1} D_p) \vec{u} = C_c \vec{x}_c + D_{c,1} C_p \vec{x} + D_{c,2} \vec{r} \quad (17.6)$$

$$\vec{u} = Z^{-1} (C_c \vec{x}_c + D_{c,1} C_p \vec{x} + D_{c,2} \vec{r}) \quad (17.7)$$

where

$$Z = I - D_{c,1} D_p \quad (17.8)$$

must be assumed to be invertible to make the overall design problem well-posed. Next, substituting into the plant model yields

$$\dot{\vec{x}} = A_p \vec{x} + B_p Z^{-1} (C_c \vec{x}_c + D_{c,1} C_p \vec{x} + D_{c,2} \vec{r}) \quad (17.9)$$

$$\dot{\vec{x}} = \left( A_p + B_p Z^{-1} D_{c,1} C_p \right) \vec{x} + B_p Z^{-1} C_c \vec{x}_c + B_p Z^{-1} D_{c,2} \vec{r} \quad (17.10)$$

and one can also substitute the system output into the controller as

$$\dot{\vec{x}}_c = A_c \vec{x}_c + B_{c,1} (C_p \vec{x} + D_p \vec{u}) + B_{c,2} \vec{r} \quad (17.11)$$

$$\dot{\vec{x}}_c = A_c \vec{x}_c + B_{c,1} \left( C_p \vec{x} + D_p Z^{-1} (C_c \vec{x}_c + D_{c,1} C_p \vec{x} + D_{c,2} \vec{r}) \right) + B_{c,2} \vec{r} \quad (17.12)$$

$$\dot{\vec{x}}_c = \left( A_c + B_{c,1} D_p Z^{-1} C_p \right) \vec{x} + B_{cl} \left( I + D_p Z^{-1} D_{c,1} \right) C_p \vec{x} + \left( B_{cl} + B_{c,1} D_p Z^{-1} \right) \vec{r} \quad (17.13)$$

Then, combining these results, the closed-loop state equation is

$$\begin{bmatrix} \dot{\vec{x}} \\ \dot{\vec{x}}_c \end{bmatrix} = \begin{bmatrix} A_p + B_p Z^{-1} D_{c,1} C_p & B_p Z^{-1} C_c \\ B_{c,1} (I + D_p Z^{-1} D_{c,1}) C_p & A_c + B_{c,1} D_p Z^{-1} C_p \end{bmatrix} \begin{bmatrix} \vec{x} \\ \vec{x}_c \end{bmatrix} + \begin{bmatrix} B_p Z^{-1} D_{c,2} \\ B_{c,2} + B_{c,1} D_p Z^{-1} D_{c,2} \end{bmatrix} \vec{r} \quad (17.14)$$

Next, consider substituting the control equation into the plant output equation

$$\vec{y} = C_p \vec{x} + D_p Z^{-1} (C_c \vec{x}_c + D_{c,1} C_p \vec{x} + D_{c,2} \vec{r}) \quad (17.15)$$

$$\vec{y} = \begin{bmatrix} C_p + D_p Z^{-1} D_{c,1} C_p & D_p Z^{-1} C_c \end{bmatrix} \begin{bmatrix} \vec{x} \\ \vec{x}_c \end{bmatrix} + D_p Z^{-1} D_{c,2} \vec{r} \quad (17.16)$$

$$\vec{y} = [(I + D_p Z^{-1} D_{c,1}) C_p \quad D_p Z^{-1} C_c] \begin{bmatrix} \vec{x} \\ \vec{x}_c \end{bmatrix} + D_p Z^{-1} D_{c,2} \vec{r} \quad (17.17)$$

Finally, by defining the augmented state vector as

$$\vec{x}_a = \begin{bmatrix} \vec{x} \\ \vec{x}_c \end{bmatrix} \quad (17.18)$$

The closed-loop state matrix as

$$A_{cl} = \begin{bmatrix} A_p + B_p Z^{-1} D_{c,1} C_p & B_p Z^{-1} C_c \\ B_{c,1} (I + D_p Z^{-1} D_{c,1}) C_p & A_c + B_{c,1} D_p Z^{-1} C_c \end{bmatrix} \quad (17.19)$$

the closed-loop input matrix as

$$B_{cl} = \begin{bmatrix} B_p Z^{-1} D_{c,2} \\ B_{c,2} + B_{c,1} D_p Z^{-1} \end{bmatrix} \quad (17.20)$$

the closed-loop output matrix as

$$C_{cl} = [(I + D_p Z^{-1} D_{c,1}) C_p \quad D_p Z^{-1} C_c] \quad (17.21)$$

and the closed-loop feedforward matrix as

$$D_{cl} = D_p Z^{-1} D_{c,2} \quad (17.22)$$

one has the closed-loop state equation written succinctly as

$$\dot{\vec{x}}_a = A_{cl} \vec{x}_a + B_{cl} \vec{r} \quad (17.23)$$

and the closed-loop output equation as

$$\vec{y} = C_{cl} \vec{x}_a + D_{cl} \vec{r} \quad (17.24)$$

The loop gain model at the plant input is formed to support frequency domain analysis of the design at the plant input loop break point. In this model, the control input to the plant can be treated as the model input  $\vec{u}_{in}$  while the control output from the controller becomes the model output  $\vec{u}_{out}$ . For this analysis one can neglect the command vector  $\vec{r}$ . In this case, the plant model is

$$\begin{aligned} \dot{\vec{x}} &= A_p \vec{x} + B_p \vec{u}_{in} \\ \vec{y} &= C_p \vec{x} + D_p \vec{u}_{in} \end{aligned} \quad (17.25)$$

and the controller model is

$$\begin{aligned} \dot{\vec{x}}_c &= A_c \vec{x}_c + B_{c,1} \vec{y} \\ \vec{u}_{out} &= C_c \vec{x}_c + D_{c,1} \vec{y} \end{aligned} \quad (17.26)$$

Then connecting these two systems with  $\vec{u}_{in}$  as the input and  $\vec{u}_{out}$  as the output, one has

$$\begin{aligned} \dot{\vec{x}}_c &= A_c \vec{x}_c + B_{c,1} C_p \vec{x} + B_{c,1} D_p \vec{u}_{in} \\ \vec{u}_{out} &= C_c \vec{x}_c + D_{c,1} C_p \vec{x} + D_{c,1} D_p \vec{u}_{in} \end{aligned} \quad (17.27)$$

which can be rewritten in LTI state-space form as

$$\begin{aligned}\begin{bmatrix} \dot{\vec{x}} \\ \vec{x}_c \end{bmatrix} &= \begin{bmatrix} A_p & 0 \\ B_{c,1}C_p & A_c \end{bmatrix} \begin{bmatrix} \vec{x} \\ \vec{x}_c \end{bmatrix} + \begin{bmatrix} B_p \\ B_{c,1}D_p \end{bmatrix} \vec{u}_{in} \\ \vec{u}_{out} &= [D_{c,1}C_p \quad C_c] \begin{bmatrix} \vec{x} \\ \vec{x}_c \end{bmatrix} + D_{c,1}D_p \vec{u}_{in}\end{aligned}\quad (17.28)$$

or generally as

$$\begin{aligned}\dot{\vec{x}}_a &= A_{L,i} \vec{x}_a + B_{L,i} \vec{u}_{in} \\ \vec{u}_{out} &= C_{L,i} \vec{x}_a + D_{L,i} \vec{u}_{in}\end{aligned}\quad (17.29)$$

where the system's **loop gain at the plant input** is

$$L_i(s) = C_{L,i}(sI - A_{L,i})^{-1}B_{L,i} + D_{L,i} \quad (17.30)$$

Similarly, the loop gain model at the plant output is formed to support frequency domain analysis of the design at the plant output loop break point. In this model, the plant output fed to the controller can be treated as the model input  $\vec{y}_{in}$ , while the plant output from the plant becomes the model output  $\vec{y}_{out}$ . For this analysis one can also neglect the command vector  $\vec{r}$ . In this case, the plant model is

$$\begin{aligned}\dot{\vec{x}} &= A_p \vec{x} + B_p \vec{u} \\ \vec{y}_{out} &= C_p \vec{x} + D_p \vec{u}\end{aligned}\quad (17.31)$$

and the controller model is

$$\begin{aligned}\dot{\vec{x}}_c &= A_c \vec{x}_c + B_{c,1} \vec{y}_{in} \\ \vec{u}_{out} &= C_c \vec{x}_c + D_{c,1} \vec{y}_{in}\end{aligned}\quad (17.32)$$

Then connecting these two systems with  $\vec{y}_{in}$  as the input and  $\vec{y}_{out}$  as the output, one has

$$\begin{aligned}\dot{\vec{x}} &= A_p \vec{x} + B_p C_p \vec{x}_c + B_p D_{c,1} \vec{y}_{in} \\ \vec{y}_{out} &= C_p \vec{x} + D_p C_c \vec{x}_c + D_p D_{c,1} \vec{y}_{in}\end{aligned}\quad (17.33)$$

which can be rewritten in LTI state-space form as

$$\begin{aligned}\begin{bmatrix} \dot{\vec{x}} \\ \vec{x}_c \end{bmatrix} &= \begin{bmatrix} A_p & B_p C_c \\ 0 & A_c \end{bmatrix} \begin{bmatrix} \vec{x} \\ \vec{x}_c \end{bmatrix} + \begin{bmatrix} B_p D_{c,1} \\ B_{c,1} \end{bmatrix} \vec{y}_{in} \\ \vec{y}_{out} &= [C_p \quad D_p C_c] \begin{bmatrix} \vec{x} \\ \vec{x}_c \end{bmatrix} + D_p D_{c,1} \vec{y}_{in}\end{aligned}\quad (17.34)$$

or generally as

$$\begin{aligned}\dot{\vec{x}}_a &= A_{L,o} \vec{x}_a + B_{L,o} \vec{y}_{in} \\ \vec{y}_{out} &= C_{L,o} \vec{x}_a + D_{L,o} \vec{y}_{in}\end{aligned}\quad (17.35)$$

where the system's **loop gain at the plant output** is

$$L_o(s) = C_{L,o}(sI - A_{L,o})^{-1}B_{L,o} + D_{L,o} \quad (17.36)$$

These two derived loop gains will be essential tools for analyzing the relative stability properties of closed-loop LTI systems in the frequency domain.

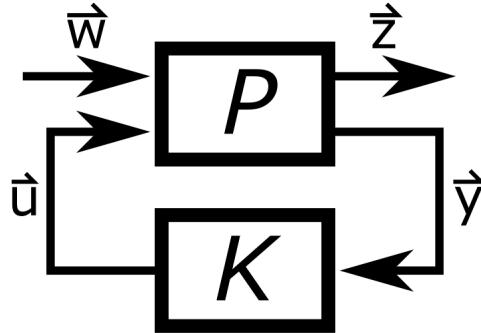
## 17.2 $H_\infty$ Optimal Control

In classical control, one can design the feedback control law,  $K(s)$ , by shaping the frequency response of the loop gain,  $L(j\omega)$ , to meet robust control system performance and robust stability requirements, i.e. **loop-shaping**. In the 1980s, this approach was extended to MIMO LTI systems and the resulting  $H_\infty$  optimal control and associated loop-shaping methods became the standard control design method for optimal solving for the feedback control law that meets robust MIMO system performance and robust stability requirements. For MIMO systems, these requirements can be related to shaping the sensitivity  $S(s)$ , the complementary sensitivity  $T(s)$ , the control effort  $U(s)$ , the loop gain  $L(s)$ , and the loop gain crossover frequency  $\omega_c$  (rad/s) to have certain characteristics. This lecture will discuss how this framework can be understood from a frequency domain perspective for MIMO systems with full state feedback.

In LTI  $H_\infty$  robust control, the LTI state-space form for the plant  $P$  is given as

$$\begin{aligned}\dot{\vec{x}} &= A\vec{x} + B\vec{u} + E\vec{w} \\ \vec{z} &= C\vec{x} + D_1\vec{u} + D_2\vec{w}\end{aligned}\tag{17.37}$$

where  $\vec{x} \in \mathbb{R}^{n_x}$  is the state vector,  $\vec{u} \in \mathbb{R}^{n_u}$  is the control input vector,  $\vec{w} \in \mathbb{R}^{n_w}$  is the external disturbance vector, and  $\vec{z} \in \mathbb{R}^{n_z}$  is the collection of signals to be regulated to zero. The design goal of  $H_\infty$  robust control is design the controller  $K$  to minimize  $\vec{z}$  in response to  $\vec{w}$  while providing system stability as shown below for **full state feedback** to the controller



This approach is equivalent to minimizing the  $\infty$ -norm of the transfer function matrix of the coupled feedback control system  $T_{z \leftarrow w}$ , i.e.

$$\frac{\|\vec{z}\|_2^2}{\|\vec{w}\|_2^2} \leq \|T_{z \leftarrow w}\|_\infty^2 \leq \gamma^2\tag{17.38}$$

### State Feedback $H_\infty$ Optimization

In the  $H_\infty$  optimization, one can solve for the arguments (i.e. the optimal control law and maximizing disturbance) that solves the min-max problem

$$\vec{u}^*, \vec{w}^* = \underset{\vec{u}}{\operatorname{argmin}} \underset{\vec{w}}{\operatorname{argmax}} J(\vec{u}, \vec{w})\tag{17.39}$$

where the cost function can be set as

$$J(\vec{u}, \vec{w}) = \frac{1}{2} \int_{t_0}^T \vec{z}^T \vec{z} - \gamma^2 \vec{w}^T \vec{w} d\tau \quad (17.40)$$

where  $\gamma \geq 0$  is chosen by the user,  $(t_0, \vec{x}_0)$  are given, and  $(T, \vec{x}(T))$  is free to vary in the optimization. Note that if  $T \rightarrow \infty$ , one has the infinite-horizon  $H_\infty$  problem.

The response  $\vec{z}$  can be related to the disturbance  $\vec{w}$  by

$$\|\vec{z}\|_2^2 \leq \|T_{z \leftarrow w}\|_\infty^2 \|\vec{w}\|_2^2 \quad (17.41)$$

Then, selecting  $\gamma \geq \|T_{z \leftarrow w}\|_\infty$ , one has

$$\|\vec{z}\|_2 - \gamma^2 \|\vec{w}\|_2 \leq 0 \quad (17.42)$$

which can be used to construct the cost function along with the  $H_\infty$  plant model to obtain

$$J(\vec{u}, \vec{w}) = \frac{1}{2} \int_{t_0}^T [C \vec{x} + D_1 \vec{u} + D_2 \vec{w}]^T [C \vec{x} + D_1 \vec{u} + D_2 \vec{w}] - \gamma^2 \vec{w}^T \vec{w} d\tau \quad (17.43)$$

$$\begin{aligned} J(\vec{u}, \vec{w}) &= \frac{1}{2} \int_{t_0}^T \vec{x}^T C^T C \vec{x} + 2 \vec{x}^T [C^T D_1 \quad C^T D_2] \begin{bmatrix} \vec{u} \\ \vec{w} \end{bmatrix} \\ &\quad + \begin{bmatrix} \vec{u} \\ \vec{w} \end{bmatrix}^T \begin{bmatrix} D_1^T D_1 & D_1^T D_2 \\ D_2^T D_1 & D_2^T D_2 - \gamma^2 I \end{bmatrix} \begin{bmatrix} \vec{u} \\ \vec{w} \end{bmatrix} d\tau \end{aligned} \quad (17.44)$$

Letting

$$S = [C^T D_1 \quad C^T D_2] \quad (17.45)$$

$$R = \begin{bmatrix} D_1^T D_1 & D_1^T D_2 \\ D_2^T D_1 & D_2^T D_2 - \gamma^2 I \end{bmatrix} \quad (17.46)$$

and

$$\tilde{u} = \begin{bmatrix} \vec{u} \\ \vec{w} \end{bmatrix} \quad (17.47)$$

one has

$$J(\vec{u}, \vec{w}) = \frac{1}{2} \int_{t_0}^T \vec{x}^T C^T C \vec{x} + 2 \vec{x}^T S \tilde{u} + \tilde{u}^T R \tilde{u} d\tau \quad (17.48)$$

which is a linear-quadratic regulator (LQR) problem that has a cross-term between the state and augmented control  $\tilde{u}$ . This infers the plant model as

$$\dot{\vec{x}} = A \vec{x} + \tilde{B} \tilde{u} \quad (17.49)$$

where

$$\tilde{B} = [B \quad E] \quad (17.50)$$

The Hamiltonian for this LQR problem is

$$H = \frac{1}{2} \left( \vec{x}^T C^T C \vec{x} + 2 \vec{x}^T S \tilde{u} + \tilde{u}^T R \tilde{u} \right) + \vec{p}^T (A \vec{x} + \tilde{B} \tilde{u}) \quad (17.51)$$

The necessary condition for optimal  $\tilde{u}$  is

$$\nabla H_{\tilde{u}} = 0 = R\tilde{u} + S^T \vec{x} + \tilde{B}^T \tilde{u} \quad (17.52)$$

Solving for the optimal  $\tilde{u}^*$  gives

$$\tilde{u}^* = -R^{-1}(S^T \vec{x} + \tilde{B}^T \vec{p}) \quad (17.53)$$

The differential equation for the costate is

$$\dot{\vec{p}} = -\nabla H_{\vec{x}} = -C^T C \vec{x} - A^T \vec{p} - S\tilde{u} \quad (17.54)$$

with  $\vec{p}(T) = 0$ . Then, by substitution and combination, one can write the Hamiltonian system as

$$\begin{bmatrix} \dot{\vec{x}} \\ \dot{\vec{p}} \end{bmatrix} = \begin{bmatrix} A & -\tilde{B}R^{-1}\tilde{B}^T \\ -C^T C + SR^{-1}S & -A^T + SR\tilde{B}^T \end{bmatrix} \begin{bmatrix} \vec{x} \\ \vec{p} \end{bmatrix} \quad (17.55)$$

The next step is to manipulate this first-order differential equation to eliminate the costate  $\vec{p}$  and create a Riccati equation whose solution will give the optimal augmented control. Assuming the state-transition matrix for the Hamiltonian system is

$$\Phi(T, t) = \begin{bmatrix} \phi_{xx}(T, t) & \phi_{xp}(T, t) \\ \phi_{px}(T, t) & \phi_{pp}(T, t) \end{bmatrix} \quad (17.56)$$

Then,  $\vec{p}(T) = \phi_{px} \vec{x} + \phi_{pp} \vec{p}$  which can be rewritten as

$$\vec{p} = \phi_{pp}^{-1} \phi_{px} \vec{x} = P \vec{x} \quad (17.57)$$

Differentiating provides

$$\dot{\vec{p}} = \dot{P} \vec{x} + P \dot{\vec{x}} \quad (17.58)$$

and from the Hamiltonian, one has

$$\dot{P} \vec{x} + P \dot{\vec{x}} = (-C^T C + SR^{-1}S) \vec{x} + (-A^T + SR\tilde{B}^T) \vec{p} \quad (17.59)$$

Substituting for  $\dot{\vec{x}}$ , replacing  $\vec{p}$ , and factoring out  $\vec{x}$  on the right results in

$$-\dot{P} = PA + A^T P + C^T C - (P\tilde{B} + S)R^{-1}(\tilde{B}^T P + S^T) \quad (17.60)$$

which is a type of Riccati equation and whose solution,  $P$  is used to form the state feedback control law as

$$\tilde{u} = -R^{-1}(B^T P + S^T) \vec{x} \quad (17.61)$$

For  $T \rightarrow \infty$ , this is an algebraic Riccati equation (ARE) and is used in the majority of applications.

For this problem to be well-posed, one must assume that for the LTI plant state-space model  $(A, B, C, D_1)$  there are no zeros on the  $j\omega$  axis,  $(A, B)$  is stabilizable, and  $(D_1^T D_1)^{-1}$  exists (i.e. injective). Then, there exists a state feedback control  $\vec{u} = -K_\infty \vec{x}$  such that the closed-loop system is internally stable and  $\|T_{z \leftarrow w}\|_\infty < \gamma$ .  $D_2^T D_2 < \gamma^2 I$  and there exists a  $P \geq 0$  that solves the following ARE:

$$PA + A^T P + C^T C - \begin{bmatrix} B^T P + D_1^T C \\ E^T P + D_2^T C \end{bmatrix}^T \begin{bmatrix} D_1^T D_1 & D_1^T D_2 \\ D_2^T D_1 & D_2^T D_2 - \gamma^2 I \end{bmatrix} \begin{bmatrix} B^T P + D_1^T C \\ E^T P + D_2^T C \end{bmatrix} = 0 \quad (17.62)$$

and the  $H_\infty$  optimal control  $\vec{u}$  is

$$\begin{aligned}\vec{u} &= \begin{bmatrix} I_{n_u \times n_u} & 0 \end{bmatrix} \tilde{u} \\ &= -\begin{bmatrix} I_{n_u \times n_u} & 0 \end{bmatrix} R^{-1} (B^T P + S^T) \vec{x} \\ &= -K_\infty \vec{x}\end{aligned}\quad (17.63)$$

However, choosing a  $\gamma$  to solve the  $H_\infty$  optimization is still required and is typically done through  **$\gamma$ -iteration**, i.e. a bisection search as

1. Initialize  $\gamma$  larger than the anticipated optimal  $\gamma$  for binary search
  - Form LQR cost matrices using  $\gamma$
  - Solve algebraic Riccati equation (ARE) for matrix  $P$
  - If  $P > 0$  and  $\text{Re}(\lambda(A - BK_\infty)) < 0$ :
    - Decrease  $\gamma$  by bisection (until convergence threshold)
  - Else:
    - Increase  $\gamma$  by bisection
2. Convergence to  $\gamma_{min}$  to form  $K_\infty$

Note that care must be taken as  $\gamma$  approaches  $\gamma_{min}$  as  $R$  typically becomes ill-conditioned. It is also typically prudent to slightly increase  $\gamma$  from  $\gamma_{min}$  to reduce feedback gain magnitudes and improve the accuracy of the numerical solver for the ARE.

Lastly, for general MIMO system  $H_\infty$  control, the weighting filters

$$\begin{aligned}\dot{\vec{x}}_S &= A_S \vec{x}_S + B_S (\vec{y} - \vec{r}) \\ \vec{z}_1 &= C_S \vec{x}_S + D_S (\vec{y} - \vec{r})\end{aligned}\quad (17.64)$$

$$\begin{aligned}\dot{\vec{x}}_T &= A_T \vec{x}_T + B_T \vec{y} \\ \vec{z}_2 &= C_T \vec{x}_T + D_T \vec{y}\end{aligned}\quad (17.65)$$

Also note that typically the control-regulating filter must be chosen to make sure that  $D_1$  is injective, i.e.  $(D_1^T D_1)^{-1}$  exists.

### 17.3 Mixed-Sensitivity $H_\infty$ Loop-Shaping

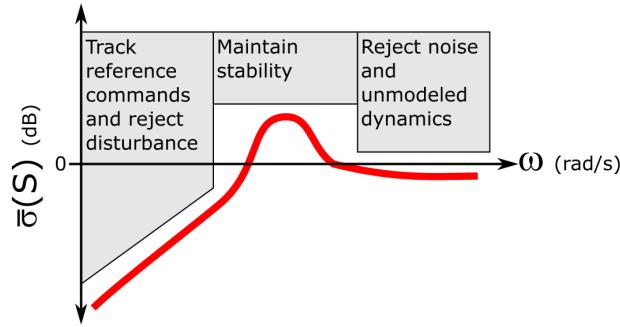
In classical control, one can design the feedback control law,  $K(s)$ , by shaping the frequency response of the loop gain,  $L(j\omega)$ , to meet robust control system performance and robust stability requirements, i.e. **loop-shaping**. In the 1980s, this approach was extended to MIMO LTI systems and the resulting  $H_\infty$  optimal control and associated loop-shaping methods became the standard control design method for optimal solving for the feedback control law that meets robust MIMO system performance and robust stability requirements. For MIMO systems, these requirements can be related to shaping the sensitivity  $S(s)$ , the complementary sensitivity  $T(s)$ , the control effort  $U(s)$ , the loop gain  $L(s)$ , and the loop gain crossover frequency  $\omega_c$  (rad/s)

to have certain characteristics. This lecture will discuss how this framework can be understood from a frequency domain perspective for MIMO systems with full state feedback.

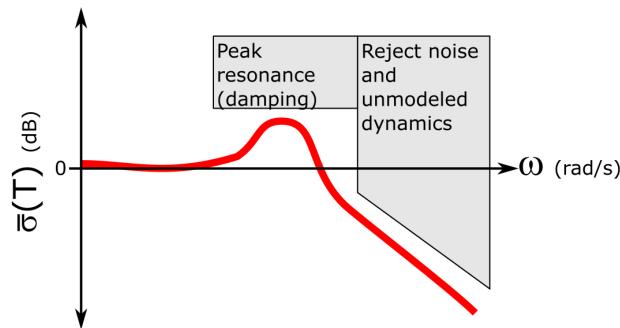
One method to set up the general LTI feedback control system in this way is to use  $\mathbf{H}_\infty$  loop-shaping to set up the input-output relationships between  $\vec{w}$  and  $\vec{z}$  to have certain MIMO frequency domain characteristics.

