NONLINEAR UAV FLIGHT CONTROL USING COMMAND FILTERED BACKSTEPPING

A Thesis

Presented to

the Faculty of California Polytechnic State University

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Aerospace Engineering

by

Brian Borra

June 2012

COMMITTEE MEMBERSHIP

TITLE:	Nonlinear UAV Flight Control Using Command Filtered Backstepping
AUTHOR:	Brian Borra
DATE SUBMITTED:	June 2012
COMMITTEE CHAIR:	Eric Mehiel, Ph.D.
COMMITTEE MEMBER:	Robert Miller, Ph.D.
COMMITTEE MEMBER:	Al Jiminéz, Ph.D.

Abstract

Nonlinear UAV Flight Control Using Command Filtered Backstepping

Brian Borra

This effort is aimed to extend the state of the art in the areas of adaptive reconfigurable flight control, specifically through implementation of Adaptive* Backstepping control architecture for use in AME UAS's *Fury 1500* Unmanned Aerial System (UAS).

Backstepping is a systematic design approach that allows the use of certain plant states to act as virtual controls for others.

Acknowledgements

Thank you...

Table of Contents

List of Tables viii

List of Figures ix

Nomenclature xi

1 Introduction 1

1.1 Significance 4

1.2 Requirements and Outcomes 4

1.3 Thesis Outline 4

2 History 5

2.1 Emergence of Adaptive Control 7

2.2 Emergence of Backstepping 9

3 Nonlinear Theory 11

3.1 Modeling 11

3.1.1 Reference Frames 14

3.1.2 Direction Cosine Matrices 17

3.1.3 Aircraft Dynamics 21

3.1.3.1 Translational Acceleration 22

3.1.3.2 Rotational Acceleration 25

3.1.3.3 Attitude Rates 27

3.1.3.4 Equations of Motion 28

3.1.4 System Observations 29

3.2 Stability 32

3.2.1 Equilibrium and Operating Points 33

3.2.2 Lyapunov Stability Definitions 36
3.2.3 Lyapunov Stability Theorems 42
3.3 Control 52
3.3.1 Requirements 52
3.3.2 Objective and Methods 54
3.3.3 Lyapunov Based Control Design 56
3.3.4 Backstepping 58
3.3.4.1 Second Order Systems 59
3.3.4.2 Higher Order Systems 62
3.3.4.3 Command Filtering 62
3.3.5 Command Filter 63
3.3.6 Control Allocation 65
4 Derivation of UAV Flight Path Controller **66**
4.1 Outer Loop 70
4.1.1 Flight Path Angle Control 71
4.1.2 Airspeed Control 73
4.2 Middle Loop 73
4.2.1 Wind-Axis Angle Control 74
4.3 Inner Loop 75
4.3.1 Body-Axis Angular Rate Control 76
4.4 Stability Proof 76
5 Application **77**
5.1 Modeling 77
5.2 Simulation 77
5.2.1 Nominal 78
5.2.2 Turbulence 78
5.2.3 Modeling Error / Unfamiliar Plant 78
5.2.4 Effector Failure 78
5.3 Results 78
6 Conclusion **79**

List of Tables

3.1 Choices for State Variables 13
3.2 Direction Cosine Matrices for Plane Rotations 19
3.3 Stability Theorem Overview 43
3.4 Nonlinear Control Method Overview 55
3.5 Backstepping Key Terms 58

List of Figures

1.1 Fury 1500 UAS 3

2.1 WPAFB: Design Methods 6

2.2 Traditional Adaptive Control Flowchart 8

3.1 Air Vehicle Reference Frames 15

3.2 Axis Relationships: Body, Stability, and Wind Axes 15

3.3 Earth Reference Frames 17

3.4 Direction Cosine Matrix Visual 20

3.5 Body Axes, Forces, and Moments 23

3.6 Orientation of Gravity Vector with Respect to Body Axes 24

3.7 Operating Surface via Phase Portraits 34

3.8 First Order Unstable Systems [1, Online Lecture Notes] 39

3.9 First Order Stable Systems [1, Online Lecture Notes] 39

3.10 First Order Asymptotically Stable Systems [1, Online Lecture Notes] 40

3.11 Concepts of Stability 40

3.12 Equilibrium Point Classification for 2^{nd} Order Linear Systems [2] 41

3.13 Geometric Interpretation of Lyapunov's Stability Theorem. 45

3.14 Asymptotic Property 1: $\dot{\boldsymbol{f}}(t) \rightarrow 0 \nRightarrow \boldsymbol{f} \rightarrow$ constant 50

3.15 Asymptotic Property 2: $\boldsymbol{f} \rightarrow$ constant $\nRightarrow \dot{\boldsymbol{f}(t)} \rightarrow 0$ 50

3.16 Backstepping Block Diagram Representation: a) Integral augmented system b) introducing $\pm\alpha(x)$ c) "backstepping" $-\alpha(x)$ through the integrator 61

3.17 Command Filter 64

4.1 Loop Interactions: Flight-Path & Airspeed, Wind, and Body 67

4.2 High Level Block Diagram 68
4.3 Command filtered adaptive backstepping control scheme [3] 69
4.4 Geometric Solution for μ_c^o and α_c^o 72

Nomenclature

α	Angle of Attack	deg
$\alpha(x)$	Stabilizing Feedback Control Law	
β	Angle of Sideslip	deg
γ	Flight-Path Angle	deg
ζ	Damping Frequency	rad/s
μ	Bank Angle	deg
ξ	Virtual Control Law	
ϕ, θ, ψ	Roll, Pitch, and Yaw Angles	deg
χ	Heading Angle	deg
ω_n	Natural Frequency	rad/s
D	Drag	$\mathrm{lb_f}$
F	Known Sub-function	
L	Lift	$\mathrm{lb_f}$
V	Airspeed	ft/s

W	Positive Definite Function	
Y	Sideforce	$\mathrm{lb_f}$
$\bar{L}$, $\bar{M}$, $\bar{N}$	Roll, Pitch, and Yaw Moments	lb-ft
f	Unknown or Partially Known Sub-function	
m	Mass	slug
u	Control Law	
u, v, w	Longitudinal, Lateral, and Normal Velocities	ft/s
z	Tracking Error	

Acronyms

CLF	Control Lyapunov Function
ECEF	Earth Centered Earth Fixed
ECI	Earth Centered Inertial

Subscript

c	Commanded
M	Magnitude
R	Rate

Superscript

$-$	Compensated Tracking Error
$\sim$	Tracking Error
o	Unfiltered

Chapter 1

Introduction

> "Sometimes I tell people that I don't understand X. Sometimes they reply, 'Well, I clearly explained it on page 4.' That'd be the wrong answer. If I (or anyone else) tells you they don't understand X, they're saying you haven't explained X well enough. Rather than getting defensive, reexamine your explanation, see how you can rewrite to make it clearer. It's not the reader's fault if they don't understand what you're saying. Your job as a writer is to make it very clear where you're coming from. Telling me that I just missed the point won't score you any points. Take another look, see what's not obvious." http://web.media.mit.edu/~intille/teaching/advising/draftAdvisorTips.htm
>
> What's the best thesis? A done one.

In all control techniques the goal is the same: to achieve a desired response even in the presence of external disturbances and sensor noise. It is the control engineer's responsibility to choose the appropriate design technique given the stability, performance, and cost requirements of the system. As technology expands so does its complexity and the level of detail placed into the system model. Typically plants with non-linearities and unknown or changing parameters are associated with this higher fidelity. These advances in control theory led to the development of adaptive control, the goal of which is to control plants with unknown or imperfect knowledge.

Focus

Implementation of a non-linear command filtered backstepping(CFBS) flight controller for use in an AME UAS program. Specifically flight path control tracking of ground track, flight path (climb rate), and airspeed commands [χ_c,γ_c,V_c] from mission planner.

The focus of this thesis is to explore backstepping, a relatively new control algorithm for non-linear systems. It's a subset of a popular modern adaptive control technique called Direct Adaptive Control (DAC) that utilizes Lyapunov synthesis to derive a stabilizing controller. "Backstepping is a recursive procedure that interlaces the choice of a Lyapunov function with the design of feedback control. It breaks a design problem for the full system into a sequence of design problems for lower-order (even scalar) systems. By exploiting the extra flexibility that exists with lower order and scalar systems, backstepping can often solve stabilization, tracking, and robust control problems under conditions less restrictive than those encountered in other methods." Khalil [1]

Appeal

Nonlinear flight control methods offer

- increase in performance
- reduction in development time

Adaptive methods offer

- management of uncertain dynamics / less precise model knowledge required
- protection against mechanical failures / damage

Constrained Backstepping

- u(t) generated guaranteed implementable
- retains useful nonlinearities*
- Lyapunov stability results provable

Goals

Attain professional proficiency in topic.

Develop a simulink based backstepping control architecture with

- Command Filtering
- Control Allocation / Distribution**
- Reproduce Simulation Analysis in "Backstepping-Based Flight Control with Adaptive Function Approximation" by J. Farrell, M. Sharma, & M. Polycarpou, 2005

Thoroughly support simulation with

- Lyapunov Control Law Synthesis / Stability Proof

Backstepping has applications in a broad spectrum of engineering disciplines, from electrical motors to jet engines, ship control to flight control just to name a few. It offers a systematically methodology for developing control schemes for non-linear systems.

Figure 1.1: Fury 1500 UAS

http://www.prnewswire.com/news-releases/fury-1500-uas-achieves-14-hour-duration-flight-html

1.1 Significance

Adaptive flight control has been viewed as an enabling technology to deal with plants with highly nonlinear, time-varying, and/or uncertain dynamical characteristics. As unmanned aerial systems (UAS) become more interactive with humans, providing safety for flight crews, it must be assured that they offer a level of reliability comparable to manned systems; this project strives to accomplish this goal, hence fostering the advancement of autonomous systems and user safety.

1.2 Requirements and Outcomes

International Traffic in Arms Regulations (ITAR) requires that information and material pertaining to defense and military related technologies may only be shared with US Persons. The proprietary AME UAS Engineering *Fury UAS* simulation model will be substituted with a publicly available one for security as well as legal reasons*. Proof of concept will be demonstrated by the dissimilarity in control effectiveness of these models. A concluding objective would include hardware implementation and flight test demonstration.

1.3 Thesis Outline

Review aircraft dynamics and supporting concepts essential to modeling and simulation. Next, results from Lyapunov theory then derive backstepping theory. Two cases were covered, a second order non-linear system and a higher order non-linear system. Each was selected to highlight a particular benefit of backstepping, however they repetitively demonstrate the steps required to manipulate the plant and develop a control law. A full state feedback, six degree of freedom, non-linear flight path control system will be derived for a UAV. To close the simulation results will be discussed with respect to feedback linearization.

Chapter 2

History

This section consists of thoughts, needs to be condensed into one flowing historical account...

Mission requirements drive vehicle requirements, and these drive control law requirements. See WPAFB Multivariate control methods

See "Constructive Nonlinear Control: Progress in the 90's"

Introduce the concept of control, lead into adaptive control section.

(1) Emergence of Adaptive Control - Krstic (2) WPAFB: MVAR CONTROL DESIGN GUIDELINES

FEEDBACK CONTROL! Feedback allows for large disturbances

"The main purpose of every feedback loop, created by nature of designed by engineers, is to reduce the effect of uncertainty on vital system functions. [4] Feedback can be used for stabilization, but inappropriately designed feedback controllers may reduce, rather than enlarge, regions of stability. "

Design Methods

Linear Quadratic Gaussian (LQG, H_2), Worst case L2 - induced norm problem (H∞), Worst case L∞ - induced norm problem (l1). "Introduction of these formal methods, together with use of state-space descriptions provided the first multivariable control design tools" (WL-TR-96-3099)

Table 1.1: Categorization of Design Methods

Formal Optimization Problems	
Basic Linear-Quadratic-Gaussian (LQG)	[Athans 1971]
LQG with Explicit/Implicit Model Following	[Asseo 1970, Tyler 1966]
Frequency Weighted LQG	[Gupta 1980]
Worst-case Induced L_2 Norm (H_∞)	[Zames 1983, Doyle 1989]
Worst-case Induced L_∞ Norm (ℓ_1)	[Dahleh 1987]
Mixed Criteria (H_2/H_∞)	[Zhou 1989, Rotea 1991, Yeh 1992]
Numerical Optimization Problems	
Fixed Structure LQ-Control	[Axsater 1966, Levine 1970, Stein 1971]
Fixed-Plus-Variable Gain LQ-Control	[VanDierendonck 1972]
Fixed Structure H_∞ Control	[Bernstein 1990]
Fixed Structure H_2/H_∞ Control	[Bernstein 1989, Ridgely 1992]
Q-Parameter Design (QDES)	[Boyd 1991]
μ-Synthesis via $D-K$-Iteration	[Doyle 1983, Stein 1991]
Frequency Domain Methods	
Diagonal Dominance/Inverse Nyquist Array	[Rosenbrock 1974]
Characteristic Loci	[Postlethwaite 1979]
Upper Triangular Structures	[Mayne 1973]
Singular-Value-Based Loop Shaping	
via Direct Inversion	[Hung 1982]
via LQG/LTR	[Stein 1987]
via H_∞	[McFarlane 1992]
via Dynamic Inversion	[Bugajski 1992b]
Quantitative Feedback Theory (QFT)	[Horowitz 1979]
Eigenstructure Assignment Methods	
via Full-State Feedback	[Andry 1983]
via Output Feedback	[Calvo-Ramon 1986, Sobel 1990]
via Quadratic Weights in LQ	[Harvey 1978]
via Numerical Optimization	[Garg 1989, Wilson 1990]
Fringe Methods	
Model Predictive Control (MPC, DMC, MAC)	[Morari 1989]
Covariance Control	[Skelton 1989]
Stochastic Parameters (maximum entropy)	[Hyland 1982]
Variable-Structure Control	[Utkin 1977]
Geometric Methods	[Wonham 1979]
Polynomial-Matrix Methods	[Peczkowski 1978, Wolovitch 1974]
Lyapunov-Based Methods	[Barmish 1985, Boyd 1989]

Figure 2.1: WPAFB: Design Methods

2.1 Emergence of Adaptive Control

Reference work from seminal papers on adaptive control: Bode lectures by Gunter Stein and Petar Kokotovic.

First implementations of feedback may be considered indirect attempts to control unknown plants

- "Even the most elementary feedback loops can tolerate significant uncertainties"

50's and 60's: Advances spawned more sophisticated systems

- Find Examples: Assume space-race drove this, Find aircraft that required adaptive control
- self-learning, self-optimizing, self-organizing, self-tuning, and adaptive control utilizing on-line identification or pattern recognition.
- Lyapunov techniques were not a major player in the control engineering until the early 1960's thanks to publications by Lur'e and a book by La Salle and Lefschetz* -Slotine and Lee

computers started to come into play

$>$ 60's: Theoretical: Stochastic and Dual Control

- gain-scheduling, fuzzy, neural, intelligent control

80's: Adaptive Linear Control or Traditional Adaptive Control

- Non-Linear Lyapunov-based control recently* achievable through recursive design procedures such as backstepping.
- "[Estimation-based designs are flexible and] achieved by treating the identifier as a separate module and guaranteeing its properties independent of the controller module." Usually referred to as modular designs; can use gradient and least squares algorithms for parameter update laws.

Traditional control relies on the certainty equivalence principal: "controller first designed as if all the plant parameters are known."

- "... control parameters are calculated by solving design equation for model matching, pole-zero placement, or optimality."
- For each adaptive control scheme it is up to the designer to choose:
  - Filters
  - Design Coefficients
  - Initialization Rules
  - etc? CONTROL ENGINEER JOB...
- Adaptive Controls Tradeoff: Transient Performance vs. Robustness

The appeal of non-linear control over linear is simple, performance. For systems which are highly structured or parametrically uncertain a nonlinear controller outperforms the best linear controller. [5]

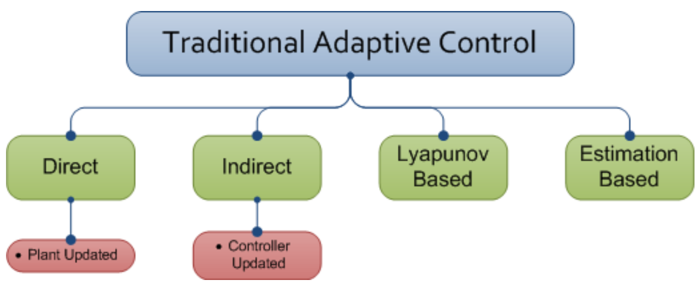


Figure 2.2: Traditional Adaptive Control Flowchart

"Progress for nonlinear systems was slow until the breakthrough Kanellakopoulos et al. [6] in adaptive non-linear control. The breakthrough was based on the nonlinear stabilization concept of "adding an integrator" as cited in **?** , Pg. 10]. The recursive application of this technique eventually led to the adaptive results of Kanellakopoulos et al. [6], is known and integrator backstepping or simply backstepping [7], [5], [8].

"backstepping led to the discovery of structural strict feedback condition (much weaker than the matching condition) under which the systematic construction of an RCLF is always possible. These first robust backstepping results appeared in... smoothness assumptions are common in recurisve backstepping designs becaue of the need to calculate derivates of various functions during the construction of the control law and Lyapunov function." [4]

2.2 Emergence of Backstepping

Development of backstepping, how it supports current requirements, and fits into current control law needs. Talk about how it is becoming a hot topic in the non-linear design world (an attempt to systematically approach non-linear control) and meets both aircraft and spacecraft control needs.

Adaptive Backstepping developed by Ioannis Kanellakopoulos [63] in collaboration with Petar Kokotovic, and Steve Morse "emerged as a confluence of the adaptive estimation idea on one side, and, on the other size, nonlinear control ideas expressed in works of [193], [12], [175], [85], [163]." Pg 11. Tuning functions were invented by Krstic [92,94] to reduce overparameterization.

- Development –¿ Kokotovic and Kanellakopoulos
  - Feedback linearization / NDI first
- Examples and reference papers used
  - Kokotovic Jet Engine Controller and other examples listed in Hrkegrd's dissertation
  - Hrkegrd's work with ADMIRE model (Canard Fighter)
  - Farrell's flying wing work which my controller is implementing

  ..
- Future of backstepping, ie. its role in future nonlinear control architecture. Do some research on alternative control techniques.

Backstepping has applications in a broad spectrum of engineering disciplines, from electrical motors to jet engines, ship control to flight control just to name a few. It offers a systematically methodology for developing control schemes for non-linear systems. The main appeal is that useful non-linearities do not have to be cancelled in the control law thereby increasing robustness to modeling errors and decreasing control effort. The alternative, feedback linearization, transforms a non-linear system into an equivalent linear system by cancelling all non-linearities via its control law.

Chapter 3

Nonlinear Theory

> "The art of flight control design is to realize a solution that achieves an acceptable compromise among the evaluation criteria: [Stability, Performance, and Flying Qualities]."
>
> Honeywell Inc. and Lockheed Martin Skunk Works [9]

3.1 Modeling

This section will introduce fundamental aircraft dynamics concepts and ultimately derive the equations of motion implemented in the backstepping control architecture. The aim is to establish a practical understanding of the equations of motion, reinforced by physical illustrations of key aspects to the derivation rather than pure mathematical formulation. The fallout allows the designer to both qualitatively *and* quantitatively evaluate the characteristic modes of motion, thereby providing an analytical playground for control design and performance evaluation via **handling qualities**[1]. Assumptions, hence consequent limitations of the derived equations, will be clearly identified with discussion immediately succeeding. No stone will be left unturned; equations will be derived from Newton's first principals and

[1] Reference handling qualities for unmanned aerial vehicles, this is a hot topic; degree of instability depends on what autopilot can handle...

include necessary supporting concepts. The philosophy of this approach and procedure itself is credited to work by [10] and [11].

Establishing a way to mathematically describe the vehicle's dynamics is a necessity for any flight control architecture. As with any dynamic system, a set of differential equations may be used to calculate an object's position, velocity, and acceleration. Typically for complex systems — such as an airplane with flexible structure, rotating internal components, and variable mass — simplifying assumptions are applied in order to use Newton's Laws to derive vehicle dynamics. These assumptions lead to a direct appreciation of important factors that govern the vehicle dynamics. This level of understanding is an "implicit requirement for effective and efficient flight control system design activities. It affords a basic understanding of the vehicle/control system interactions and of the flight controller possibilities which are most likely to succeed." [11]

Assumption 3.1. Airframe is a rigid body.

"Rigid body models are described by six degrees of freedom and include forces and moments caused by gravity, aerodynamics, and propulsion." [9] The distances between any two points are fixed, hence forces acting between those points due to elastic deformation are absent. Consequently, the air vehicle may be modeled as an individual element of mass. In reality air vehicles diverge from the rigid body assumption in two ways: aeroelastic phenomena due to airframe structure deformation (such as wing bending due to air loads) and relative motion of components (engine, propeller, and control surfaces).