For SISO systems,  $L(s)$  is a scalar with singular value  $\sigma(L) = \bar{\sigma}(L) = |L|$  and the loop gain is shaped using *control stages* to set the crossover frequency, increase the gain at low frequencies, decrease the gain at high frequencies, and reduce the slope of the crossover region to achieve good classical stability margins.

For MIMO systems, the singular value stability margins are computed from the sensitivity  $S(s) = (I + L(s))^{-1}$  and complementary sensitivity  $T(s) = (I + L(s))^{-1}L(s)$ .

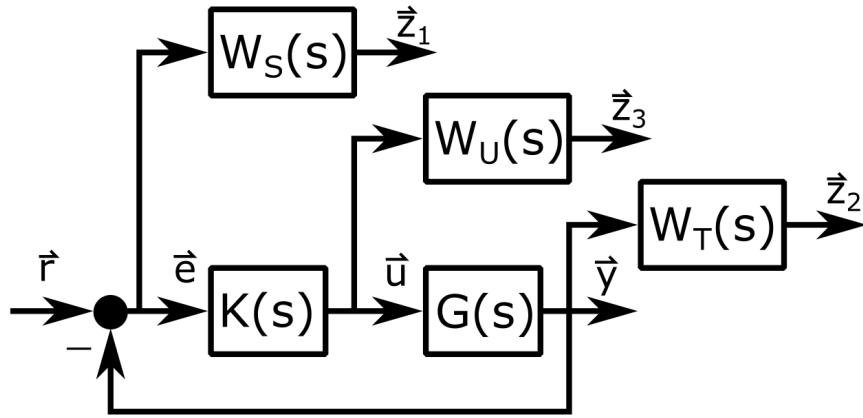


and

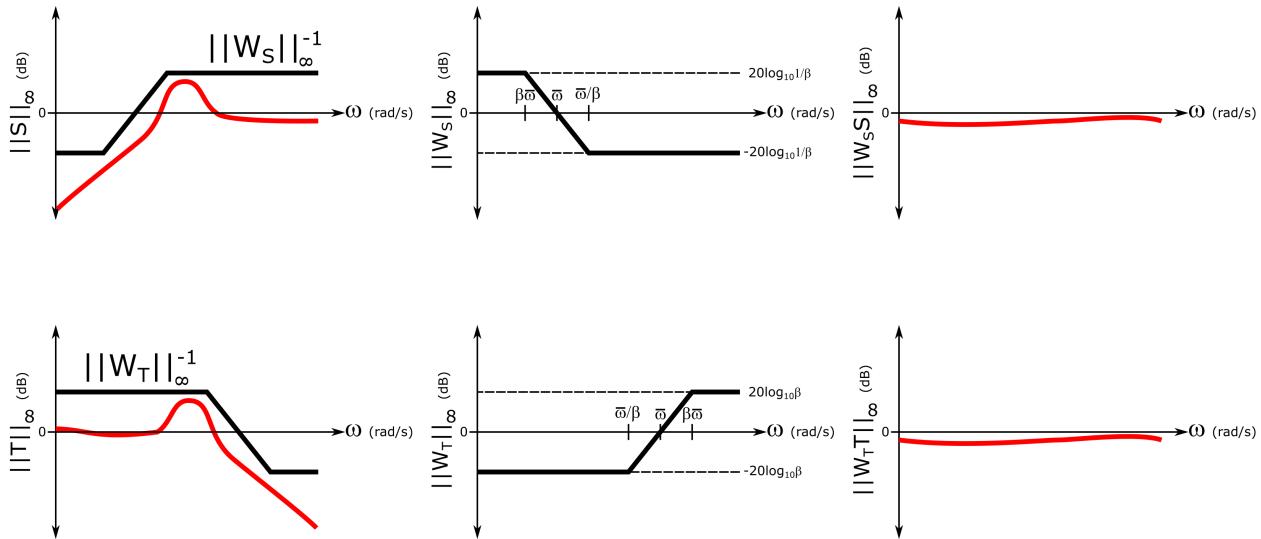


At low frequency, where  $L(s)$  needs to be large,  $S(s)$  needs to be small as  $S(s)$  is also the transfer function for the tracking error and  $T(s)$  is near unity or 0 dB. At high frequencies where  $L(s)$  needs to be small,  $S(s)$  is near unity or 0 dB and  $T(s)$  needs to roll off to reject high-frequency noise and unmodeled dynamics. Recall that the near singularity of the return difference  $I + L(s) = S^{-1}(s)$  is measured by the minimum of  $\underline{\sigma}(I + L(j\omega))$  across all frequencies which equates to the maximum or peak value of  $S(j\omega)$ , i.e.  $\|S\|_\infty$ . At the peak of  $T(s)$ ,  $\|T\|_\infty$  denotes the closed-loop system's peak resonance at the frequency for which inputs are most amplified. Thus  $\|T\|_\infty$  is analogous to the idea of damping for second-order systems, i.e. the dominant poles are close to the  $j\omega$ -axis.

In  $H_\infty$  robust control, one can set up an optimization so that the optimal control law achieves a performance requirement on the  $\infty$ -norm of the transfer function matrix which is a balancing act between the different individual transfer functions which produce the output of the plant,  $\vec{z}$ . To accomplish this transfer function balancing for the general MIMO feedback control system, one may consider directly weighting the different signals that correspond to  $S(s)$ ,  $U(s)$ , and  $T(s)$  requirements in order to set up an optimization routine such as used in  $H_\infty$  optimal control, a process called **mixed-sensitivity  $H_\infty$  loop-shaping**. These weights can be designed as weight filters which weigh different signals in the feedback control system and output them as vectors,  $\vec{z}_1$ ,  $\vec{z}_2$ , and  $\vec{z}_3$ .



With this frequency-dependent weights framework, the optimization routine will attempt to minimize  $\|W_S S\|_\infty$ ,  $\|W_T T\|_\infty$ , and  $\|W_U U\|_\infty$ , as demonstrated in the following diagrams.



and as one typically desires to minimize the control effort,  $\|U\|$ , across all frequencies to ensure that any

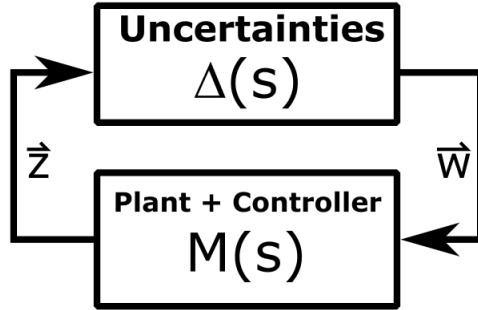
actuators are not position or rate saturated. Thus, one typically sets a constant weight on  $W_U$ . Note that here the selection of the  $W_S$  and  $W_T$  weighting filters are first order lag and lead transfer functions, i.e.

$$W(s) = \frac{\beta s + \bar{\omega}}{s + \beta \bar{\omega}} \quad (17.66)$$

where for  $0 < \beta < 1$ , this corresponds to a lag control (e.g.  $W_S$ ) and  $\beta > 1$  corresponds to a lead control (e.g.  $W_T$ ). Thus, the degree of difficulty in choosing these weighting filters can be comparable to choosing the lead-lag filters in classical control design for improving the gain or phase margins of SISO feedback control systems. Typically design iteration with the weighting filters is used in conjunction with the  $H_\infty$  optimization to perform  $H_\infty$  loop-shaping control design.

## 17.4 $\mu$ -Synthesis Robust Control

Previous sections considered the effects of structured and unstructured uncertainties in a generalized  $\Delta M$  framework which has the following block diagram



where the structured singular value (SSV)  $\mu_\Delta$  can be used to compute an upper and lower bound for the inverse of the minimum possible  $\bar{\sigma}\Delta$  which causes the  $\Delta M$  model to becomes unstable. This is related to the maximum singular value of  $M$  by

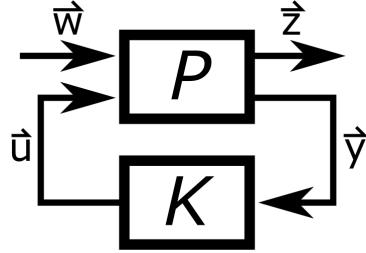
$$\bar{\rho}(M) \leq \mu_\Delta(M) \leq \bar{\sigma}(M) \quad (17.67)$$

which is numerically approximated by

$$\max_Q \lambda(QM) \leq \mu_\Delta(M) \leq \inf_D \bar{\sigma}(DMD^{-1}) \quad (17.68)$$

where the frequency-dependent  $D$  matrices, which commute with  $\Delta$  (i.e.  $D\Delta = \Delta D$ ), are called the  **$D$  scalings**.

The previous section also considered the general  $H_\infty$  control framework which has the following block diagram



where the optimal control problem (OCP) was to find

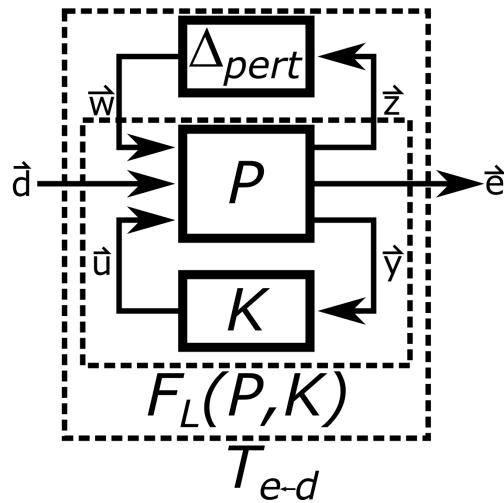
$$\gamma^* = \min_{K \text{ stabilizing}} \|T_{z \leftarrow w}\|_\infty \quad (17.69)$$

which is typically approximated by  $\gamma$ -iteration, i.e. given a  $\gamma > 0$ , find

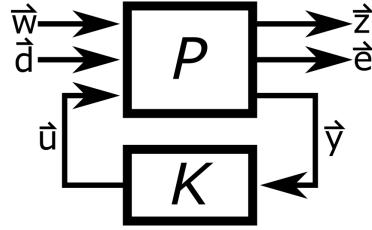
$$K_\infty = \operatorname{argmin}_{K \text{ stabilizing}} \|T_{z \leftarrow w}\|_\infty < \gamma \quad (17.70)$$

or determine that no stabilizing  $K$  exists and iterate over a bisection search for  $\gamma_{\min}$ . Note that this is slightly different version of the  $H_\infty$  OCP which uses  $K_\infty$  explicitly instead of  $\vec{u}^*$ .

This lecture will discuss the combination of these two robust performance and robust stability control frameworks which has the general block diagram



where  $\vec{d}$  and  $\vec{e}$  are now introduced as the disturbance input and error output vectors,  $\vec{w}$  and  $\vec{z}$  are the perturbation input and output vectors to some plant perturbation uncertainty,  $\Delta_{pert}$ , and  $\vec{u}$  and  $\vec{y}$  are the control input to the plant and plant output to the controller vectors as previously. Note that the lower transfer function  $F_L(P, K)$  is the equivalent to  $M$  in the  $\Delta M$  robust analysis model. Here one can then set up the model  $F_L(P, K)$  as



where one has for the augmented uncertainty block matrix,  $\Delta$ , in the  $\Delta M$  robust analysis

$$\Delta = \begin{bmatrix} \Delta_{pert} & 0 \\ 0 & \Delta_F \end{bmatrix} \quad (17.71)$$

which takes into account both input and output vectors in the  $\mu$ -synthesis model as  $\vec{e} = \Delta_F \vec{d}$ . Thus, the  **$\mu$ -synthesis** OCP attempts to minimize over all stabilizing controllers,  $K_\mu$ , the peak value of  $\mu_\Delta$  of the closed-loop transfer function, i.e.

$$K_\mu = \underset{K \text{ stabilizing}}{\operatorname{argmin}} \max_{\omega} \mu_\Delta (F_L(P, K)(j\omega)) \quad (17.72)$$

Note that  $F_L(P, K)$  is a type of **linear fractional transformation** (LFT) which is a generalized feedback interconnection of two models. MATLAB computes these types of system connections using the command `lft()`.

### D-K Iteration

To derive a tractable approximation for this  $\mu$ -synthesis optimization known as  $D - K$  iteration, recall the upper bound definition for  $\mu_\Delta$ , one can rewrite this minimization of the maximum possible  $\mu_\Delta$  as

$$K_\mu = \underset{K \text{ stabilizing}}{\operatorname{argmin}} \max_{\omega} \min_{D_\omega} \bar{\sigma} \left( D_\omega F_L(P, K)(j\omega) D_\omega^{-1} \right) \quad (17.73)$$

where  $D_\omega$  is chosen from all possible scalings independently as every  $\omega$ . Then, rearranging, one has

$$K_\mu = \underset{K \text{ stabilizing}}{\operatorname{argmin}} \min_{D_\omega} \max_{\omega} \bar{\sigma} \left( D_\omega F_L(P, K)(j\omega) D_\omega^{-1} \right) \quad (17.74)$$

or

$$K_\mu = \underset{K \text{ stabilizing}}{\operatorname{argmin}} \min_{D_\omega} \|DF_L(P, K)(j\omega)D_\omega^{-1}\|_\infty \quad (17.75)$$

Thus, one can see that this optimization can be constructed as minimizing two different parameters,  $D$  and  $K$ , which is done as an iterative procedure, i.e. holding  $D$  as fixed and finding the optimal  $K$  using  $H_\infty$ , then holding  $K$  fixed and finding the optimal  $D$  that minimizes the transformed  $\infty$ -norm. Note that this  $D$  procedure serves as an approximation for maximizing the  $\mu_\Delta$  upper bound, i.e. maximizing the “distance” to singularity/instability of  $F_L(P, K)$  for the unstructured uncertainty  $\Delta$  (i.e. fully complex block matrix as defined previously). This approximation is typically very close. However, it should also be noted that  $D - K$

iteration is not guaranteed to converge to a global or even a local minimum which is a serious shortcoming of this approach, but has been shown to work well in many different engineering problems including highly flexible airplanes, missile autopilots, modern fighter airplanes, and the space shuttle flight control system.

An example of using  $\mu$ -synthesis using D-K iteration for robust control of the lateral-directional axis of an airplane in MATLAB is `openExample('robust/MuSynthesisAircraftExample')`.

# Chapter 18

## Introductory Adaptive Control

### 18.1 Introduction to Adaptive Control

Flight control systems must provide closed-loop stability, adequate reference command tracking and robustness to model uncertainties, system failures, and environmental disturbances. The previous robust control techniques discussed allow one to accomplish these requirements through different loop-shaping approaches to setting up an  $H_\infty$  optimization that allows the computed controller to balance these requirements as well as assess the robust stability and performance of these controllers in the presence of uncertainties.

However, robust control law are not designed to handle any specific uncertainty in the system. Thus, adaptive control laws are an alternative control strategy that can be used when the uncertainties appear only where the control inputs exist in the system dynamics, i.e. if the system uncertainties are known, the controller could cancel out the uncertainty, also known as **matched uncertainties**. Adaptive control laws alter the control law so that a desired closed-loop performance is maintained while operating under matched uncertainties. This course will consider a reference model for specifying the desired closed-loop performance.

The adaptive control development was largely motivated for aircraft that operate in a wide flight envelope with large range of speeds and altitudes defining flight conditions which dramatically change the aircraft aerodynamic and propulsive forces and moments which are typically modeled as stability and control derivatives. Adaptive control was proposed as one control strategy to solve this problem. This led to the control concept known as **explicit model-following** which allows one to specify the desired command-to-output performance of a tracking system using a differential/difference equation that defines the ideal response, i.e. the **reference model**. The control system architecture corresponding to this model-following is known as **model reference adaptive control (MRAC)**.

As aerodynamic stability and control derivatives are rarely exactly known, aircraft typically have **parametric uncertainty** as the parameters that are used in the nominal control system design vary from the assumed constant values. This lecture will provide an example of a SISO MRAC system for an airplane with parametric uncertainty in its roll dynamics and present some general conclusions from the example.

## MRAC Example for Roll Dynamics

Suppose the airplane roll dynamics can be approximated by a scalar LTI ODE as

$$\dot{p} = L_p p + L_{\delta_a} \delta_a \quad (18.1)$$

where  $p$  is the roll rate in the stability frame,  $\delta_a$  is the aileron deflection,  $L_p$  is the roll damping derivative, and  $L_{\delta_a}$  is the rolling moment derivative due to a differential aileron deflection. For conventional open-loop-stable airplane,  $L_p$  is negative and  $L_{\delta_a}$  is positive. Then, consider that the two aerodynamic stability and control derivatives,  $L_p$  and  $L_{\delta_a}$ , in the roll dynamics are constant but otherwise completely unknown, with the exception that one does know the *sign* of the aileron control derivative  $L_{\delta_a}$  is positive. The model reference control problem now is to force the airplane to roll like some reference model, i.e.

$$\dot{p}_{ref} = a_{ref} p_{ref} + b_{ref} p_c \quad (18.2)$$

which correspond to some prescribed reference values of  $a_{ref} < 0$  and  $b_{ref}$  by finding a control law for  $\delta_a$  such that  $p$  tracks the reference roll rate  $p_{ref}$  which itself is driven by a bounded, but possibly time-varying reference command,  $p_c$ . Note that if one knew the roll dynamics model, then a feedback-feedforward control law as

$$\delta_a = k_p p + k_{p_c} p_c \quad (18.3)$$

is one possible control law which would allow one to match any reference model, i.e.

$$\dot{p} = (L_p + L_{\delta_a} k_p) p + L_{\delta_a} k_{p_c} p_c \quad (18.4)$$

with choosing  $a_{ref} = L_p + L_{\delta_a} k_p$  and  $b_{ref} = L_{\delta_a} k_{p_c}$ . However, since the system parameters are unknown, the ideal control law gains,  $k_p$  and  $k_{p_c}$  to accomplish this cannot be computed directly. Instead, consider an adaptive control law in the form

$$\delta_a = \hat{k}_p p + \hat{k}_{p_c} p_c \quad (18.5)$$

where  $\hat{k}_p$  and  $\hat{k}_{p_c}$  represent the *estimated* feedback and feedforward control gains, respectively. Substituting this control law gives the closed-loop system

$$\dot{p} = (L_p + L_{\delta_a} \hat{k}_p) p + L_{\delta_a} \hat{k}_{p_c} p_c \quad (18.6)$$

Next, the reference model can still be regarded as some

$$\dot{p}_{ref} = (L_p + L_{\delta_a} k_p) p_{ref} + L_{\delta_a} k_{p_c} p_c \quad (18.7)$$

where  $a_{ref} = L_p + L_{\delta_a} k_p$  and  $b_{ref} = L_{\delta_a} k_{p_c}$ . Then, the gain estimation errors can be defined as

$$\begin{aligned} \Delta k_p &= \hat{k}_p - k_p \\ \Delta k_{p_c} &= \hat{k}_{p_c} - k_{p_c} \end{aligned} \quad (18.8)$$

and the closed-loop system can be written as

$$\dot{p} = (L_p + L_{\delta_a} k_p) p + L_{\delta_a} k_{p_c} p_c + L_{\delta_a} (\Delta k_p p + \Delta k_{p_c} p_c) \quad (18.9)$$

Finally, one can write the tracking error dynamics,  $e = p - p_{ref}$ , as

$$\dot{e} = a_{ref}e + L_{\delta_a}(\Delta k_p p + \Delta k_{p_c} p_c) \quad (18.10)$$

where  $a_{ref}$  has been chosen by the user.

In this construction, one has three error signals which should be driven to zero by the adaptive control law for  $\hat{k}_p$  and  $\hat{k}_{p_c}$ , in both a global and asymptotic sense, i.e. for any initial conditions and as  $t \rightarrow \infty$ . To prove this, one must use a Lyapunov function candidate  $V$  which represents the total “kinetic energy” of all the errors in the system, i.e.

$$V(e, \Delta k_p, \Delta k_{p_c}) = \frac{e^2}{2} + \frac{|L_{\delta_a}|}{2\gamma_p} \Delta k_p^2 + \frac{|L_{\delta_a}|}{2\gamma_{p_c}} \Delta k_{p_c}^2 \quad (18.11)$$

where the constant scalar weights  $\gamma_p > 0$  and  $\gamma_{p_c} > 0$  will be shown to be the rates of adaptation for the control gains,  $\hat{k}_p$  and  $\hat{k}_{p_c}$ , respectively. Note that this energy function is a weighted sum of squares of all the errors in the system and, by construction,  $V \geq 0$ . Consequently the system “power” can be written as

$$\dot{V}(e, \Delta k_p, \Delta k_{p_c}) = e\dot{e} + \frac{|L_{\delta_a}|}{\gamma_p} \Delta k_p \hat{k}_p + \frac{|L_{\delta_a}|}{\gamma_{p_c}} \Delta k_{p_c} \hat{k}_{p_c} \quad (18.12)$$

However, substituting the closed-loop dynamics of the system, i.e. evaluating the power “along” the trajectories of the system, one has

$$\dot{V}(e, \Delta k_p, \Delta k_{p_c}) = a_{ref}e^2 + eL_{\delta_a}(\Delta k_p p + \Delta k_{p_c} p_c) + \frac{|L_{\delta_a}|}{\gamma_p} \Delta k_p \hat{k}_p + \frac{|L_{\delta_a}|}{\gamma_{p_c}} \Delta k_{p_c} \hat{k}_{p_c} \quad (18.13)$$

Rearranging, one has

$$\dot{V}(e, \Delta k_p, \Delta k_{p_c}) = a_{ref}e^2 + \Delta k_p |L_{\delta_a}| \left( \text{sign}(L_{\delta_a}) p e + \frac{\dot{\hat{k}}_p}{\gamma_p} \right) + \Delta k_{p_c} |L_{\delta_a}| \left( \text{sign}(L_{\delta_a}) p_c e + \frac{\dot{\hat{k}}_{p_c}}{\gamma_{p_c}} \right) \quad (18.14)$$

Naturally, for the three errors to reach zero one desires the system energy to dissipate as  $t \rightarrow \infty$ . Thus, the system power should be non-positive, when evaluated along the system trajectories, i.e. the system equations of motion. The nonpositivity of  $\dot{V}$  can be achieved if one chooses

$$\begin{aligned} \dot{\hat{k}}_p &= -\gamma_p p e \text{ sign}(L_{\delta_a}) \\ \dot{\hat{k}}_{p_c} &= -\gamma_{p_c} p_c e \text{ sign}(L_{\delta_a}) \end{aligned} \quad (18.15)$$

or if assumes  $\text{sign}(L_{\delta_a})$  is positive, one has

$$\begin{aligned} \dot{\hat{k}}_p &= -\gamma_p p e \\ \dot{\hat{k}}_{p_c} &= -\gamma_{p_c} p_c e \end{aligned} \quad (18.16)$$

Thus, one has

$$\dot{V}(e, \Delta k_p, \Delta k_{p_c}) = a_{ref}e^2 \leq 0 \quad (18.17)$$

as  $a_{ref} < 0$  by construction for a stable (by design) reference model. Thus, as  $V$  is a non-increasing function of time, the three error signals in the closed-loop are bounded functions of time including  $e$ . For bounded reference commands  $p_c, p_{ref}$  is also bounded, thus  $p$  is bounded and consequently  $\delta_a$  is bounded and  $\dot{p}$  is bounded. Furthermore, as  $k_p$  and  $k_{p_c}$  are constant,  $\hat{k}_p$  and  $\hat{k}_{p_c}$  are bounded and since  $\dot{p}_{ref}$  is bounded, then  $\dot{e}$  is bounded.