Under this assumption, the equations of motion may be decoupled into translational and rotational equations if the coordinate origin is chosen to coincide with the center of gravity. Two possible system descriptions are described in Table 3.1. It implies that there are 9 equations necessary for control, list items 1–3, and 3 for navigation, list item 4. The implementation herein is concerned with control, therefore the **state-vector** [2] of this controller will consist of the first 9 variables.

Table 3.1: Choices for State Variables

		Body		Flight-Path / Wind	
x_1	Translational	Longitudinal Velocity	u	Heading Angle	χ
		Lateral Velocity	v	Flight-Path Angle	γ
		Normal Velocity	w	Velocity	V
x_2	Attitudes	Euler Roll	ϕ	Bank Angle	μ
		Euler Pitch	θ	Angle of Attack	α
		Euler Yaw	ψ	Sideslip Angle	β
x_3	Rotational	Roll Rate	P	Roll Rate	P
		Pitch Rate	Q	Pitch Rate	Q
		Yaw Rate	R	Yaw Rate	R
x_4	Navigation	North Position	ξ	North Position	ξ
		East Position	η	East Position	η
		Altitude	h	Altitude	h

Body

- 3 components of attitude to specify orientation relative to the gravity vector
- 3 components of velocity to specify translational kinetic energy
- 3 components of angular velocity to specify rotational kinetic energy
- 3 components of position to specify potential energy in earth's gravity field

Flight Path / Wind

- 3 components of attitude to specify orientation relative to the velocity vector
- 1 component of velocity magnitude, 2 components of velocity direction to specify translational kinetic energy
- 3 components of angular velocity to specify rotational kinetic energy
- 3 components of position to specify potential energy in earth's gravity field

[2] The minimal set of system variables necessary to indicate the energy of the system, potential and kinetic, and its distribution at any given time.

Again the flight path controller herein will only be using the first three subsets of Table 3.1:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \end{bmatrix} = \begin{bmatrix} (\chi\ \gamma\ V)^T \\ (\mu\ \alpha\ \beta)^T \\ (P\ Q\ R)^T \end{bmatrix} \begin{array}{r} \text{Translational} \\ \text{Attitude} \\ \text{Rotational} \end{array} \tag{3.1}$$

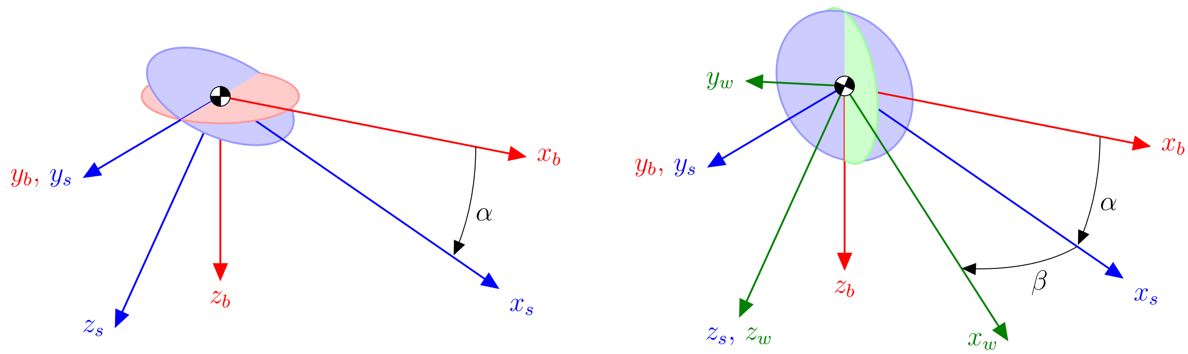
3.1.1 Reference Frames

Just as language is needed to express thoughts, a reference frame is necessary to convey motion. The relationship between an object and the space it resides in is relative; choosing a reference frame, or coordinate system, enables an observer to describe the motion of an object over time. Selecting an appropriate reference frame can greatly simplify the description of this relationship.

Assumption 3.2. Earth is an inertial reference frame.

When earth is considered as an inertial frame of reference, one that is *fixed* or moving at a constant velocity (non-rotating and non-accelerating) relative to earth, it permits accurate short-term control and navigation analysis. Conversely, an inertial frame of reference is unacceptable for air vehicles that require long-term navigation, especially for high-speed flight, or extra-atmospheric operation; for most UAVs this assumption is fairly accurate however. As this situation dictates, there are numerous reference systems in aerospace applications. The frames applicable to the equations of motion derivation herein are: body, stability, wind or flight-path, and earth-centered-inertial or earth-fixed. Additionally, north-east-down or local-tangent-plane, vehicle-carried-vertical, and earth-centered-earth-fixed frames will be covered. All coordinate systems follow the right hand rule and are orthogonal.

Body, stability, and wind axes are attached to the airframe at the center of gravity as depicted in Figure 3.1. By convention, body axis x_b points out the nose, y_b out the right



(a) Body to Stability Axes

(b) Stability to Wind Axes

Figure 3.1: Air Vehicle Reference Frames

wing, and z_b down the bottom of the aircraft. Stability axes are defined by a rotation of the body axes in the x_b-z_b plane by an angle-of-attack, α, that trims the air vehicle, ie. zero pitching moment; axis x_s points into the direction of steady flow, $y_s = y_b$, and z_s is perpendicular to the x_s-y_s plane in the direction following the right handed sign convention. Note that sideslip angle, β, is zero in stability axes. In wind, or flight-path, axes the x_w axis always points into the relative wind. This is defined by a rotation of the body axes through angle-of-attack and sideslip angle with $z_w = z_s$ and y_w following the right hand rule as shown in Figure 3.2:

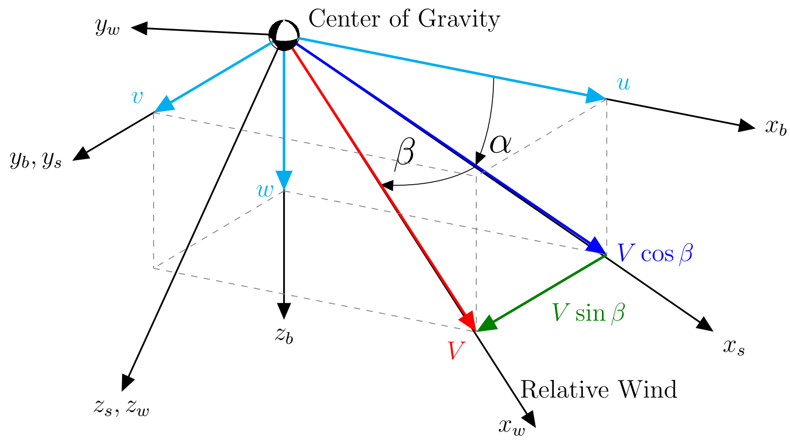


Figure 3.2: Axis Relationships: Body, Stability, and Wind Axes

The body state variables can be derived from the flight-path state variables as follows

[9], recall Table 3.1:

$$u = V \cos\beta \cos\alpha \tag{3.2}$$

$$v = V \sin\beta \tag{3.3}$$

$$w = V \cos\beta \sin\alpha \tag{3.4}$$

$$\phi = \tan^{-1}\left(\frac{\cos\gamma \sin\mu \cos\beta - \sin\gamma \sin\beta}{-\cos\gamma \sin\mu \sin\alpha \sin\beta + \cos\gamma \cos\alpha \cos\mu - \sin\gamma \sin\alpha \cos\beta}\right) \tag{3.5}$$

$$\theta = \sin^{-1}\left(\cos\gamma \sin\mu \cos\alpha \sin\beta + \cos\gamma \cos\mu \sin\alpha + \sin\gamma \cos\alpha \cos\beta\right) \tag{3.6}$$

$$\psi = \tan^{-1}\left\{\frac{(\sin\mu \sin\alpha - \cos\alpha \cos\mu \sin\beta)\cos\chi + [\cos\gamma \cos\alpha \cos\beta - \sin\gamma(\sin\alpha \cos\mu + \sin\beta \cos\alpha \sin\mu)]\sin\chi}{-(\sin\mu \sin\alpha - \cos\alpha \cos\mu \sin\beta)\sin\chi + [\cos\gamma \cos\alpha \cos\beta - \sin\gamma(\sin\alpha \cos\mu + \sin\beta \cos\alpha \sin\mu)]\cos\chi}\right\} \tag{3.7}$$

Flight-path variables ca be derived from the body state variables as follows [9], recall Table 3.1:

$$V = \sqrt{u^2 + v^2 + w^2} \tag{3.8}$$

$$\alpha = \tan^{-1}\left(\frac{w}{u}\right) \tag{3.9}$$

$$\beta = \sin^{-1}\left(\frac{v}{\sqrt{u^2 + v^2 + w^2}}\right) \tag{3.10}$$

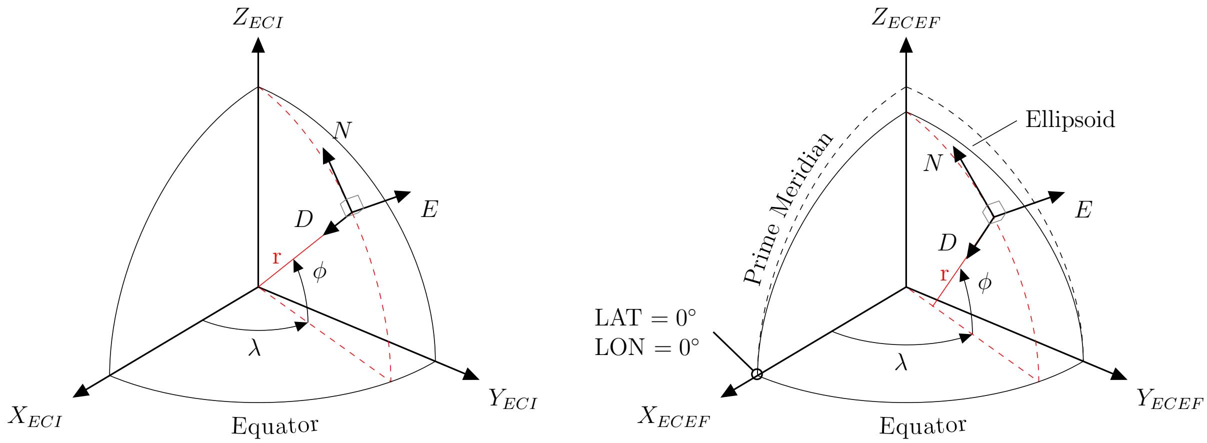
$$\mu = \tan^{-1}\left[\frac{uv \sin\theta + (u^2 + w^2)\sin\phi \cos\theta - vw \cos\phi \cos\theta}{\sqrt{u^2 + v^2 + w^2}\,(w \sin\theta + u \cos\phi \cos\theta)}\right] \tag{3.11}$$

$$\gamma = \sin^{-1}\left(\frac{u \sin\theta - v \sin\phi \cos\theta - w \cos\phi \cos\theta}{\sqrt{\mathrm{u}^2 + v^2 + w^2}}\right) \tag{3.12}$$

$$\chi = \tan^{-1}\left[\frac{u \cos\theta \sin\psi + v\,(\sin\phi \sin\theta \sin\psi + \cos\phi \cos\psi) + w\,(\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi)}{u \cos\theta \cos\psi + v\,(\sin\phi \sin\theta \cos\psi - \cos\phi \sin\psi) + w\,(\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi)}\right] \tag{3.13}$$

North East Down (NED), also known as Local Tangent Plane (LTP), is positioned on the surface of earth with it's origin vertically aligned to the aircraft's center of gravity. North is parallel to lines of longitude (λ), east is parallel to lines of latitude (ϕ), and down completes the right hand rule pointing into earth. Vehicle Carried Vertical (VCV) shares the NED orientation definition, with the exception of a shift in origin from Earth's surface to the vehicle's center of gravity, as the name suggests.

The Earth Centered Inertial (ECI) frame is considered *fixed* in space with its origin at the



(a) Earth Centered Inertial (Fixed)

(b) Earth Centered Earth Fixed (Rotates)

Figure 3.3: Earth Reference Frames

center of Earth; it does not rotate with Earth and is oriented to suit the situation. Typically the z_{ECI} axis is aligned along Earth's spin axis pointing toward the North Pole. Consult Stevens and Lewis [12, Pg. 20], for alternative ECI orientations.

Earth Centered Earth Fixed (ECEF) is a non-inertial frame that rotates with earth. This reference system aligns x_{ECEF} to the intersection of the zero-longitude prime meridian and zero-latitude equator. y_s lies in the equatorial plane and z_{ECEF} points toward the Earth's North Pole. Note how the radius endpoint is not coincident with the center of the ellipsoid; this is because the radius emanates from a plane tangent to the ellipsoid surface.

3.1.2 Direction Cosine Matrices

If reference frames were languages, direction cosine matrices would be interpreters. It allows a vector's orientation to be expressed as components among relative coordinate systems. As the name suggests, rotations are achieved by defining a matrix of direction cosines[3] that relate *unit vectors* in one axis system to those in another, preserving the length of the rotated vector. The determination of matrix elements may be accomplished by inspection; McRuer et al. [11] and Stevens and Lewis [13] note several general properties for construction

[3] Definition

of these matrices:

- "The one is always associated with the axis about which rotation occurs."
- "The remaining elements in the row and column containing the one are all zeros."
- The remaining main diagonal terms are the cosine of the angle of rotation.
- The remaining matrix elements contain the sine of the angle of rotation and are always symmetrically placed relative to the cosine terms; this is done so that zero rotation produces an identity matrix.
- "In the direct right-handed rotation the negative sign always appears in the row above the one (this is to be interpreted as the third row if the one is in the first)."
- "Changing the sign of the rotation angle yields the matrix transpose."

A coordinate rotation example from the body axis frame, $\boldsymbol{F_B}$, to north-east-down frame, $\boldsymbol{F_{NED}}$, is illustrated by three plane rotations in Table 3.2.

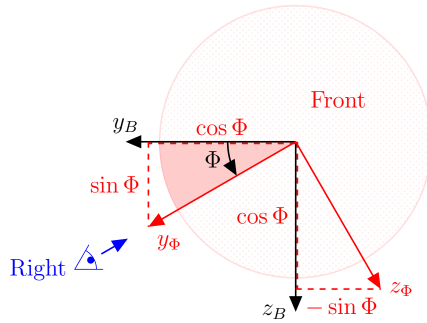
As an example, the array C_Φ from Table 3.2 may be read either left to right or down as $\boldsymbol{y_\Phi} = \boldsymbol{x_B}\ 0 + \boldsymbol{y_B}\ \cos\Phi - \boldsymbol{z_B}\ \sin\Phi$ and $\boldsymbol{y_B} = \boldsymbol{x_\Phi}\ 0 + \boldsymbol{y_\Phi}\ \cos\Phi + \boldsymbol{z_\Phi}\ \sin\Phi$ respectively. The transpose of C_Φ, ie. C_Φ^T allows us to get to $\boldsymbol{x_\Phi}$, $\boldsymbol{y_\Phi}$, $\boldsymbol{z_\Phi}$ from $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$, to be proven later. Doing the left to right read for the remaining rows and corresponding columns leads to a convenient matrix formulation of these equations:

$$\begin{bmatrix} \boldsymbol{x_\Phi} \\ \boldsymbol{y_\Phi} \\ \boldsymbol{z_\Phi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Phi & \sin\Phi \\ 0 & -\sin\Phi & \cos\Phi \end{bmatrix} \begin{bmatrix} \boldsymbol{x_B} \\ \boldsymbol{y_B} \\ \boldsymbol{z_B} \end{bmatrix} \quad \Longleftrightarrow \quad \boldsymbol{F_\Phi} = C_\Phi \boldsymbol{F_B} \tag{3.14}$$

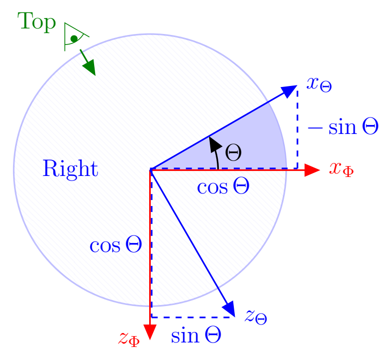
In this formulation C_Φ gets us to $\boldsymbol{x_B}$, $\boldsymbol{y_B}$, $\boldsymbol{z_B}$ from $\boldsymbol{x_\Phi}$, $\boldsymbol{y_\Phi}$, $\boldsymbol{z_\Phi}$. A variety of notations exist for direction cosine matrices, Stevens would write $C_{\boldsymbol{F_\Phi}/\boldsymbol{F_B}}$ instead of C_Φ which explicitly expresses the coordinate frame transformation in the subscript, ie. from $\boldsymbol{F_B}$ to $\boldsymbol{F_\Phi}$. Less trivial than notation are the properties which these rotation matrices possess:

Table 3.2: Direction Cosine Matrices for Plane Rotations

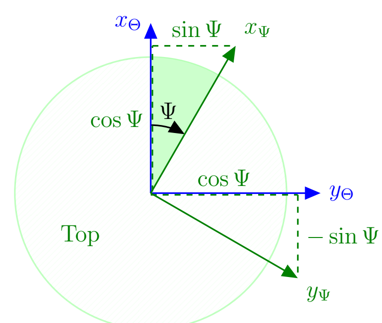
C_Φ	x_B	y_B	z_B
x_Φ	1	0	0
y_Φ	0	$\cos\Phi$	$\sin\Phi$
z_Φ	0	$-\sin\Phi$	$\cos\Phi$



C_Θ	x_Φ	y_Φ	z_Φ
x_Θ	$\cos\Theta$	0	$-\sin\Theta$
y_Θ	0	1	0
z_Θ	$\sin\Theta$	0	$\cos\Theta$



C_Ψ	x_Θ	y_Θ	z_Θ
x_Ψ	$\cos\Psi$	$\sin\Psi$	0
y_Ψ	$-\sin\Psi$	$\cos\Psi$	0
z_Ψ	0	0	1



1. *Orthogonality*

If we let Q be square, $n \times n$, matrix and suppose $Q^{-1} = Q^T$ then Q is orthogonal if and only if:

$$QQ^T = Q^TQ = I \tag{3.15}$$

where I is the identity matrix. This implies that the rotated vector length is unchanged. Alternatively, an orthogonal matrix may be defined as a square matrix with entries whose rows and columns are perpendicular and of unit length, ie. orthogonal unit vectors or orthonormal vectors.

2. *Non-Commutative*

Direction cosine matrices do not commute:

$$C_1 C_2 \neq C_2 C_1 \tag{3.16}$$

3. *Successive rotations may be described the by product of individual plane rotation matrices.*

The orientation of a three-dimensional coordinate frame to another may be obtained by a sequence of three successive rotations. By tradition, aerospace applications perform these transformations through right handed rotations in each coordinate planes, referred to earlier as plane rotations, in the Z-Y-X order Stevens and Lewis [13]; alternate sign convention and planes of rotation exist in other fields, eg. 3D modeling in computer science. Right handed rotation about the z-axis is positive yaw, right handed rotation about the y-axis is positive pitch, and right handed rotation about the x-axis is positive roll. Order of rotation sequence is arbitrary, Figure 3.4 depicts a complete coordinate transformation in a X-Y-Z (Roll-Pitch-Yaw) manner. This rotation sequence is suitable for calculating aircraft attitudes

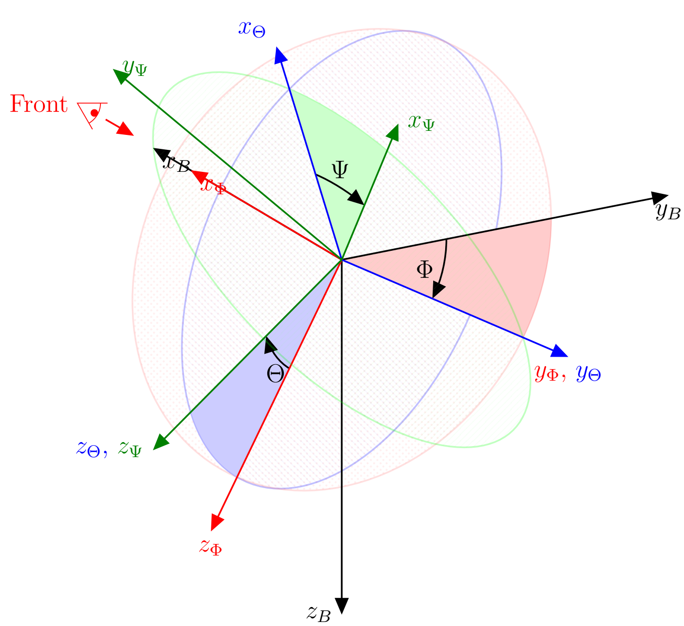


Figure 3.4: Direction Cosine Matrix Visual

with respect to the north-east-down frame. These angles of rotation are called **Euler angles**.