Thus, from these bounded errors, one has

$$\ddot{V}(e, \Delta k_p, \Delta k_{p_c}) = 2a_{ref}e\dot{e} \quad (18.18)$$

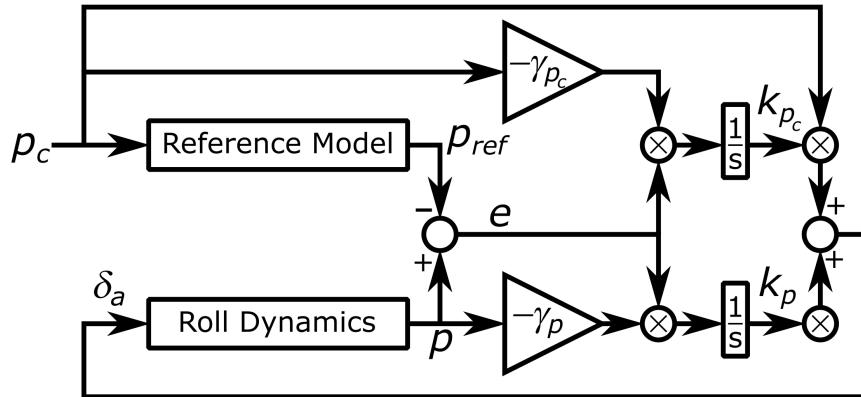
is a uniformly bounded function of time which implies that  $\dot{V}$  is a uniformly continuous function of time. Lastly, as  $V \geq 0$  and  $\dot{V} \leq 0$ ,  $V$  tends to a limit as  $t \rightarrow \infty$  which may not be zero which can be written mathematically as

$$0 \leq \lim_{t \rightarrow \infty} V(e(t), \Delta k_p(t), \Delta k_{p_c}(t)) < \infty \quad (18.19)$$

then according to Barbalat's lemma,  $\dot{V} \rightarrow 0$  as  $t \rightarrow \infty$  and

$$\lim_{t \rightarrow \infty} e(t) = 0 \quad (18.20)$$

Thus, the adaptive control law forces  $p$  to track  $p_{ref}$  globally and asymptotically. At the same time, all signals remain uniformly bounded. These prove closed-loop stability and tracking performance of the closed-loop adaptive control system which can be depicted as a block diagram



From this figure, one can see that the control system design utilizes the reference model dynamics along with the original dynamics which adapts the control gains,  $k_p$  and  $k_{p_c}$ , through the integration by some chosen rates of adaptation  $\gamma_p$  and  $\gamma_{p_c}$ . The external input to the control system is the roll rate command  $p_c$  and one has for the closed-loop dynamics

$$\begin{aligned} \dot{p} &= (L_p + L_{\delta_a} \hat{k}_p)p + L_{\delta_a} \hat{k}_{p_c} p_c \\ \dot{p}_{ref} &= a_{ref} p_{ref} + b_{ref} p_c \\ \dot{k}_p &= -\gamma_p p(p - p_{ref}) \\ \dot{\hat{k}}_{p_c} &= -\gamma_{p_c} p_c(p - p_{ref}) \end{aligned} \quad (18.21)$$

which can also be written as closed-loop error dynamics

$$\begin{aligned}\dot{e} &= (a_{ref} + L_{\delta_a} \Delta k_p) e + L_{\delta_a} (\Delta k_p p_{ref} + \Delta k_{p_c} p_c) \\ \frac{d}{dt} \Delta k_p &= -\gamma_p (e + p_{ref}) e \\ \frac{d}{dt} \Delta k_{p_c} &= -\gamma_{p_c} p_c e\end{aligned}\tag{18.22}$$

Note that if state regulation is of interest (i.e.  $x \rightarrow 0$ ), then  $p_{ref} = p_c = 0$  and  $\hat{k}_{p_c} = k_{p_c} = 0$ . In this case, the closed-loop system simplifies to a second-order nonlinear ODE as

$$\begin{aligned}\dot{p} &= (L_p + L_{\delta_a} \hat{k}_p) p \\ \dot{\hat{k}}_p &= -\gamma_p p^2\end{aligned}\tag{18.23}$$

Thus, the time-varying adaptive feedback gain  $\hat{k}_p(t)$  will monotonically decrease its value until  $L_p + L_{\delta_a} \hat{k}_p < 0$ . As a result, the roll rate  $p(t) \rightarrow 0$  as  $t \rightarrow \infty$ . The magnitude of the constant rate of adaptation,  $\gamma_p > 0$ , defines the rate at which the adaptive gain decreases. As shown, the derivation of the properties of this system inherently use Lyapunov function or “energy-based” arguments which are a standard tool for nonlinear control systems which are sometimes necessary for advanced flight vehicles. The subsequent lecture will provide an overview for the closed-loop stability of MRAC systems using Lyapunov stability theory.

## MRAC for SISO LTI Systems

To summarize the results of this lecture for SISO systems consider the first-order LTI uncertain system

$$\dot{x} = ax + bu\tag{18.24}$$

where  $x$  is the state,  $u$  is the control input, and  $(a, b)$  are uncertain parameters (constant and unknown) with known sign( $b$ ).

In MRAC, first, one must choose the desired reference model

$$\dot{x}_{ref} = a_{ref} x_{ref} + b_{ref} r\tag{18.25}$$

with  $a_{ref} < 0$  and  $r$  is the reference command. Here the reference parameters are chosen so  $x_{ref}$  tracks  $r$  with some design criteria. For example,  $b_{ref} = -a_{ref}$  for unity gain for a step input from  $r$  to  $x_{ref}$  and  $a_{ref} = 1/\tau_{des}$  chosen such that a desired time constant,  $\tau_{des}$ , is achieved.

Second, one defines the model reference adaptive control law as

$$u = \hat{k}_x x + \hat{k}_r r\tag{18.26}$$

where  $\hat{k}_x$  and  $\hat{k}_r$  are two adaptive gains and follow differential equation laws

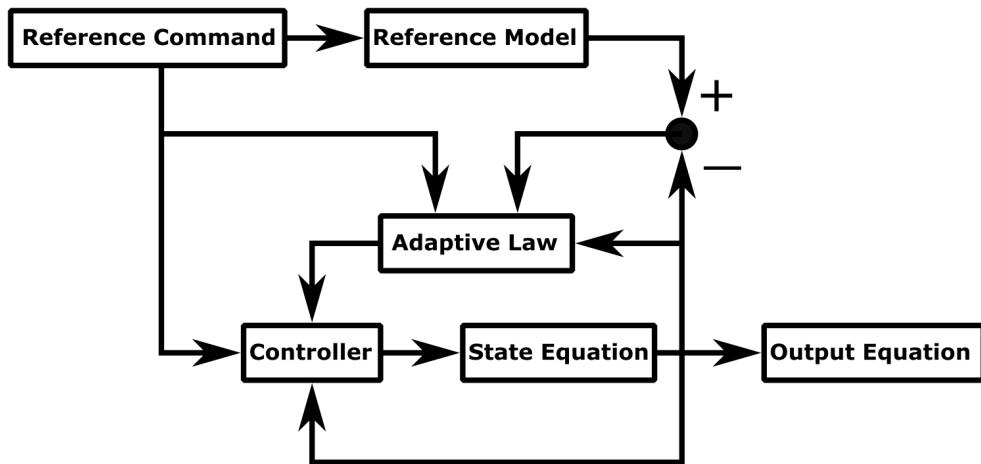
$$\begin{aligned}\dot{\hat{k}}_x &= -\gamma_x x (x - x_{ref}) \text{sign}(b) \\ \dot{\hat{k}}_r &= -\gamma_r r (x - x_{ref}) \text{sign}(b)\end{aligned}\tag{18.27}$$

where  $\gamma_x > 0$  and  $\gamma_r > 0$  are the **rates of adaptation**. The larger their values, the faster the system will adapt to the parametric uncertainties. Here the control law gains are adapted directly in order to enforce the desired closed-loop tracking performance, i.e. a direct MRAC. Alternatively, one could design an indirect adaptive control where one estimates the unknown parameter uncertainties ( $a, b$ ) online and then uses their estimated values to calculate control law gains.

Finally, using energy-based arguments, one can formally prove that the adaptive control law provides desired closed-loop tracking performance, in the sense that the system state  $x$  globally and asymptotically tracks the state  $x_{ref}$  of the reference model while keeping all signals in the corresponding closed-loop dynamics uniformly bounded in time. However, one should note that a characterization of the transient dynamics of the tracking error remains an open problem. Furthermore, the adaptive parameters  $\hat{k}_x$  and  $\hat{k}_r$  are not guaranteed to converge to their true unknown values, nor even to any constant values in any way. All that can be stated is that they will remain bounded in time. Sufficient conditions for parameter convergence are known as **persistency of excitation** and have been proven for only particular dynamic systems.

## 18.2 Direct SISO Model Reference Adaptive Control

A generic block diagram for a system operating with Model Reference Adaptive Control (MRAC) is



In essence, an MRAC system consists of a controller whose gains are updated online using an adaptive law. The adaptive law is a function of the system output, the reference command, and the tracking error, i.e. the difference between the system output and the reference model output which specifies the desired trajectories for the system to follow. Finally, the controller computes its commands based on the reference command, the system output, and the online adjusted parameters from the adaptive law. Per design, the adaptive controller forces the system output to follow the desired reference commands while operating in the presence of system parameter uncertainties. The main objective of the controller is to maintain consistent performance of the closed-loop system in the presence of uncertainties and unknown variations in the system parameters. When the true system parameters are unknown, one may attempt to estimate control gains online using available measurements. This approach is referred to as **direct adaptive control**. Alternatively, the

gains can be estimated online by solving system design equations that relate the uncertain parameters to the measured signals in the system. This is referred to as **indirect adaptive control**. MRAC systems can be designed using either approach as well as combining both approaches as **hybrid adaptive control**.

This course will consider command tracking algorithms for continuous-time dynamical systems, i.e.

$$\begin{aligned}\dot{\vec{x}} &= f(t, \vec{x}, \vec{u}, \vec{\theta}, \vec{\eta}) \\ \vec{y} &= h(t, \vec{x}, \vec{u}, \vec{\theta}, \vec{\eta})\end{aligned}\quad (18.28)$$

with constant vector of uncertain parameters,  $\vec{\theta}$ , bounded environmental disturbances,  $\vec{\eta}$ , and  $\vec{x}$  is available for full state feedback control, i.e.  $\vec{y}(t) = \vec{x}(t)$ . Then, the command tracking control problem involves designing the control input  $\vec{u}$  so that the regulated output  $\vec{y}(t)$  tracks a given bounded reference signal  $\vec{r}(t) \in \mathbb{R}^n$  in the presence of system uncertainties,  $\vec{\theta}$  and environmental disturbances,  $\vec{\eta}(t)$ . Specifically, one must find  $\vec{u}(t)$  such that the command tracking error

$$\vec{e} = \vec{y}(t) - \vec{r}(t) \quad (18.29)$$

becomes sufficiently small as  $t \rightarrow \infty$  and all the signals in the corresponding closed-loop system remain uniformly bounded in time. Note that if one could achieve  $\vec{e}(t) \rightarrow 0$  as  $t \rightarrow \infty$ , then asymptotic command tracking would be achieved. However, this may not be feasible, in which case, the control objective would be to achieve uniform ultimate boundedness of the command tracking error, i.e.

$$\|\vec{e}\| \leq \epsilon, \quad \forall t \geq T \quad (18.30)$$

where  $\epsilon > 0$  is the desired tracking tolerance and  $T$  is some finite time.

## Direct SISO MRAC

This section will consider a single input, single output (SISO) plant with dynamics

$$\dot{x} = ax + b(u + f(x)) \quad (18.31)$$

where  $x$  is the system state and  $u$  is the control input, while  $a$  and  $b$  are the unknown constant parameters. Assuming the sign of  $b$  is known and the system is controllable. The system dynamics depend on the unknown function  $f(x)$  defined as a linear combination of  $N$  known basic functions  $\phi_i(x)$  with  $N$  unknown constants,  $\vec{\theta}_i$ , i.e.

$$f(x) = \sum_{i=1}^N \vec{\theta}_i \phi_i(x) = \vec{\theta}^T \Phi(x) \quad (18.32)$$

where  $\Phi(x) = [\phi_1 \dots \phi_N]^T \in \mathbb{R}^n$  is the known **regressor vector**, whose components  $\phi_i(x)$  are assumed to be Lipschitz-continuous in  $x$ . Thus, the SISO plant model considered here is

$$\dot{x} = ax + b(u + \vec{\theta}^T \Phi(x)) \quad (18.33)$$

Furthermore, suppose a stable reference model is given. Its dynamics are described by a first-order differential equation in the form

$$\dot{x}_{ref} = a_{ref}x_{ref} + b_{ref}r(t) \quad (18.34)$$

where  $a_{ref} < 0$  and  $b_{ref}$  are the desired constants chosen to represent the desired response due to bounded commands. For example,  $b_{ref} = -a_{ref}$  for unity DC gain and, then selecting  $a_{ref}$  such that the reference time constant is as small as desired and would be indirectly indicative of the control effort possible.

The control objective of interest in the SISO case is to asymptotically track the state  $x_{ref}$  of the reference model which can be driven by any bounded command  $r(t)$ . In other words, one requires a control law  $u(t)$  such that the state tracking error  $e(t) = x(t) - x_{ref}(t)$  globally uniformly asymptotically tends to zero as  $t \rightarrow \infty$ , while all other signals remain uniformly ultimately bounded and must be accomplished in the presence of  $N + 2$  unknown constant parameters  $\{a, b, \theta_1, \dots, \theta_N\}$ .

First, define the ideal feedback and feedforward control law as if the unknown parameters were known as

$$u_{ideal} = k_x x + k_r r - \theta^T \Phi(x) \quad (18.35)$$

where  $k_x$  and  $k_r$  are the ideal feedback and feedforward gains, respectively. Next, substitute this into the plant, one has

$$\dot{x} = (a + bk_x)x + bk_r r(t) \quad (18.36)$$

Then, comparing with the desired reference model dynamics, it follows that the ideal gains  $k_x$  and  $k_r$  must satisfy the following two algebraic equations

$$a + bk_x = a_{ref} \quad bk_r = b_{ref} \quad (18.37)$$

These relations are called the **matching conditions**, then it is clear that for SISO plants, the unknown ideal gains,  $k_x$  and  $k_r$ , always exist. However, for MIMO dynamics, this is not the case. Thus, one can propose a tracking control law as

$$u = \hat{k}_x x + \hat{k}_r r - \hat{\theta}^T \Phi(x) \quad (18.38)$$

where the adaptive feedback gain,  $\hat{k}_x$ , the adaptive feedforward gain,  $\hat{k}_r$ , and the vector of estimated parameters,  $\vec{\theta}$ , are determined to achieve global uniform asymptotic tracking of the reference model trajectories. To do show, substitute into the system dynamics

$$\dot{x} = (a + b\hat{k}_x)x + b(\hat{k}_r r - (\hat{\theta} - \vec{\theta})^T \Phi(x)) \quad (18.39)$$

then, rewriting using the matching conditions

$$\dot{x} = a_{ref}x + bk_r r + b(\hat{k}_x - k_x)x + b(\hat{k}_r - k_r)r - b(\hat{\theta} - \vec{\theta})^T \Phi(x) \quad (18.40)$$

where the control gain estimation errors are

$$\Delta k_x = \hat{k}_x - k_x \quad (18.41)$$

and

$$\Delta k_r = \hat{k}_r - k_r \quad (18.42)$$

and the parameter estimation error vector is

$$\Delta \vec{\theta} = \hat{\theta} - \vec{\theta} \quad (18.43)$$

Then, the closed-loop dynamics of the system tracking error signal

$$e(t) = x(t) - x_{ref}(t) \quad (18.44)$$

can be written as

$$\dot{e}(t) = a_{ref}e + b(\Delta k_x x + \Delta k_r r - \Delta \vec{\theta}^T \Phi(x)) \quad (18.45)$$

where these are going to choose adaptive gains  $\hat{k}_x, \hat{k}_r, \hat{\vec{\theta}}$  to enforce global uniform asymptotic stability of the origin. This will accomplished through the inverse Lyapunov design approach in which one chooses a Lyapunov function candidate and then selects adaptive laws such that Lyapunov function time derivative evaluated along the trajectories of the error dynamics becomes nonpositive. Then, by Lyapunov stability theory, the tracking error would asymptotically converge to the origin and the system state would asymptotically track the state of the reference model.

Consider a quadratic Lyapunov function candidate in the form

$$V(e, \Delta k_x, \Delta k_r, \Delta \vec{\theta}) = e^2 + |b|(\gamma_x^{-1} \Delta k_x^2 + \gamma_r^{-1} \Delta k_r^2 + \Delta \vec{\theta}^T \Gamma_\theta^{-1} \Delta \vec{\theta}) \quad (18.46)$$

where  $\gamma_x > 0, \gamma_r > 0$  and  $\Gamma_\theta \in \mathbb{R}^{n \times n}$  are the **rates of adaptation** and are tunable by the control system designer. Taking the time derivative of  $V$  along the trajectories of the current SISO MRAC problem, one has

$$\begin{aligned} \dot{V}(e, \Delta k_x, \Delta k_r, \Delta \vec{\theta}) &= 2e\dot{e} + 2|b|(\gamma_x^{-1} \Delta k_x \dot{\hat{k}}_x + \gamma_r^{-1} \Delta k_r \dot{\hat{k}}_r + \Delta \vec{\theta}^T \Gamma_\theta \dot{\hat{\theta}}) \\ \dot{V}(e, \Delta k_x, \Delta k_r, \Delta \vec{\theta}) &= 2e(a_{ref}e + b(\Delta k_x x + \Delta k_r r - \Delta \vec{\theta}^T \Phi(x))) \\ &\quad + 2|b|(\gamma_x^{-1} \Delta k_x \dot{x} + \gamma_r^{-1} \Delta k_r \dot{r} + \Delta \vec{\theta}^T \Gamma_\theta^{-1} \dot{\hat{\theta}}) \\ \dot{V}(e, \Delta k_x, \Delta k_r, \Delta \vec{\theta}) &= 2a_{ref}e^2 + 2|b|(\Delta k_x (xe \operatorname{sign}(b) + \gamma_x^{-1} \dot{\hat{k}}_x) \\ &\quad + 2|b|(\Delta k_r (resign(b) + \gamma_r^{-1} \dot{\hat{k}}_r)) + 2|b|\Delta \vec{\theta}^T (-\Phi(x)e \operatorname{sign}(b) + \Gamma_\theta^{-1} \dot{\hat{\theta}})) \end{aligned} \quad (18.47)$$

By Lyapunov stability theory, it is sufficient to choose adaptive laws such that  $\dot{V}(e, \Delta k_x, \Delta k_r, \Delta \vec{\theta}) \leq 0$  which will occur if one selects

$$\begin{aligned} \dot{\hat{k}}_x &= -\gamma_x x e \operatorname{sign}(b) \\ \dot{\hat{k}}_r &= -\gamma_r r e \operatorname{sign}(b) \\ \dot{\hat{\theta}} &= \Gamma_\theta \Phi(x) e \operatorname{sign}(b) \end{aligned} \quad (18.48)$$

then,

$$\dot{V}(e, \Delta k_x, \Delta k_r, \Delta \vec{\theta}) = 2a_{ref}e(t)^2 \leq 0 \quad (18.49)$$

as  $a_{ref} < 0$  as specified. This immediately implies that the signals  $(e, \Delta k_x, \Delta k_r, \Delta \vec{\theta})$  are bounded in time. The latter, coupled with the facts that  $x_{ref}$  and  $r$  are bounded and  $\vec{\theta}$  is a constant vector, means that the system state  $x$  and the estimated vector of parameters  $\hat{\theta}$  are uniformly bounded. Moreover, since the components  $\phi_i(x)$  of  $\Phi(x)$  are Lipschitz-continuous functions of  $x$  (which is bounded), then all  $\phi_i(x)$  are uniformly bounded. Hence, the control signal  $u$  is uniformly bounded. Lastly, both  $\dot{x}$  and  $\dot{x}_{ref}$  are bounded.

Furthermore, differentiating results in

$$\ddot{V}(e, \Delta k_x, \Delta k_r, \Delta \vec{\theta}) = 4a_{ref}e(t)\dot{e}(t) \quad (18.50)$$

which is bounded and consequently,  $\dot{V}$  is a continuous function of time. Furthermore, as  $V$  is lower bounded and  $\dot{V} \leq 0$ , then  $V$  must have a finite limit and with Barbalot's lemma, one has

$$\lim_{t \rightarrow \infty} \dot{V}(t) = 0 \quad (18.51)$$

which infers  $e(t) \rightarrow 0$  as  $t \rightarrow \infty$ . Furthermore, as the Lyapunov function is radially unbounded and does not depend explicitly on time, the attained stability property is global and uniform, i.e. the closed-loop tracking error dynamics are globally uniformly asymptotically stable. The command tracking problem is solved.

It should be noted that the estimated parameters  $\hat{\theta}$  are not guaranteed to converge to their ideal parameters  $\vec{\theta}$ ; however, they will be uniformly bounded. The persistency of excitation provide sufficient conditions for these estimates to converge. Lastly, it should be noted that the MRAC “tuning knobs” are the rates of adaptation,  $\gamma_x$ ,  $\gamma_r$ , and  $\Gamma_\theta$  where the larger the rates, the faster the adaptive laws will evolve and will yield fast tracking; however, this can lead to undesirable oscillations during transient times as the system output is being forced closer to the command. The trade-off between fast tracking and smooth transients is a design-dependent challenge.

### Helicopter Pitch Example

The angular motion of a helicopter is achieved by tilting its main rotor and as a result altering the direction of the rotor thrust vector which induces a change in the angular moments acting on the vehicle and results in a change in angular velocity. For a helicopter in hover, the helicopter pitch dynamics depend primarily on the vehicle pitch rate  $q$  and on the applied longitudinal control input,  $\delta$ , which affects the longitudinal cyclic pitch of the rotor. This is equivalent to the elevator command for an airplane. Assuming constant thrust and neglecting small forward and vertical velocity components, the pitch dynamics of a helicopter during hover can be approximated by

$$\dot{q} = M_q q + M_\delta (\delta + f(q)) \quad (18.52)$$

where  $M_q$  is the pitch stability derivative and  $M_\delta$  is the cyclic pitch-to-pitch control derivative. The system also depends on the unknown function  $f(q)$  which models inherent uncertainties in the helicopter dynamics, both linear and nonlinear.

Consider that the unknown parameters of this model are  $M_q = -0.61$  rad/s and  $M_\delta = -6.65$  rad/s<sup>2</sup>. Also, assume that

$$f(q) = -0.01 \tanh\left(\frac{360}{\pi}q\right) = \theta\Phi(q) \quad (18.53)$$

where  $\theta = -0.01$  is unknown and  $\Phi(q)$  is the known regressor. All together, one has

$$\dot{q} = -0.61q - 6.65 \left( \delta - 0.01 \tanh\left(\frac{360}{\pi}q\right) \right) \quad (18.54)$$

whose origin is locally unstable for  $\delta = 0$  and requires active control for stabilization and command tracking of some commanded pitch rate,  $q_c(t)$ .

Using the direct SISO MRAC design, let the reference model be

$$\dot{q}_{ref} = 4(q_c - q_{ref}) \quad (18.55)$$

the controller be

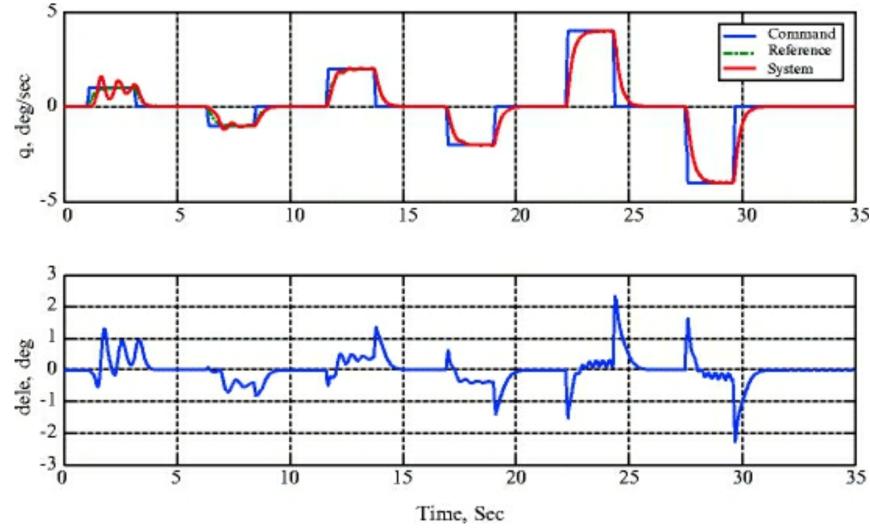
$$\delta = \hat{k}_q q + \hat{k}_{q_c} q_c - \hat{\theta}^T \Phi(q) \quad (18.56)$$

and adaptive law be

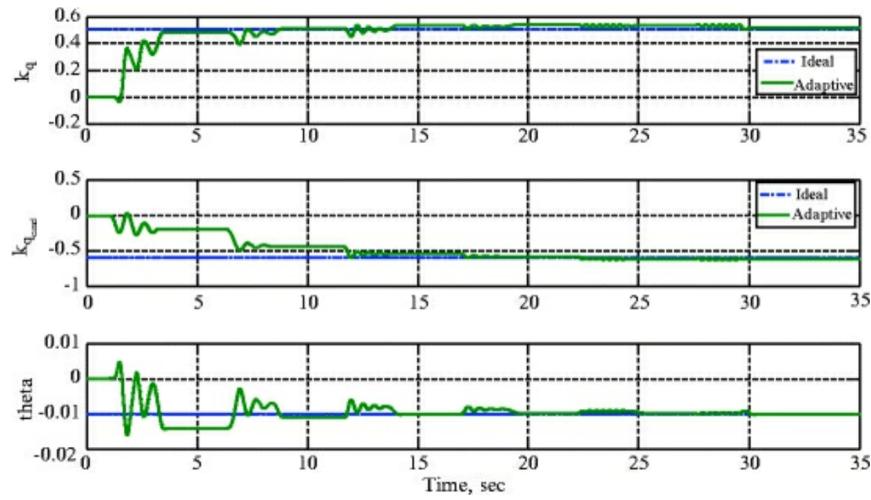
$$\begin{aligned} \dot{\hat{k}}_q &= \gamma_q q (q - q_{ref}) \\ \dot{\hat{k}}_{q_c} &= \gamma_{q_c} q_c (q - q_{ref}) \\ \dot{\hat{\theta}} &= -\Gamma_\theta \Phi(q) (q - q_{ref}) \end{aligned} \quad (18.57)$$

as the sign of  $b$  is negative for this system. Note that  $a_{ref} = -b_{ref} = -4$ .