In terms of coordinate transformations

$$\boldsymbol{F_B} = (C_\Phi C_\Theta C_\Psi)\ \boldsymbol{F_{NED}} \tag{3.17}$$

where Equation 3.18 is the complete coordinate transformation from north-east-down to the body frame, ie. $C_{\boldsymbol{F_{NED}}/\boldsymbol{F_B}} = C_\Phi C_\Theta C_\Psi$

$$C_\Phi C_\Theta C_\Psi = \begin{bmatrix} \cos\Theta\cos\Psi & \cos\Theta\sin\Psi & -\sin\Theta \\ -\cos\Phi\sin\Psi + \sin\Phi\sin\Theta\cos\Psi & \cos\Phi\cos\Psi + \sin\Phi\sin\Theta\sin\Psi & \sin\Phi\cos\Theta \\ \sin\Phi\sin\Psi + \cos\Phi\sin\Theta\cos\Psi & -\sin\Phi\cos\Psi + \cos\Phi\sin\Theta\sin\Psi & \cos\Phi\cos\Theta \end{bmatrix} \tag{3.18}$$

3.1.3 Aircraft Dynamics

With Assumptions 3.1 & 3.2 in our front pocket, which is more accessible than the back pocket, we now have an idealized rigid body and live in a world suited for the application Newton's Laws. With this we can describe translational and rotational motion of the aircraft by its kinematic analogs: linear momentum, $\boldsymbol{p}$, and angular momentum, $\boldsymbol{H}$ respectively.

"By Newton's second law the time rate of change of linear momentum equals the sum of all *externally* applied forces, [$\boldsymbol{F}$].

$$\boldsymbol{F} = \frac{d}{dt}(\boldsymbol{p}) = \frac{d}{dt}(m\boldsymbol{V}) \tag{3.19}$$

and the rate of change of angular momentum is equal to the sum of all applied torques

$$\boldsymbol{M} = \frac{d}{dt}(\boldsymbol{H}) \tag{3.20}$$

These vector differential equations provide the starting point for a complete description of the rigid body motions of the vehicle." [11]

Assumption 3.3. Mass is considered constant

In most aerospace systems thrust is generated by an expenditure of vehicle mass; an exception being electric powered applications. Whether that trade in mass directly contributes to linear momentum or not needs to be considered. In the present propulsion case a heavy fuel piston engine turns a propeller, therefore the thrust generated may be considered an external force. Alternatively, if a jet engine were used there would be a component of thrust due to expulsion of vehicle mass[4].

TODO: Rectilinear acceleration eg, to get to expanded forms of Equation 3.19 and Equation 3.20 introduced in beginning of next two sections?

3.1.3.1 Translational Acceleration

The goal is to derive equations for translation accelerations in the wind axis reference frame – $\dot{V}$, $\dot{\alpha}$, $\dot{\beta}$. Picking up where Equation 3.19 left off, along with Assumption 3.3, we may expand the expression to

$$\boldsymbol{F} = m\left[\frac{d}{dt}(\boldsymbol{V}) + \boldsymbol{\Omega} \times \boldsymbol{V}\right] \tag{3.21}$$

where $\boldsymbol{F}$ is the total force acing on the vehicle, m is the vehicle mass, $\boldsymbol{V}$ is the total vehicle velocity, and $\boldsymbol{\Omega}$ is the total vehicle angular velocity:

$$\boldsymbol{F} = \begin{bmatrix} \sum F_x \\ \sum F_y \\ \sum F_z \end{bmatrix} = \begin{bmatrix} F_{x_T} + F_{x_A} + F_{x_G} \\ F_{y_T} + F_{y_A} + F_{y_G} \\ F_{z_T} + F_{z_A} + F_{z_G} \end{bmatrix} \tag{3.22}$$

$$\boldsymbol{V} = [\,u,\, v,\, w\,]^T \tag{3.23}$$

$$\boldsymbol{\Omega} = [\,P,\, Q,\, R\,]^T \tag{3.24}$$

The elements of $\boldsymbol{F}$ are summations of externally applied propulsive (T), aerodynamic (A), and gravitational (G) forces respective to each body axis, (B). It will be assumed

4 McRuer et al. [11] derives a modified extension of Equation 3.19 to take this into account.

that the engine is mounted to align with body axes, therefore there are no thrust-angles or $F_{y_T} = F_{z_T} = 0$.

The body axis aerodynamic forces can be transformed to their equivalent stability axis components lift L, drag D, and side-force Y as Figure 3.5 indicates.

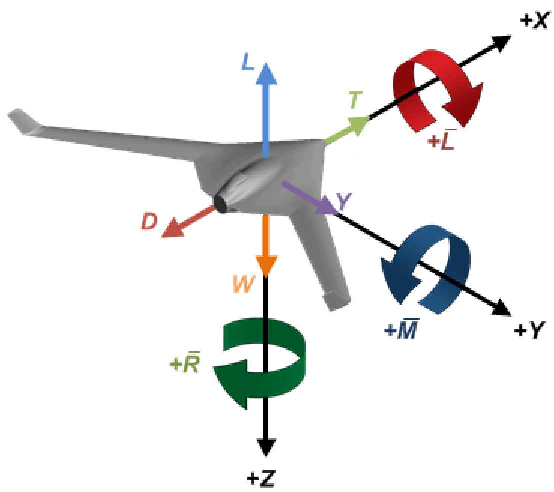


Figure 3.5: Body Axes, Forces, and Moments

$$F_{x_A} = -D\cos\alpha + L\sin\alpha \tag{3.25}$$

$$F_{y_A} = Y \tag{3.26}$$

$$F_{z_A} = -D\sin\alpha - L\cos\alpha \tag{3.27}$$

The gravitational forces can be resolved into body axis components such that

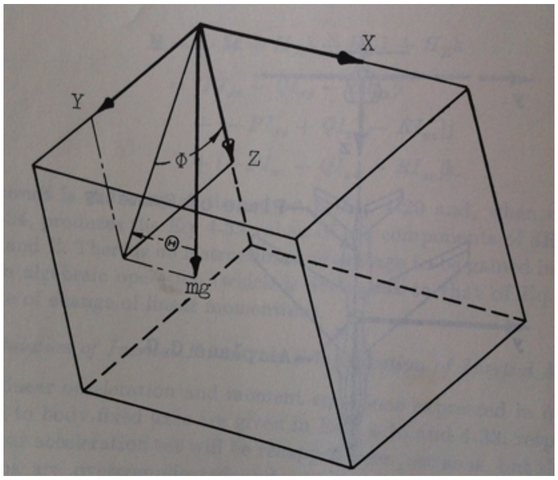


Figure 3.6: Orientation of Gravity Vector with Respect to Body Axes

$$F_{x_G} = -\, mg\sin\theta \tag{3.28}$$

$$F_{y_G} = \quad mg\sin\phi\cos\theta \tag{3.29}$$

$$F_{z_G} = \quad mg\cos\phi\cos\theta \tag{3.30}$$

Combining these we arrive at an expression that considers all external forces acting on the airframe.

$$\begin{bmatrix} \sum F_x \\ \sum F_y \\ \sum F_z \end{bmatrix} = \begin{bmatrix} F_{x_T} - D\cos\alpha + L\sin\alpha - mg\sin\theta \\ Y + mg\sin\phi\cos\theta \\ -D\sin\alpha - L\cos\alpha + mg\cos\phi\cos\theta \end{bmatrix} \tag{3.31}$$

By rearranging Equation 3.21 to solve for translational acceleration, $d\boldsymbol{V}/dt$, we can express body axis accelerations in terms of body axis forces, angular rates, and velocities:

$$\frac{d}{dt}(\boldsymbol{V}) = \frac{1}{m}\boldsymbol{F} - \boldsymbol{\Omega} \times \boldsymbol{V} \tag{3.32}$$

Substitution of Equations 3.23, 3.24, and 3.22 yields

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} \frac{1}{m}\left(F_{x_T} + F_{x_A} + F_{x_G}\right) + Rv - Qw \\ \frac{1}{m}\left(F_{y_T} + F_{y_A} + F_{y_G}\right) + Pw - Ru \\ \frac{1}{m}\left(F_{z_T} + F_{z_A} + F_{z_G}\right) + Qu - Pv \end{bmatrix} \tag{3.33}$$

In order to express Equation 3.33 in the wind axis system, will need to use Equations 3.2, 3.3, and 3.4 as transforms

$$
\begin{aligned}
u =& V\cos\beta\cos\alpha \\
v =& V\sin\beta \\
w =& V\cos\beta\sin\alpha
\end{aligned}
$$

and Equations 3.8, 3.9, and 3.10

$$
\begin{aligned}
V =& \sqrt{u^2+v^2+w^2} \\
\alpha =& \tan^{-1}\left(\frac{w}{u}\right) \\
\beta =& \sin^{-1}\left(\frac{v}{\sqrt{u^2+v^2+w^2}}\right)
\end{aligned}
$$

Leads to $\dot{V}$, $\dot{\alpha}$, and $\dot{\beta}$ equations

3.1.3.2 Rotational Acceleration

The goal is to derive rotational acceleration equations – $\dot{p}$, $\dot{q}$, $\dot{r}$. Picking up where Equation 3.20 left off and substituting total angular momentum for the product of the moment of inertia matrix and rotational velocity vector, $\boldsymbol{H} = \boldsymbol{I\Omega}$, we may expand the expression to

$$
\boldsymbol{M} = \left[\frac{d}{dt}(\boldsymbol{I\Omega}) + \boldsymbol{\Omega} \times \boldsymbol{I\Omega}\right] \tag{3.34}
$$

where $\boldsymbol{M}$ is the total moment acing on the vehicle, $\boldsymbol{I}$ is the inertia matrix (alternatively referred to as tensor or dyad), and $\boldsymbol{\Omega}$ is the total vehicle angular velocity:

$$
\boldsymbol{M} = \begin{bmatrix} \sum L \\ \sum M \\ \sum N \end{bmatrix} = \begin{bmatrix} L + L_T \\ M + M_T \\ N + N_T \end{bmatrix} \tag{3.35}
$$

L, M ,and N are the total aerodynamic moments about the x_B, y_B, and z_B body axes with T subscript indicates moments induced by the power-plant.

$$\boldsymbol{I} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \tag{3.36}$$

Elements along the main diagonal are called the **moments of inertia** with respect to the x, y, and z axes respectively and are always positive. The off diagonal terms are referred to as the **products of inertia** and may be positive, negative, or zero; they are measures of the imbalance in mass distribution. Note that it is possible to orient the axes in such a way that the products of inertia are zero. In this case the diagonal terms are called the principal moments of inertia.

Assumption 3.4. The XZ plane is a plane of symmetry.

This is a very good approximation for most air vehicles, and leads to $I_{yz} = 0$ as well as $I_{xy} = 0$. If we assume that the inertia tensor is constant, as we did with mass in the translational acceleration derivation, then Equation 3.34 may be rewritten as

$$\frac{d}{dt}\boldsymbol{\Omega} = \boldsymbol{I}^{-1}\left(\boldsymbol{M} - \boldsymbol{\Omega} \times \boldsymbol{I}\boldsymbol{\Omega}\right) \tag{3.37}$$

$$\frac{d}{dt}\boldsymbol{\Omega} = [\,\dot{p},\, \dot{q},\, \dot{r}\,]^T \tag{3.38}$$

$$\boldsymbol{I}^{-1} = \frac{1}{\det I}\begin{bmatrix} I_1 & I_2 & I_3 \\ I_4 & I_5 & I_6 \\ I_7 & I_8 & I_9 \end{bmatrix} \tag{3.39}$$

Leads to $\dot{p}$, $\dot{q}$, and $\dot{r}$ equations

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \dots \tag{3.40}$$

3.1.3.3 Attitude Rates

Relative to velocity vector...

Leads to $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$ equations, but I need to derive in terms of flight path components $\dot{\mu}$, $\dot{\gamma}$, and $\dot{\chi}$

$$\begin{bmatrix} \dot{\mu} \\ \dot{\gamma} \\ \dot{\chi} \end{bmatrix} = \dots \tag{3.41}$$

3.1.3.4 Equations of Motion

Complete Set of Nonlinear EOM

$$\begin{aligned}
\dot{\chi} &= \frac{1}{mV\cos\gamma}\left[D\sin\beta\cos\mu + Y\cos\beta\cos\mu + L\sin\mu\right. && (3.42a)\\
&\quad \left. + \, T\left(\sin\alpha\sin\mu - \cos\alpha\sin\beta\cos\mu\right)\right] \\
\dot{\gamma} &= \frac{1}{mV}\left[-D\sin\beta\sin\mu - Y\cos\beta\sin\mu + L\cos\mu\right. && (3.42b)\\
&\quad \left. + \, T\left(\sin\alpha\cos\mu + \cos\alpha\sin\beta\sin\mu\right)\right] - \frac{g}{V}\cos\gamma \\
\dot{V} &= \frac{1}{m}\left(-D\cos\beta + Y\sin\beta + T\cos\alpha\cos\beta\right) - g\sin\gamma && (3.42c)
\end{aligned}$$

$$\begin{aligned}
\dot{\mu} &= \frac{1}{mV}\left[D\sin\beta\tan\gamma\cos\mu + Y\cos\beta\tan\gamma\cos\mu + L\left(\tan\beta + \tan\gamma\sin\mu\right)\right. \\
&\quad \left. + \, T\left(\sin\alpha\tan\gamma\sin\mu + \sin\alpha\tan\beta - \cos\alpha\sin\beta\tan\gamma\cos\mu\right)\right] && (3.42d)\\
&\quad - \frac{g\tan\beta\cos\gamma\cos\mu}{V} + \frac{P\cos\alpha + R\sin\alpha}{\cos\beta} \\
\dot{\alpha} &= -\frac{1}{mV\cos\beta}\left(L + T\sin\alpha\right) + \frac{g\cos\gamma\cos\mu}{V\cos\beta} + Q && (3.42e)\\
&\quad - \tan\beta\left(P\cos\alpha + R\sin\alpha\right) \\
\dot{\beta} &= \frac{1}{mV}\left(D\sin\beta + Y\cos\beta - T\sin\beta\cos\alpha\right) + \frac{g\cos\gamma\sin\mu}{V} && (3.42f)\\
&\quad P\sin\alpha - R\cos\alpha
\end{aligned}$$

$$\begin{aligned}
\dot{P} &= \left(c_1 R + c_2 P\right) Q + c_3\bar{L} + c_4\bar{N} && (3.42g)\\
\dot{Q} &= c_5 PR - c_6\left(P^2 - R^2\right) + c_7\bar{M} && (3.42h)\\
\dot{R} &= \left(c_8 P - c_2 R\right) Q + c_4\bar{L} + c_9\bar{N} && (3.42i)
\end{aligned}$$

where [12] defines c terms as

$$
\begin{aligned}
\Gamma &= I_{xx}I_{zz} - I_{xz}^2 \\
\Gamma c_1 &= \left(I_{yy} - I_{zz}\right) I_{zz} - I_{xz}^2 \\
\Gamma c_2 &= \left(I_{yy} - I_{zz}\right) I_{zz} - I_{xz}^2 \\
\Gamma c_3 &= I_{zz} \\
\Gamma c_4 &= I_{xz} \\
c_5 &= \frac{I_{zz} - I_{xx}}{I_{yy}} \\
c_6 &= \frac{I_{xz}}{I_{yy}} \\
c_7 &= \frac{1}{I_{yy}} \\
\Gamma c_8 &= I_{xx}\left(I_{xx} - I_{yy}\right) + I_{xz}^2 \\
\Gamma c_9 &= I_{xx}
\end{aligned} \tag{3.43}
$$

3.1.4 System Observations

The nonlinear differential equations summarized in § 3.1.3.4 may be reduced to the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}\left(\boldsymbol{x}, t\right) \tag{3.44}$$

where $\dot{\boldsymbol{x}}$ is the $n \times 1$ derivative of the state vector with respect to time, $\boldsymbol{f}$ is an $n \times 1$ non-linear vector function expressing the six-degree of freedom rigid body equations, and $\boldsymbol{x}$ is the $n \times 1$ state-vector with respect to time. Additionally, the state-vector is defined as a set of real numbers, $\left(x_1, \ldots, x_n\right)^T$, contained in n-dimensional Euclidean space, denoted by the symbol $\mathbb{R}^n$, and is formally referred to as **state-space**. The parameter n is the **system order**[5] and refers to the number of first order differential equations required to represent an equivalent n^{th} order ordinary differential equation (ODE).

[5] The highest derivative of the dependent variable with respect to the independent variable appearing in the equation.

Equation 3.44 represents the closed-loop time-variant dynamics of a feedback control system, even though it does not explicitly contain a control input vector $\boldsymbol{u}$. This is because the control input may be considered a function of state $\boldsymbol{x}$ and time t, therefore *disappearing* in the closed-loop dynamics. Showing this mathematically, if the plant dynamics are

$$\dot{\boldsymbol{x}} = \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{u}, t\right) \tag{3.45}$$

and some control law $\boldsymbol{u}$ has been selected as

$$\boldsymbol{u} = \boldsymbol{g}\left(\boldsymbol{x}, t\right) \tag{3.46}$$

then the closed-loop dynamics are

$$\dot{\boldsymbol{x}} = \boldsymbol{f}\left[\boldsymbol{x}, \boldsymbol{g}\left(\boldsymbol{x}, t\right), t\right] \tag{3.47}$$

which can be rewritten in the form of Equation 3.44; since $\boldsymbol{g}(\boldsymbol{x}, t)$ is a function of the state $\boldsymbol{x}$, which is already represented in the expression, it may be discarded. Naturally, Equation 3.44 may also represent a system without control input, such as a freely moving spring-mass damper or pendulum. These examples, as with all physical systems, are time dependent.

Given a particular initial condition, an ODE may have several system trajectories. Continuity of $\boldsymbol{f}$, ie. $\lim\limits_{x \to a} \boldsymbol{f}(x) = \boldsymbol{f}(a)$, ensures that there is at least one solution but does not ensure uniqueness of the solution. A stronger and therefore more frequently used condition, that guarantees *Lipschitz* continuity, may be used to prove existence *and* uniqueness as well as protect against the possibility of $\boldsymbol{f}(x)$ having an infinite slope, eg. a discontinuity.

Definition 3.1.1 (Lipschitz Condition). Khalil [1, §2.2]

If there exists a strictly positive Lipschitz constant L such that $\boldsymbol{f}(\boldsymbol{x}, t)$ satisfies the inequality,

$$\|\boldsymbol{f}(\boldsymbol{x}, t) - \boldsymbol{f}(\boldsymbol{y}, t)\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\| \qquad \forall\, \boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}$$

then the function $\boldsymbol{f}(\boldsymbol{x}, t)$ is said to be *Lipschitz in* $\boldsymbol{x}$ for all points in the domain $\mathcal{D}$.

Further specifying conditions on the domain $\mathcal{D}$, over which the Lipschitz condition holds, imposes restrictions on input values for the function $\boldsymbol{f}(\boldsymbol{x}, t)$. A function is said to be *locally Lipschitz in* $\boldsymbol{x}$ if that for each point $\boldsymbol{x} \in \mathcal{D} \subset \mathbb{R}^n$ there exists a finite neighborhood $\mathcal{D}_0 \in \mathcal{D}$ such that the Lipschitz condition holds for all points in $\mathcal{D}_0$ with a corresponding Lipschitz constant L_0.

Theorem 3.1.1 (Global Existence and Uniqueness). *Khalil [1, Thm 2.4]*
Let $\boldsymbol{f}(\boldsymbol{x}, t)$ be piecewise continuous in t and ***locally Lipschitz in $\boldsymbol{x}$*** *for all $t \geq t_0$ and all $\boldsymbol{x}$ in a domain $D \subset \mathbb{R}^n$ and let W be a* ***compact (closed and bounded) subset*** *of $\mathcal{D}$. If for $x_0 \subset W$ it is known that every solution of*

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, t), \quad \boldsymbol{x}(t_0) = \boldsymbol{x_0} \qquad \forall t \geq t_0$$

lies entirely in W. Then there is a ***unique solution*** *that is defined for all $t \geq t_0$*

Proof: Refer to Khalil [1, Pg. 77] □

"The trick in applying Theorem 3.1.1 is in checking the assumption that every solution lies in a compact set without actually solving the differential equation." This concept is introduced in Lyapunov's stability definitions to come.

Definition 3.1.2 (Autonomous and Non-Autonomous Systems). Slotine and Weiping [2]
The non-linear system $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, t)$ is said to be **autonomous** if $\boldsymbol{f}$ does not depend explicitly on time, ie. if the system's state equation can be written

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) \tag{3.48}$$

otherwise, the system is called **non-autonomous**.

Again, all real-world systems are non-autonomous, but "in practice system properties often change very slowly, so we can neglect their time variation without causing any practically meaningful error."[2] Most importantly, Definition 3.1.2 implies that solutions, or system trajectories, of autonomous ODEs are independent of initial time, thereby greatly simplifying stability analysis. In other words, stability does not depend on initial conditions!

3.2 Stability

The goal is to familiarize the reader with concepts required to prove stability for the backstepping control architecture. Proofs will not be included[6], but will be referenced immediately succeeding theorems. It is assumed that the reader has a good understanding of solution properties to ordinary differential equations such as existence, uniqueness, and continuity. Mathematical notation and terminology tied to these properties may be used in subsequent theorems and definitions without introduction. A great overview of mathematical preliminaries pertinent to nonlinear systems and [backstepping] control is included in [1, Chp. 2]. The following standard mathematical abbreviation symbols shall be used:

$\forall$	"for all"
$\exists$	"there exists"
$\ni$	"such that"
$\in$	"in" or "belongs to"
$\subset$	"a subset of"
$\Rightarrow$	"implies"

Backstepping control designs are constructed using Lyapunov stability theory. This is currently the most useful and general approach [2] to establishing stability for non-linear systems and may also be extended to linear systems. It was published in 1892 by Russian mathematician Alexandr Lyapunov in *The General Problem of Motion Stability for systems without inputs*. As a consequence it has traditionally been applied to closed-loop control systems, ie. those in which a feedback control law has already been selected. Later, in § 3.3.3, a method for designing a feedback control law in conjunction with Lyapunov theory will be introduced. Furthermore, Lyapunov stability theory provides two methods for stability

[6] Consider including in appendix?

analysis, indirect and direct. The first method[7], indirect or linearization, determines *local* stability properties; eigenvalues of a linear (approximate, hence indirect) system reveals stability in the immediate vicinity of an equilibrium point. The second method, direct, determines *regional* stability properties; the aim is to make the system act like a function whose time derivative guarantees some form of stability. Since our system equations are non-linear only Lyapunov's direct method will be covered.

Ultimately, this section condenses the philosophy, definitions, and theorems of Slotine and Weiping [2, Chp. 3], Khalil [1, Chp. 3], Härkegå rd [14, Chp. 3], Krstic et al. [8, Chp. 2] and Farrell and Polycarpou [15, Appendix A] in a brief and clear manner.

3.2.1 Equilibrium and Operating Points

A system trajectory may correspond to only a single point $\boldsymbol{x}^*$, called an **equilibrium point**, if once $\boldsymbol{x}(t)$ is equal to $\boldsymbol{x}^*$ it remains equal to $\boldsymbol{x}^*$ for all time. For the non-autonomous system in Equation 3.44, the equilibrium points are the real roots (x-intercepts (for $n = 2$ systems)) of the differential equation, that is

$$\boldsymbol{f}\left(\boldsymbol{x}^*, t\right) = 0, \quad \forall t \geq 0 \tag{3.49}$$

An **operating point** is a region of stability formally defined as "any state space location at which the system can be forced into equilibrium *by choice of control signal.*" For the generalized system containing control input $\boldsymbol{u}$ in Equation 3.46, the vectors $(\boldsymbol{x_0}, \boldsymbol{u_0})$ are operating points if

$$\boldsymbol{f}\left(\boldsymbol{x_0}, \boldsymbol{u_0}, t\right) = 0, \quad \forall t \geq 0 \tag{3.50}$$

[7] In my research I found conflicting evidence on what Lyapunov's first method actually was. [2] warns the reader that linearization is sometimes incorrectly referred to as the first method, which should be Lyapunov's method of exponents. This was confirmed, as many other sources I found on the topic referred to the indirect method as the first method.

Varying the control input changes the operating point, implying that these points are not isolated. A collection of these points is called a surface of operating points, and is illustrated in the following example through multiple phase portraits.

For a second order system ($n = 2$), solutions of an ODE may be realized in **phase-space**[8] as trajectories from $t = (0, \infty)$.

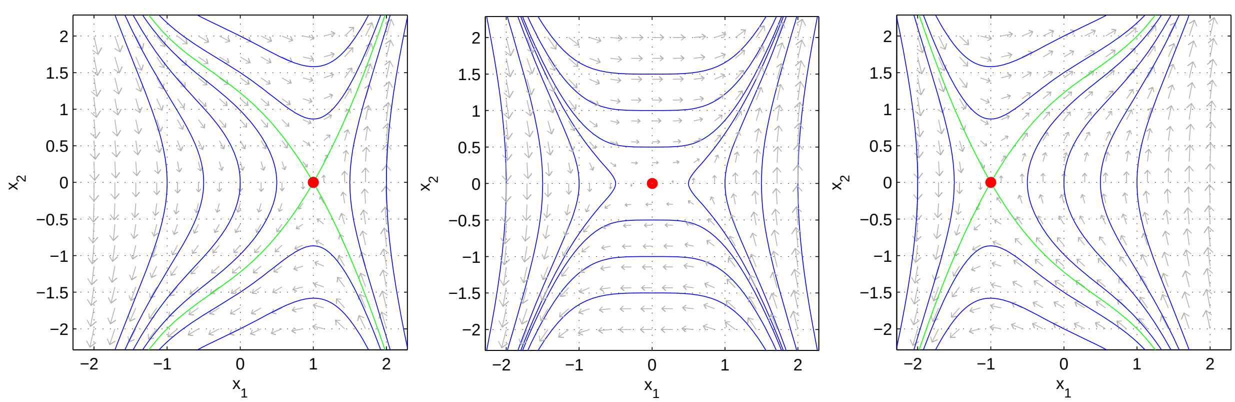
Example 3.2.1 (Operating Points)**.**

Given the system

$$\dot{x_1} = x_2$$

$$\dot{x_2} = x_1^3 + u$$

and applying the operating point definition for an arbitrary control signal $\boldsymbol{u_0} = -1$, an operating point emerges at $\boldsymbol{x_0} = [1, 0]^T$. If the control input is varied, we can see the surface of operating points:



(a) $\boldsymbol{u_0} = -1, \quad \boldsymbol{x_0} = [1, 0]^T$ (b) $\boldsymbol{u_0} = 0, \quad \boldsymbol{x_0} = [0, 0]^T$ (c) $\boldsymbol{u_0} = 1, \quad \boldsymbol{x_0} = [-1, 0]^T$

Figure 3.7: Operating Surface via Phase Portraits

Visualization of trajectories, through phase portraits[9], provides an intuitive feel for the

[8] A vector field of derivatives $[f_2(x_1, x_2),\ f_1(x_1, x_2)]$ at respective state variable $(x_2,\ x_1)$ locations that allows for visualization of the qualitative behavior of the system. Especially useful for classifying stability of equilibrium points [9] Made using pplane8, http://math.rice.edu/~dfield/index.html

stability of an operating point. Red dots (•) are operating points, green lines (—) are stable or unstable manifolds, blue lines (—) are solution trajectories, and grey vectors (→) indicate solution direction tangent to trajectories. Qualitative graphical insight is often more clear, simple, and useful than an analytical approach. You can see that the operating point moves along the x_1 axis as control input is varied and that trajectories flow away from these operating points, therefore they're unstable. The surface of operating points may be expressed as $\boldsymbol{x_0} = [a, 0]^T$ with $\boldsymbol{u_0} = -a^3$. This example proves that operating points are not isolated. Most importantly, the system could be forced to operate at any point on the surface if a stabilizing controller, $u = -x_1^3 - (x_1 - a) - x_2$, was selected.

As Figure 3.7 shows, the operating point is not always coincident with the state-space origin, ie. $\boldsymbol{x} = \boldsymbol{0}$. For the sake of notational and analytical simplicity one may translate the equilibrium point to the origin *without loss of generality* by redefining the state-vector. This allows one to analyze the local stability of the system, neglecting possible higher order terms $\mathcal{O}^3$ and above. Using notation from [8, Pg. 23]:

$$\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{x}^* \tag{3.51}$$

To prove this statement, substitute a reformulation of previous expression, $\boldsymbol{x} = \boldsymbol{z} + \boldsymbol{x}^*$, into Equation 3.44 as shown:

$$\dot{\boldsymbol{z}} = \boldsymbol{f}(\boldsymbol{z} + \boldsymbol{x}^*, t) \tag{3.52}$$

It is clear that by substituting Equation 3.51 into Equation 3.52 one would arrive at a system equivalent to the original, ie. $\dot{\boldsymbol{z}} = \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, t)$. "In addition $\boldsymbol{z} = 0$, the solution corresponding to the original system's equilibrium point $\boldsymbol{x} = \boldsymbol{x}^*$, is an equilibrium point of Equation 3.52; [recall Equation 3.49]. Therefore, instead of studying the behavior of the original system, Equation 3.44, in the neighborhood of $\boldsymbol{x}^*$, one can equivalently study behavior of the redefined system, Equation 3.52, in the neighborhood of the origin"[2] ; what was meant by, "without loss of generality."

In cases like aircraft trajectory control, the concept of stability about a point is not what we're concerned about. In the presence of atmospheric disturbances trajectory perturbations will arise, and it is the stability of *motion* that is important: Will the system remain on its desired trajectory if slightly perturbed away from it? Lyapunov synthesis will answer this question.

3.2.2 Lyapunov Stability Definitions

A **stable system** is one that starts near a desired operating point and stays within a bound of that point ever after; unstable otherwise. Linear stability is evaluated in terms of a single equilibrium point, while non-linear stability is based on the idea of boundedness as these systems can have several isolated equilibrium points. This implies much more complex and unfamiliar behavior for non-linear systems, therefore requiring more refined stability concepts. This section will formally introduce these concepts in the Lyapunov sense, thereby providing the tools necessary to prove stability for backstepping control architectures.

To begin, let ϵ denote a spherical region defined by $\|\boldsymbol{x}\| < \epsilon$ in state-space and δ denote a spherical region that is generally within ϵ; δ is called a domain of attraction. Recall that equilibrium points may be translated to the origin by redefining the state-vector as shown in Equation 3.51. The theorems in this section will be presented as if the equilibrium point was translated to the origin; note that $\boldsymbol{z}$ notation will be dropped, and the usual $\boldsymbol{x}$ representation will be used to represent a system with redefined equilibrium point.

Definition 3.2.1 (Stability in the Sense of Lypaunov). Khalil [1, Def 3.2]

For the non-autonomous system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, t)$$

where $x \in \mathbb{R}^n$, and$f : \mathbb{R}^n \times \mathbb{R}_+ \rightarrow \mathbb{R}^n$ is piecewise continuous in t and locally Lipschitz in x, the equilibrium point $\boldsymbol{x}^* = 0$ is

- **stable** if for each $\epsilon > 0$, there exists a variable $\delta = \delta(\epsilon, t_0) > 0$ such that

$$\|\boldsymbol{x}(t_0)\| < \delta \quad \Rightarrow \quad \|\boldsymbol{x}(t)\| < \epsilon \quad , \qquad \forall\, t \geq t_0 \geq 0 \tag{3.53}$$

- **uniformly stable** if for each $\epsilon > 0$, there exists a variable $\delta = \delta(\epsilon) > 0$, *independent of* t_0 such that Equation 3.53 is satisfied
- **unstable** if not stable.
- **asymptotically stable** if it is stable and there exists a constant $c = c(t_0) > 0 \ni$

$$\|\boldsymbol{x}(t_0)\| < c \quad \Rightarrow \quad \lim_{t \to \infty} \boldsymbol{x}(t) = \boldsymbol{x}^* = 0 \tag{3.54}$$

- **uniformly asymptotically stable** if it is uniformly stable and there exists a constant $c > 0$ *independent of* t_0, such that Equation 3.54 is satisfied uniformly in t_0; that is, for each $\eta > 0$, there is $T(\eta) > 0$ such that

$$\|\boldsymbol{x}(t)\| < \eta \quad , \qquad \forall\, t \geq t_0 + T(\eta) \quad , \quad \forall\, \|\boldsymbol{x}(t_0)\| < c \tag{3.55}$$

- **globally uniformly asymptotically stable** if it is uniformly asymptotically stable and for each $\eta > 0$ and $c > 0$ there is $T(\eta, c) > 0$ such that

$$\|\boldsymbol{x}(t)\| < \eta \quad , \qquad \forall\, t \geq t_0 + T(\eta, c) \quad , \quad \forall\, \|\boldsymbol{x}(t_0)\| < c \tag{3.56}$$

- **exponentially stable** if for any $\epsilon > 0$ and some $\lambda > 0$ there exists $\delta = \delta(\epsilon) > 0 \ni$

$$\|\boldsymbol{x}(t_0)\| < \delta \quad \Rightarrow \quad \|\boldsymbol{x}(t)\| < \epsilon e^{-\lambda(t-t_0)} \quad , \quad \forall\, t > t_0 \geq 0 \tag{3.57}$$

I think that asymptotic stability uses c–η instead of ϵ–δ because its already used in the stability def and asymptotic stability requires stability. Also, if you chose delta and epsilon large enough couldn't any system be stable?

Again these definitions are developed with respect to a non-autonomous, or time-variant, system therefore include initial time, t_0. It is presented in this form for the sake of generality, as these definitions are also applicable to autonomous systems by the substitution $t_0 = 0$. It is clear through these definitions that Lyapunov evaluated stability by ensuring that solutions were not only bounded, but also that "the bound on the solution [could] be made as small as desired by restriction of the size of the initial condition."[15].

With regards to *uniform* stability, the additional stipulation over ordinary stability is that δ is independent of t_0; all properties are uniform if the system is time-invariant. Important in the equilibrium points with forms of asymptotically stable is the additional stipulation on initial time or states, ie. $\|\boldsymbol{x}(t_0)\| = \|\boldsymbol{x_0}\| < c(t_0)$. This implies that an attractive region, c, *for every initial time*, $c(t_0)$, exists: $c(t_0) > 0$. Stability properties are said to be *global* when the domain is equal to $\mathbb{R}^n$. I DONT SEE THIS EXPLICITLY IN KHALIL'S DEF. Terms like asymptotic and global stability are foreign in classic controls sense because all linear time-invariant (LTI) solutions are global and exponential. To conclude comments on Definition 3.2.1, the state vector of the exponentially stable system converges faster than an exponential function, where the positive number λ is the rate of exponential convergence. "By writing the positive constant ϵ as $\epsilon = e^{\lambda(t-t_0)}$ it is easy to see that after a time of $\tau_0 + (1/\lambda)$, the magnitude of the state vector decreases to less than 35% ($\approx e^{-1}$) of its original value; this is similar to the notion of a time-constant in a linear system. After $\tau_0 + (3/\lambda)$, the state magnitude $\|\boldsymbol{x}(t)\|$ will be less than 5% ($\approx e^{-3}$) of $\|\boldsymbol{x}(0)\|$" [2]. Also, exponential stability implies asymptotic stability, but not the other way 'round.

It turns out that operating points which are uniformly asymptotically stable are more robust to disturbances than those that are merely stable, especially important for adaptive designs [1, Sec. 2.1]. It's also important to remember that these are merely definitions; positive constants δ, ϵ, c, and η are *arbitrarily fixed bounds* that are used to evaluate solution trajectories.

For systems up to second order ($n = 1, 2$), a two-dimensional[10] graphical analysis via vector fields and solution trajectories, called **phase portraits**, may be utilized to determine stability. First order systems may be represented as a vector field on the "x-axis": it dictates

[10] However, technically speaking phase portraits may be drawn in three dimensional space. This is not very common, and can be complicated to generate as well as interpret.

the velocity vector $\dot{\boldsymbol{x}}$ at each $\boldsymbol{x}$. "The behavior of the solution in the neighborhood of the origin can be determined by examining the sign of $\boldsymbol{f}(x)$. The $\epsilon - \delta$ requirement for stability is violated if $\boldsymbol{x}(\boldsymbol{f}(x)) > 0$ on either side of the origin."

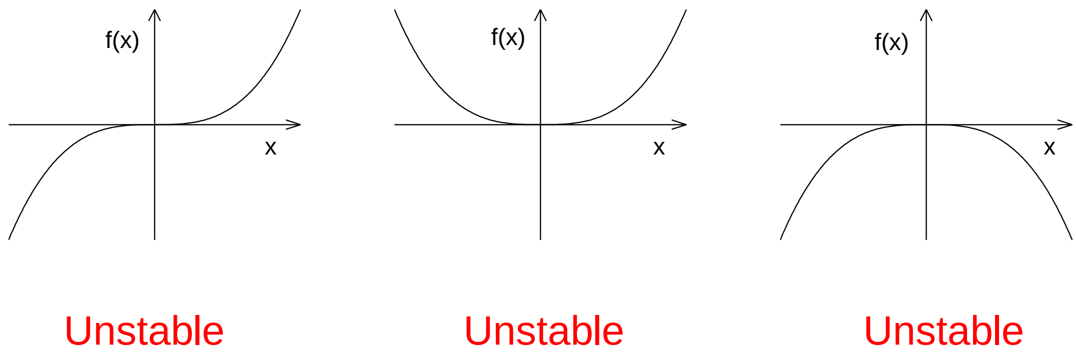


Figure 3.8: First Order Unstable Systems [1, Online Lecture Notes]

"The origin is stable if and only if $\boldsymbol{x}\boldsymbol{f}(x) \leq 0$ in some neighborhood of the origin"

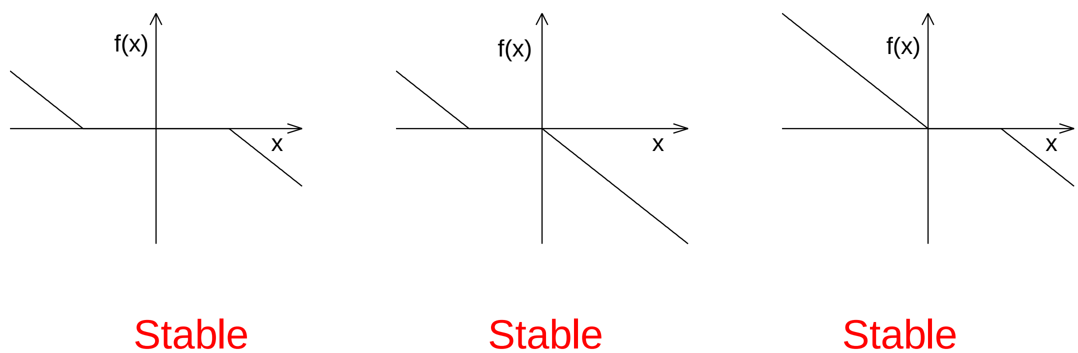
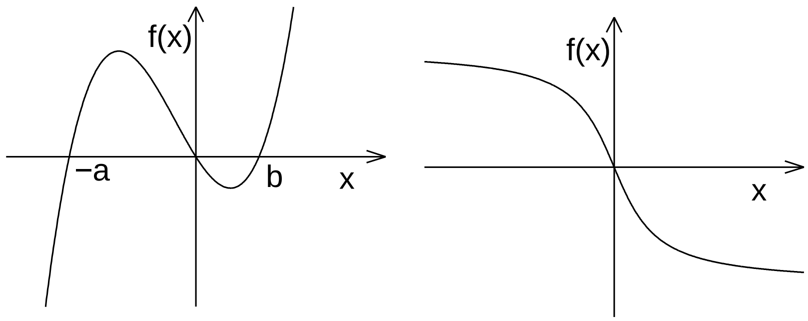


Figure 3.9: First Order Stable Systems [1, Online Lecture Notes]

"The origin is asymptotically stable if and only if $\boldsymbol{x}\boldsymbol{f}(x) < 0$ in some neighborhood of the origin"



(a) Asymptotically Stable (b) Globally Asymptotically Stable

Figure 3.10: First Order Asymptotically Stable Systems [1, Online Lecture Notes]

In two-dimensions, Figure 3.11 depicts stable (S), unstable (US), and asymptotically stable (AS) trajectories in phase-space with respect to ϵ and δ contours in Definition 3.2.1; the initial condition is $x(0)$, solid line is the trajectory $x(t)$, and origin is the equilibrium point x^*.

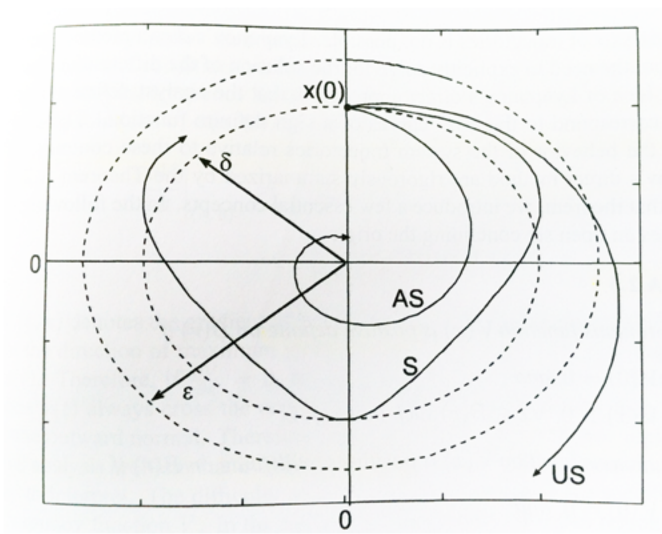


Figure 3.11: Concepts of Stability

This representation is a great conceptual tool, as it brings tangible meaning to the definitions. It may be used to analyze non-linear systems, especially useful for those with multiple equilibrium points, as around each point a linear system approximation is valid. In practice, many non-linear aircraft control architectures are linearized about a trim condition in order to apply classical frequency domain control techniques such as Bode or Root-Locus analysis. These are formalized and well understood techniques which use performance metrics, such as gain and phase margin, to ensure a particular degree of stability.