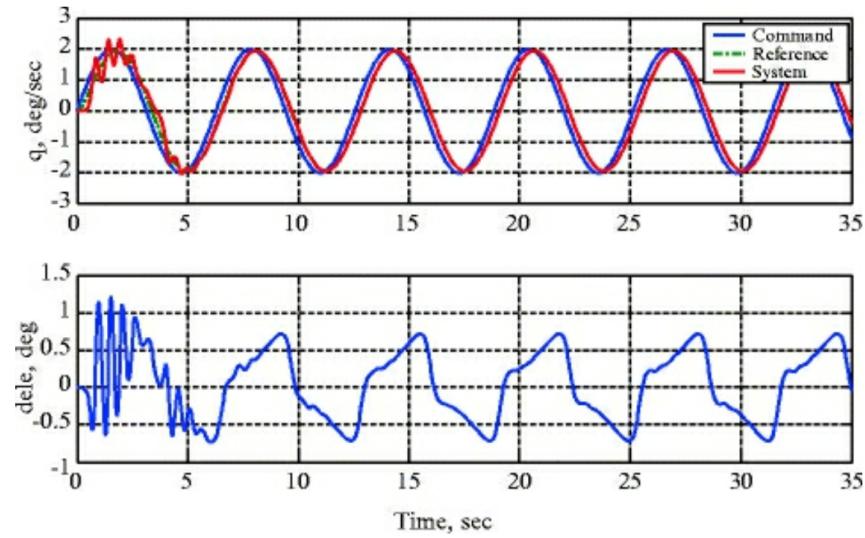
Choosing  $\gamma_q = \gamma_{q_c} = 6000$  and  $\Gamma_\theta = 8$  after several iterations, one can obtain the following results for a step input command to  $q_c$



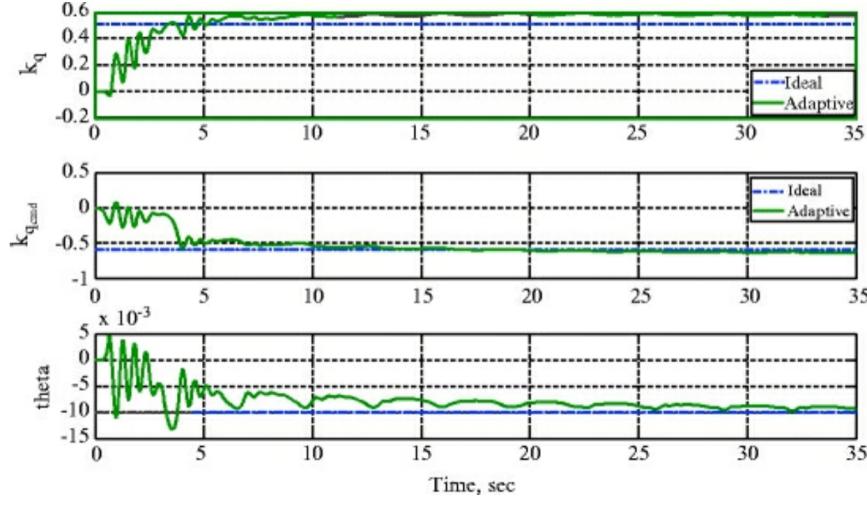
where the estimated parameters eventually reach their ideal values.



As well as the following results for a sinusoidal command to  $q_c$



where the estimated parameters do not reach their ideal values, but do remain ultimately bounded.



### 18.3 Direct MIMO Model Reference Adaptive Control

This section will consider the extension of the direct MRAC design to a multiple input, multiple output (MIMO) nonlinear plant with dynamics

$$\dot{\vec{x}} = A\vec{x} + B\Lambda(\vec{u} + f(\vec{x})) \quad (18.58)$$

where  $\vec{x} \in \mathbb{R}^n$  is the system state,  $\vec{u} \in \mathbb{R}^p$  is the control input,  $B \in \mathbb{R}^{n \times p}$  is the known control matrix,  $A \in \mathbb{R}^{n \times n}$  is the unknown state matrix, and  $\Lambda \in \mathbb{R}^{p \times p}$  is the unknown diagonal matrix of control uncertainties with diagonal elements  $\lambda_i > 0$ . This uncertainty can be due to modeling errors or model control failures. It is assumed that the pair  $(A, B\Lambda)$  is controllable. In addition, the unknown nonlinear vector function  $f(\vec{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^n$  represents the system matched uncertainty where each individual component  $f_i(\vec{x})$  of  $f(\vec{x})$  can be written as a linear combination of  $N$  known locally Lipschitz-continuous basis function  $\phi_i(\vec{x})$  with unknown constant coefficients, i.e.

$$f(\vec{x}) = \vec{\theta}^T \Phi(\vec{x}) \quad (18.59)$$

where  $\vec{\theta} \in \mathbb{R}^N$  is constant unknown matrix and  $\Phi(\vec{x}) = [\phi_1(\vec{x}) \dots \phi_N(\vec{x})]^T$  is the known regressor vector.

A direct MIMO state feedback MRAC is required to force the system state  $\vec{x}$  to globally uniformly asymptotically track the reference model state  $\vec{x}_{ref}$  given by the **reference model**

$$\dot{\vec{x}}_{ref} = A_{ref}\vec{x}_{ref} + B_{ref}\vec{r}(t) \quad (18.60)$$

where  $A_{ref} \in \mathbb{R}^n$  is chosen such that all eigenvalues are in the LHP,  $B_{ref} \in \mathbb{R}^{n \times m}$  is the reference input matrix, and  $\vec{r}(t) \in \mathbb{R}^p$  is the external, bounded reference command. One also requires that during tracking in an MRAC system all signals in the closed-loop system remain uniformly bounded. Note that the reference command has the same dimension as the plant control input. Thus, given any bounded command  $\vec{r}(t)$ , the control input  $\vec{u}(t)$  needs to be chosen such that the state tracking error,  $\vec{e}(t) = \vec{x}(t) - \vec{x}_{ref}(t)$ , globally uniformly asymptotically tends to zero, i.e.

$$\lim_{t \rightarrow \infty} \|\vec{x}(t) - \vec{x}_{ref}\| = 0 \quad (18.61)$$

First, note that if matrices  $A$  and  $\Lambda$  were known, one can apply the ideal fixed-gain control law as

$$\vec{u} = K_x^T \vec{x} + K_r^T \vec{r} - \vec{\theta}^T \Phi(\vec{x}) \quad (18.62)$$

and obtain the closed-loop system

$$\dot{\vec{x}} = (A + B\Lambda K_x^T) \vec{x} + B\Lambda K_r^T \vec{r} \quad (18.63)$$

Comparing this with the desired reference dynamics, for the existence of a controller of the ideal fixed-gain form, the ideal unknown control gains,  $K_x$  and  $K_r$ , must satisfy the **matching conditions**

$$\begin{aligned} A + B\Lambda K_x^T &= A_{ref} \\ B\Lambda K_r^T &= B_{ref} \end{aligned} \quad (18.64)$$

Assuming these hold, by inspection, the ideal fixed-gain control law will result in a closed-loop system which is exactly the same as the reference model. Consequently, for any bounded reference signal input,  $\vec{r}(t)$ , the fixed-gain controller provides global uniform asymptotic tracking performance. It is important to note that given a general  $A$ ,  $B$ ,  $\Lambda$ ,  $A_{ref}$  and  $B_{ref}$ , there is no guarantee that the ideal MIMO gains  $K_x$  and  $K_r$  exist such that the matching conditions are satisfied and the chosen adaptive control law may not be able to meet the design objective. However, in practice, the structure of  $A$  is known and the reference model matrices  $A_{ref}$  and  $B_{ref}$  are chosen so that the system has at least one ideal solution for  $K_x$  and  $K_r$ .

Assuming  $K_x$  and  $K_r$  do exist, consider the following control law based on the ideal fixed-gain control law, i.e.

$$\vec{u} = \hat{K}_x^T \vec{x} + \hat{K}_r^T \vec{r} - \hat{\theta}^T \Phi(\vec{x}) \quad (18.65)$$

where  $\hat{K}_x \in \mathbb{R}^{n \times p}$ ,  $\hat{K}_r \in \mathbb{R}^{p \times p}$ , and  $\hat{\theta} \in \mathbb{R}^{N \times n}$  are the estimates of the ideal unknown matrices  $K_x$ ,  $K_r$  and  $\vec{\theta}$ , respectively, based on Lyapunov stability analysis. Substituting this control law into the plant dynamics results in the closed-loop system dynamics as

$$\dot{\vec{x}} = (A + B\Lambda \hat{K}_x^T) \vec{x} + B\Lambda \left( \hat{K}_r^T \vec{r} - (\hat{\theta} - \vec{\theta})^T \Phi(\vec{x}) \right) \quad (18.66)$$

and subtracting the reference model from these closed-loop system dynamics, one has the closed-loop tracking error dynamics as

$$\dot{\vec{e}} = (A + B\Lambda \hat{K}_x^T) \vec{x} + B\Lambda \left( \hat{K}_r^T \vec{r} - (\hat{\theta} - \vec{\theta})^T \Phi(\vec{x}) \right) - A_{ref} \vec{x}_{ref} - B_{ref} \vec{r} \quad (18.67)$$

Including the matching conditions, one has

$$\begin{aligned} \dot{\vec{e}} &= (A_{ref} + B\Lambda (\hat{K}_x - K_x)) \vec{x} - A_{ref} \vec{x}_{ref} + B\Lambda (\hat{K}_r - K_r) \vec{r} - B\Lambda (\hat{\theta} - \vec{\theta})^T \Phi(\vec{x}) \\ \dot{\vec{e}} &= A_{ref} \vec{e} + B\Lambda \left( (\hat{K}_x - K_x)^T \vec{x} + (\hat{K}_r - K_r)^T \vec{r} - (\hat{\theta} - \vec{\theta})^T \Phi(\vec{x}) \right) \end{aligned} \quad (18.68)$$

where defining  $\Delta K_x = \hat{K}_x - K_x$ ,  $\Delta K_r = \hat{K}_r - K_r$ , and  $\Delta \vec{\theta} = \hat{\theta} - \vec{\theta}$  as the parameter estimation errors, one has

$$\dot{\vec{e}} = A_{ref} \vec{e} + B\Lambda \left( \Delta K_x^T \vec{x} + \Delta K_r^T \vec{r} - \Delta \vec{\theta}^T \Phi(\vec{x}) \right) \quad (18.69)$$

Next, one can define the rates of adaptation as  $\Gamma_x = \Gamma_x^T > 0$ ,  $\Gamma_r = \Gamma_r^T > 0$ , and  $\Gamma_\theta = \Gamma_\theta^T > 0$ . Then, consider a globally radially unbounded quadratic Lyapunov function candidate in the form

$$V(\vec{e}, \Delta K_x, \Delta K_r, \Delta \vec{\theta}) = \vec{e}^T P \vec{e} + \text{Tr} \left( \left( \Delta K_x^T \Gamma_x^{-1} \Delta K_x + \Delta K_r^T \Gamma_r^{-1} \Delta K_r + \Delta \vec{\theta}^T \Gamma_\theta^{-1} \Delta \vec{\theta} \right) \Lambda \right) \quad (18.70)$$

where  $P = P^T > 0$  satisfies the **algebraic Lyapunov equation**

$$PA_{ref} + A_{ref}^T P = -Q \quad (18.71)$$

for some  $Q = Q^T > 0$ . Then, the time derivative of  $V$  evaluated along the state trajectories can be written as

$$\dot{V} = \dot{\vec{e}}^T P \vec{e} + \vec{e}^T P \dot{\vec{e}} + 2\text{Tr} \left( \left( \Delta K_x^T \Gamma_x^{-1} \dot{K}_x + \Delta K_r^T \Gamma_r^{-1} \dot{K}_r + \Delta \vec{\theta}^T \Gamma_\theta^{-1} \dot{\vec{\theta}} \right) \Lambda \right) \quad (18.72)$$

$$\begin{aligned} \dot{V} = & \left( A_{ref} \vec{e} + B \Lambda (\Delta K_x^T \vec{x} + \Delta K_r^T \vec{r} - \Delta \vec{\theta}^T \Phi(\vec{x})) \right)^T P \vec{e} \\ & + \vec{e}^T P \left( A_{ref} \vec{e} + B \Lambda (\Delta K_x^T \vec{x} + \Delta K_r^T \vec{r} - \Delta \vec{\theta}^T \Phi(\vec{x})) \right) \\ & + 2\text{Tr} \left( \left( \Delta K_x^T \Gamma_x^{-1} \dot{K}_x + \Delta K_r^T \Gamma_r^{-1} \dot{K}_r + \Delta \vec{\theta}^T \Gamma_\theta^{-1} \dot{\vec{\theta}} \right) \Lambda \right) \end{aligned} \quad (18.73)$$

$$\begin{aligned} \dot{V} = & \vec{e}^T (A_{ref} P + PA_{ref}) \vec{e} + 2\vec{e}^T P B \Lambda (\Delta K_x^T \vec{x} + \Delta K_r^T \vec{r} - \Delta \vec{\theta}^T \Phi(\vec{x})) \\ & + 2\text{Tr} \left( \left( \Delta K_x^T \Gamma_x^{-1} \dot{K}_x + \Delta K_r^T \Gamma_r^{-1} \dot{K}_r + \Delta \vec{\theta}^T \Gamma_\theta^{-1} \dot{\vec{\theta}} \right) \Lambda \right) \end{aligned} \quad (18.74)$$

$$\begin{aligned} \dot{V} = & -\vec{e}^T Q \vec{e} + \left( 2\vec{e}^T P B \Lambda \Delta K_x^T \vec{x} + 2\text{Tr} \left( \Delta K_x^T \Gamma_x^{-1} \dot{K}_x \Lambda \right) \right) \\ & + \left( 2\vec{e}^T P B \Lambda \Delta K_r^T \vec{r} + 2\text{Tr} \left( \Delta K_r^T \Gamma_r^{-1} \dot{K}_r \Lambda \right) \right) \\ & + \left( -2\vec{e}^T P B \Lambda \Delta \vec{\theta}^T \Phi(\vec{x}) + 2\text{Tr} \left( \Delta \vec{\theta}^T \Gamma_\theta^{-1} \dot{\vec{\theta}} \Lambda \right) \right) \end{aligned} \quad (18.75)$$

Next, one can use the vector trace identity  $\vec{a}^T \vec{b} = \text{Tr}(\vec{b} \vec{a}^T)$  to form

$$\begin{aligned} \dot{V} = & -\vec{e}^T Q \vec{e} + 2\text{Tr} \left( \Delta K_x^T \left( \Gamma_x^{-1} \dot{K}_x + \vec{x} \vec{e}^T P B \right) \Lambda \right) \\ & + 2\text{Tr} \left( \Delta K_r^T \left( \Gamma_r^{-1} \dot{K}_r + \vec{r} \vec{e}^T P B \right) \Lambda \right) + 2\text{Tr} \left( \Delta \vec{\theta}^T \left( \Gamma_\theta^{-1} \dot{\vec{\theta}} - \Phi(\vec{x}) \vec{e}^T P B \right) \Lambda \right) \end{aligned} \quad (18.76)$$

Thus, if the **Direct MIMO adaptive laws** are chosen as

$$\begin{aligned} \dot{K}_x &= -\Gamma_x \vec{x} \vec{e}^T P B \\ \dot{K}_r &= -\Gamma_r \vec{r}(t) \vec{e}^T P B \\ \dot{\vec{\theta}} &= \Gamma_\theta \Phi(\vec{x}) \vec{e}^T P B \end{aligned} \quad (18.77)$$

Then,

$$\dot{V} = -\vec{e}^T Q \vec{e} \leq 0 \quad (18.78)$$

Therefore, the closed-loop error dynamics are uniformly stable. So, the tracking error  $\vec{e}(t)$  and the parameter estimation errors  $\Delta K_x(t)$ ,  $\Delta K_r(t)$ , and  $\Delta \theta(t)$  are uniformly bounded and so are the parameter estimates  $\hat{K}_x(t)$ ,

$\hat{K}_r(t)$ , and  $\hat{\theta}(t)$ . Since  $\vec{r}(t)$  is bounded and  $A_{ref}$  has all LHP eigenvalues, then  $\vec{x}_{ref}$  and  $\dot{\vec{x}}_{ref}$  are bounded. Hence, the system state  $\vec{x}(t)$  is uniformly bounded, the control input  $\vec{u}(t)$  is bounded, and  $\vec{x}(t)$  is bounded, and thus,  $\vec{e}(t)$  is bounded. Furthermore, the second time derivative of  $V(t)$

$$\ddot{V} = -2\vec{e}^T Q \dot{\vec{e}} \quad (18.79)$$

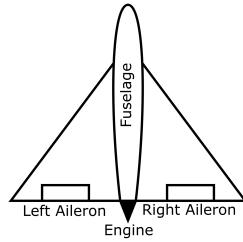
is bounded, and so  $\dot{V}(t)$  is uniformly continuous. Since, in addition,  $V(t)$  is lower bounded and  $\dot{V}(t) \leq 0$ , then using Barbalat's lemma  $\lim_{t \rightarrow \infty} \dot{V}(t) = 0$ . Thus, it has been formally proven that the state tracking error  $\vec{e}(t)$  tends to the origin globally, uniformly, and asymptotically, i.e.

$$\lim_{t \rightarrow \infty} \|\vec{x}(t) - \vec{x}_{ref}(t)\| = 0 \quad (18.80)$$

Lastly, note that the tuning knobs in the MIMO case are  $\Gamma_x$ ,  $\Gamma_r$ , and  $\Gamma_\theta$  as before, but also  $Q$  for the algebraic Lyapunov equation, all of which must be symmetric, positive definite matrices.

## MRAC Control of Delta Wing

Consider a delta wing airplane as shown in the following figure.



Delta wings are known to be unstable, yet their primary advantage is aerodynamic efficiency in high-speed flight. A delta wing's open-loop instability at high angles of attack is in the roll plane due to unsteady aerodynamic effects acting asymmetrically on the delta wing. This is called the **wing rock phenomenon**. A generic delta wing rock dynamic model is

$$\begin{aligned} \dot{p} &= a_1 p + a_2 \phi + (\theta_1 |\phi| + \theta_2 |p|) p + \theta_3 \phi^3 + b \delta_a \\ \dot{\phi} &= p \end{aligned} \quad (18.81)$$

where  $\phi$  is the roll angle (rad),  $p$  is the roll rate (rad/s), and  $\delta_a$  is the differential aileron control input (rad).

To actively control a delta wing airplane, one can design an MRAC system by rewriting this dynamics in the direct MRAC plant dynamics form

$$\dot{\vec{x}} = A \vec{x} + B \Lambda \left( \vec{u} + \vec{\theta}^T \Phi(\vec{x}) \right) \quad (18.82)$$

as

$$\begin{bmatrix} \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} p \\ \phi \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} b \left( \delta_a + \frac{1}{b} \left( (\theta_1 |\phi| + \theta_2 |p|) p + \theta_3 \phi^3 \right) \right) \quad (18.83)$$

where the unknown parameters for this model are  $a_1$ ,  $a_2$ ,  $\theta_1$ ,  $\theta_2$ , and  $\theta_3$ . Note that the uncertain function  $f(x) = \vec{\theta}^T \Phi(\vec{x})$  implies that

$$\vec{\theta} = \begin{bmatrix} \theta_1 \\ \frac{\theta_2}{b} \\ \frac{\theta_3}{b} \end{bmatrix} \quad (18.84)$$

and

$$\Phi(\vec{x}) = \begin{bmatrix} |\phi|p \\ |p|p \\ \phi^3 \end{bmatrix} \quad (18.85)$$

For the reference model, consider the second-order transfer function for a commanded roll angle,  $\phi_c$ , as

$$\frac{\phi_{ref}(s)}{\phi_c(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (18.86)$$

where  $\phi_{ref}$  is the reference roll angle,  $\omega_n$  is the reference natural frequency, and  $\zeta$  is the reference damping ratio. By also defining  $p_{ref} = \dot{\phi}_{ref}$ , one can write this in state-space form as

$$\begin{bmatrix} \dot{p}_{ref} \\ \ddot{\phi}_{ref} \end{bmatrix} = \begin{bmatrix} -2\zeta\omega_n & -\omega_n^2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} p_{ref} \\ \phi_{ref} \end{bmatrix} + \begin{bmatrix} \omega_n^2 \\ 0 \end{bmatrix} \phi_c \quad (18.87)$$

which has the reference model form

$$\dot{\vec{x}}_{ref} = A_{ref} \vec{x}_{ref} + B_{ref} r \quad (18.88)$$

Checking the matching conditions,

$$\begin{aligned} \begin{bmatrix} a_1 & a_2 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} b K_x^T &= \begin{bmatrix} -2\zeta\omega_n & -\omega_n^2 \\ 1 & 0 \end{bmatrix} \\ \begin{bmatrix} 1 \\ 0 \end{bmatrix} b K_r^T &= \begin{bmatrix} \omega_n^2 \\ 0 \end{bmatrix} \end{aligned} \quad (18.89)$$

one can see that these hold with

$$\begin{aligned} K_x &= -\frac{1}{b} \begin{bmatrix} 2\zeta\omega_n + a_1 \\ \omega_n^2 + a_2 \end{bmatrix} \\ K_r &= \frac{\omega_n^2}{b} \end{aligned} \quad (18.90)$$

where in this example, one has selected  $\omega_n = 1$  rad/s and  $\zeta = 0.7$ .

For this example, consider the following constant parameters

$$a_1 = 0.015 \quad a_2 = -0.018 \quad \theta_1 = -0.062 \quad \theta_2 = 0.009 \quad \theta_3 = 0.021 \quad b = 0.75 \quad (18.91)$$

which has an unstable equilibrium at  $\delta_a = 0$ . However, if one selects  $\omega_n = 1$  rad/s and  $\zeta = 0.7$  for the reference model, then the ideal gains are

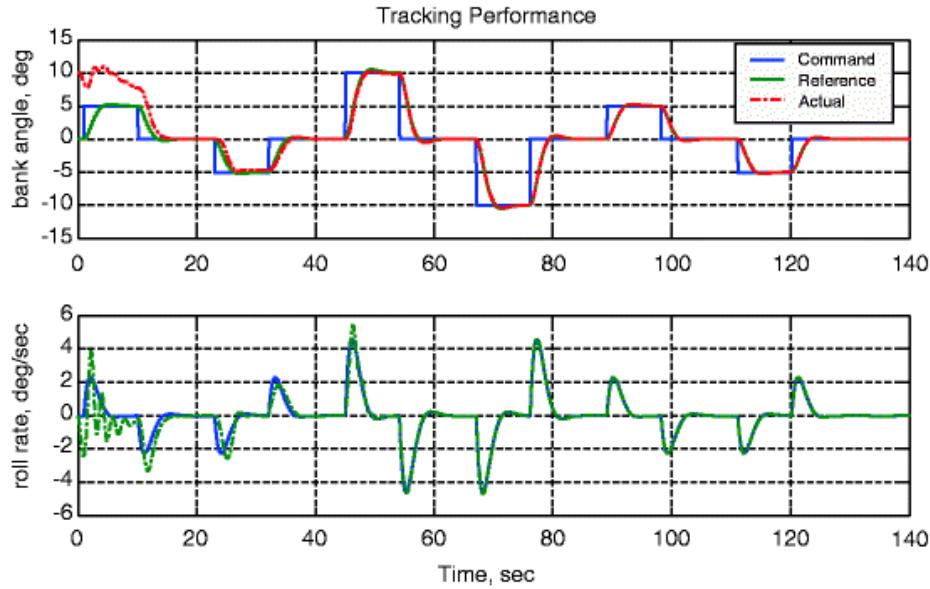
$$K_x = \begin{bmatrix} -1.8867 \\ -1.3093 \end{bmatrix} \quad K_r = 1.3333 \quad (18.92)$$

to achieve the reference model command tracking performance.