Figure 3.12 illustrates possible equilibrium point classifications in equivalent root-loci and phase-portraits:

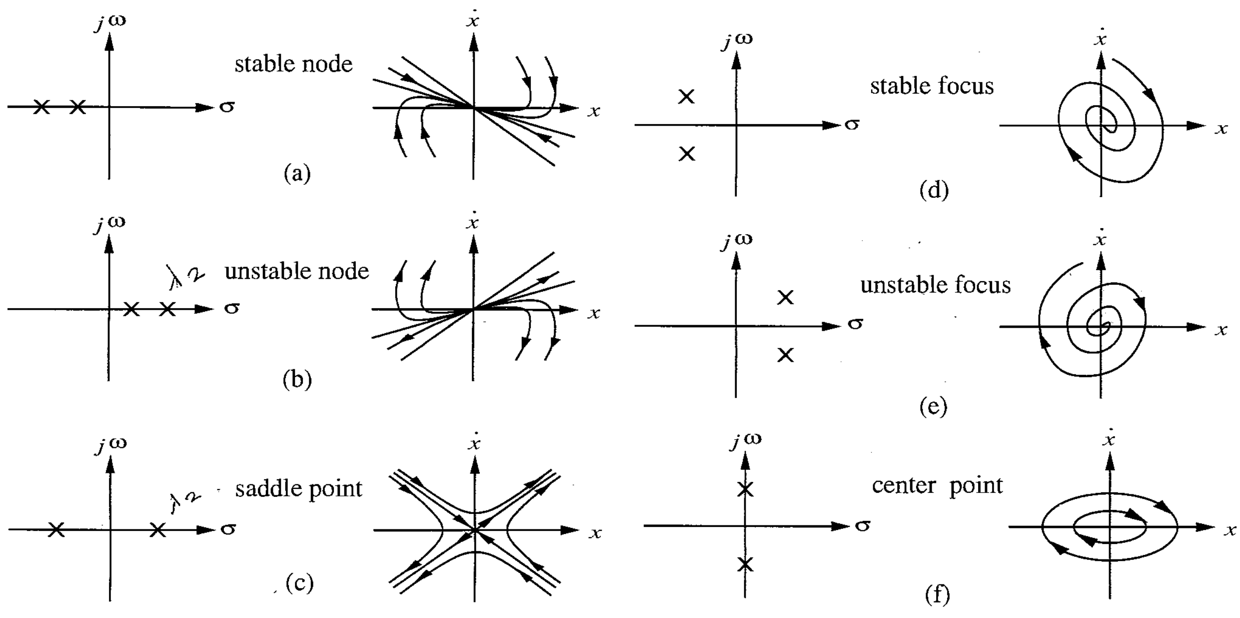


Figure 3.12: Equilibrium Point Classification for 2^{nd} Order Linear Systems [2]

These concepts evaluate the stability of an *ideal* system. Undoubtedly there are model uncertainties[11] and disturbances that effect the true stability of the air vehicle. Boundedness concepts from stability analysis in the Lyapunov sense may be applied to a reformulated system if uncertainties do not effect system order, as:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, t) + \boldsymbol{g}(\boldsymbol{x}, t) \tag{3.58}$$

where $\boldsymbol{f}(\boldsymbol{x}, t)$ is the nominal system and $\boldsymbol{g}(\boldsymbol{x}, t)$ is the perturbation term. Typically we don't know $\boldsymbol{g}(\boldsymbol{x}, t)$ but we know something about its bounds, for example an upper bound

[11] Errors in modeled parameters are called parametric uncertainties, while neglected or unknown parameters are non-parametric uncertainties.

on $\|\boldsymbol{g}(\boldsymbol{x},t)\|$. Note that if $\boldsymbol{g}(\boldsymbol{x},t) \neq 0$ the origin is not necessarily an equilibrium point of Equation 3.58. If a solution trajectory is kept arbitrarily close to an equilibrium point in the presence of **sufficiently small disturbances** then the system may be considered **totally stable**. The following definition formalizes this concept:

Definition 3.2.2 (Total Stability). Slotine and Weiping [2, Defn. 4.13]
The equilibrium point $\boldsymbol{x}^* = \boldsymbol{x}(0) = 0$ for the unperturbed system in Equation 3.48, $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$, is said to be totally stable if for every $\epsilon \geq 0$, two numbers exist δ_1 and δ_2 exist such that $\|\boldsymbol{x}(t_0)\| < \delta_1$ and $\|\boldsymbol{g}(\boldsymbol{x},t)\| < \delta_2$ imply that every solution $\boldsymbol{x}(t)$ of the perturbed system Equation 3.58 satisfies the condition $\|\boldsymbol{x}(t_0)\| < \epsilon$

"Note that total stability is simply a local version (with small input) of BIBO (bounded-input bounded-output) stability." Furthermore, equilibrium points that are uniformly asymptotically stable, therefore exponentially stable points as well, may be proven totally stable by use of converse Lyapunov theorems, mentioned in § 3.2.3.

Theorem 3.2.1 (Asymptotic Stablity and Total Stability). *[2, Thm. 4.14]*
If the equilibrium point $\boldsymbol{x}^* = \boldsymbol{x}(0) = 0$ for the unperturbed system in Equation 3.48, $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$, is uniformly asymptotically stable, then it is totally stable.

Proof: Refer to Khalil [1, Chp. 5, Stability of Perturbed Systems] □

Unfortunately however, most physical systems of interest are higher than second order and are therefore incapable of being displayed graphically. Lyapunov's direct method provides an analytical tool for this case.

3.2.3 Lyapunov Stability Theorems

Aleksandr Lyapunov realized that stability of an equilibrium point may be established if one can make the system act like a scalar, energy-like function, $V(x)$, and examine its time derivative along trajectories of the system. If this function's time derivative, $\dot{V}(x)$, is decreasing over time then it may be asserted that the system will eventually reach an equilibrium condition. Table 3.3 outlines the theorems that will be covered.

Table 3.3: Stability Theorem Overview

$\dot{\boldsymbol{x}} = \boldsymbol{f}(x)$	$V > 0$	$\dot{V} \leq 0$	S	Lyapunov's Direct Method	Thm. 3.2.2
	$V > 0$	$\dot{V} < 0$	AS	Lyapunov's Direct Method	Thm. 3.2.2
	$V > 0$	$\dot{V} < 0$	GAS	Lyapunov's Direct Method	Thm. 3.2.2
	$V \geq 0$	$\dot{V} \leq 0$	CONV	LaSalle	Thm. 3.2.3
	$V > 0$	$\dot{V} \leq 0$	GAS	Barbashin–Krasovskii	Cor. 3.2.1
$\dot{\boldsymbol{x}} = \boldsymbol{f}(x, t)$	$V \geq 0$	$\dot{V} \leq 0$	CONV	Barbalat's Lemma	Lem. 3.2.2
	$V > 0$	$\dot{V} \leq 0$	GUAS	LaSalle-Yoshizawa	Thm. 3.2.4

Before presenting stability theorems, some function related terminology must be introduced:

Definition 3.2.3. A scalar function $V(x)$ is

- **positive definite** if $V(0) = 0$ and $V(x) > 0$, $x \neq 0$
- **negative definite** if $V(0) = 0$ and $V(x) < 0$, $x \neq 0$
- **positive semidefinite** if $V(0) = 0$ and $V(x) \geq 0$, $x \neq 0$
- **negative semidefinite** if $V(0) = 0$ and $V(x) \leq 0$, $x \neq 0$
- **sign indefinite** if it is not any of the above
- **radially unbounded** if $V(x) \to \infty$ as $\|\boldsymbol{x}\| \to \infty$

Where $V(x)$ is assumed to be a scalar function on $\mathcal{D}$ into $\mathbb{R}$, ie. $V : \mathcal{D} \to \mathbb{R}$, is continuously differentiable, and is defined in a domain $\mathcal{D} \subset \mathbb{R}^n$ that contains the origin $\boldsymbol{x} = 0$. When a function is positive or negative (semi)definite there is one and only one unique global minimum or maximum respectively; this is the mathematical justification for stability. The derivative of $V(x)$ along the system trajectory for the autonomous system in Equation 3.48 is obtained by the chain rule:

$$\dot{V}(\boldsymbol{x}) = \frac{d}{dt}V(\boldsymbol{x}) = \frac{\partial V}{\partial \boldsymbol{x}}\frac{d\boldsymbol{x}}{dt} = \frac{\partial V}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} = \frac{\partial V}{\partial \boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x}) \tag{3.59}$$

The scalar function $V(\boldsymbol{x})$ has an implicit dependence on time and its derivative is dependent on the system's equation, therefore each system will have a different $\dot{V}(\boldsymbol{x})$. Not to mention that the form of $V(\boldsymbol{x})$ is anything but consistent, as will be covered later.

Theorem 3.2.2 (Lyapunov's Direct Method). *Khalil [1, Thm. 3.1]*
Let the origin be an equilibrium point, $\boldsymbol{x}^* = \boldsymbol{x}(0) = 0$, for an autonomous system, $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ and $V : \mathbb{R}^n \to \mathbb{R}$ a continuously differentiable, ***positive definite*** *function, then*

- *the origin is* ***stable*** *if*

$$\dot{V}(\boldsymbol{x}) = \frac{\partial V}{\partial \boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x}) \leq 0 \quad , \quad \forall\, \boldsymbol{x} \in \mathcal{D} \tag{3.60}$$

- *the origin is* ***asymptotically stable*** *if*

$$\dot{V}(\boldsymbol{x}) = \frac{\partial V}{\partial \boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x}) < 0 \quad , \quad \forall\, \{\boldsymbol{x} \in \mathcal{D} \,|\, \boldsymbol{x} \notin 0\} \tag{3.61}$$

Proof: Refer to Khalil [1, Pg. 101] □

Remark (Theorem 3.2.2) Because the system is time-invariant these equilibrium point properties are *uniform*, ie. uniformly stable and uniformly asymptotically stable. At its core, Lyapunov's direct method determines stability properties of $\boldsymbol{x}$ through a relationship between $\boldsymbol{f}$ and a positive definite function, $V(\boldsymbol{x})$. This theorem asserts that if the system loses energy over time it will eventually reach an equilibrium condition.

A function $V(\boldsymbol{x})$ *satisfying Theorem 3.2.2* is called a **Lyapunov function**, otherwise a **potential function**. From a physics approach, it is essentially a model of system energy and therefore typically takes the form of a quadratic, kinetic-energy like term. It is most easily understood from a geometric perspective through a two-dimensional phase-portrait.

A Lyapunov function may be visualized as a *collection* of closed concentric[12] contours in $\mathbb{R}^3$; in the simplest sense, imagine looking down a bowl-like shape with numerous rings drawn on the inside that specify height in that space. Figure 3.13 demonstrates this analogy for decreasing values of c.

Formally, each contour, c, characterizes a level set[13] of the Lyapunov function $V(\boldsymbol{x}) = c$ for some $c > 0$, and is called a **Lyapunov or level surface**. It's very important to understand that a Lyapunov function is a projection of the state variables x_1 and x_2 onto a third dimension representing the energy of the system; in other words, the state solution is not fit to a Lyapunov function. It is directly dependent on system trajectories, ie. $V(\boldsymbol{x})$, which are solutions to the differential equations that govern system dynamics, thereby indirectly dependent on system dynamics. The take away is that V is unique to every system and measures how far from equilibrium the system is.

I think I've cleared up my initial confusion. You don't pick an energy function you want and make the system fit it, you alter the system until you're satisfied with its energy function.

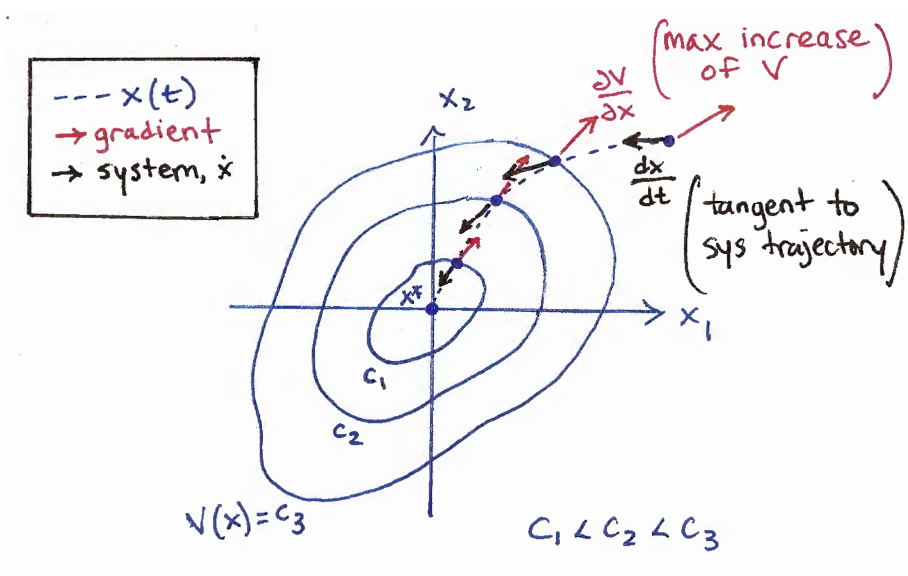


Figure 3.13: Geometric Interpretation of Lyapunov's Stability Theorem.

[12] Not in a circular sense by any means, the contours may comprise any closed shape. The main point is that they share a common center. [13] A level set is a collection of equilibrium points.

The partial derivative term in Equation 3.59 may be considered as the gradient of V with respect to $\boldsymbol{x}$, representing a vector pointing in the direction of maximum increase of V. The vector $d\boldsymbol{x}/dt$ represents the system dynamics, which could've equivalently been labeled $\boldsymbol{f}(\boldsymbol{x})$, and is tangent to the solution $\boldsymbol{x}(t)$. The condition imposed by Lyapunov, ie. $\dot{V}(\boldsymbol{x}) = (\partial V/\partial \boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}) \leq 0$, implies that solutions $\boldsymbol{x}(t)$ always cross the contours of V with an angle greater than or equal to 90° relative to the outward normal; if $d\boldsymbol{x}/dt$ points inward then system trajectories will always move to smaller and smaller values of V. $\dot{V}(\boldsymbol{x}) \leq 0$ does not ensure that a trajectory will get to the origin[14], but does imply the origin is stable, since by Definition 3.2.1: when a trajectory starts within δ (a level surface for this case), it will stay within ϵ.

When an equilibrium point is classified as asymptotically stable it requires the initial condition to be within some domain $\mathcal{D}$, but how big can this domain be? Establishing *global* asymptotic stability expands the region of attraction to the whole space $\mathbb{R}^n$ by an extra condition, radial unboundedness, as described previously in Definition 3.2.3.

Theorem 3.2.2 (continued) *Khalil [1, Thm. 3.2]*

*Let the origin be an equilibrium point, $\boldsymbol{x}^* = \boldsymbol{x}(0) = 0$, for an autonomous system, $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ and $V : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable, positive definite, **radially unbounded** function such that*

$$\dot{V}(\boldsymbol{x}) = \frac{\partial V}{\partial \boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x}) < 0 \quad , \quad \forall\, \boldsymbol{x} \neq 0 \tag{3.62}$$

*then the origin is **globally asymptotically stable**.*

Proof: Refer to Khalil [1, Pg. 111] □

As shown, "the direct method of Lyapunov replaces [an] n-dimensional analysis problem that is difficult to visualize with a lower dimensional problem that is easy to interpret." [15]

[14] LaSalle's Invariance Principle [1, Thm. 3.4] may be used to prove convergence of a solution to the largest invariant set for all point within a region of attraction where $\dot{V}(\boldsymbol{x}) = 0$.

The large appeal of this method is that stability of an equilibrium point may be inferred without explicitly solving system equations. The difficulty is that finding a Lyapunov function for complex systems is like [insert funny simile here]. Most significantly, this theorem is *a*) **non-constructive** – there is no systematic method for finding a V to satisfy stability requirements – and *b*) **only offers sufficient conditions** for stability – it does not say whether the given conditions are also necessary. In other words, if a Lyapunov function does not satisfy the conditions for stability or asymptotic stability, then *no conclusion* can be made about the stability properties of the system.

However, there are tools that not only support the search for V, ie. addressing *a*), but also overturning *b*), referred to as *Converse Lyapunov Theorems*[15]. If a sub-system of a non-linear system exhibits stability then converse theorems may be used to generate a Lyapunov function for the sub-system. This implies that a Lyapunov function for the whole system may exist, thereby making Lyapunov stability conditions necessary; it's nice to know there's hope! Unfortunately however, almost always they assume some knowledge of the solution to the differential equation.

For cases where $\dot{V}$ is only negative semi-definite, ie. $\dot{V} \leq 0$, global asymptotic stability may still be established through LaSalle's[16] Invariance Principle, an invariant set theorem:

Lemma 3.2.1 (Fundamental Property of Limit Sets). *Khalil [1, Lem. 3.1]*

If a solution $\boldsymbol{x}(t)$ to the autonomous system $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ is bounded and belongs to $\mathcal{D}$ for $t \geq 0$, then its positive limit set L^+ is a nonempty, compact, invariant set. Moreover,

$$\lim_{t \to \infty} \boldsymbol{x}(t) = L^+ \tag{3.63}$$

Theorem 3.2.3 (LaSalle's Theorem). *Khalil [1, Thm. 3.4]*

Let $\Omega \subset \mathcal{D}$ be a compact set that is positively invariant for an autonomous system, $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$.

[15] Converse Theorems, see Khalil [1, §3.6]

[16] Joseph P. LaSalle received a mathematics doctoral degree in 1941 from Caltech and worked alongside Lefschetz at Brown University in the 1960's. Oddly there wasn't much more information about him.

Let $V : \mathcal{D} \to \mathbb{R}$ be a continuously differentiable function such that $\dot{V}(\boldsymbol{x}) \leq 0$ in Ω. Let $\mathcal{E}$ be the set of all points in Ω where $\dot{V}(\boldsymbol{x}) = 0$. Let $\mathcal{M}$ be the largest invariant set in $\mathcal{E}$. Then every solution starting in Ω approaches $\mathcal{M}$ as $t \to \infty$.

Proof: Refer to Khalil [1, Pg. 115] □

LaSalle's theorem also extends Lyapunov's theorem in two ways by *a*) providing an estimate to the region of attraction specified as any compact positively invariant set and *b*) allowing Theorem 3.2.2 to be applied for cases where the system has an equilibrium set, ie. dynamic convergence or limit cycles, rather than a single equilibrium point.

Corollary 3.2.1 (Barbashin-Krasovskii). *Khalil [1, Cor. 3.2]*

Let the origin be an equilibrium point, $\boldsymbol{x}^* = \boldsymbol{x}(0) = 0$, for an autonomous system, $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ and $V : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable, positive definite, radially unbounded function such that

$$\dot{V}(\boldsymbol{x}) = \frac{\partial V}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}) \leq 0 \quad , \quad \forall\, \boldsymbol{x} \tag{3.64}$$

*Let $\mathcal{S} = \{x \in \mathbb{R}^n \,|\, \dot{V}(\boldsymbol{x}) = 0\}$ and suppose that no solution can stay identically in $\mathcal{S}$, other than the trivial solution. Then the origin is **globally asymptotically stable**.*

Remark (Corollary 3.2.1) When $\dot{V}(\boldsymbol{x}) \leq 0$ and $\mathcal{S} = \{0\}$, Corollary 3.2.1 coincides with Theorem 3.2.2. It is also referred to as the Krasovskii-LaSalle method in some textbooks. LaSalle published this theorem in the west, unaware that it was earlier published in Russia; most likely attributed to a language barrier or lack of cooperation due to political tension of the 1950's when this theorem was developed.

LaSalle's invariant set based theorem is applicable to autonomous systems that desire state convergence to a constant, therefore time-invariant, reference signal. If the control objective is tracking of a time-varying reference signal then LaSalle's theorem is insufficient because the system is then non-autonomous. Subsequently, stability analysis is more difficult when the system is time-variant because it's harder to find a Lyapunov function, now dependent on both $\boldsymbol{x}$ and t, ie. $V(\boldsymbol{x}, t)$, that has a negative-definite derivative. For tracking analysis, tools developed by LaSalle, Yoshizawa, and Barbalat are relied upon.