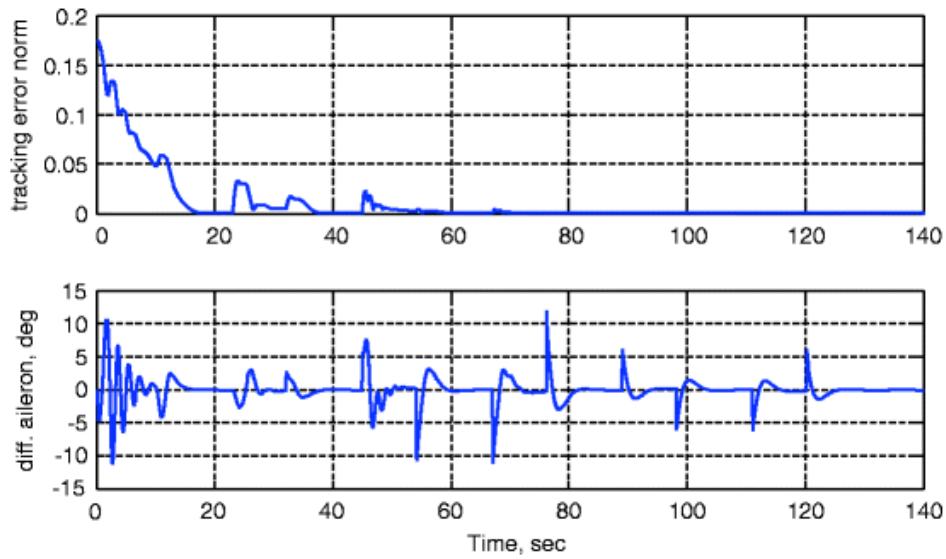
For the MRAC system, the tuning knobs are  $Q = Q^T > 0$ ,  $\Gamma_x = \Gamma_x^T > 0$ ,  $\Gamma_r = \Gamma_r^T > 0$ , and  $\Gamma_\theta = \Gamma_\theta^T > 0$ . Consider the selection

$$Q = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix} \quad \Gamma_x = 100I_{2 \times 2} \quad \Gamma_r = 100 \quad \Gamma_\theta = 100I_{3 \times 3} \quad (18.93)$$

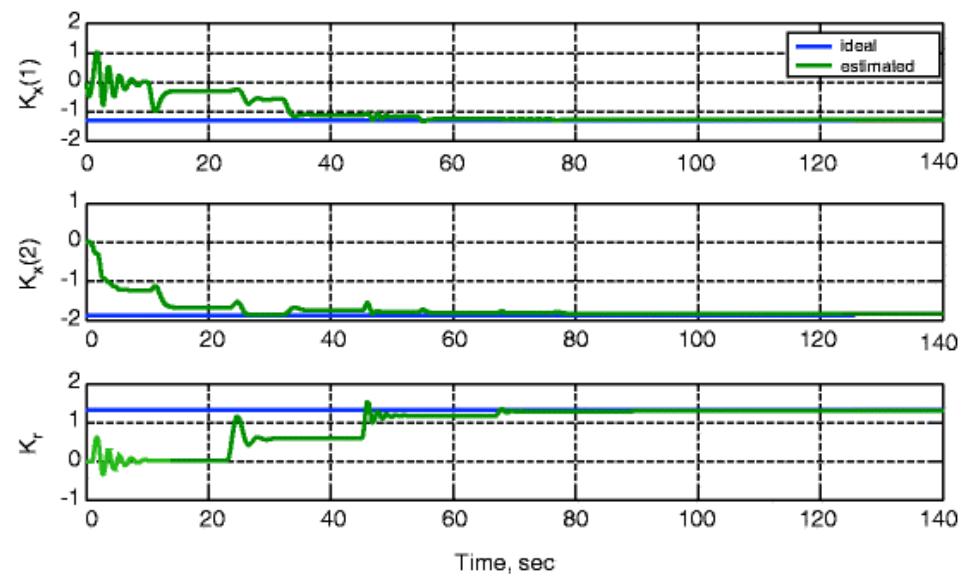
Then, with an initial bank angle set to  $10^\circ$ , one has for the MRAC system response



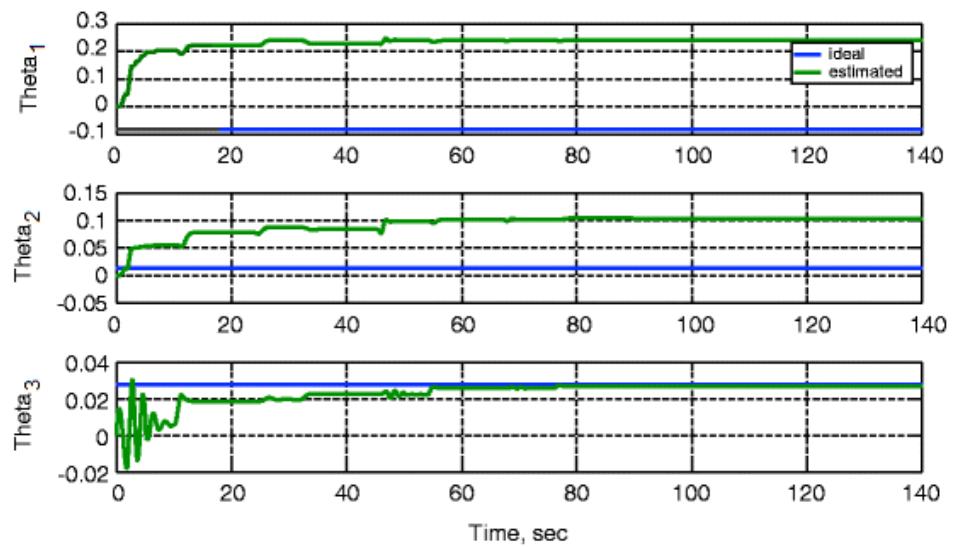
which shows that the tracking error quickly converges to zero while the aileron control inputs stay reasonable



Lastly, while the estimated feedback and feedforward gains do converge to their true values (not guaranteed)



the other estimated  $\vec{\theta}$  parameters all do not



## **Part IV**

# **Appendices**

## **Appendix A**

# **Dynamical Systems Programming**

### **A.1 Dynamical Systems in MATLAB**

# dynamical\_systems\_in\_matlab.m

## Table of Contents

MATLAB is prepared for new session. ....	1
MATLAB code style examples for AEM 468 are shown. ....	1
Show examples for improving readability and speed using functions. ....	2
Functions for LTI Systems and linear algebra are demonstrated. ....	5
An example demonstrating the effects of an additional pole. ....	10
An example demonstrating the effects of an additional LHP zero. ....	11
An example demonstrating the effects of an additional RHP zero. ....	12
Solve/simulate a differential equation both nonlinearly and linearly. ....	13
This section provides the function for ode45 function. ....	14

To publish this script as a PDF, navigate to the "Publish" tab, then hit publish. Make sure that the "Publishing Options" output file settings are set to "pdf."

## MATLAB is prepared for new session.

```
% Close all windows.
close all;

% Delete all workspace variables.
clear;

% Clear the command window.
clc;
```

## MATLAB code style examples for AEM 468 are shown.

```
% Functions, scripts, and variables should use snake_case, i.e.
% lowercase
% letters with underscores. Avoid uncommon abbreviations and make the
% length of a variable roughly the same length as its scope.
example_var = 5.5;

% Constants that should not change value are written in all capitals
% with
% underscores.
ACCEL_GRAVITY = 9.81;

% Instances of classes and structures should use mixedCase. These are
% recommended for passing data to and from functions.
parametersFDC.A = [3 4; 2 1];
parametersFDC.B = [0; 1];

% An example of a continuation line is shown for keeping the maximum
% characters under 75 characters for a single statement. The gray
% line to
```

```
% the right shows this limit in the editor. Note that for strings,
one
% must use the brackets for continuing the line. Also, the doubled
% apostrophe allows for the string to contain that symbol.
max_characters = 75;
fprintf(['Don''t go beyond %2d characters without using a ' ...
    'continuation line!\n'], max_characters);

% An example of a blank lines being omitted between one-liners, and
single
% lines used between disconnected lines of code. Note, however, that
the
% sections in this script end with two blank lines. Note that the
% assignment operator has whitespace on both sides. This should also
be
% done for comparisons and booleans. Lastly, note that absence of
extra
% blank lines, but that each statement is on its own line.
a = 1;
b = 2;
c = a + b;
disp(c);

% Note that comments come before the code they describe and inline
comments
% should only be used sparingly. Use complete sentences and two
spaces
% after a sentence-ending period in multi-sentence comments.

% The end of sections uses two blank lines.
```

*Don't go beyond 75 characters without using a continuation line!*

3

## Show examples for improving readability and speed using functions.

```
% Create time array using linspace().
time_start = 0;
time_final = 4;
num_points = 100;
time = linspace(time_start, time_final, num_points);

% Create 3x2 matrix of NaN (not-a-number).
nan_mat = nan(3, 2);
disp(nan_mat);

% Create 3x2 matrix of all zeros.
zero_mat = zeros(3, 2);
disp(zero_mat);
```

```
% Create 3x2 matrix of all ones.
one_mat = ones(3, 2);
disp(one_mat);

% Create 2x2 identity matrix (ones along the diagonal).
identity = eye(2);
disp(identity);

% Replicate a matrix 3x2 times.
tiled_mat = repmat(identity, 3, 2);
disp(tiled_mat);

% Reshape vector into a matrix. An error results if the vector does
% not
% have the correct number of elements.
vector = [1 2 3 4 5 6];
sample_mat = reshape(vector, 3, 2);
disp(sample_mat);

% Find the index and value of the last 1 element that satisfies a
% boolean
% argument.
[last_ind, last_value] = find(vector < 3.5, 1, 'last');
fprintf('The last value is %d and occurs at index %d\n',
       last_value, ...
       last_ind);

% Examples of matrix operations such as products, sums, and cumulative
% sums along the first (column) dimension.
sample_mat = [0 1 2; 3 4 5];
disp(sample_mat);
disp(prod(sample_mat, 1));
disp(sum(sample_mat, 1));
disp(cumsum(sample_mat, 1));

% An example of a native MATLAB interpolation function is given. The
% default is a linear interpolation.
coarse_inputs = 0:10;
coarse_outputs = sin(coarse_inputs);
new_inputs = 0:.25:10;
new_outputs = interp1(coarse_inputs, coarse_outputs,
                     new_inputs, 'spline');

figure(1)
plot(coarse_inputs, coarse_outputs, 'r*', 'LineWidth', 2);
grid on; hold on;
plot(new_inputs, new_outputs, 'b.-');
xlabel('Angle (rad)');
ylabel('Sine');
title('Interpolation of Sine using Spline', 'FontSize', 14);

NaN    NaN
NaN    NaN
NaN    NaN
```

```
0      0
0      0
0      0

1      1
1      1
1      1

1      0
0      1

1      0      1      0
0      1      0      1
1      0      1      0
0      1      0      1
1      0      1      0
0      1      0      1

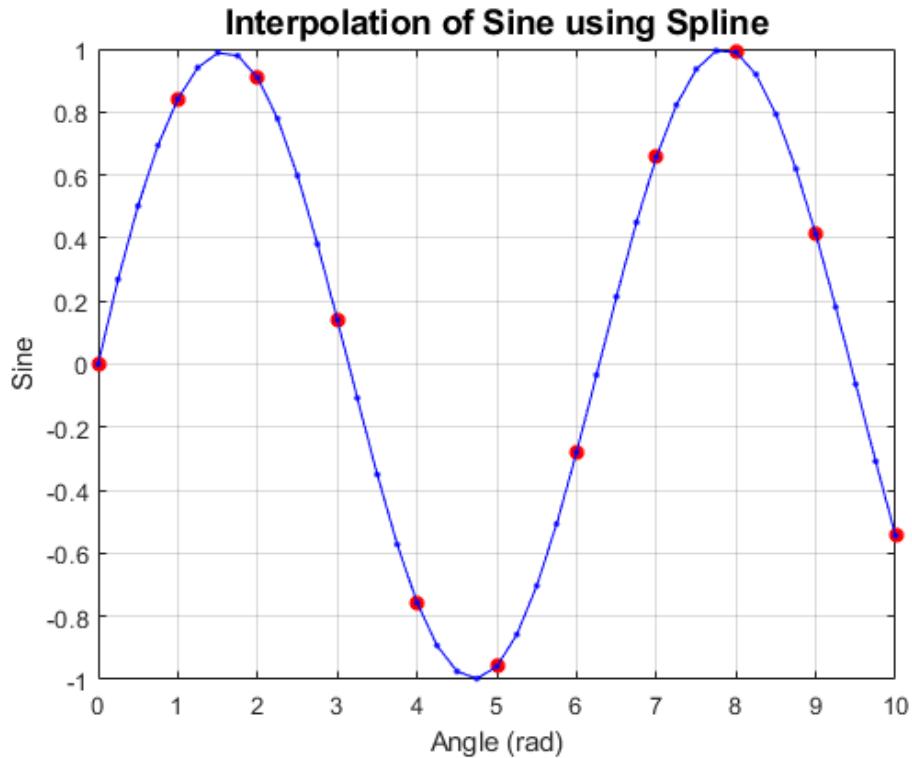
1      4
2      5
3      6
```

The last value is 3 and occurs at index 1

```
0      1      2
3      4      5

0      4      10
3      5      7

0      1      2
3      5      7
```



**Functions for LTI Systems and linear algebra are demonstrated.**

```
% Create a transfer function object.
numerator = [0.3 7];
denominator = [2 6 20];
transfer_function = tf(numerator, denominator);
disp(transfer_function);

% Convert transfer function object to state-space matrices.
[state_mat, input_mat, output_mat, feed_mat] = tf2ss(numerator, ...
denominator);

% Generate state-space system object.
system = ss(state_mat, input_mat, output_mat, feed_mat);
disp(system);

% Convert state-space object to transfer function for the 1st input.
Note
% that transfer functions for all outputs will be calculated. Thus,
% numerator and denominator may be vectors of coefficient arrays.
[numerator, denominator] = ss2tf(system.A, system.B, system.C, ...
system.D, 1);
fprintf('The LTI system is %3.1f s^2 + %4.2f s + %3.1f\n', ...
...
```

```

    numerator(1, 1), numerator(1, 2), numerator(1, 3));
fprintf('-----\n');
fprintf('      %3.1f s^2 + %1.1f s + %3.1f\n', ...
denominator(1, 1), denominator(1, 2), denominator(1, 3));

% Plot poles and zeros of the LTI system.
figure(2)
pzmap(system);

% Calculate zeros (and gain) of the LTI system.
system_zeros = zero(system);
fprintf('The zero of the system is %3.1f.\n', system_zeros);

% Calculate poles of the LTI system.
system_poles = pole(system);
fprintf(['The poles of the system are %5.3f + %5.3fj and %5.3f + ' ...
' %5.3fj.\n'], real(system_poles(1, 1)), imag(system_poles(1,
1)), ...
real(system_poles(2, 1)), imag(system_poles(2, 1)));

% Calculate the natural frequencies and damping of the LTI system.
Note
% the number of frequencies and damping ratios corresponds to the
% number of modes of the system.
[system_freq, system_damping] = damp(system);
fprintf('The natural frequency of the system is %3.1f.\n',
system_freq);
fprintf('The damping ratio of the system is %3.1f.\n',
system_damping);

% Calculate the eigenvalues of the state matrix, A. Note that these
% are different than the poles of the LTI system due to the output
matrix
% effects on the system output versus the state.
[eigenvectors, eigenvalues] = eig(system.A);
fprintf('The eigenvalues of A are %5.2f +%5.2fj and %5.2f +%5.2fj.
\n', ...
real(eigenvalues(1, 1)), imag(eigenvalues(1, 1)), ...
real(eigenvalues(2, 2)), imag(eigenvalues(2, 2)));
fprintf(['The corresponding eigenvectors are [%5.2f %5.2f+%5.2fj]^T
' ...
' and [%5.2f %5.2f+%5.2fj]^T.\n'], eigenvectors(1, 1), ...
real(eigenvectors(2, 1)), imag(eigenvectors(2, 1)), ...
eigenvectors(1, 2), real(eigenvectors(2, 2)), ...
imag(eigenvectors(2, 2)));

% Solve/simulate free response for a particular initial condition.
initial_conditions = [5.5 2.3];
output = initial(system, initial_conditions, time);

% Plot free response.
figure(3)
plot(time, output, 'r-', 'LineWidth', 2);
grid on;

```

```

xlabel('Time (sec)');
ylabel('Free Response, y(t)');

% Solve/simulate step response and get characteristics.
output = step(system, time);
step_info = stepinfo(system, 'RiseTimeLimits', [0, 1], ...
    'SettlingTimeThreshold', 0.05);

% Estimate the steady-state output, a.k.a. final value, a.k.a. DC
% gain.
final_value = dcgain(system);

% Extract rise time, settling time, peak time, and peak value.
rise_time = step_info.RiseTime;
settling_time = step_info.SettlingTime;
peak_time = step_info.PeakTime;
peak_value = step_info.Peak;

% Plot step response.
figure(4)
plot(time, output, 'r-', 'LineWidth', 2);
grid on; hold on;
plot(rise_time, final_value, 'r*');
xline(settling_time, 'Color', 'r', 'LineStyle', '--', 'LineWidth', 2);
plot(peak_time, peak_value, 'r*');
plot(time(end), final_value, 'r*', 'LineWidth', 2);
xlabel('t (sec)');
ylabel('y(t)');
title('Step Responses', 'FontSize', 14);

tf with properties:

    Numerator: {[0 0.3000 7]}
    Denominator: {[2 6 20]}
        Variable: 's'
        IODelay: 0
        InputDelay: 0
        OutputDelay: 0
            Ts: 0
            TimeUnit: 'seconds'
        InputName: {}
        InputUnit: {}
        InputGroup: [1x1 struct]
        OutputName: {}
        OutputUnit: {}
        OutputGroup: [1x1 struct]
            Notes: [0x1 string]
        UserData: []
            Name: ''
        SamplingGrid: [1x1 struct]

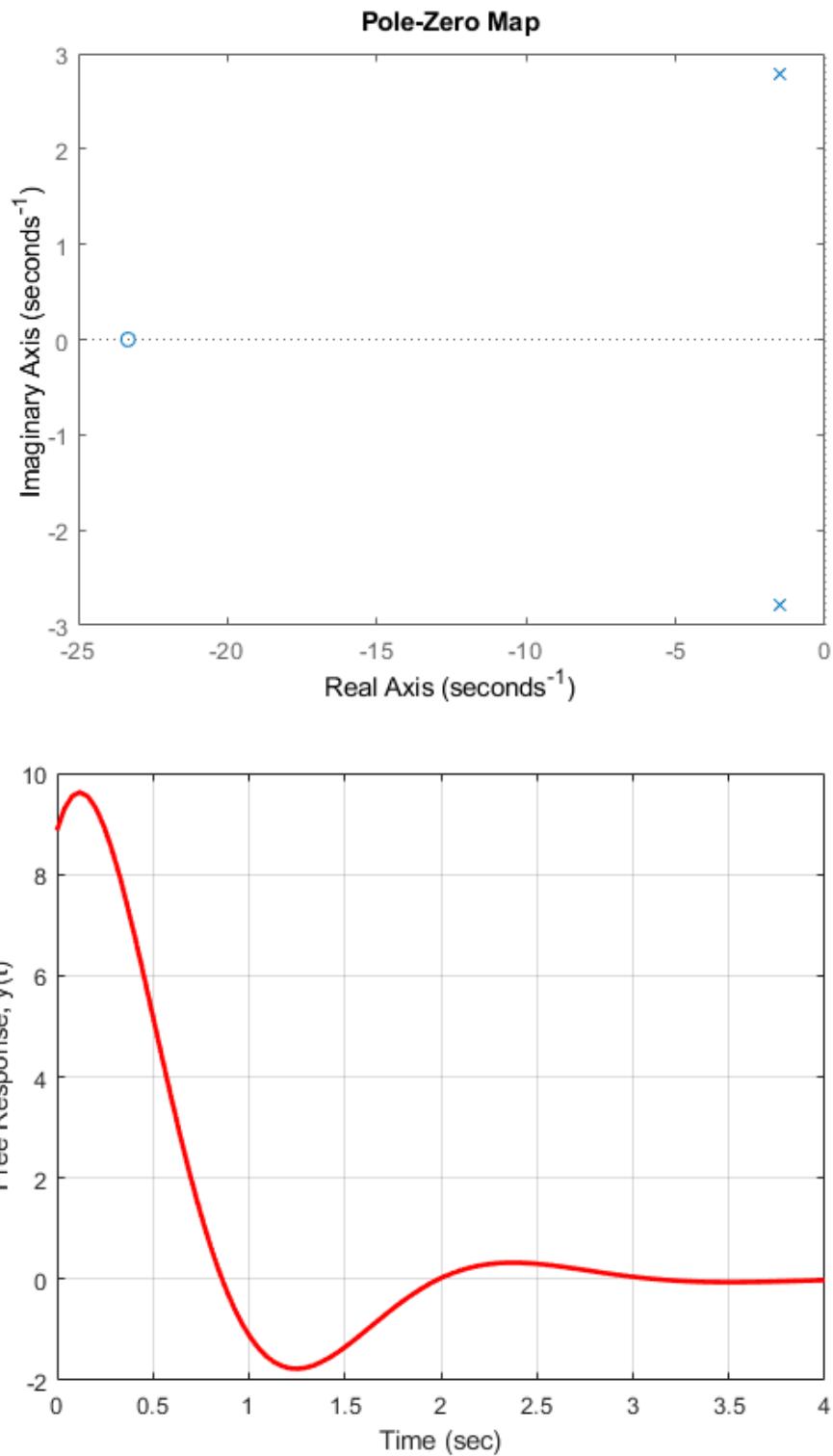
ss with properties:

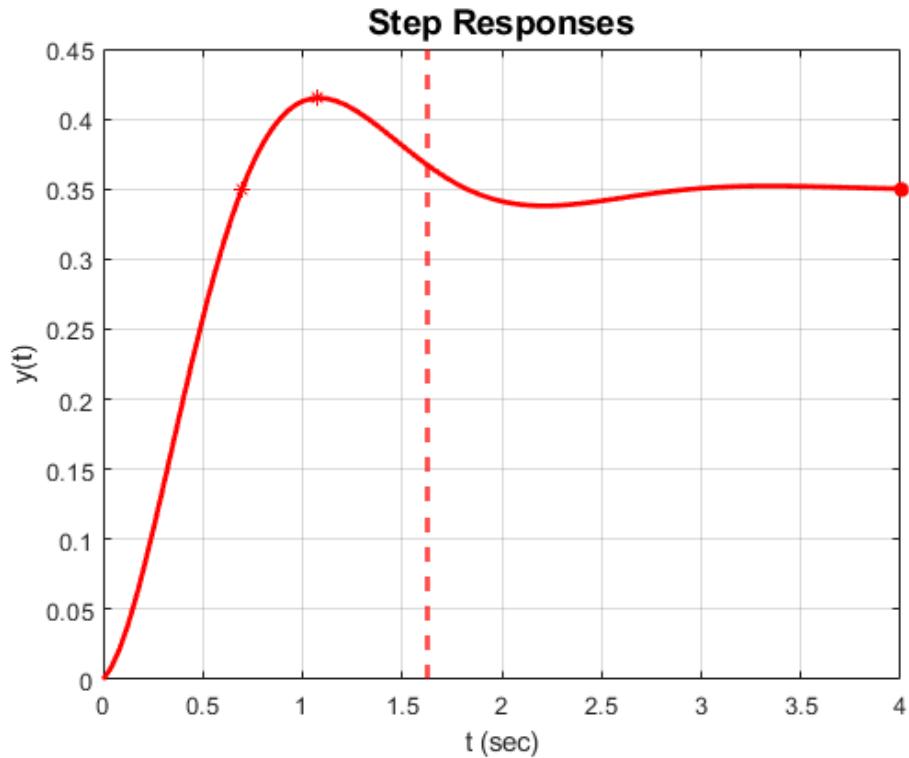
A: [2x2 double]

```

```
B: [2x1 double]
C: [0.1500 3.5000]
D: 0
E: []
Scaled: 0
StateName: {2x1 cell}
StatePath: {2x1 cell}
StateUnit: {2x1 cell}
InternalDelay: [0x1 double]
InputDelay: 0
OutputDelay: 0
Ts: 0
TimeUnit: 'seconds'
InputName: {''}
InputUnit: {''}
InputGroup: [1x1 struct]
OutputName: {''}
OutputUnit: {''}
OutputGroup: [1x1 struct]
Notes: [0x1 string]
UserData: []
Name: ''
SamplingGrid: [1x1 struct]

The LTI system is 0.0 s^2 + 0.15 s + 3.5
-----
1.0 s^2 + 3.0 s + 10.0
The zero of the system is -23.3.
The poles of the system are -1.500 + 2.784j and -1.500 + -2.784j.
The natural frequency of the system is 3.2.
The natural frequency of the system is 3.2.
The damping ratio of the system is 0.5.
The damping ratio of the system is 0.5.
The eigenvalues of A are -1.50 + 2.78j and -1.50 +-2.78j.
The corresponding eigenvectors are [ 0.95 -0.14+-0.27j]^T and [ 0.95
-0.14+ 0.27j]^T.
```





## An example demonstrating the effects of an additional pole.

```
% Create a second system with another faster pole.
system2 = tf(1, [1/6 1])*transfer_function;

% Solve/simulate step response and get characteristics.
output2 = step(system2, time);
step_info2 = stepinfo(system2, 'RiseTimeLimits', [0, 1], ...
    'SettlingTimeThreshold', 0.05);

% Estimate the steady-state output, a.k.a. final value, a.k.a. DC
% gain.
final_value2 = dcgain(system2);

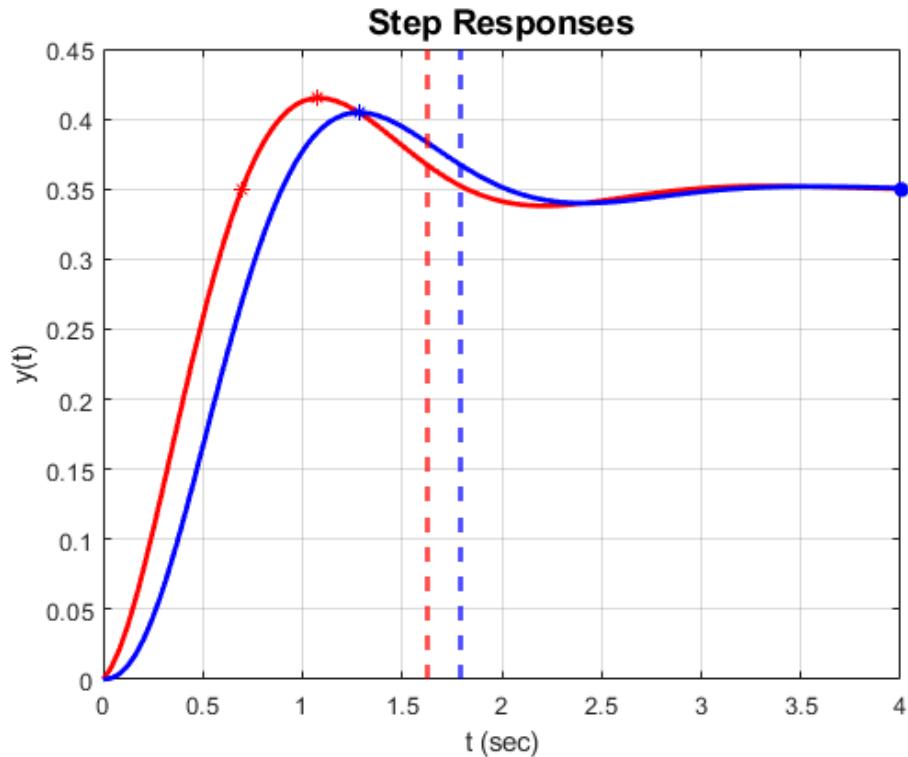
% Extract rise time, settling time, peak time, and peak value.
rise_time2 = step_info2.RiseTime;
settling_time2 = step_info2.SettlingTime;
peak_time2 = step_info2.PeakTime;
peak_value2 = step_info2.Peak;

% Plot step response.
plot(time, output2, 'b-', 'LineWidth', 2);
```

```

xline(settling_time2, 'Color', 'b', 'LineStyle', '--', 'LineWidth',
2);
plot(peak_time2, peak_value2, 'b*');
plot(time(end), final_value2, 'b*', 'LineWidth', 2);

```



## An example demonstrating the effects of an additional LHP zero.

```

% Create a third system with a left half plane (LHP) zero.
system3 = tf([1/6 1], 1)*transfer_function;

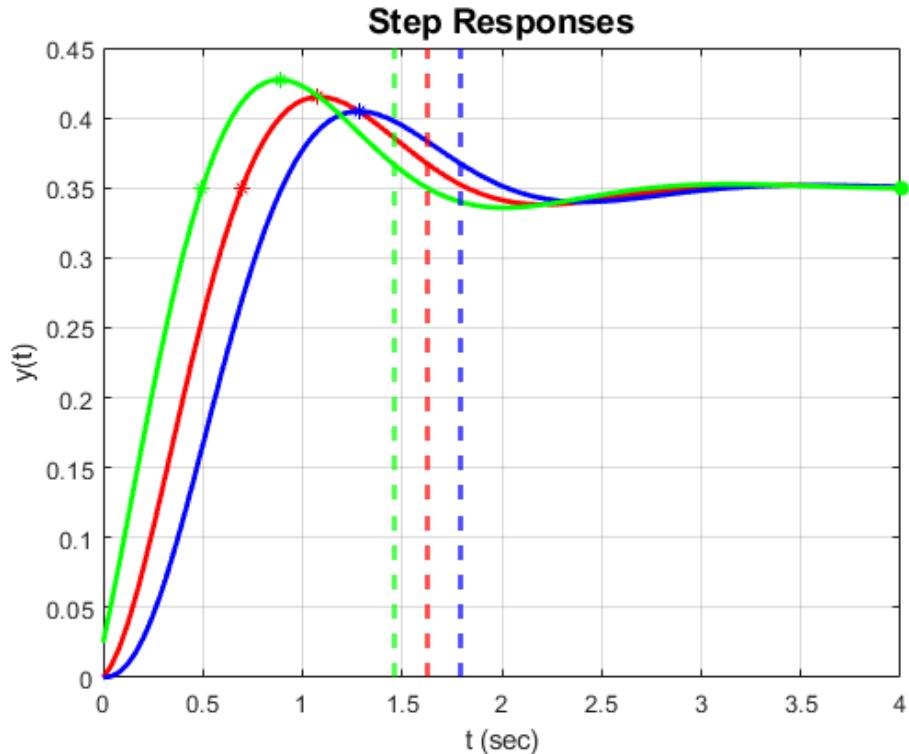
% Solve/simulate step response and get characteristics.
output3 = step(system3, time);
step_info3 = stepinfo(system3, 'RiseTimeLimits', [0, 1], ...
    'SettlingTimeThreshold', 0.05);

% Estimate the steady-state output, a.k.a. final value, a.k.a. DC
% gain.
final_value3 = dcgain(system3);

% Extract rise time, settling time, peak time, and peak value.
rise_time3 = step_info3.RiseTime;
settling_time3 = step_info3.SettlingTime;
peak_time3 = step_info3.PeakTime;
peak_value3 = step_info3.Peak;

```

```
% Plot step response.
plot(time, output3, 'g-', 'LineWidth', 2);
plot(rise_time3, final_value3, 'g*');
xline(settling_time3, 'Color', 'g', 'LineStyle', '--', 'LineWidth',
2);
plot(peak_time3, peak_value3, 'g*');
plot(time(end), final_value3, 'g*', 'LineWidth', 2);
```



## An example demonstrating the effects of an additional RHP zero.

```
% Create a third system with a right half plane (RHP) zero.
system4 = tf([-1/6 1], 1)*transfer_function;

% Solve/simulate step response and get characteristics.
output4 = step(system4, time);
step_info4 = stepinfo(system4, 'RiseTimeLimits', [0, 1], ...
    'SettlingTimeThreshold', 0.05);

% Estimate the steady-state output, a.k.a. final value, a.k.a. DC
% gain.
final_value4 = dcgain(system4);

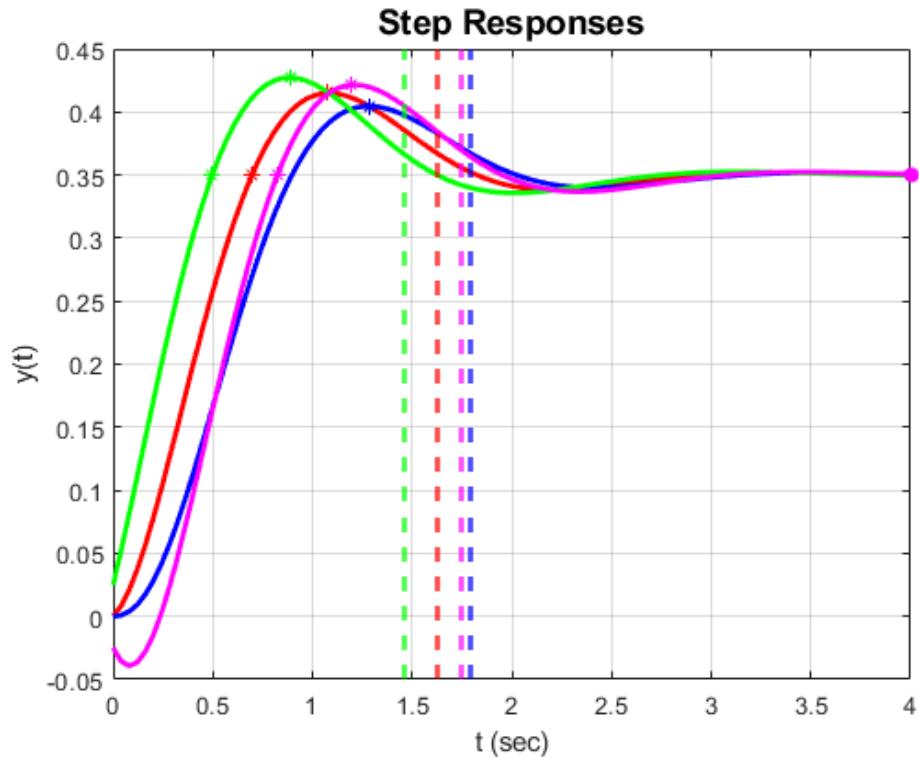
% Extract rise time, settling time, peak time, and peak value.
```

```

rise_time4 = step_info4.RiseTime;
settling_time4 = step_info4.SettlingTime;
peak_time4 = step_info4.PeakTime;
peak_value4 = step_info4.Peak;

% Plot step response.
plot(time, output4, 'm-', 'LineWidth', 2);
plot(rise_time4, final_value4, 'm*');
xline(settling_time4, 'Color', 'm', 'LineStyle', '--', 'LineWidth',
2);
plot(peak_time4, peak_value4, 'm*');
plot(time(end), final_value4, 'm*', 'LineWidth', 2);

```



## Solve/simulate a differential equation both non-linearly and linearly.

This system is based on the nonlinear state equation in Lecture 3, Example Problem 1.

```

% Set time, coefficients, input, and initial conditions for system.
time_start = 0;
time_final = 5;
parameters.time = linspace(0, 5, 100);
parameters.k_D = 0.1388;
parameters.k_L = 0.7654;
parameters.g = 9.81;

```

```

input_array = 3 + cos(2*parameters.time).^2;
initial_conditions = [1.5, 1.5];

% Set trim point of this system.
trim_state = [1 2];
trim_input = 3;

% ode45() solves a differential equation over a certain time span for
% certain initial conditions and input. This function uses the Runge-
Kutta
% (4,5) formula for the numerical integration. The differential
equation
% is specified as a function (given at the bottom of this script).
[time, nonlin_sol] = ode45(@(time, state) state_equation(time,
state, ...
    input_array, parameters), [time_start, time_final], ...
    initial_conditions);

% lsim() solves a linear differential equation over a certain time
span for
% certain initial conditions. The differential equation is specified
as a
% a state-space or transfer function. The state and input matrices
come
% from lecture 3, example problem 1. The output and feedthrough
matrices
% are chosen so that lin_sol contains the solution for both states.
state_mat = [-4 4; -9 0];
input_mat = [0; 6];
output_mat = eye(2);
feed_mat = zeros(2, 1);
lin_sol = lsim(state_mat, input_mat, output_mat, feed_mat, ...
    input_array - trim_input, parameters.time, initial_conditions ...
    - trim_state);

% The nonlinear and linear simulations are plotted for the second
state.
figure(5)
hold on; grid on;
plot(time, nonlin_sol(:, 2), 'r-', 'LineWidth', 2);
plot(parameters.time, trim_state(2) + lin_sol(:, 2), 'b-', 'LineWidth', 2);
xlabel('Time (sec)', 'FontSize', 14);
ylabel('x_2 Response', 'FontSize', 14);
title('Nonlinear vs Linear Simulation', 'FontSize', 14);
legend('NL', 'Lin', 'Location', 'Best');

```

## This section provides the function for ode45 function.

```

function dstate_dtime = state_equation(time, state, input_array, ...
parameters)

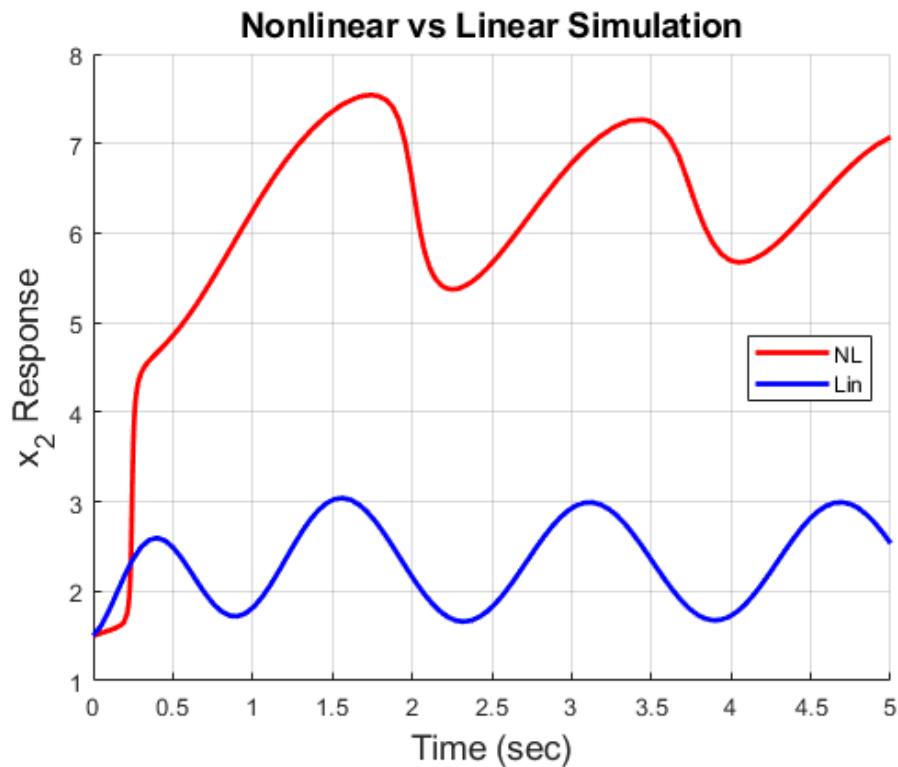
```

```
% STATE_EQUATION
% This function contains the state equation used with the MATLAB
% function
% ode45() as an example.

% Interpolate the input array at the current time.
input = interp1(parameters.time, input_array, time);

dstate_dtime = [-parameters.k_D*(state(1))^2 -
parameters.g*sin(state(2)) ...
+ input; parameters.k_L*state(1) - (parameters.g/state(1))...
*cos(state(2))];

end
```



Published with MATLAB® R2021a

## A.2 Dynamical Systems in Python

Python is an open-source, multi-purpose, scripting language where one can execute any line of code without having to compile it into an executable. A **script** in Python is a collection of lines of code and denoted with file extension ‘.py.’ To run a script, one must use an interpreter in order that the computer through the command line can interpret the Python code. An alternative to the command line is a integrated development environment (IDE) which has a built-in interpreter, interface, and other features for writing and executing scripts. One popular Python distribution is Anaconda which installs the Python 3.7 interpreter within the Spyder IDE with an interface similar to the MATLAB IDE. Spyder’s editor also has a feature which will automatically check that your code adheres to the PEP-008 style which is often used in Python software development.

Python was designed to be a language for fast prototyping of software. Thus, readability and succinct syntax was paramount. However, since Python was open-source, many developers in the mathematics and scientific world developed two big libraries known as NumPy and SciPy to give Python functions to perform mathematical computations on par with other languages. **NumPy (Numerical Python)** is the fundamental package for numerical computations using matrices and basic operations for them. This package will provide us the basic linear algebra functions to design and test GNC algorithms. **SciPy (Scientific Python)** is a collection of numerical algorithms and domain-specific toolboxes in mathematics and science. This includes algorithms in signal processing, optimization, and control theory. Both of these are continually in development by the Python community. Another important package is **Matplotlib** which provides functionality for publication-quality 2D plots and some 3D plotting tools which still needs further development to reach the levels of other tools.

This appendix provides the basic syntax of Python and NumPy/SciPy which does not contain many complex operators (like in C/C++), but is intended as a more readable language. Specifically, many times Python uses explicit words for some operations. It uses indentation explicitly for conditionals, loops, functions, and continuation lines, thus minimizing the extra characters needed for syntax. There’s no end line character, such as the ‘;’ in C/C++ or MATLAB. Next, comparisons between NumPy/SciPy and MATLAB will be provided which can be further explored at [scipy.org](http://scipy.org). Lastly, this section will discuss matplotlib and provide an example. More resources for Matplotlib are available at [https://matplotlib.org/2.0.0/users/pyplot\\_tutorial.html](https://matplotlib.org/2.0.0/users/pyplot_tutorial.html).

The basic syntax for basic input/output and reading/writing data is given as follow.

- Output:

```
print('GNC')
```

```
>> GNC
```

- Input:

```
a = input('Enter value:  ')
```

- Open file:

```
f = open('file.txt')
```

- Can read or write on file

- Read file:

```
f = open('file.txt', 'r')
```

- All content:

```
print(f.read())
```

- One line:

```
print(f.readline())
```

- Write file:

```
f = open('file.txt', 'w')
```

and

```
f.write('GNC \n')
```

- Close file:

```
f.close()
```

Python has six basic data types which are often used for control and estimation.

- Numbers

- Integer:

```
a = 1
```

- Float:

```
a = 1.1
```

- Complex:

```
a = 1 + 1j
```

- String

- Examples:

```
a = 'Welcome'
```

or

```
b = "Home"
```

- Concatenate:

```
print(a+b)
```

```
>> WelcomeHome
```

- Repeat:

```
print(b*3)
```

```
>> HomeHomeHome
- Slice:
  print(a[0:1])
>> We
- Immutable:
  b[0] = 'N'
Out: TypeError
```

- List

- Example:
 

```
a = [2, 3, 'red']
```
- Mutable:
 

```
a[1] = 'blue'
print(a)
Out: [2, 'blue', 'red']
```

- Tuple

- Example:
 

```
a = (2, 3, 'red')
```
- Immutable:
 

```
a[1] = 'blue'
TypeError
```

- Set

- Examples:
 

```
a = {2, 'red'}
or
b = {3, 2}
```
- Union:
 

```
print(a | b)
Out: {2, 3, 'red'}
```
- Intersection:
 

```
print(a & b) {2}
```
- Unordered:

```
a[0] = 3
```

Out: TypeError

- Dictionary

- Example:

```
a = {1:2.2, 2:'red', 3:8}
```

- First integer is the key

- Retrieval:

```
print(a[2])
```

Out: red

It is important to note that String, List, Tuple, Set, Dictionary are **sequences** data types and that sequences can also contain other sequences.

Python has seven primary operator types denoted as follow.

- **Arithmetic Operators**

- Addition: +
- Subtraction: -
- Multiplication: \*
- Division: /
- Modulus: %
- Floor division: //
- Exponentiation: \*\*

- **Bitwise Operators** (Calculations on binaries of numbers)

- AND: &
- OR: |
- NOT: ~
- XOR: ^
- Right shift: >>
- Left shift: <<

- **Assignment Operator =**

- Any arithmetic or bitwise operation can be combined with assignment
  - \* E.g. a /= 1 equivalent to a = a // 1

- **Comparison Operators** (Boolean output, i.e. True or False)

- Equal to: ==
- Not equal to: !=
- Greater than: >
- Greater than or equal to: >=
- Less than: <
- Less than or equal to: <=

- **Logic Operators** (Boolean output, i.e. True or False)

- AND: and
- OR: or
- NOT: not

- **Identity Operators** (Boolean output, i.e. True or False)

- Checks memory location identity
- `a is b`  
 Out: True if a and b are identical  
`a is not b`  
 Out: True if a and b are not identical

- <5-> **Membership in Sequence Operators** (Boolean output, i.e. True or False)

- `a in b`  
 Out: True if value a is found in sequence b  
`a not in b`  
 Out: True if value a is not found in sequence b

For conditionals, loops, and functions, Python uses indentation and the ‘:’ symbol. The only *natural* conditional in Python is the If-Else block, an example of which is

```
if(Boolean expression):
    # Block of code
elif(Boolean expression):
    # Block of code
else:
    # Block of code
```

There are two loops in Python, the While and For loops. The While loop syntax is

- `while(expression):`

# Block of code

and the For loop syntax is

`for variable in sequence:`

# Block of code

For functions in Python, one uses the following syntax

```
def function_name(arguments):
    # Block of code
    return x, y (optional)
```

The arguments for functions can be any data type and can be specified in one of the following four forms: required, default, variable-length non keyworded, and variable-length keyworded.

An example of required arguments is `def my_function(a, b):` which can be called with `x, y = my_function(2, 3)` or `x, y = my_function(a = 2, b = 3)` which assigns a and b with **keyworded arguments**. This is often abbreviated as `kwargs` in Python documentation.

An example of default arguments is `def my_function(a=1):` which can be called with `x, y = my_function(2)` which assigns `a = 2`. Alternatively, it can be called with `x, y = my_function()` which assigns default `a = 1`.

Non-keyworded variable-length arguments are denoted using an `*`, e.g. `def my_function(*a):`, and keyworded variable-length arguments are denoted using `**` `def my_function(**a):`.

Lastly, Python allows for importing other scripts to be used in the current session of Python. These are typically designated as modules, packages, or libraries. A **Module** contains definitions (e.g. constants, functions) which can be imported to other scripts. It contain a list of multiple functions which can be accessed using the `'.'` notation (e.g. `my_module.my_function()`) or can be imported individually (e.g. `from my_module import my_function1, my_function2`). **Packages** are a specific type of module which uses a particular directory structure to make organization and importing easier for users. The term **Library** is used for a ‘published’ module or package and typically contains no scripts which are meant to be ‘run.’ It simply provides definitions.

`matplotlib` whose primary package will be `pyplot`. This is often imported similar to `import matplotlib.pyplot as plt`. `pyplot` primarily contains functions which serve as wrappers around `matplotlib`’s object-oriented interface with every `figure` object. This allows `matplotlib` to function similar to the basic plotting tools in MATLAB. Some of these functions are

- `plot`
- `scatter`
- `hist`
- `title`

- xlabel
- ylabel
- legend

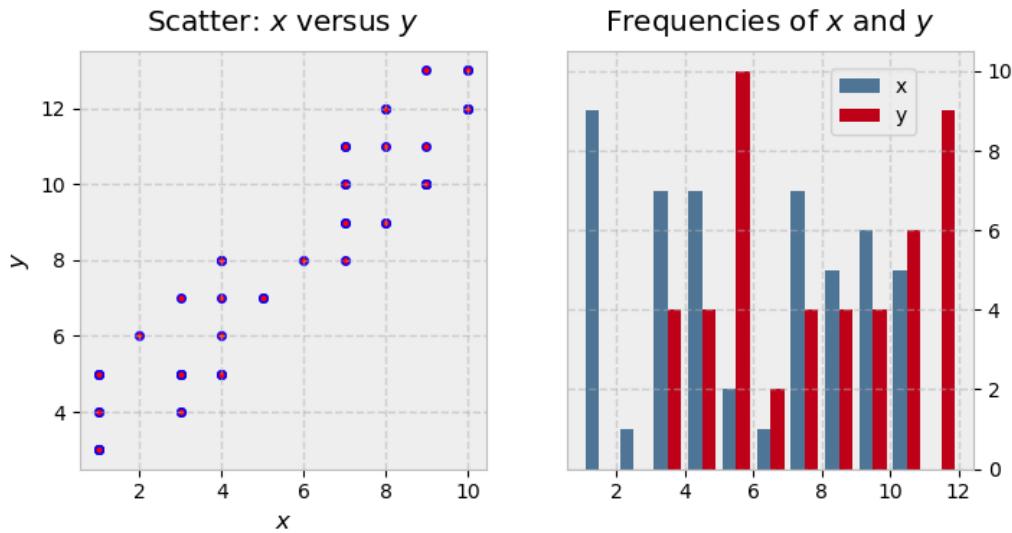
Each `figure` object also contains objects including `axes` and `axis`. By using the `subplot` command, one is able to use multiple `Axes` objects in one `Figure` object. Below is an example of `pyplot` for subplots.

```
import matplotlib.pyplot as plt
import numpy as np

# Code for generating x, y, and data

fig, (ax1, ax2) = plt.subplots(nrows=1, ncols=2, figsize=(8, 4))
ax1.scatter(x=x, y=y, marker='o', c='r', edgecolor='b')
ax1.set_title('Scatter: $x$ versus $y$')
ax1.set_xlabel('$x$')
ax1.set_ylabel('$y$')
ax2.hist(data, bins=np.arange(data.min(), data.max()), label=('x', 'y'))
ax2.legend(loc=(0.65, 0.8))
ax2.set_title('Frequencies of $x$ and $y$')
ax2.yaxis.tick_right()
```

which generates the following figure.



### A.3 MATLAB and Python Comparison

MATLAB and NumPy/SciPy have many commonalities, but with noticeable differences as NumPy/SciPy were created to do numerical and scientific computing in the most natural way with the Python language, and not to be MATLAB clones. Some of the main programming differences between the two are provided in the following table.