Theorem 3.2.4 (LaSalle–Yoshizawa). *Krstic et al. [8, Thm. 2.1/A.8]*
Let the origin be an equilibrium point, $\boldsymbol{x}^* = \boldsymbol{x}(0) = 0$, for a non-autonomous system, $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, t)$ and suppose $\boldsymbol{f}$ is locally Lipschitz in $\boldsymbol{x}$ uniformly in t. Let $V : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}_+$ be a continuously differentiable, positive definite, and radially unbounded function $V = V(\boldsymbol{x}, t)$, then

$$\dot{V}(\boldsymbol{x}, t) = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}, t) \leq -W(x) \leq 0 \quad , \quad \forall\, t \geq 0 \quad , \quad \forall\, \boldsymbol{x} \in \mathbb{R}^n \tag{3.65}$$

Where $W(x)$ is a continuous function. Then all solutions of [Equation 3.65] are globally uniformly bounded and satisfy

$$\lim_{t \to \infty} W(\boldsymbol{x}(t)) = 0 \tag{3.66}$$

*In addition, if $W(x) > 0$ ie. positive definite, then the equilibrium $\boldsymbol{x}^* = \boldsymbol{x}(0) = 0$ is **globally uniformly asymptotically stable**.*

Proof: Refer to Krstic et al. [8, Appendix A, Pg. 492] □

A technical lemma by Barbalat usually precedes Theorem 3.2.4 which is a purely mathematical result inferred by asymptotic properties of functions and their derivatives:

- If $\dot{\boldsymbol{f}}(t) \to 0$ it does not imply that $\boldsymbol{f}(t)$ converges

  Example 3.2.2. As the derivative term converges to zero, the solution does not, for the system:

$$\boldsymbol{f}(t) = \sin(\log(t)) \quad ; \quad \dot{\boldsymbol{f}}(t) = (\cos(\log(t)))/t$$

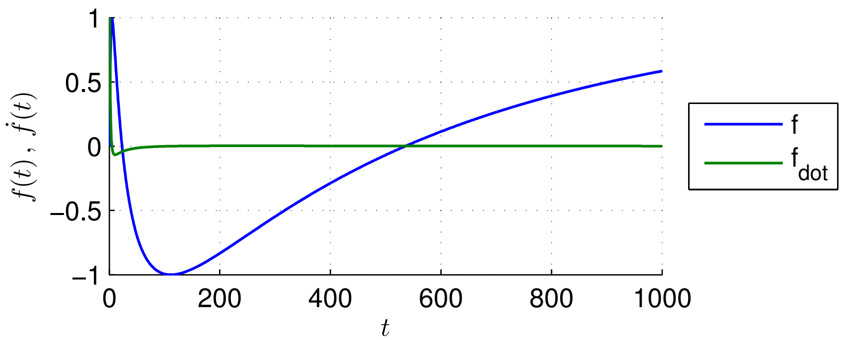


Figure 3.14: Asymptotic Property 1: $\dot{\boldsymbol{f}}(t) \to 0 \not\Rightarrow \boldsymbol{f} \to$ constant

- If $\boldsymbol{f}(t)$ converges it does not; imply that $\dot{\boldsymbol{f}}(t) \to 0$

  Example 3.2.3. As the solution tends to zero, the derivative is unbounded, for the system:

$$\boldsymbol{f}(t) = e^{-t}\sin^2(e^{2t}) \quad ; \quad \dot{\boldsymbol{f}}(t) = 2e^{t}\sin(2e^{2t}) - e^{-t}\sin^2(e^{2t})$$

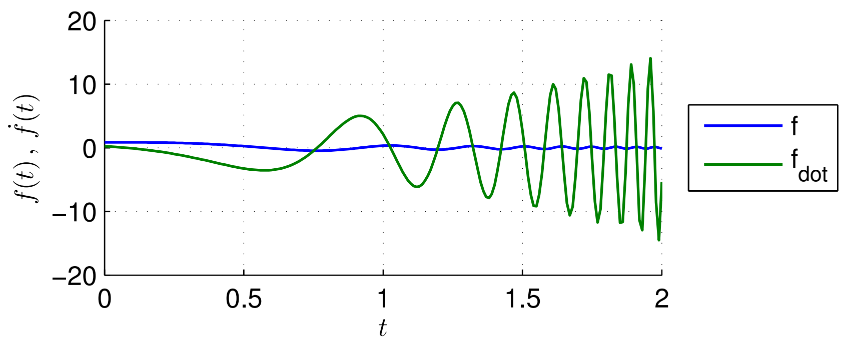


Figure 3.15: Asymptotic Property 2: $\boldsymbol{f} \to$ constant $\not\Rightarrow \dot{\boldsymbol{f}}(t) \to 0$

- If $\boldsymbol{f}(t)$ is lower bounded and decreasing, ie. $\dot{\boldsymbol{f}}(t) \leq 0$, then $\boldsymbol{f}(t)$ converges to a limit, but $\dot{\boldsymbol{f}}(t)$ **is not guaranteed to diminish if at all**.

An additional "smoothness" property on the Lyapunov derivative imposed by Barbalat's lemma guarantees that $\dot{\boldsymbol{f}}(t)$ actually converges to zero. This result ensures that a system will fulfill its tracking requirement.

Lemma 3.2.2 (Barbalat's Lemma)**.**

If it can be shown that the differentiable function is bounded, then it may be considered uniformly continuous and convergence may be established.

- ***Form 1:*** *Examine the Function's Derivative* — *Slotine and Weiping [2]*

  "If the differentiable function $\boldsymbol{f}(t)$ has a finite limit as $t \to \infty$, and is such that $\ddot{\boldsymbol{f}}$ exists and is bounded, [ie. uniformly continuous], then:"

$$\lim_{t\to\infty} \dot{\boldsymbol{f}}(t) = 0$$

- ***Form 2:*** *"Lyapunov-Like Lemma"* — *Slotine and Weiping [2]*

  If a scalar function $V(\boldsymbol{x}, t)$ satisfies the following conditions:

  - $V(\boldsymbol{x}, t)$ *is lower bounded*
  - $\dot{V}(\boldsymbol{x}, t)$ *is negative semi-definite*
  - $\dot{V}(\boldsymbol{x}, t)$ *is uniformly continuous in time (by proving $\ddot{V}$ is bounded)*

  then $\dot{V}(\boldsymbol{x}, t) \to 0$ as $t \to \infty$.

- ***Form 3:*** *$\mathcal{L}_p$ Space Representation* — *Farrell and Polycarpou [15, A.2.2.3]*

  Consider the function $\phi(t) : \mathbb{R}_+ \to \mathbb{R}$ be in $\mathcal{L}_\infty$, $d\phi/dt \in \mathcal{L}_\infty$, and $d\phi/dt \in \mathcal{L}_2$, then

$$\lim_{t\to\infty} \phi(t) = 0$$

- ***Form 4:*** *Initial Version, Barbalat 1959* — *Khalil [1, Lemma 4.2]*

  Let $\phi(t) : \mathbb{R}_+ \to \mathbb{R}$ be a uniformly continuous function. Suppose that $\lim_{t\to\infty} \int_0^t \phi(\tau) d\tau$ exists and is finite, then:

$$\lim_{t\to\infty} \phi(t) = 0$$

Proof: Refer to Slotine and Weiping [2, Sec. 4.5.2, Pg. 124] □

In the next sections you will see how backstepping may be applied to control of nonlinear systems. The procedure involves choosing a $V(x)$ that retains useful non-linearities and developing a stabilizing feedback control law.

3.3 Control

The aim of this section is to introduce control law design considerations and thoroughly present backstepping control theory. A simplified first order non-linear system will serve as the basis for development, in order to clearly conceptualize motivating factors. Key features of the method will be cited along the way, eventually leading to a command filtered backstepping technique for the ninth order non-linear equations of motion formulated in § 3.1.3.4: Equation 3.42.

3.3.1 Requirements

> "The most important design specification is to achieve asymptotic tracking of a known reference trajectory with the strongest possible form of stability. The designed controller should provide effective means for shaping the transient performance and thus allow different performance-robustness trade-offs."
>
> –Krstic et al. [8]

Loosely speaking, the approach to non-linear control design is qualitative, while linear control is quantitative. Numerous analysis tools are available for linear control system design that have very specific metrics common in the controls community; in the time domain step response analysis yields terms like rise time; overshoot; and settling time, in the frequency-domain Bode analysis yields terms like phase margin; gain margin; and bandwidth, and root-locus / Nichols analysis represents a mixture of aforementioned methods. Through use of tools like these control system requirements may be imposed *systematically* on closed-loop controllers for linear systems, hence the term quantitative. For non-linear systems a frequency-domain description is not possible, therefore requirements are typically evaluated using time-domain analyses of transient system response for a given control input. Furthermore, non-linear systems often act in peculiar ways, expressing erratic behavior sensitive to even the smallest changes in initial condition and system parameters, or commands, (think chaotic systems, ie. Lorenz Attractor). This demonstrated inconsistency is a reason why a analysis tools are limited for non-linear systems, each system is unique and exhibited motion is dependent on inputs. As a result, for non-linear systems, *qualitative* requirements for desired behavior are specified within the operating region of interest.

The implications are that on the designers end a much deeper understanding of vehicle dynamics are necessary for non-linear system control development; one can't simply set and forget system equations as with linear systems and then rely on analysis tools to help tune a controller. In fact in backstepping designs if useful, ie. stabilizing, non-linearities are recognized then they may be retained, thereby reducing *a*) the overall control effort needed and *b*) the level of modeling fidelity. The problem is that experience drives these considerations, hence the motivation for including the equations of motion derivation in **??**. Other benefits and disadvantages of backstepping will be covered, but first non-linear control design considerations and options are briefly introduced.

As outlined by Slotine and Weiping [2], the following characteristics are considered by designers when evaluating nonlinear control system requirements and behavior.

- **Stability** *must be guaranteed for the nominal model, either in a local or global sense. The region of stability and convergence are also of interest.*
- **Accuracy and Speed of Response** *may be considered for some motion trajectories, typically derived from mission requirements, in the region of operation.*
- **Robustness** *is the sensitivity to effects which are not considered in the design, such as disturbances, measurement noise, unmodeled dynamics, etc. Leads to robust or adaptive control problem formulations.*
- **Cost** *of a control system is determined by the number and type of actuators, sensors, computers, and time necessary to implement it.*

These metrics are in direct competition with one another, as control systems cannot exhibit each of these characteristics to the full extent. The designer is responsible for making effective trade-offs for conflicting requirements and most importantly recognizing when to freeze design iterations.

Stability in the non-linear sense does not imply that a system is capable of handling *constant* disturbances; recall from Definitions 3.2.1 and 3.2.2 that stability is defined with respect to initial conditions. For example, a system is stable in the Lyapunov sense if a trajectory starts *within* δ and stays within ϵ; a persistent wind-shear – erratic disturbance due to thermals, downdraft, inversion layers, etc. – may shift the equilibrium point, therefore starting within the δ region does not mean you're starting within some bounds of the true equilibrium point. The effects of persistent disturbances are resolved through robustness techniques. "Robustness is a property which guarantees that essential functions of the designed system are maintained under adverse conditions in which the model no longer accurately reflects reality. [4]"

3.3.2 Objective and Methods

Definition 3.3.1. Control Tasks

- **Regulation**: Reference signal is constant.
- **Tracking**: Reference signal is time-varying.

In general, there are two tasks for any flight control system: regulation, sometimes referred to as stabilization, aims to hold a particular state to a time-independent reference value, common examples being temperature control and aircraft altitude control. If the objective of the controller is tracking, commonly called a tracker, then the aim is to make a particular state to follow a time-dependent reference signal, examples include making a robot hand draw circles or aircraft flight path following. The goal in both these cases is to drive the deviation from a desired reference value/signal to zero. Convergence for regulation problems may be achieved using LaSalle's Theorem 3.2.3, while tracking problems rely on Barbalat's Lemma 3.2.2.

One of the simplest and aging rivals of backstepping is feedback linearization, or nonlinear dynamic inversion. In backstepping literature this non-linear control architecture usually plays the victim; it's a point solution, because it's linear, which constructs control laws that cancels non-linear plant dynamics with feedback, just as the term reads. Consequently it's considered wasteful and inflexible, requiring more control effort and exhibiting a lack of robustness to uncertainties. It can get especially complicated for high order systems, over an order of two. Table **??** covers features of a few non-linear control techniques.

Backstepping and feedback linearization cancel *known* nonlinearities... Farrell et al. [16]

Table 3.4: Nonlinear Control Method Overview

Method	Advantage	Disadvantage
Trial and Error	• Visual stability analysis via phase portraits	• Applicable only to simple systems up to second order
Small Singular Linearization / Gain Scheduling	• Good closed-loop performance for a equilibrium point (SSL). • Good closed-loop performance over many equilibrium points (GS).	• Accurate only in a neighborhood around operating point(s) • Controller parameters fixed online • A lot of offline validation required
Feedback Linearization	• Globally stable with exponential tracking error • Linear in modeled domain • Bandwidth theoretically infinite for input signal tracking	• Lack of controllability at singularities • Requires exact knowledge and special class of system • More control effort is required • Not robust to uncertainties
Backstepping / Robust / Adaptive	• Globally asymptotically stable • Model uncertainties well handled • Systematic procedures • Potential reduction in development time • Useful non-linearities retained	• Analytic derivative calculation • Feedback control algorithm complex, especially for high order systems

But how is feedback linearization better than small singular linearization at the same equilibrium point especially since both systems are linear. In the former, exact state transformations and non-linear feedback are used, rather than the latter where linear *approxi-*

mations of the nonlinear system and linear feedback are used. The downside is that all non-linearities must be precisely known. Sonneveldt et al. [3]

"One of the main problems with applying feedback linearization techniques is that the process produces a system with the same **relative degree** as the original system, but usually with an order that is less. This process results in zero or internal dynamics, which are modes that are effectively rendered unobservable by the linearization process. If the system is non-minimum phase, then the zero dynamics are unstable. The analogy with linear systems is that a zero-pole system is linearized into an all-pole system by selecting the pole-zero excess as the order of the approximating system. In order to produce linearized systems that have no internal dynamics, techniques which preserve the dynamic order of the system are needed."

3.3.3 Lyapunov Based Control Design

There are two ways to apply Lyapunov's direct method based on where you begin: *1*) If a control law is hypothesized then a valid Lyapunov function needs to be found to justify the choice. *2*) If a Lyapunov function is hypothesized then a control law needs to be found to make the Lyapunov function valid. This section will apply the latter technique using a **control Lyapunov function** (CLF).

Up to this point Lyapunov theorems have been used to prove stability of a given system, however the main objective is to *design* closed-loop systems with desirable stability properties. The following point by Freeman and Kokotovic [4] clearly summarizes the parallelism between Lyapunov functions and CLFs: "Just as the existence of a Lyapunov function is necessary and sufficient for the stability of a system without inputs [closed-loop], the existence of a CLF is necessary and sufficient for the *stabilizability* of a system with a control input."

The "control" prefix implies that the non-linear system it's applied to has an explicit dependence on u, that is:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, u) \quad , \quad \boldsymbol{x} \in \mathbb{R}^n \quad , \quad \boldsymbol{u} \in \mathbb{R} \quad , \quad \boldsymbol{f}(0,0) = 0 \tag{3.67}$$

If a stabilizing feedback control law $\alpha(\boldsymbol{x})$ is chosen for control input u such that the inequality in Equation 3.68 holds, implying $\dot{\boldsymbol{x}} = \boldsymbol{f}(x, \alpha(\boldsymbol{x}))$ where $\boldsymbol{x} = 0$ is an equilibrium point, then the origin is globally asymptotically stable.

$$\dot{V}(\boldsymbol{x}, \alpha) = \frac{\partial V}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}, \alpha(\boldsymbol{x})) \leq -W(\boldsymbol{x}) \tag{3.68}$$

where $W(x)$ is a positive definite function; see Krstic et al. [8, §2.1.2].

Definition 3.3.2 (Control Lyapunov Function (CLF))**.** Krstic et al. [8, Def. 2.4],[14] A positive definite, radially unbounded, smooth scalar function $V = V(\boldsymbol{x})$ is called a CLF for $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, u)$ if there exists a u such that:

$$\inf_{u \in \mathbb{R}} \left\{ \dot{V}(\boldsymbol{x}, u) = \frac{\partial V}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}, u) < 0 \right\} \quad , \quad \forall\, \boldsymbol{x} \neq 0 \tag{3.69}$$

where **inf** denotes infimum, the greatest lower bound. For systems affine in control, ie.

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})u \quad , \quad \boldsymbol{f}(0) = 0\,, \tag{3.70}$$

the CLF inequality in Equation 3.68 becomes

$$\dot{V}(\boldsymbol{x}, \alpha) = \frac{\partial V}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}) + \frac{\partial V}{\partial \boldsymbol{x}} \boldsymbol{g}(\boldsymbol{x}) \alpha(\boldsymbol{x}) \leq -W(\boldsymbol{x}) \tag{3.71}$$

For this system the only way to satisfy Definition 3.3.2 is if:

$$\frac{\partial V}{\partial \boldsymbol{x}} \boldsymbol{g}(\boldsymbol{x}) = 0 \quad \Rightarrow \quad \frac{\partial V}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}) < 0 \quad , \quad \forall\, \boldsymbol{x} \neq 0$$

The problem with the CLF concept is that for most non-linear systems it is unknown. "The task of finding an appropriate CLF may be as complex as that of designing a stabilizing feedback law. For several important classes of nonlinear systems, we will solve these two tasks simultaneously using a backstepping procedure. [8]"

3.3.4 Backstepping

Key features of backstepping will be verified alongside implementation of the control architecture. Defining attributes and commonly mentioned benefits of backstepping are:

Table 3.5: Backstepping Key Terms

Useful Nonlinearities	Less control effort and less precise model information required
Flexible	Less restrictive, more freedom in choosing stabilizing function $\alpha(x)$ and Lyapunov function $V(x)$
Recursive	System is augmentable by a chain of integrators, creating intermediate states called virtual control laws ξ_k that assist in control
Constructive	Systematic design procedure

Backstepping offers a systematic design procedure via the addition of integrators, and is introduced for two general classes of non-linear systems. The first is a *strict feedback* system, as indicated by Krstic et al. [8, §2.3.1], and is of the form:

$$
\begin{aligned}
\dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\xi_1 \\
\dot{\xi}_1 &= f_1(\boldsymbol{x},\xi_1) + g_1(\boldsymbol{x},\xi_1)\xi_2 \\
\dot{\xi}_2 &= f_2(\boldsymbol{x},\xi_1,\xi_2) + g_2(\boldsymbol{x},\xi_1,\xi_2)\xi_3 \\
&\vdots \\
\dot{\xi}_{k-1} &= f_{k-1}(\boldsymbol{x},\xi_1,\cdots,\xi_{k-1}) + g_{k-1}(\boldsymbol{x},\xi_1,\cdots,\xi_{k-1})\xi_k \\
\dot{\xi}_k &= f_k(\boldsymbol{x},\xi_1,\cdots,\xi_k) + g_k(\boldsymbol{x},\xi_1,\cdots,\xi_k)u
\end{aligned}
\tag{3.72}
$$

where $x \in \mathbb{R}^n$, $\xi_1,\cdots,\xi_k$ are scalar virtual control laws, and the $\boldsymbol{x}$-subsystem satisfies assumptions necessary to apply backstepping, to be introduced in Assumption 3.6. The ξ-system is referred to as "strict-feedback" because "nonlinearities f_i and g_i in the $\dot{\xi}_i$-equation $(i = 1,\cdots,k)$ depend only on $x,\xi_1,\cdots,\xi_i$ that is, on state variables that are fed back."

The second is a *pure feedback* system, as indicated by Krstic et al. [8, §2.3.2], and is of the form:

$$\begin{aligned}
\dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}, \xi_1) \\
\dot{\xi}_1 &= f_1(\boldsymbol{x}, \xi_1, \xi_2) \\
\dot{\xi}_2 &= f_2(\boldsymbol{x}, \xi_1, \xi_2, \xi_3) \\
&\vdots \\
\dot{\xi}_{k-1} &= f_{k-1}(\boldsymbol{x}, \xi_1, \cdots, \xi_k) \\
\dot{\xi}_k &= f_k(\boldsymbol{x}, \xi_1, \cdots, \xi_k, u)
\end{aligned} \tag{3.73}$$

where $\xi_i \in \mathbb{R}^n$ and the $\boldsymbol{x}$-subsystem again satisfies upcoming Assumption 3.6. The form of this system represents a more general class of "triangular" systems, specifically lower triangular systems. In comparison to strict-feedback systems, System 3.73 lacks the *affine*[17] appearance of variables ξ_k and u. Krstic et al. [8, §2.3] explicitly shows the recursive design procedures in which a stabilizing control law $\alpha(x)$ from Lyapunov function V is generated for each intermediate virtual control law ξ; to be demonstrated in the derivation of the flight-path controller in chapter 4.

The control architecture will first be applied to a general second order system, so that the reader may develop a sense of procedure and terminology. An extension of this special case to a higher order system will follow, where a clear step-by-step procedure will be defined.

3.3.4.1 Second Order Systems

Typically in feedback control architectures the objective is to create a control law that cancels known dynamics and imposes variables that transform the system into a tracking problem. The key term here is *known*, implying that complete model information is available. Feedback linearization is one such case, where the exact knowledge of *non-linear* system dynamics is required; if one of the functions is uncertain then cancellation is not possible. The key idea here is *tracking*, with the goal of driving the error between an actual and desired

[17] "An affine function is just a linear function plus a translation." http://cfsv.synechism.org/c1/sec15.pdf

value to zero.