Characteristic	MATLAB	NumPy/SciPy
Function/File Access	anywhere on path	require import statements, e.g. <code>import numpy as np</code> or <code>from numpy import linalg</code>
Basic type	multidimensional floating point array	<code>np.array</code> object
Standard operators	matrix operations	element-wise
Indexing starts at	1, e.g. <code>a(1)</code>	0, e.g. <code>a[0]</code>
Scan order	Fortran	'C'
Parameter passing	call-by-value	call-by-object
Array slicing	copy parts of an array	views into an array

Some useful linear algebra equivalents between NumPy and MATLAB are

MATLAB	NumPy/SciPy
<code>[1 2 3; 4 5 6]</code>	<code>np.array([[1., 2., 3.], [4., 5., 6.]])</code>
<code>[a b; c d]</code>	<code>np.block([[a, b], [c, d]])</code>
<code>ndims(a)</code>	<code>np.ndim(a)</code> <code>a.ndim</code>

<code>size(a)</code>	<code>np.shape(a)</code> <code>a.shape</code>
<code>numel(a)</code>	<code>np.size(a)</code> <code>a.size</code>
<code>a(end)</code>	<code>a[-1]</code>
<code>a(2, 5)</code>	<code>a[1, 4]</code>
<code>a(2, :)</code>	<code>a[1]</code> <code>a[1, :]</code>
<code>a(1:5, :)</code>	<code>a[0:5]</code> <code>a[:5]</code> <code>a[0:5, :]</code>
<code>a(end-4:end, :)</code>	<code>a[-5:]</code>
<code>a(1:3, 5:9)</code>	<code>a[0:3][:, 4:9]</code>
<code>a([2, 4, 5], [1, 3])</code>	<code>a[np.ix_([1, 3, 4], [0, 2])]</code>
<code>a(3:2:21, :)</code>	<code>a[2:21:2, :]</code>
<code>a(1:2:end, :)</code>	<code>a[::-2, :]</code>
<code>flipud(a)</code>	<code>a[::-1, :]</code>
<code>a([1:end 1], :)</code>	<code>a[np.r_[:len(a), 0]]</code>
<code>a.'</code>	<code>a.transpose()</code> <code>a.T</code>
<code>a'</code>	<code>a.conj().transpose()</code> <code>a.conj().T</code>
<code>a * b</code>	<code>a @ b</code>
<code>a .* b</code>	<code>a * b</code>
<code>a ./ b</code>	<code>a / b</code> <code>a[1, :]</code>
<code>a.^b</code>	<code>a**b</code>
<code>(a&gt;2)</code> gives matrix of 0s and 1s	<code>(a&gt;2)</code> gives matrix of False and True
<code>find(a&gt;2)</code>	<code>np.nonzero(a&gt;2)</code>
<code>a = b</code>	<code>a = b.copy()</code>
<code>a = b(:)</code>	<code>a = b.flatten()</code>
<b>MATLAB</b>	<b>NumPy/SciPy</b>
<code>0:9</code>	<code>np.arange(10.)</code> <code>np.r_[10.]</code>
<code>1:10</code>	<code>np.arange(1., 11.)</code> <code>np.r_[1.:11.]</code>
<code>[1:10]'</code>	<code>np.arange(1., 11.)[:, newaxis]</code>
<code>[a b]</code>	<code>np.concatenate((a, b), 1)</code> <code>np.hstack((a, b))</code> <code>np.c_[a, b]</code>
<code>[a; b]</code>	<code>np.concatenate((a, b), 1)</code>

	<code>np.vstack((a, b))</code> <code>np.r_[a, b]</code>
<code>zeros(2, 3, 4)</code>	<code>np.zeros((2, 3, 4))</code>
<code>ones(2, 3)</code>	<code>np.ones((2, 3))</code>
<code>eye(2)</code>	<code>np.eye(2)</code>
<code>diag(a)</code>	<code>np.diag(a)</code>
<code>linspace(2, 3, 4)</code>	<code>np.linspace(2, 3, 4)</code>
<code>repmat(a, m, n)</code>	<code>np.tile(a, (m, n))</code>
<code>sort(a)</code>	<code>np.sort(a)</code>
<code>[x, y] = meshgrid(0:5, 0:3)</code>	<code>x, y = np.ix_(np.r_[0:6.], r_[0:4.])</code>
<code>[x, y] = meshgrid([1, 2, 3], [2, 3, 4])</code>	<code>x, y = np.ix_([1, 2, 3], [2, 3, 4])</code>
<code>max(a)</code>	<code>a.max(0)</code>
<code>max(max(a))</code>	<code>a.max()</code>
<code>max(a, [], 2)</code>	<code>a.max(1)</code>
<code>max(a, b)</code>	<code>np.maximum(a, b)</code>
<code>norm(a)</code>	<code>np.linalg.norm(a)</code>
<code>inv(a)</code>	<code>np.linalg.inv(a)</code>
<code>pinv(a)</code>	<code>np.linalg.pinv(a)</code>
<code>rank(a)</code>	<code>np.linalg.matrix_rank(a)</code>
<code>a\b</code>	<code>np.linalg.solve(a, b) if a is square</code> <code>np.linalg.lstsq(a, b) otherwise</code>
<code>b\ a</code>	<code>np.linalg.solve(a.T, b.T).T if a is square</code> <code>np.linalg.lstsq(a.T, b.T).T otherwise</code>
<code>[V, D] = eig(a, b)</code>	<code>np.linalg.eig(a, b)</code>
<code>squeeze(a)</code>	<code>a.squeeze()</code>
<code>rand(2, 3)</code>	<code>np.random.rand(3, 4)</code>
<code>rng(100)</code>	<code>np.random.seed(3)</code>

## A.4 Optimal Control and Estimation in Python

In Python, the function

```
scipy.linalg.solve_continuous_are(a, b, q, r, e=None, s=None, balanced=True)
```

can be used to solve the CARE for the unconstrained infinite horizon continuous-time LQR. This function uses a solver that forms the extended Hamiltonian matrix pencil, and then uses a QZ decomposition method. It should be noted that the `e` in this function is not used for LQR control.

An example script of this is

```
import numpy as np
from scipy import linalg as la
A = np.array(...)
B = np.array(...)
```

```

Q = np.array(...)

R = np.array(...)

la.solve_continuous_are(A, B, Q, R, S=np.array(...))

K = la.inv(R)@(B.T@P + S.T)

```

For the finite horizon continuous-time LQR, one must solve the Riccati differential equation *backwards* in-time, thus one can form the reverse of the matrix  $P$  as

$$\dot{P}' = P'A + A^TP' - (P'B + S)R^{-1}(B^TP' + S^T) + Q \quad (\text{A.1})$$

and using `scipy.integrate.odeint()` as a numerical integrator with “initial” condition

$$P'(0) = E \quad (\text{A.2})$$

which reflects the final condition of the LQR problem. One can obtain  $P'$  which can be reversed to get  $P$  for the LQR controller. One can solve the infinite horizon LQR OCP using a function within the SciPy package. Similar to continuous-time, SciPy has a solver for the DARE using the function

`scipy.linalg.solve_discrete_are(a, b, q, r, e=None, s=None, balanced=True)`

which solves the problem using by forming the extended symplectic matrix pencil and uses a QZ decomposition method. Again, it should be noted that `e` is never used for the LQR DARE.

An example script using this function is given below

```

import numpy as np

from scipy import linalg as la

F = np.array(...)

G = np.array(...)

Q = np.array(...)

R = np.array(...)

P = la.solve_discrete_are(F, G, Q, R, S=np.array(...))

K = la.inv(G.T@P@G + R)@(G.T@P@F + S.T)

```

For the finite horizon discrete-time LQR OCP, one finds the solution by starting with the final condition

$$P_N = E \quad (\text{A.3})$$

and using the dynamic Riccati equation as a backwards recursive in time, i.e.

$$P_{k-1} = F^TP_kF + Q - \left( F^TP_kG + S \right) \left( G^TP_kG + R \right)^{-1} \left( G^TP_kF + S^T \right) \quad (\text{A.4})$$

which is simply a matrix equation. Then, using the sequence of  $P_k$ , one can then compute the *forward* sequence of control gain matrices as

$$K_k = \left( G^T P_k G + R \right)^{-1} \left( G^T P_k F + S^T \right) \quad (\text{A.5})$$

which are used in the state feedback control, i.e.

$$u_k^* = -K_k \vec{x}_k \quad (\text{A.6})$$

The function `scipy.optimize.minimize` can be used for optimal control problems. An example script is as follows:

```
from scipy.optimize import minimize, rosen, rosen_der

x0 = [1.3, 0.7, 0.8, 1.9, 1.2]

result = minimize(rosen, x0, method='BFGS', jac=rosen_der, tol=1e-6)

result.x
```

which would output the following

```
array([ 1.,  1.,  1.,  1.,  1.])
```

It should be noted that the Rosenbrock function is used in this example, a standard function used to test various optimization methods. This function is given by the following equation

$$\sum_{i=1}^{N-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_i^2) \quad (\text{A.7})$$

Secondly, it should be noted that the ‘BFGS’ is a quasi-Newton method is an optimization method which approximates the Hessian of the function to be optimized and will be discussed for nonlinear OCPs in future sections.

The SciPy package `scipy.stats` contains *many* distribution **objects**, including the uniform distribution object given by `scipy.stats.uniform`, and the normal distribution object given by `scipy.stats.norm` which requires the use of `loc` and `scale` to use the general Gaussian distribution as the default is for the standard normal.

In this package, each distribution object has many common **methods** of which the most relevant to this textbook are: generating random samples using `rvs(loc=0, scale=1, size=1, random_state=None)`, computing values of the PDF using `pdf(x, loc=0, scale=1)`, and computing values fo the CDF using `cdf(x, loc=0, scale=1)`.

The SciPy stats package has only one multivariate distribution which is the multivariate normal, i.e. the standard version of the multivariate Gaussian with  $\vec{\mu} = 0$  and  $\Sigma = I$ . The call `scipy.stats.multivariate_normal` creates the distribution object which has the following relevant methods:

- CDF value at  $\vec{x}$ : `cdf(x, mean=None, cov=1, allow_singular=False,`

```
maxpts=1000000*dim, abseps=1e-5, releps=1e-5)
```

- PDF value at  $\vec{x}$ : `pdf(x, mean=None, cov=1)`
- Random samples: `rvs(mean=None, cov=1)`

The OLS problem can be solved in Python using the SciPy `optimize` package. For the OLS problem, the `scipy.optimize.lsq_linear()` function can be used which handles constraints in the free variables. This calls the `numpy.linalg.lstsq()` or the `numpy.linalg.lsmr()` functions to solve the unconstrained problem and iterates intelligently until the constraints are satisfied. Lastly, it should be noted that for each of these functions, SciPy includes a factor of  $\frac{1}{2}$  to the cost function, which is commonly found in quadratic optimization problem formulations due to the derivative canceling the  $\frac{1}{2}$  factor in the computations.

The Nonlinear LS problems can be solved in Python using the SciPy `optimize` package. For the Nonlinear LS, the `scipy.optimize.least_squares()` function can be used which uses trust-region solvers. This function uses a slightly different notation for the residual as any function `f()`, not necessarily the difference function above. It also includes a loss function option `rho(z)` which by default is just `z`. Lastly, it should be noted that for each of these functions, SciPy includes a factor of  $\frac{1}{2}$  to the cost function, which is commonly found in quadratic optimization problem formulations due to the derivative canceling the  $\frac{1}{2}$  factor in the computations.

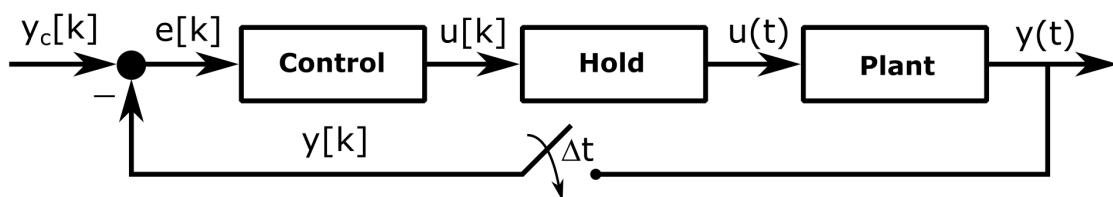
## Appendix B

# Miscellaneous Topics

### B.1 Discrete-Time Feedback Control Systems

Before modern computers, control design was performed using standard circuit components to implement continuous-time control laws, i.e. “classical” control. However, modern feedback control systems are implemented by microprocessors or digital computers, (a.k.a. digital control where time and values are discrete). Thus, there are additional considerations that must be taken for hardware implementation of the control system concepts discussed in this course. One approach is to use the continuous-time control design methods in this course and then convert the continuous-time control system to a discrete-time/digital control system, i.e. **discretization**. However, one may also directly design the digital feedback control system by modeling the discrete-time effects as a continuous-time process and using discrete-time control methods for the controller. In either case, one should be generally familiar with digital system modeling. The additional effects of discrete-values in truly digital systems is left to the reader, but generally has little effect if the numerical precision of the digital computer is high enough relative to the system dynamics.

In general, the block diagram for a **digital feedback control system** can be constructed as the following.



This system notably consists of the following digital processing steps in addition to the continuous-time plant.

- **Sampling:** The output of the system,  $y(t)$  is updated in the digital system every  $\Delta t$  forming the  $k^{\text{th}}$  sampled output,  $y[k]$ , where  $k = 1, 2, \dots$  is the time step, i.e.  $t = k\Delta t$ . This is typically performed by an analog-to-digital converter (ADC).

- **Control Update:** The control input at time step  $k$ ,  $u[k]$ , is updated using a digital computer. This is typically performed by a microprocessor with a “real-time” operating system (RTOS).
- **Hold:** The updated control input,  $u[k]$ , is converted into a continuous-time signal  $u(t)$  for the physical (e.g. electro-mechanical) system. This is typically performed by a digital-to-analog converter (DAC).

It should be noted that the **digital signal processing** (DSP) from system to system will also cause some small time delays that should be modeled in the full system analysis and typically would also include an actuator in the plant model. Furthermore, if one is simply discretizing a continuous-time control law to discrete-time without this additional analysis, one must ensure that the sampling time,  $\Delta t$ , is much faster than any relevant dynamics. However, if this is not that case, one must consider the real continuous-time effects of the digital feedback control system primarily including the sampling and hold steps. This lecture will provide a brief introduction to discrete-time LTI system representations, stability, and discretization.

## Difference Equations and Z-Transform

Analogous to continuous-time differential equations for LTI SISO systems, discrete-time LTI SISO systems use **difference equations** whose standard LTI form is

$$\begin{aligned} & y[k] + a_{m-1}y[k-1] + \cdots + a_1y[k-m+1] + a_0y[k-m] \\ & = b_p u[k] + b_{p-1}u[k-1] + \cdots + b_1u[k-p+1] + b_0u[k-p] \end{aligned} \quad (\text{B.1})$$

Similarly, the Laplace transform for representing ODEs as continuous-time transfer functions is analogous to the **z-transform** for representing difference equations as **discrete-time transfer functions**. For a discrete-time signal  $x[n]$ , the z-transform is given by

$$X(z) = \mathcal{Z}\{x[k]\} = \sum_{k=-\infty}^{\infty} x[k]z^{-k} \quad (\text{B.2})$$

where  $z$  is a complex number. This also has an inverse z-transform

$$x[k] = \mathcal{Z}^{-1}\{X(z)\} = \frac{1}{2\pi j} \oint_C X(z)z^{k-1} dz \quad (\text{B.3})$$

where  $C$  is a counterclockwise closed path encircling the origin and entirely in the **region of convergence (ROC)**, i.e. the set of points in the complex plane for which the z-transform summation converges. Thus, for a  $m^{\text{th}}$  order LTI difference equation, the **discrete-time transfer function** is

$$H(z) = \frac{Y(z)}{U(z)} = \frac{b_p z^p + b_{p-1} z^{p-1} + \cdots + b_1 z + b_0}{z^m + a_{m-1} z^{m-1} + \cdots + a_1 z + a_0} \quad (\text{B.4})$$

For reference, some functions in the  $k$  and  $z$  domains are provided in the following table.

Function	$k$ Domain ( $\forall k \geq 0$ )	$z$ Domain
Unit Step	1	$\frac{z}{z-1}$
Power	$a^k$	$\frac{z}{z-a}$
Sine	$\sin(\omega k)$	$\frac{z \sin \omega}{z^2 - 2z \cos \omega + 1}$
Cosine	$\cos(\omega k)$	$\frac{z^2 - z \cos \omega}{z^2 - 2z \cos \omega + 1}$
Power Sine	$a^k \sin(\omega k)$	$\frac{z^2 - z \cos \omega}{z^2 - 2az \cos \omega + a^2}$
Power Cosine	$a^k \cos(\omega k)$	$\frac{az \sin \omega}{z^2 - 2az \cos \omega + a^2}$

One way to describe the similarity between the  $s$  and  $z$  transforms is to see that as the  $s$  variable operates in multiplication and division as a differentiator and integrator in continuous-time, respectively, the  $z$  variable operates in multiplication and division as a backward and forward **time shift operator** in discrete-time, respectively.

For the frequency response of a discrete-time system, one substitutes  $z = e^{j\omega}$ . Thus, the **discrete-time Fourier transform (DTFT)** with periodicity of  $2\pi$  can be written as

$$H(\omega) = \sum_{k=-\infty}^{\infty} h[k] e^{-j\omega k} \quad (\text{B.5})$$

and the inverse discrete-time Fourier transform

$$H[k] = \frac{1}{2\pi j} \int_{-\pi}^{\pi} h(e^{j\omega}) e^{j\omega k} d\omega \quad (\text{B.6})$$

where the substitution  $z = e^{j\omega}$  has set the closed path  $C$  to the unit circle. It should be noted that the **discrete Fourier transform (DFT)** discretely samples the continuous DTFT, i.e.

$$H[n] = \sum_{k=0}^{N-1} h[k] e^{-\frac{j2\pi}{N} nk} \quad (\text{B.7})$$

where  $N$  is the number of evenly-spaced samples. In this way, one can use the z-transform and apply SISO loop-shaping methods based on discrete-time frequency response requirements.

## Discrete-Time LTI State-Space Modeling

The difference equations for LTI SISO systems can be further extended to a discrete-time LTI state-space model which only requires the state samples from the previous time step  $k - 1$  to the current one  $k$ , i.e.

$$\begin{aligned} \vec{x}[k] &= F \vec{x}[k-1] + G \vec{u}[k-1] \\ \vec{y}[k] &= H \vec{x}[k] \end{aligned} \quad (\text{B.8})$$

where  $F$  is the **state transition matrix**,  $G$  is the **discrete-time input matrix**, and  $H$  is the **output matrix**, also known as the **measurement matrix**.

For an initial condition at  $\vec{x}[0]$  and free response dynamics (i.e. no control), the discrete-time state-space can be reduced to

$$\vec{x}[k] = F \vec{x}[k-1] \quad (\text{B.9})$$

which has the simple solution for the state at time step  $k$  as

$$\vec{x}[k] = F^k \vec{x}[0] \quad (\text{B.10})$$

This solution can also be analyzed similar to the continuous-time case by transforming to the Jordan canonical form

$$\vec{x}[k] = V \vec{z}[k] \quad (\text{B.11})$$

where  $V$  is the matrix of  $n$  eigenvectors of  $F$ . Substituting into the equation above, one has

$$\vec{z}[k+1] = \Lambda \vec{z}[k] \quad (\text{B.12})$$

where  $\Lambda$  is in Jordan Form, i.e. diagonal or nearly diagonal. Rewriting the free response, one has

$$\vec{z}[k] = \Lambda^k \vec{z}[0] \quad (\text{B.13})$$

If  $F$  is diagonalizable, then  $\Lambda$  is diagonal and the free response is

$$\vec{z}[k] = \begin{bmatrix} \lambda_1^k & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n^k \end{bmatrix} \vec{z}[0] \quad (\text{B.14})$$

and substituting the inverse transformation,  $\vec{z}(t) = V^{-1} \vec{x}(t)$ , one can solve for the free response solution in the original state as

$$V^{-1} \vec{x}[k] = \begin{bmatrix} \lambda_1^k & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n^k \end{bmatrix} V^{-1} \vec{x}[0] \quad (\text{B.15})$$

$$\vec{x}[k] = V \begin{bmatrix} \lambda_1^k & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n^k \end{bmatrix} V^{-1} \vec{x}[0] \quad (\text{B.16})$$

$$\vec{x}[k] = [\vec{v}_1 \ \cdots \ \vec{v}_n] \begin{bmatrix} \lambda_1^k & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n^k \end{bmatrix} V^{-1} \vec{x}[0] \quad (\text{B.17})$$

$$\vec{x}[k] = [\lambda_1^k \vec{v}_1 \ \cdots \ \lambda_n^k \vec{v}_n] V^{-1} \vec{x}[0] \quad (\text{B.18})$$

and representing

$$V^{-1} \vec{x}[0] = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} \quad (\text{B.19})$$

where  $c_1, \dots, c_n$  are scalar constants, one has

$$\vec{x}[k] = [\lambda_1^k \vec{v}_1 \ \cdots \ \lambda_n^k \vec{v}_n] \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} \quad (\text{B.20})$$

$$\vec{x}[k] = c_1 \lambda_1^k \vec{v}_1 + \dots + c_n \lambda_1^k \vec{v}_n \quad (\text{B.21})$$

which is very similar to the continuous-time solution except the exponential terms have become power terms of the eigenvalues. The same sort of modal characteristics apply here as well. Furthermore, by inspection of the free response solution

$$\vec{x}[k] = c_1 \lambda_1^k \vec{v}_1 + \dots + c_n \lambda_1^k \vec{v}_n \quad (\text{B.22})$$

as  $k$  approaches  $\infty$  each  $i^{\text{th}}$  component with  $|\lambda_i| < 1$  will go  $\rightarrow 0$  while each  $|\lambda_i| = 1$  will remain constant as  $c_i \vec{v}_i$ . Since both of these are bounded, one can assume that discrete-time LTI systems are stable if and only if the eigenvalues of  $F$  have magnitude  $\leq 1$ , though a full proof of this result must include non-diagonalizable  $F$  which can be done in a similar fashion. For control of these discrete-time LTI systems, a discrete-time full-state feedback control law could be designed similarly to the continuous-time case as

$$\vec{u}[k-1] = -K \vec{x}[k-1] \quad (\text{B.23})$$

which can place the eigenvalues accordingly within the complex plane, notably in this case within the unit circle (i.e.  $|\lambda| < 1$ ).

## Discretization

Often one wishes to discretize a continuous-time LTI system, i.e. convert it to a discrete-time LTI system. To discretize general MIMO systems, recall the continuous-time state equation

$$\dot{\vec{x}}(t) = A \vec{x}(t) + B \vec{u}(t) \quad (\text{B.24})$$

which has the general solution

$$\vec{x}(t) = e^{At} \vec{x}(0) + \int_0^t e^{A(t-\tau)} B \vec{u}(\tau) d\tau \quad (\text{B.25})$$

Thus, for every time step from  $k-1$  to  $k$  and  $t = \Delta t$ , one can write

$$\vec{x}[k] = e^{A\Delta t} \vec{x}[k-1] + \int_0^{\Delta t} e^{A(\Delta t-\tau)} B \vec{u}(\tau) d\tau \quad (\text{B.26})$$

which infers that

$$F = e^{A\Delta t} \quad (\text{B.27})$$

Note that this explains the connection between the linear stability conditions on  $F$  and the corresponding linear stability condition on  $A$ , i.e. magnitude of eigenvalues of  $F$  are less than one and the real part of the eigenvalues of  $A$  are negative.

However, for the input matrix, one must assume that  $\vec{u}(\tau)$  follows some type of piecewise function between times steps. To only use the previous time step to approximate the integral, one may use a **zero-order hold (ZOH)** which assumes  $\vec{u}(\tau)$  is a piecewise constant input  $\vec{u}[k-1]$  from  $k-1$  to  $k$ , then

$$\int_0^{\Delta t} e^{A(\Delta t-\tau)} B \vec{u}(\tau) d\tau = \left( \int_0^{\Delta t} e^{A(\Delta t-\tau)} B d\tau \right) \vec{u}[k-1] \quad (\text{B.28})$$

and one has for the discrete-time input matrix

$$G = \int_0^{\Delta t} e^{A(\Delta t - \tau)} B(\tau) d\tau \quad (\text{B.29})$$

Other alternatives include higher-order holds, including a **first-order hold (FOH)** which assumes  $\vec{u}(\tau)$  is a piecewise linear input from  $k - 1$  to  $k$ . However, other holds require knowledge of the inputs at additional time steps, e.g. FOH requires  $u[k]$  and  $u[k - 1]$ . Lastly, note that  $H = C$  in this discretization.

Analogous to the ZOH and FOH, one can also approximate the transfer functions of SISO systems, using the **Euler Forward approximation**, i.e.

$$s \approx \frac{z - 1}{\Delta t} \quad (\text{B.30})$$

or the **Tustin approximation**, i.e.

$$s \approx \frac{2(z - 1)}{\Delta t(z + 1)} \quad (\text{B.31})$$

and another common method is the **Euler Backward approximation**, i.e.

$$s \approx \frac{z - 1}{z\Delta t} \quad (\text{B.32})$$

which is numerically more stable than the forward version due to the backward requiring smaller sampling time  $\Delta t$ . Even so, using these latter two approximation methods to convert controllers typically requires that the sampling time satisfy

$$\Delta t < \frac{\pi}{5\omega_c} \quad (\text{B.33})$$

in order for the controller to *likely* produce satisfactory closed-loop behavior where  $\omega_c$  is the crossover frequency of the open-loop transfer function.