Backstepping synthesis efficiently handles these two critical objectives. Stabilizing non-linear terms in the dynamics, if recognized, may be retained hence less precise modeling information and less control effort is necessary. The inclusion of non-linearities improves transient performance; furthermore even the "worst case performance attained by a "non-linear controller coincides with the performance attained by the best linear design." Kokotovic [5, Linear Versus Nonlinear]

The term backstepping is derived from the recursive nature of the procedure. Specifically when a stabilizing feedback control term is proposed for a virtual control law, one subsequently "back-steps" this stabilizing control through an integrator – to be illustrated by block diagrams as shown by Khalil [1, §13.2] and Krstic et al. [8, §2.2].

To begin, we'll select a general second order system already augmented by an integrator for which backstepping[18] will be applied:

$$\begin{aligned} \dot{x} =& f(x) + g(x)\xi \\ \dot{\xi} =& u \end{aligned} \tag{3.74}$$

where $[x^T, \xi]^T \in \mathbb{R}^{n+1}$ is the state and $u \in \mathbb{R}$ is the control input. The function $f : \mathcal{D} \to \mathbb{R}^n$ and $g : \mathcal{D} \to \mathbb{R}^n$ are smooth in a domain $\mathcal{D} \subset \mathbb{R}^n$ that contains $x = 0$ and $f(0) = 0$. The aim is to design a state feedback control law u to force the variable ξ to stabilize the origin $(x = 0\,, \xi = 0)$

Assumption 3.5 (*Integrator* Backstepping)**.**

- Full State Feedback
- System in Lower-Triangular Form
- Lyapunov Function Known*
- System Dynamics Known*
- Smooth Feedback Control

Assumption 3.6 (Backstepping System)**.** Krstic et al. [8, Asmp. 2.7]

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})u \quad , \quad f(0) = 0$$

[18] Alternatively referred to as *integrator* backstepping.

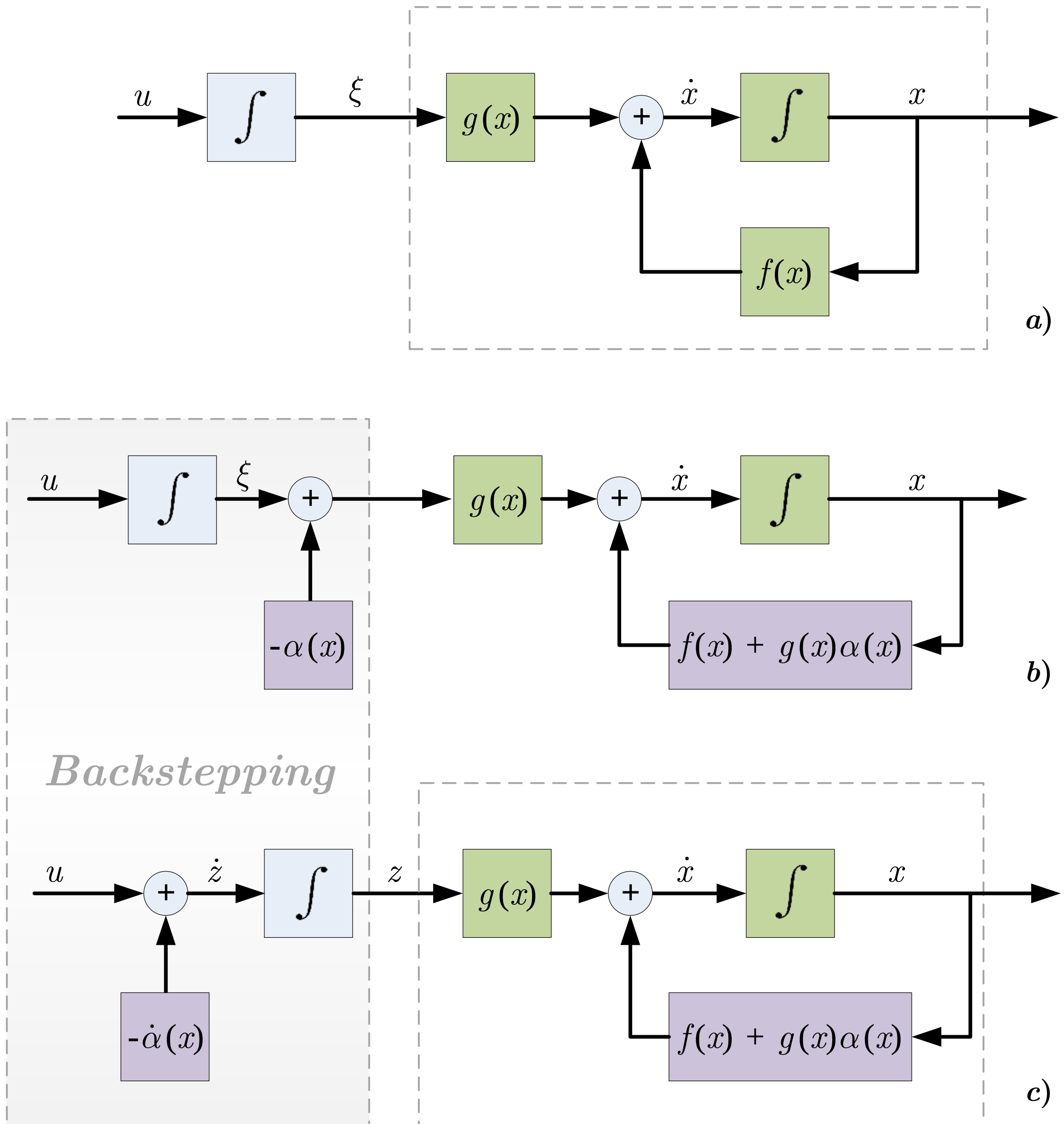


Figure 3.16: Backstepping Block Diagram Representation: *a*) Integral augmented system *b*) introducing $\pm\alpha(x)$ *c*) "backstepping" $-\alpha(x)$ through the integrator

where $\boldsymbol{x} \in \mathbb{R}^n$ is the state and $u \in \mathbb{R}$ is the control input. There exists a continuously differentiable feedback control law

$$u = \alpha(\boldsymbol{x}) \quad , \quad \alpha(0) = 0$$

and a smooth, positive definite, radially unbounded function $V : \mathbb{R}^n \to \mathbb{R}$ such that

$$\frac{\partial V}{\partial \boldsymbol{x}}(\boldsymbol{x}) \left[\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x}) \alpha(\boldsymbol{x}) \right] \leq -W(\boldsymbol{x}) \quad , \quad \forall \, \boldsymbol{x} \in \mathbb{R}^n$$

where $W : \mathbb{R}^n \to \mathbb{R}$ is positive semidefinite.

"The main result of backstepping is not the specific form of the control law, but rather the construction of a Lyapunov function whose derivative can be made negative definite by a wide variety of control laws." Krstic et al. [8].

3.3.4.2 Higher Order Systems

This should really be called something like, "intermediate example," but I want to highlight the difference between this eg. and the last: multiple loops where the commands are generated for loops after it (the heart of backstepping and exemplification of recursive nature)

3.3.4.3 Command Filtering

Avoids analytic computation of derivative, instead uses relatively simple low pass filter.

Insert figure showing generation of x_c and $\dot{x}_c$ by integrators only.

$$\dot{x}_1 = f_1(x_1) + g_1(x_1) x_2 \tag{3.75}$$

$$\dot{x}_2 = \tag{3.76}$$

1. Define

$$ \tag{3.77} $$

2. Define the compensated tracking error as

$$ \tag{3.78} $$

3. Define

$$ \tag{3.79} $$

$$\frac{\partial V_1}{\partial \tilde{x}} \left[f_1 + g_1 \alpha - \dot{x}_{1c} \right] = -W(\tilde{x}_1) \tag{3.80}$$

Lemma 3.3.1 (Command Filtered Backstepping)**.** *Farrell and Polycarpou [15, Lem. 5.3.2] Let the control law α_1 solve the tracking problem for the system*

$$\dot{x}_1 = f_1(x_1) + g_1(x_1)\alpha_1 \quad \textit{with} \quad x_1 \in \mathbb{R}^{n-1}$$

with Lyapunov function V_1 satisfying Equation 3.80. Then the controller of Equation 3.77 to 3.79 solves the tracking problem (ie., guarantees that $x_1(t)$ converges to $y_d(t)$) for the system described by Equation 3.76.

3.3.5 Command Filter

This filter was developed in Farrell et al. [17, Appendix A] and provides bounded and continuous command and command-derivative signals. There are two significance features of this filter: the first is the ability to generate derivatives of intermediate control signals, alleviating the need for analytic derivative calculation as mentioned in the preceding sections;

the second is magnitude, rate, and bandwidth limiting of state and actuator commands, ensuring that control signals generated are implementable. Some control allocation procedures make provisions for rate and magnitude limiting, so its up to the designer whether this filter should be used for actuator commands or not. The non-linear state space representation of this filter is as follows: Do I need q_1 and q_2, couldn't I just use x_c and $\dot{x}_c$?

$$\begin{bmatrix} x_c \\ \dot{x}_c \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \tag{3.81}$$

$$\begin{bmatrix} \dot{q}_1(t) \\ \dot{q}_2(t) \end{bmatrix} = \begin{bmatrix} q_2 \\ 2\zeta\omega_n \left(S_R \left\{ \frac{\omega_n^2}{2\zeta\omega_n} \left[S_M(x_c^o) - q_1 \right] \right\} - q_2 \right) \end{bmatrix} \tag{3.82}$$

where S_M and S_R are the magnitude and rate limit functions:

$$S_M = \begin{cases} M & \text{if} \quad x \geq M \\ x & \text{if} \quad \|x\| < M \\ -M & \text{if} \quad x \leq -M \end{cases} \qquad S_R = \begin{cases} R & \text{if} \quad x \geq R \\ x & \text{if} \quad \|x\| < R \\ -R & \text{if} \quad x \leq -R \end{cases} \tag{3.83}$$

Equation 3.82 may be represented in block diagram form as:

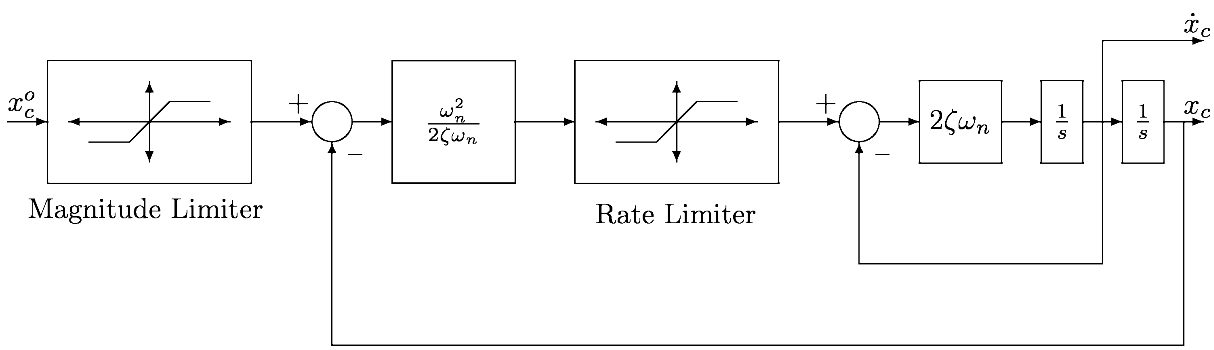


Figure 3.17: Command Filter

As shown in Figure 3.17, this filter generates command (x_c) and command derivative ($\dot{x}_c$) from an unfiltered command (x_c^o) while enforcing magnitude, rate, and bandwidth constraints.

3.3.6 Control Allocation

v = Bu

pinv(B)*v = u

Least squares minimization via pseudo inverse, simplest technique.

Crucial to the robustness of the design, and attempted to implement simplest form of control allocation: not the focus of my thesis, and all the papers I read didn't fully address it, with the exception of Harkegard's dissertation.

Chapter 4

Derivation of UAV Flight Path Controller

This derivation is based on work by Farrell et al. [16], [17], [15] and Sonneveldt et al. [3]. In the Farrell et al. papers online approximation based, command filtered, backstepping flight path control is developed alongside control laws for three feedback loops. The translational, attitude, and rotational subsets of the state-vector described in Table 3.1 and Equation 3.1 are employed in that order, ie. the state-vector for the flight path controller is:

$$\boldsymbol{x} = [\chi\ \gamma\ V\ \mu\ \alpha\ \beta\ P\ Q\ R]^T \tag{4.1}$$

The inputs to the controller are commanded heading χ_c, climb rate or glide-path angle γ_c, airspeed V_c, and the bounded first derivatives of these signals; the subscripts c here mean commanded. The outer loop controller generates roll angle μ_c and angle-of-attack α_c commands for the middle loop along with a thrust command T_c that is not passed to succeeding loops. The objective is coordinated flight, hence angle-of-sideslip command β_c is always set to zero. The middle loop generates roll rate P_c, pitch rate Q_c, and yaw rate R_c commands that serve as inputs to a control allocator which produces actuator deflection commands $\delta_1, \delta_2, \cdots, \delta_n$, with n being the number of available actuators.

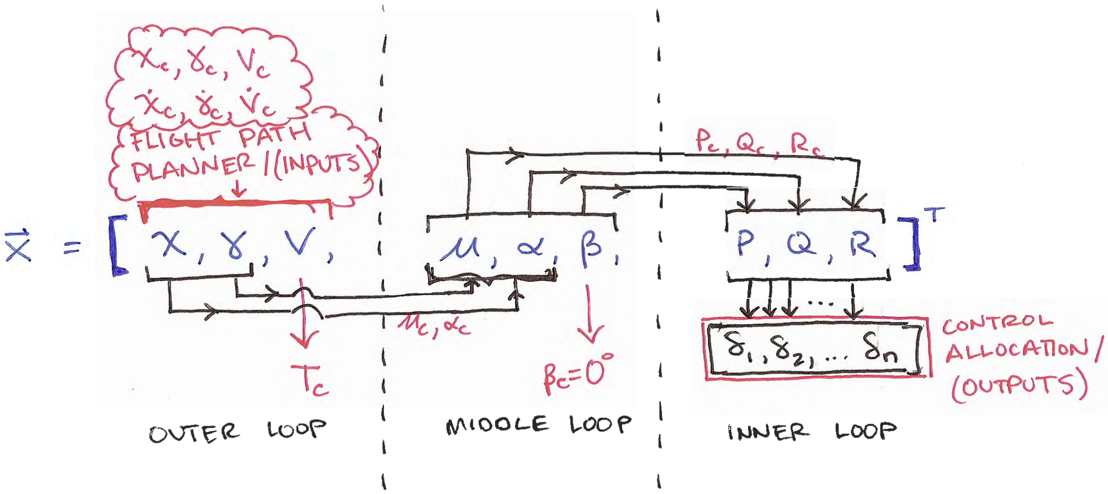


Figure 4.1: Loop Interactions: Flight-Path & Airspeed, Wind, and Body

Design of the flight-path controller herein compliments their work, with the exception of adaptive approximation for aerodynamic force and moment coefficients and dynamic control allocation. For the air vehicle this control architecture was closed around an exhaustive set of wind tunnel data was available and the aerodynamics model was validated through full-scale flight testing; a static control allocation method was also in place and utilized for this initial development. Even for a well characterized model, online approximation would still be advantageous as aerodynamic parameters always contain some degree of uncertainty. The motive of these choices was simplification, the goal being a baseline backstepping controller; these features are independent of the control law design procedure[1] and may be implemented in future phases of development.

Assumption 4.1 (*Command Filtered* Backstepping)**.**

- Full State Feedback
- System Dynamics Known*
- System in Lower-Triangular Form*?
- Smooth Feedback Control
- Lyapunov Function Known*
- Actuator Dynamics Neglected

A major contribution of this thesis is the comprehensible, simulink driven, graphical block diagram implementation of the control architecture. It offers an intuitive visualization of the

[1] However, adaptive parameter estimation is not independent of the stability analysis.

procedure and illustrates signal interdependencies that are not self-evident through pages and pages of equations; especially useful for higher order systems. A journal publication by Sonneveldt et al. [3, Figure 1] contains one of the clearest block diagram representation of a backstepping flight controller I've found.

Mine...

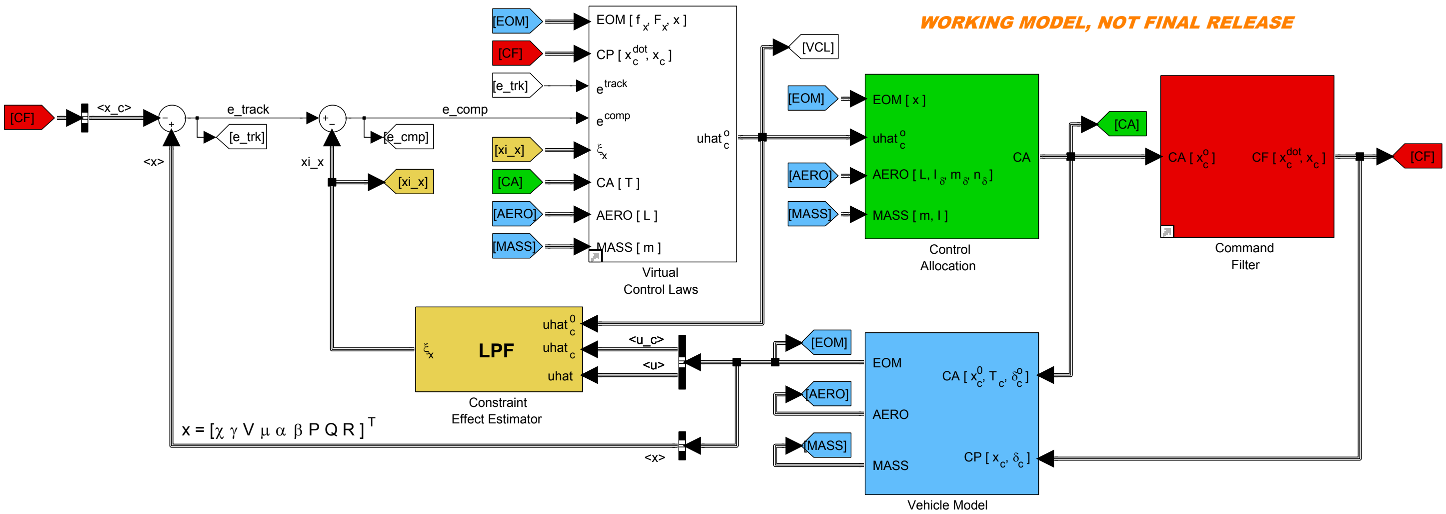


Figure 4.2: High Level Block Diagram

His...

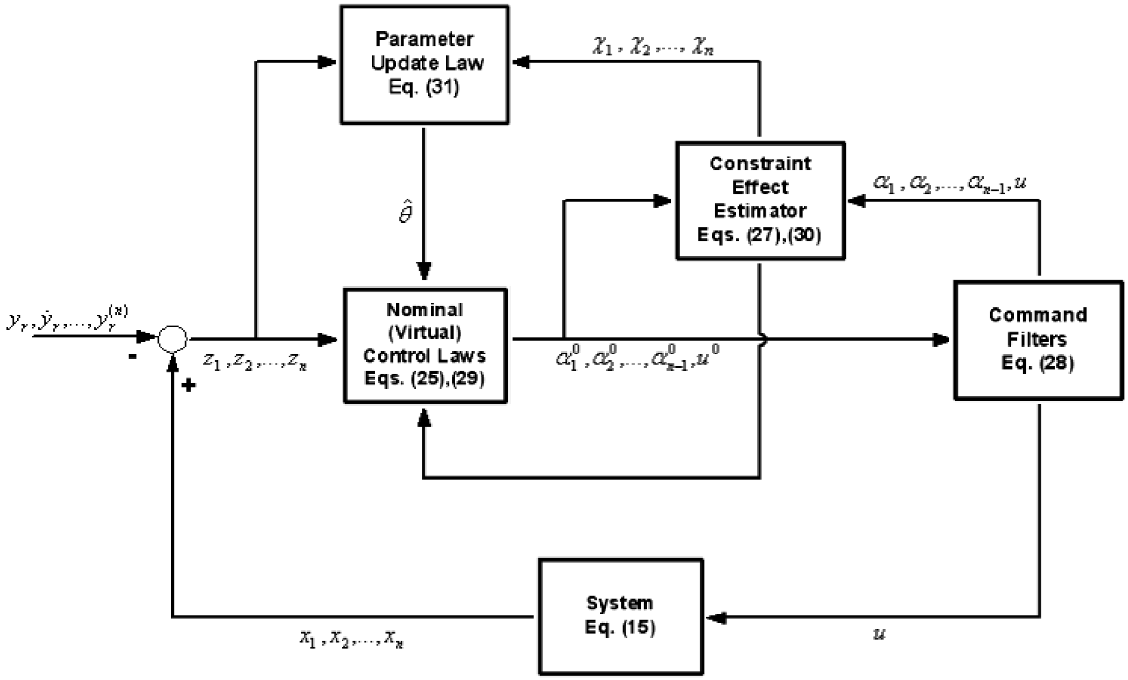


Figure 4.3: Command filtered adaptive backstepping control scheme [3]

As cited within Farrell et al. [17], "the main advantages of the approach include ... 2) Lyapunov stability results are provable and 3) state and control constraints can be enforced while maintaining Lyapunov stability.[2]" Robustness to large disturbances, extreme cases being airframe battle damage or actuator failure, could be ensured by implementing an adaptive online approximation and dynamic control allocation. If this were the case, then conceivably the control architecture could be applied to variants of the base vehicle configuration without redesign. This implies that the control designer would need less input from the aerodynamicist, hence less analysis or wind-tunnel testing resulting in a lower-fidelity aero model. Consequently this could increase cost savings and decrease development time, *if* the overall investment in adaptive backstepping development was cheaper than an exhaustive aerodynamic evaluation.

The following sections use a stability result initially developed in Farrell et al. [16],

[2] If online approximation were implemented then: "1) the aerodynamic force and moment models are automatically adjusted to accommodate changes to the aerodynamic properties of the vehicle"

formalized in a textbook by Farrell and Polycarpou [15, §5.3.3], and summarized here in § 3.3.4.3, Lemma 3.3.1. **The block vector representation of equations within Farrell et al. [17] are expanded with respect to each state to make the control law derivation easier to understand. Dynamics will be grouped into sub-functions consisting of known F_x and partially or completely unknown variables f_x, even though online approximation is not used, ie. it is assumed that there are no parametric uncertainties in the aerodynamic coefficients; this was done for the sake of scalability.**

4.1 Outer Loop

Wind-axis angles from the middle loop, μ and α, drive flight-path angles χ and γ, while airspeed, V, is managed via thrust, T. Remember that this is a feedback control architecture, do not confuse state commands with state dynamics: state commands, $\boldsymbol{x}_c$, are fed top-down in the controller as Figure 4.1 illustrates while state dynamics, $\boldsymbol{x}$, are fed bottom-up, or back.

To begin, flight-path and airspeed dynamics for the outer loop, Equations 3.42a, 3.42b, and 3.42c respectively, will be grouped into sub-functions of their partially or completely unknown f, known F, and control signals u:

$$\dot{\chi} = f_\chi + F_\chi + u_\chi \tag{4.2}$$

$$f_\chi = \frac{\cos\mu}{mV\cos\gamma}\left(D\sin\beta + Y\cos\beta\right) \tag{4.3}$$

$$F_\chi = \frac{1}{mV\cos\gamma}\left(-T\cos\alpha\sin\beta\cos\mu\right) \tag{4.4}$$

$$u_\chi = \frac{\sin\mu}{mV\cos\gamma}\left(L + T\sin\alpha\right) \tag{4.5}$$

$$\dot{\gamma} = f_\gamma + F_\gamma + u_\gamma \tag{4.6}$$

$$
\begin{aligned}
f_\gamma &= \frac{\sin\mu}{mV}\left(-D\sin\beta - Y\cos\beta\right) && (4.7)\\
F_\gamma &= \frac{1}{mV}\left(T\cos\alpha\sin\beta\sin\mu - mg\cos\gamma\right) && (4.8)\\
u_\gamma &= \frac{\cos\mu}{mV}\left(L + T\sin\alpha\right) && (4.9)
\end{aligned}
$$

$$\dot{V} = f_V + F_V + u_V \tag{4.10}$$

$$
\begin{aligned}
f_V &= \frac{1}{m}\left(-D\cos\beta + Y\sin\beta\right) && (4.11)\\
F_V &= -g\sin\gamma && (4.12)\\
u_V &= \frac{1}{m}\left(T\cos\alpha\cos\beta\right) && (4.13)
\end{aligned}
$$

In a modified block vector notation Equations 4.2, 4.6, and 4.10 may be simplified to:

$$\dot{x}_1 = f_1(x) + F_1(x) + u_1(x) \tag{4.14}$$

4.1.1 Flight Path Angle Control

Control laws for the flight-path angle controller are developed in this sub section. Tracking errors are defined as

$$\tilde{\chi} = \chi - \chi_c\,, \qquad \tilde{\gamma} = \gamma - \gamma_c\,, \qquad \tilde{V} = V - V_c \tag{4.15}$$

and additionally, compensated tracking errors defined as

$$\bar{\chi} = \tilde{\chi} - \xi_\chi\,, \qquad \bar{\gamma} = \tilde{\gamma} - \xi_\gamma\,, \qquad \bar{V} = \tilde{V} - \xi_V \tag{4.16}$$

which are tracking errors that *compensate* for a state or actuator signal that is limited by rate, magnitude, and/or bandwidth constraints; ξ terms will be defined shortly.

The goal is to compute unfiltered wind-axis angle commands, μ_c^o and α_c^o.

Virtual control laws are defined as

$$u^o_{\chi_c} \quad = \quad -f_\chi - F_\chi + \dot{\chi}_c - k_\chi \bar{\chi} \tag{4.17}$$

$$u^o_{\gamma_c} \quad = \quad -f_\gamma - F_\gamma + \dot{\gamma}_c - k_\gamma \bar{\gamma} \tag{4.18}$$

Control signals, used in control allocation, are equivalent to their counterparts in Equations 4.5 and 4.9:

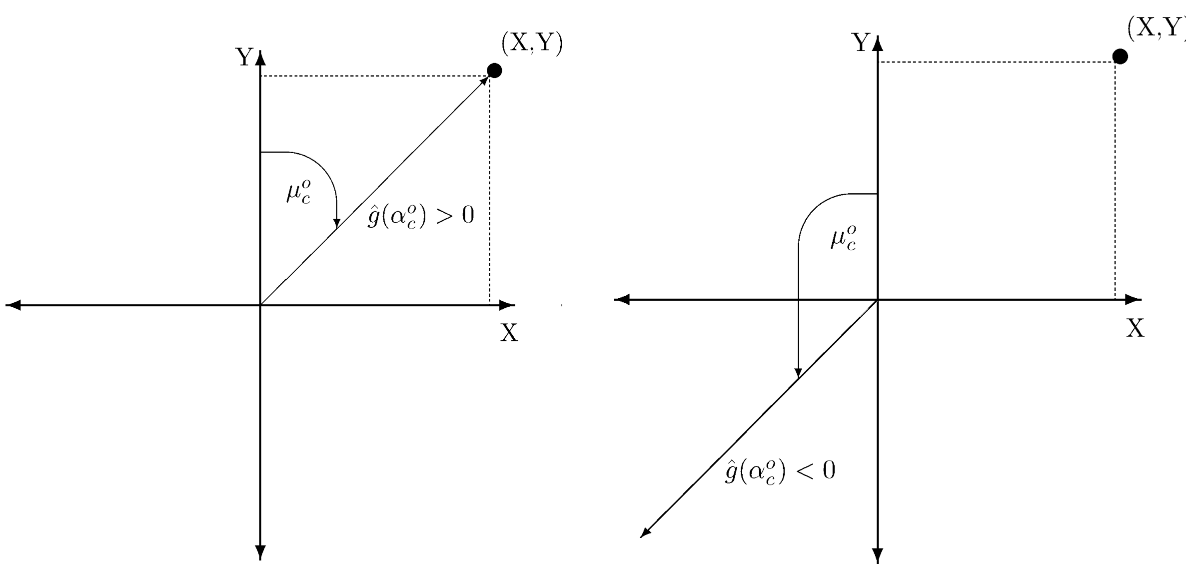
$$u^o_{\chi_c} \equiv \frac{g(\alpha^o_c)\sin\mu}{mV\cos\gamma} \tag{4.19}$$

$$u^o_{\gamma_c} \equiv \frac{g(\alpha^o_c)\cos\mu}{mV} \tag{4.20}$$

where $g(\alpha^o_c) = L(\alpha^o_c) + T\sin\alpha^o_c$. What's critical in the definition of g is realizing that lift is dependent on alpha, not just the thrust term; this interdependency was not transparent in Equations 4.5 and 4.9.

$$X \equiv mV\cos(\gamma)u_\chi = g(\alpha^o_c)\sin(\mu^o_c) \tag{4.21}$$

$$Y \equiv mVu_\gamma = g(\alpha^o_c)\cos(\mu^o_c) \tag{4.22}$$



(a) Positive lift case

(b) Negative lift case

Figure 4.4: Geometric Solution for μ^o_c and α^o_c

$$g(X,Y) = \sqrt{X^2 + Y^2} \tag{4.23}$$

$$\mu_c^o = \tan^{-1}\left(\frac{Y}{X}\right) \tag{4.24}$$

$$\alpha_c^o = \text{interp1}\left(g(\boldsymbol{\alpha}),\ \boldsymbol{\alpha},\ g(X,Y)\right) \tag{4.25}$$

Command Filter

4.1.2 Airspeed Control

4.2 Middle Loop

$$\dot{\mu} = f_\mu + F_\mu + u_\mu \tag{4.26}$$

with

$$\begin{aligned} f_\mu &= \frac{1}{mV}\left[D\sin\beta\tan\gamma\cos\mu + Y\cos\beta\tan\gamma\cos\mu + L\left(\tan\beta + \tan\gamma\sin\mu\right)\right] & (4.27) \\ F_\mu &= \frac{1}{mV}\left[T\left(\sin\alpha\tan\gamma\sin\mu + \sin\alpha\tan\beta - \cos\alpha\sin\beta\tan\gamma\cos\mu\right)\right. & (4.28) \\ &\quad \left. - \, mg\tan\beta\cos\gamma\cos\mu\right] & \\ u_\mu &= \frac{P\cos\alpha + R\sin\alpha}{\cos\beta} \equiv \frac{P_s}{\cos\beta} & (4.29) \end{aligned}$$

$$\dot{\alpha} = f_\alpha + F_\alpha + u_\alpha \tag{4.30}$$

with

$$\begin{aligned} f_\alpha &= -\frac{L}{mV\cos\beta} & (4.31) \\ F_\alpha &= \frac{1}{mV\cos\beta}\left[-T\sin\alpha + mg\cos\gamma\cos\mu\right] - \tan\beta\left(P\cos\alpha + R\sin\alpha\right) & (4.32) \\ u_\alpha &= Q & (4.33) \end{aligned}$$

$$\dot{\beta} = f_\beta + F_\beta + u_\beta \tag{4.34}$$

with

$$f_\beta = \frac{1}{mV}\left(D \sin\beta + Y \cos\beta\right) \tag{4.35}$$

$$F_\beta = \frac{1}{mV} - T \sin\beta \cos\alpha + mg \cos\gamma \sin\mu \tag{4.36}$$

$$u_\beta = P \sin\alpha - R \cos\alpha \equiv -R_s \tag{4.37}$$

4.2.1 Wind-Axis Angle Control

Tracking errors are defined as

$$\tilde{\mu} = \mu - \mu_c\,, \qquad \tilde{\alpha} = \alpha - \alpha\,, \qquad \tilde{\beta} = \beta - \beta_c \tag{4.38}$$

and compensated tracking errors defined as

$$\bar{\mu} = \tilde{\mu} - \xi_\mu\,, \qquad \bar{\alpha} = \tilde{\alpha} - \xi_\alpha\,, \qquad \bar{\beta} = \tilde{\beta} - \xi_\beta \tag{4.39}$$

The goal is to compute unfiltered wind-axis angle commands, P_c^o,Q_c^o, and R_c^o.

Virtual control laws are defined as

$$u^o_{\mu_c} = -f_\mu - F_\mu + \dot{\mu}_c - k_\mu \bar{\mu} \tag{4.40}$$

$$u^o_{\alpha_c} = -f_\alpha - F_\alpha + \dot{\alpha}_c - k_\alpha \bar{\alpha} \tag{4.41}$$

$$u^o_{\beta_c} = -f_\beta - F_\beta + \dot{\beta}_c - k_\beta \bar{\beta} \tag{4.42}$$

Control signals, used in control allocation, are equivalent to their counterparts in Equations 4.29, 4.33, and 4.37:

$$u^o_{\mu_c} \equiv \tag{4.43}$$

$$u^o_{\alpha_c} \equiv \tag{4.44}$$

$$u^o_{\beta_c} \equiv \tag{4.45}$$

4.3 Inner Loop

$$\dot{P} = f_P + F_P + u_P \tag{4.46}$$

with

$$f_P = c_3\bar{L}^{'} + c_4\bar{N}^{'} \tag{4.47}$$

$$F_P = (c_1 R + c_2 P)\, Q \tag{4.48}$$

$$u_P = c_3 \sum_{i=1}^{n} \bar{L}_{\delta_i}\delta_i + c4 \sum_{i=1}^{n} \bar{N}_{\delta_i}\delta_i \tag{4.49}$$

where the rolling and yawing moments were decomposed into moments due to aerodynamics and deflections as $\bar{L} = \bar{L}^{'} + \sum_{i=1}^{n} \bar{L}_{\delta_i}\delta_i$ and $\bar{N} = \bar{N}^{'} + \sum_{i=1}^{n} \bar{N}_{\delta_i}\delta_i$.

$$\dot{Q} = f_Q + F_Q + u_Q \tag{4.50}$$

with

$$f_Q = c_7\bar{M}^{'} \tag{4.51}$$

$$F_Q = c_5 PR - c_6 \left(P^2 - R^2\right) \tag{4.52}$$

$$u_Q = c_7 \sum_{i=1}^{n} \bar{M}_{\delta_i}\delta_i \tag{4.53}$$

where the pitching moment was decomposed into moment due to aerodynamics and deflections as $\bar{M} = \bar{M}^{'} + \sum_{i=1}^{n} \bar{M}_{\delta_i}\delta_i$

$$\dot{R} = f_R + F_R + u_R \tag{4.54}$$

with

$$f_R = c_4\bar{L}^{'} + c_9\bar{N}^{'} \tag{4.55}$$

$$F_R = (c_8 P - c_2 R)\, Q \tag{4.56}$$

$$u_R = c_4 \sum_{i=1}^{n} \bar{L} + c_9 \sum_{i=1}^{n} \bar{N} \tag{4.57}$$

where the rolling and yawing moments were decomposed as in Equation 4.47.

4.3.1 Body-Axis Angular Rate Control

4.4 Stability Proof

Chapter 5

Application

VALIDATION STEPS!

Will eventually test in hardware, at minimum hardware in loop at AME SIL. In the meanwhile and for the thesis I will be running a purely computer simulation.

5.1 Modeling

Describe simulink environment and any models referenced from AME (i.e the aerodynamics model for Fury based on windtunnel data / CFD analysis padded by Wurts)

Introduce 3 view of air vehicle

5.2 Simulation

List simulation stipulations: ie. back up assumptions (constant mass, deflections assumed to be measured, ideal actuators, constant density therefore altitude, what have you)

Possibly discuss simulation configuration –¿ ODE solver, considerations or lessons learned for those who follow. ie. dave suggested typecasting vars for future encapsulation into hardward (FCAS), library block usage, unit testing, initialization w/ memory block (avoid algebraic loops), how to send items to command workspae for plotting nicely, ...

5.2.1 Nominal

Plot time histories between command and response [$\gamma\mu V$]

5.2.2 Turbulence

Throw in atmospheric model to test turbulence? Currently model has constant rho... hmmm

5.2.3 Modeling Error / Unfamiliar Plant

Won't work if I don't have aero function approximations

5.2.4 Effector Failure

Won't work unless control allocation robust

5.3 Results

1. Here is a list item.
    (a) Here is a sub list item.
        i. Here is a sub sub list item.

Chapter 6

Conclusion

Could've covered...

1. adaptive aero coefficient modeling
2. linearization around operating points to evaluate PM/GM in classic controls sense
3. New(er) derivation for backstepping that uses tikanov's theorem to guaranteed GES.
4. hardware in the loop testing

GES stability vis Tikhonov's Theorem: Farrell et al. [18], Farrell et al. [19]

Bibliography

[1] Hassan K. Khalil. *Nonlinear Systems*. Prentice-Hall, Upper Saddle River, New Jersey, 2nd edition, 1996. ISBN 0132280248. URL http://www.worldcat.org/oclc/757428832.

[2] Jean-Jacques E Slotine and Li Weiping. *Applied Nonlinear Control*. Prentice-Hall, Englewood Cliffs, NJ, 1st edition, 1991. ISBN 0130408905. URL http://www.worldcat.org/oclc/636122876.

[3] L. Sonneveldt, Q. P. Chu, and J. A. Mulder. Nonlinear Flight Control Design Using Constrained Adaptive Backstepping. *Journal of Guidance, Control, and Dynamics*, 30 (2):322–336, March 2007. ISSN 0731-5090. doi: 10.2514/1.25834. URL http://doi.aiaa.org/10.2514/1.25834.

[4] Randy A. Freeman and Petar V. Kokotovic. *Robust Nonlinear Control Design*. Birkhäuser Boston, Boston, MA, 2008 edition, 1996. ISBN 978-0-8176-4758-2. doi: 10.1007/978-0-8176-4759-9. URL http://www.springerlink.com/index/10.1007/978-0-8176-4759-9.

[5] Petar V. Kokotovic. The joy of feedback: nonlinear and adaptive. *IEEE Control Systems Magazine*, 12(3):7–17, June 1992. ISSN 02721708. doi: 10.1109/37.165507. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=165507.

[6] Ioannis Kanellakopoulos, Petar V. Kokotovic, and A.S. Morse. Systematic design of

adaptive controllers for feedback linearizable systems. *IEEE Transactions on Automatic Control*, 36(11):1241–1253, 1991. ISSN 00189286. doi: 10.1109/9.100933. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=100933.

[7] Ioannis Kanellakopoulos. A toolkit for nonlinear feedback design. *Systems & Control Letters*, 18(2):83–92, February 1992. ISSN 01676911. doi: 10.1016/0167-6911(92)90012-H. URL http://linkinghub.elsevier.com/retrieve/pii/016769119290012H.

[8] Miroslav Krstic, Ioannis Kanellakopoulos, and Petar V. Kokotovic. *Nonlinear and adaptive control design*. John Wiley & Sons, Inc., New York, 1995. ISBN 0471127329. URL http://www.worldcat.org/title/nonlinear-and-adaptive-control-design/oclc/32166728?referer=di&ht=edition.

[9] Honeywell Inc. and Lockheed Martin Skunk Works. Application of Multivariable Control Theory to Aircraft Control Laws. Final Report: Multivariable Control Design Guideline. Technical Report March 1993, Wright Laboratory, Palmdale, CA, 1996. URL http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA315259.

[10] Courtland D. Perkins and Robert E. Hage. *Airplane Performance, Stability, and Control*. Wiley, New York, 1949. ISBN 047168046X. URL http://www.worldcat.org/oclc/1650766.

[11] Duane T. McRuer, Irving Ashkenas, and Dunstan Graham. *Aircraft Dynamics and Automatic Control*. Princeton University Press, New Jersey, 1973. ISBN 0691080836. URL http://www.worldcat.org/oclc/900904.

[12] Brian Stevens and Frank Lewis. *Aircraft Control and Simulation*. Wiley, Hoboken, NJ, 1st edition, 1992. ISBN 0471613975. URL http://www.worldcat.org/oclc/23356757.

[13] Brian Stevens and Frank Lewis. *Aircraft Control and Simulation*. Wiley, Hoboken, NJ, 2nd edition, 2003. ISBN 978-0-471-37145-8. URL http://www.worldcat.org/oclc/51751879.

[14] Ola Härkegå rd. *Backstepping and Control Allocation with Applications to Flight Control*. Dissertation, Linköping University, 2003. URL http://www.control.isy.liu.se/research/reports/Ph.D.Thesis/PhD820.pdf.

[15] Jay A. Farrell and Marios M. Polycarpou. *Adaptive Approximation Based Control*. Wiley-Interscience, Hoboken, NJ, 1st edition, 2006. ISBN 0471727881. URL http://www.worldcat.org/oclc/61169767.

[16] Jay A. Farrell, Marios M. Polycarpou, and Manu Sharma. On-Line Approximation Based Control of Uncertain Nonlinear Systems with Magnitude , Rate and Bandwidth Constraints on the States and Actuators. In *Proc. of the American Control Conference*, pages 2557–2562, Boston, MA, 2004. IEEE. doi: 0.1109/ACC.2004.182491.

[17] Jay A. Farrell, Marios M. Polycarpou, and Manu Sharma. Backstepping-Based Flight Control with Adaptive Function Approximation. *Journal of Guidance, Control, and Dynamics*, 28(6):1089–1102, November 2005. ISSN 0731-5090. doi: 10.2514/1.13030. URL http://doi.aiaa.org/10.2514/1.13030.

[18] Jay A. Farrell, Marios M. Polycarpou, Manu Sharma, and Wenjie Dong. Command Filtered Backstepping. *2008 American Control Conference*, pages 1923–1928, June 2008. doi: 10.1109/ACC.2008.4586773. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4586773.

[19] Jay A. Farrell, Marios M. Polycarpou, and Manu Sharma. Command Filtered Backstepping. *IEEE Transactions on Automatic Control*, 54(6):1391–1395, June 2009. ISSN 0018-9286. doi: 10.1109/TAC.2009.2015562. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4982643.