# Bibliography

- [1] R. Nelson, *Flight Stability and Automatic Control*. McGraw-Hill Companies, Inc., 2nd ed., 1998.
- [2] D. K. Schmidt, *Modern Flight Dynamics*. McGraw-Hill Companies, Inc., 1st ed., 2012.
- [3] B. Stevens, F. Lewis, and E. Johnson, *Aircraft Control and Simulation*. John Wiley & Sons, Inc., 3rd ed., 2016.
- [4] E. Lavretsky and K. Wise, *Robust and Adaptive Control with Aerospace Applications*. Springer-Verlag London, 2013.
- [5] D. Simon, *Optimal State Estimation: Kalman,  $H_\infty$ , and Nonlinear Approaches*. Springer Science & Business Media, 2006.
- [6] V. Klein and E. A. Morelli, *Aircraft System Identification: Theory and Practice*. American Institute of Aeronautics and Astronautics, 2006.
- [7] S. Gleason and D. Gebre-Egziabher, *GNSS Applications and Methods*. Artech House, 2009.
- [8] P. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2nd ed., 2013.

# Index

- $L^2$ -norm, 233
- $L^\infty$ -norm, 233
- $L^p$ -norms, 233
- $L^{2,2}$ -norm, 234
- $L^{\infty,\infty}$ -norm, 234
- $\tau$  empirical parameter, 119
  - table, 119
- $p$ -norm, 233
- acceleration, 61
  - centrifugal, 62
  - Coriolis, 62
  - Euler, 62
  - fictitious, 62
- Ackermann's formula, 145
- actuation
  - system, 58
- actuation system, 211
- actuator, 136
- adaptive control, 211
- aerodynamic chord
  - airfoil, 77
- aerodynamic moments, 53
- aerodynamic twisting, 79
- affine transformation, 261
- aileron, 75
- aileron-rudder interconnect, 225
- air density, 77
- airfoil, 76
- airplane trim, 87
- airspeed, 56
- altitude hold, 220
- analog system, 7
- angle of attack, 64
- airfoil, 77
- angular acceleration, 61
- angular velocity
  - three-dimensional, 68
  - two-dimensional, 61
- anhedral, 102
- aspect ratio
  - wing, 79
- asymptotic stability, 243
- attitude control
  - pitch, 219
- auto-correlation function, 266, 280
- auto-covariance function, 267
- autonomous dynamics, 232
- autopilot, 53
- average power
  - random process, 268
- axis
  - lateral, 55
  - longitudinal, 55
  - vertical, 56
- bank angle, 63
- basic rotation matrices, 62
- Bayes filter, 323
- Bayes' Rule
  - events, 263
- Bayes' rule, 284
  - continuous random variables, 263
- Bayesian least-squares
  - estimator, 285
- Bayesian methods, 263
- Bernoulli distribution, 257
- Bernoulli process, 269

- best linear unbiased estimator, 274  
Binomial distribution, 257  
block diagram, 6  
Bode gain-phase formula, 201  
Bode plot, 39  
    asymptotic slope, 47  
    first order, 41  
    higher order, 47  
    real zero, 44  
    second order underdamped, 45  
body frame, 55  
boundary condition, 7  
bounded set, 245  
Brownian Motion, 269  
Brownian motion process, 271  
  
cambered wing, 78  
candidate function  
    Lyapunov, 244  
cardinality, 266  
cascade interconnection, 165  
cascade-loop control, 210  
Cauchy problem, 234  
Cauchy's argument principle, 181  
Cauchy-Peano theorem, 234  
Cayley-Hamilton definition, 250  
center of pressure  
    airfoil, 77  
characteristic equation  
    ordinary differential equation, 8  
    roots, 8  
    SISO feedback control system, 168  
Cholesky decomposition, 242  
closed set, 245  
closed-loop transfer function, 174  
cockpit, 74  
coefficient of determination, 281  
column rank, 249  
compact set, 245  
complementary sensitivity  
    transfer function, 174  
conditional PDF, 263  
confidence region, 265  
  
confidence testing, 265  
consistency  
    estimator, 273  
constrained finite horizon LQR OCP, 311  
constraints  
    inequality, 312  
continuous- to discrete-time, 239  
control  
    automatic, 53  
    closed-loop, 137  
    feedback, 138  
    open-loop, 136  
    semi-automatic, 53  
    system, 58  
control effort, 161  
control gain, 135  
control law, 135  
control power  
    aileron, 88, 124  
    elevator, 88, 119  
    rudder, 88, 125  
controllability, 249  
    matrix, 144  
    output, 252  
    state, 249  
    under constraints, 252  
controllability Gramian, 252  
controllability matrix, 251  
controllable, 144  
Controllable Canonical Form, 18  
convolution integral, 38  
Cooper-Harper Rating Scale, 131  
coordinated flight, 86  
correlation  
    random process, 267  
correlation coefficient, 261  
correlation matrix, 261  
countable set, 256  
countably infinite set, 256  
covariance, 262  
covariance stationarity, 267  
cross-correlation  
    function, 267

- cross-covariance matrix, 262
- cumulative distribution function
  - joint, 260
- damped least-squares, 289
- DC gain, 39
- decibel table, 40
- decision altitude, 215
- decision height, 215
- deconvolution, 38
- definiteness
  - function, 244
  - matrix, 240
- delay margin, 187
- delayed gratification, 289
- delta function
  - Kronecker, 270
- deterministic, 255
- diagonal matrix, 241
- diagonalizable, 240
- difference equation, 231
- digital system, 7
- dihedral angle, 102
- dihedral effect, 102
- dimension
  - vector space, 249
- Dirac delta, 36
- direct Lyapunov method, 244
- direction cosine matrix, 62
- directional plane, 93
- disk margin, 187
- distance measuring equipment, 217
- downwash angle, 93
- drag coefficient
  - induced, 78
  - parasitic, 78
  - wing, 77
  - zero-lift, 78
- drag force, 53
  - airfoil, 77
  - airplane, 81
- dutch roll mode, 128
- dynamic pressure
  - free-stream, 77
- dynamic stability, 99
- dynamical system, 231
  - deterministic, 255
  - random, 255
  - stochastic, 255
  - tychastic, 255
- dynamics
  - autonomous, 232
  - equation, 6
  - time-invariant, 232
  - unforced, 232
- dynamics equation, 15, 231
- Earth mean radius, 71
- Earth-Centered Inertial, 54
- effectiveness
  - aileron, 124
  - elevator, 119
  - rudder, 125
- efficient estimator, 272
- eigenvalue, 144, 239
  - equation, 144
- eigenvalue decomposition, 239
- eigenvalue placement, 144
- eigenvalue problem, 26, 239
- eigenvector, 144, 239
- elevator, 75
- elevons, 76
- elliptical wing, 79
- empennage, 74
- engine, 74
- ensemble average, 268
- entry-wise matrix norm, 234
- equation of motion, 6
- equations of motion
  - Newton-Euler, 67
  - Newton-Euler, rotating frame, 69
- rigid airplane, 80
- rigid flight vehicle, 72
- rotation equation, 67
- six degrees-of-freedom, 69
- translation equation, 67

- equilibrium
  - flight conditions, 85
- equilibrium point, 235
  - autonomous ODE, 10
  - non-autonomous ODE, 10
  - stability, 243
- ergodic, 268
- error
  - estimator, 273
- error ellipse, 264
- error transfer function, 174
- estimator
  - consistency, 273
  - efficient, 272
  - error, 273
  - linear, 274
  - mean, 273
  - variance, 273
- estimator gain matrix, 274
- Euclidean norm, 233
- Euler angle
  - navigation-to-body, 63
  - navigation-to-wind, 63
  - wind-to-body, 64
- Euler angle ambiguity, 63
- Euler angle rates, 70
- Euler angles, 62
- Euler integration, 70
- Euler's formula, 36
- event, 255
  - intersection, 262
  - mutually independent, 263
  - union, 262
- expectation
  - random process, 266
- expectation function, 266
- fair, 256
- feedback control
  - state-space, 143
- feedback interconnection, 166
- feedforward control, 140
- feedforward gain, 140
- feedthrough matrix, 16, 237
- final value, 31
- final value theorem, 176
- finite set, 256
- finite wing theory, 76
- first order approximation, 8
- fit error, 281
- flap, 76
- flat Earth approximation, 55
- flight dynamics and control, 53
- flight path angle, 63
- flying qualities, 129
- force
  - gravitational, 71
  - propulsive, 71
- forces
  - fictitious, 62
- forces and moments
  - aerodynamic, 53
  - gravitational, 53
  - propulsive, 53
  - thrust, 53
  - weight, 53
- Fourier inversion theorem, 36
- Fourier series, 35
- Fourier transform, 35
- free response, 237
  - LTI MIMO continuous-time, 247
  - LTI MIMO discrete-time, 248
- free-stream airflow, 56
- free-stream velocity
  - airfoil, 77
- frequency
  - angular, 36
  - periodic, 36
- Frobenius norm, 234
- full rank, 249
- function definiteness, 244
- fuselage, 74
- gain margin, 184
- gain scheduling, 211
- gambler's ruin, 270

- Gauss-Markov process, 271  
Gauss-Markov theorem, 280  
Gauss-Newton algorithm, 288  
Gaussian distribution, 258  
    multivariate, 264  
Gaussian process, 270  
generalized eigenvectors, 241  
generalized least-squares  
    estimator, 283  
geodesy, 54  
geodetic coordinates, 54  
geoid, 54  
glideslope, 216  
global asymptotic stability, 243  
global stability, 243  
global uniform asymptotic stability, 243  
globally negative definite, 244  
globally negative semi-definite, 244  
globally positive semi-definite, 244  
Gramian  
    controllability, 252  
    observability, 254  
great circle, 212  
ground speed, 57  
guidance  
    system, 58  
guidance law  
    line-following, 212  
guidance, navigation, and control, 57
- heading angle, 63  
heading hold, 227  
Hermitian matrix, 241  
Hermitian transpose, 241  
higher order terms, 8  
hold  
    zero-order, 239  
hold control  
    altitude, 220  
    heading, 227  
    speed, 220  
horizontal tail  
    efficiency, 97
- volume ratio, 97  
Hurwitz matrix, 240
- identity matrix, 16  
impulse function, 36  
increment, 266  
indefinite matrix, 240  
independence  
    pairwise, 262  
    random processes, 267  
independent and identically distributed, 272  
index set, 266  
indirect Lyapunov, 243  
induced matrix norm, 233  
inertia  
    matrix, 68  
    moment tensor, 68  
    moments, 68  
    products, 68  
inertia matrix  
    airplane, 79  
    stability frame, 105  
initial condition, 7  
initial value problem, 20, 234  
inner-outer loop control, 210  
input, 231  
    doublet, 30  
    impulse, 37  
    signal, 7  
    step, 29  
    vector, 15  
input matrix  
    continuous-time, 16, 237  
    discrete-time, 237  
instrument guidance system, 216  
instrument landing systems, 216  
inverse  
    matrix, 241  
iterative least-squares  
    estimator, 288
- Jacobian linearization  
    univariate, 8  
joint CDF

- random process, 266
- joint cumulative distribution function, 260
- joint PDF
  - random process, 266
- joint PMF
  - random sequence, 266
- joint probability density function, 260
- Jordan blocks, 240
- Jordan canonical form, 241
- Jordan Chains, 247
- Krasovskii-LaSalle theorem, 245
- kurtosis, 259
- Laplace transform, 14
  - inverse, 14
- lateral plane, 94
- latitude, 54
- law of gravitation, 71
- law of total probability
  - continuous random variables, 263
  - events, 263
- lead-lag control, 199
- least upper bound, 234
- least-squares
  - Bayesian, 285
  - constrained problem, 289
  - damped, 289
  - estimator, 279
  - iterative, 288
  - nonlinear, 287
  - problem, 276
- level flight, 87
- level set, 245
- Levenburg-Marquardt algorithm, 289
  - modified, 289
- lift coefficient
  - wing, 77
- lift force, 53
  - airfoil, 77
  - airplane, 81
- lifting-line theory, 78
- linear estimator, 274
- linear inequality, 312
- linear least-squares
  - estimator, 279
- linear matrix inequality, 312
- linear system, 7
- linear, parameter-varying system, 237
- linear, time-invariant system, 237
- linear, time-varying system, 237
- linearization
  - cosine, 9
  - error, 8
  - Jacobian, 17
  - sine, 9
  - theorem, 11
- linearized dynamics
  - lateral-directional, 109
  - longitudinal, 108
- Lipschitz condition, 234
- localizer, 216
- localizer-type direction aid, 216
- locally negative definite, 244
- locally negative semi-definite, 244
- locally positive definite, 244
- locally positive semi-definite, 244
- long-period mode, 126
- longitudinal plane, 93
- loop bandwidth, 202
- loop-shaping
  - control stages, 193
  - SISO, 172
- lower triangular matrix, 242
- LTI MIMO continuous-time stability, 248
- LTI MIMO discrete-time stability, 248
- LTI MIMO free response
  - continuous-time, 247
  - discrete-time, 248
- Lyapunov function, 244
- Lyapunov function candidate, 244
- Lyapunov stability, 243
  - equilibrium point, 243
- Mahalanobis distance, 265, 279
- marginal PDF, 263
- marker beacons, 216

- Markov chain, 268  
Markov process, 268  
Markov property, 268  
Marquardt parameter, 289  
matplotlib, 477, 482  
matrix  
    correlation, 261  
    covariance, 262  
    cross-covariance, 262  
    diagonal, 241  
    Hermitian, 241  
    lower triangular, 242  
    norm, 233  
    orthogonal, 241  
    square, 242  
    symmetric, 241  
    transfer function, 238  
    unitary, 241  
    upper triangular, 241  
    variance-covariance, 262  
matrix decomposition  
    Cholesky, 242  
    eigenvalue, 239  
    QR, 242  
    singular value, 242  
matrix definitions, 241  
matrix exponential, 237  
    Cayley-Hamilton, 250  
matrix inverse, 241  
matrix rank, 249  
matrix transpose, 241  
    conjugate, 241  
    Hermitian, 241  
maximum *a posteriori* estimator, 284  
maximum likelihood estimator, 272  
maximum norm  
    matrix, 234  
    vector, 233  
mean, 259  
    estimator, 273  
    function, 266  
    random process, 266  
    random vector, 261  
measurable, 255  
measurement noise, 255  
median, 259  
memoryless process, 268  
minimum mean square error, 273  
minimum phase system, 202  
minimum variance unbiased estimator, 273  
missions, 129  
modal analysis, 247  
mode, 247, 259  
mode approximation  
    long-period, 127  
    roll, 128  
    short-period, 127  
    spiral, 129  
model evidence, 284  
Moore-Penrose inverse, 277  
Multi-Bernoulli distribution, 257  
multiple inputs, 232  
multiple outputs, 232  
mutual independence, 263  
navigation frame, 55  
negative definite  
    function globally, 244  
    function locally, 244  
negative definite matrix, 240  
negative semi-definite  
    function globally, 244  
    function locally, 244  
negative semi-definite matrix, 240  
nested-loop control, 210  
neutral point, 101  
no wind approximation, 57  
noise  
    measurement, 255  
    observation, 255  
    process, 255  
nonlinear least-squares  
    estimator, 287  
nonlinear system, 7  
norm  
     $L^2$ , 233

- $L^\infty$ , 233
- $L^p$ , 233
- $L^{2,2}$ , 234
- $L^{\infty,\infty}$ , 234
- $p$ , 233
- entry-wise matrix, 234
- Euclidean, 233
- Frobenius, 234
- induced matrix, 233
- matrix, 233
- matrix maximum, 234
- taxicab, 233
- vector, 232
- vector maximum, 233
- normal distribution, 258
- nose, 74
- numpy, 477
- Nyquist plot, 177
- Nyquist stability criterion
  - simplified SISO, 182
  - SISO, 182
- oblate spheriod, 54
- observability, 145, 249
  - matrix, 145
  - state, 252
- observability Gramian, 254
- observation error, 274
- observation matrix, 274
- observation noise, 255
- ODE solution
  - homogeneous, 28
  - particular, 28
- open-loop transfer function, 172
- operator, 53
- optimal estimation
  - parameter, 72, 104
- optimal estimator
  - parameter, 272
- ordinary differential equation, 7, 231
  - autonomous, 7
  - linear, 8
  - order, 7
- ordinary least-squares
  - estimator, 280
  - problem, 276
  - solution, 277
- Orlicz theorem, 235
- orthogonal matrix, 241
- orthogonality
  - random processes, 267
- outcomes, 255
- output, 231
  - signal, 7
  - vector, 15
- output controllability, 252
- output equation, 15, 231
- output feedback control, 143
- output matrix, 16, 237
- parallel interconnection, 165
- parameter estimator, 272
  - optimal, 272
- penalty function, 312
- perturbation
  - form, 8
  - scalar, 8
  - theory, 9
  - vector, 17
- phase margin, 186
- phases of flight, 85
- phugoid mode, 127
- pilot, 53
- pitching moment, 77
- pitching moment coefficient
  - wing, 77
- planning
  - path, 58
  - system, 58
  - trajectory, 58
- plant, 136, 167
- pole placement, 144
- pole-zero cancellation, 15, 166
- position deviation
  - lateral-directional, 228
  - longitudinal, 222

- positive definite
  - function globally, 244
  - function locally, 244
- positive definite matrix, 240
- positive semi-definite
  - function locally, 244
- positive semi-definite matrix, 240
- precision approach, 214
- principle of superposition, 7, 8
- probability density function
  - joint, 260
- probability distribution
  - Bernoulli, 257
  - Binomial, 257
  - Gaussian, 258
  - Multi-Bernoulli, 257
  - normal, 258
  - uniform, 258
- process noise, 255
- proportional control, 137
- proportional gain, 138
- proportional-derivative control, 156
- proportional-integral control, 148
- proportional-integral-derivative control, 158
- propulsive force
  - airplane, 81
- pseudoinverse, 253, 277
  - left, 277
- pyplot, 482
- QR decomposition, 242
- quaternions, 63
- radially unbounded, 245
- random dynamical system, 255
- random process, 266
  - average power, 268
  - ensemble average, 268
  - time average, 268
- random sequence, 266
- random variable, 255
  - central moment, 259
  - continuous, 257
  - discrete, 256
- kurotsis, 259
- mean, 259
- median, 259
- mode, 259
- moment, 259
- skewness, 259
- standard deviation, 259
- standardized moment, 259
- variance, 259
- random vector, 260
- random walk, 270
  - damped, 271
- rank
  - column, 249
  - deficient, 249
  - full, 249
  - matrix, 145, 249
  - row, 249
- rate feedback control, 156
- reachability
  - state, 249
- realization, 256
  - random process, 266
  - vector, 260
- recursive Bayes estimator, 323
- recursive least-squares, 285
- recursive relations, 231
- reference ellipsoid, 54
- reference frame
  - Earth-centered, Earth-fixed, 54
  - international celestial reference frame, 54
  - international terrestrial reference frame, 54
  - rotations, 59
- reference frames, 53
  - Earth, 53
- residual
  - least-squares, 276
  - sample, 279
- resolvent matrix, 238
- restricted-step methods, 289
- rigid body, 67
- rise time, 31
- robustness

- SISO feedback control system, 184
- roll mode, 128
- rolling moment
  - wing, 77
- rolling moment coefficient
  - wing, 77
- root locus, 139, 152
- rotation matrix, 60
- row rank, 249
- rudder, 75
- ruddervator, 76
- Runge-Kutta
  - first order, 70
- sample, 266
- sample path, 266
- sample trajectory, 266
- scipy, 477
- script, 477
- search vector, 288
- sensitivity
  - transfer function, 174
- sensor, 138
- sequence
  - data type, 480
- serial interconnection, 165
- set
  - bounded, 245
  - closed, 245
  - compact, 245
  - countable, 256
  - countably infinite, 256
  - level, 245
- setpoints, 312
- settling time, 31
- short-period mode, 126
- side force, 53
- sideslip angle, 64
- sidewash angle, 94
- signal, 6
  - analog, 6
  - digital, 6
  - discrete-time, 6
- periodic, 35
- single input, 232
- single output, 232
- singular value, 242
- singular value decomposition, 242
- SISO feedback control system
  - characteristic equation, 168
  - stability, 168
- skew-symmetric matrix operation, 68
- skewness, 259
- slat, 76
- small angle approximation, 9
- span
  - wing, 77
- speed hold, 220
- spiral mode, 128
- spoiler, 76
- square matrix, 242
- stability
  - asymptotic, 243
  - bounded input, bounded output, 10
  - continuous-time LTI, 21
  - dynamic, 126
  - global, 243
  - global asymptotic, 243
  - global uniform asymptotic, 243
  - globally asymptotically, 10
  - lateral-directional, 128
  - longitudinal, 126
  - LTI MIMO continuous-time, 248
  - LTI MIMO discrete-time, 248
  - Lyapunov, 10, 243
  - SISO feedback control system, 168
  - uniform, 243
  - uniform asymptotic, 243
- stability and control coefficients
  - longitudinal, 113
- stability and control derivatives, 104
  - lateral-directional, 120
  - longitudinal, 113
- stability augmentation system, 145
- stability augmentation systems, 129
- stability coefficients

- static, 99
- stability derivative
  - weathercock, 102
- stability frame, 104
- stability margin
  - (time) delay, 187
  - disk, 187
  - gain, 184
  - phase, 186
  - SISO, 184
- stabilizability, 252
- stall, 78
- standard deviation, 259
- standard gravitational acceleration, 71
- standard gravitational parameter, 71
- standard normal distribution, 258
- state, 231
  - Markov process, 268
  - vector, 15
- state controllability, 249
- state equation, 15, 231
- state estimation
  - vehicle, 58
- state feedback control, 143
- state matrix
  - continuous-time, 16, 237
  - discrete-time, 237
- state observability, 252
- state reachability, 249
- state trajectory, 231
- state transition
  - Markov process, 268
- state transition matrix
  - Markov process, 268
- state transition probability, 268
- state-space
  - continuous-time, 15
  - continuous-time linear, 236
  - continuous-time LTI, 16
  - discrete-time linear, 237
  - linear, 236
  - stochastic, 255
- state-space model, 231
- state-transition matrix, 237
- static control coefficients, 88
- static margin, 100
- static stability, 99
  - airplane, 100
  - directional, 101
  - lateral, 101
  - longitudinal, 100
- static stability equations, 88
- stationarity
  - covariance, 267
  - strict-sense, 267
  - wide-sense, 267
- statistic, 259
- steady flight conditions, 85
- steady flight equations
  - body frame, 85
  - performance, 87
- steady-state
  - condition, 31
  - gain, 31
  - input, 31
  - output, 31
- steady-state error, 137
- steady-state output
  - vector, 144
- steady-state sinusoidal response, 39
- steepest gradient, 289
- stochastic dynamical system, 255
- stochastic process, 266
- straight flight, 87
- strict-sense stationarity, 267
- supremum, 234
- symmetric matrix, 241
- symmetric wing, 78
- system, 6
  - dynamical, 6
  - dynamics equation, 231
  - input, 231
  - linear, parameter-varying, 237
  - linear, time-invariant, 237
  - linear, time-varying, 237
  - mimo, 7

- miso, 7
- mode, 21
- multiple inputs, 232
- multiple outputs, 232
- order, 231
- output, 231
- output equation, 231
- response, 6
- simo, 6
- single input, 232
- single output, 232
- siso, 6
- state, 231
- state trajectory, 231
- state-space model, 231
- static, 6
- system identification, 6
  - aircraft, 72
  - airplane, 104
- system response
  - forced, 28
  - free, 20, 237
  - frequency, 39
  - impulse, 37
  - pole effects, 31
  - step, 29
  - unit step, 30
  - zero effects, 31
- zero-input, 237
- zero-state, 237
- system robustness, 165
- tail
  - horizontal, 74
  - vertical, 74
- tailerons, 76
- taxicab norm, 233
- Taylor series
  - multivariate, 17
  - univariate, 8
- thrust, 71
- thrust vectoring, 72
- time, 231
- time average, 268
- time sample, 266
- time step, 231
- time-invariant dynamics, 232
- tracking error, 138
- transfer function, 14
  - form, 14
  - matrix, 238
  - pole, 14
  - zero, 14
- transpose
  - matrix, 241
- trial step vector, 290
- triangle inequality, 233
- trim point, 10
- trust-region, 290
- trust-region methods, 289
- turn compensation, 227
- tychastic dynamical system, 255
- unbiased, 273
- uncorrelation, 261
- unforced dynamics, 232
- uniform asymptotic stability, 243
- uniform distribution, 258
- uniform stability, 243
- unitary matrix, 241
- upper triangular matrix, 241
- v-tail, 74
- validation and verification, 235
- variance, 259
  - estimator, 273
- variance-covariance, 262
- vector
  - norm, 232
- vector space, 249
  - dimension, 249
- vehicle, 52
  - aerospace, 53
  - flying, 53
- vehicle dynamics and control, 53
- velocity, 61
- vertical tail

- efficiency, 98
- volume ratio, 99
- volume ratio
  - horizontal tail, 97
  - vertical tail, 99
- waypoint, 212
- weight, 71
- white noise process, 270
- wide-sense stationarity, 267
- Wiener process, 269
- wind frame, 56
- wind speed, 57
- wings, 74
- zero
  - left half plane, 31
  - right half plane, 31
- zero matrix, 16
- zero-input response, 237
- zero-order hold, 239
- zero-state response, 237