Lecture 1: Dynamical Systems, Equilibrium, and Stability

Textbook Sections 1.1, 1.2, & 1.3

Dr. Jordan D. Larson

Model-Based Design

- **Model-based design**: common design environment facilitating communication, data analysis, design verification and validation (V&V) between various development groups for entire system

Model-Based Design

- **Model-based design**: common design environment facilitating communication, data analysis, design verification and validation (V&V) between various development groups for entire system
- Modern computers allow use of complex structures and extensive software code, engineers typically use model-based design with advanced simulation tools to provide rapid prototyping of new systems, including aerospace systems
    - Engineers can locate and correct errors early in system design, when impact of design modifications minimized
    - Design reuse for upgraded or expanded systems made easier

Model-Based Design

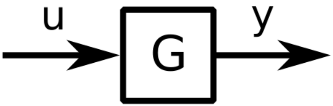
- **Model-based design**: common design environment facilitating communication, data analysis, design verification and validation (V&V) between various development groups for entire system
- Modern computers allow use of complex structures and extensive software code, engineers typically use model-based design with advanced simulation tools to provide rapid prototyping of new systems, including aerospace systems
  - Engineers can locate and correct errors early in system design, when impact of design modifications minimized
  - Design reuse for upgraded or expanded systems made easier
- Model-based design uses coverall approach to standard embedded systems development $\rightarrow$ time to port between modeling software and embedded systems can outweigh temporal value for alternative lab-based design

Model-Based Design Software

- Uses graphical modeling tools: generic and unified graphical modeling environment
  - Reduces complexity of model designs by breaking into hierarchies of individual design blocks
  - Helps design engineers to conceptualize entire system, typically with **graphical user interface (GUI)**
  - Connection between signals and systems in these GUIs typically depicted using **block diagrams**, for which each “block” represents system and each “arrow” represents signal

Model-Based Design Software

- Uses graphical modeling tools: generic and unified graphical modeling environment
  - Reduces complexity of model designs by breaking into hierarchies of individual design blocks
  - Helps design engineers to conceptualize entire system, typically with **graphical user interface (GUI)**
  - Connection between signals and systems in these GUIs typically depicted using **block diagrams**, for which each “block” represents system and each “arrow” represents signal
- Example of basic block diagram:
  - System: G
  - Input signal: u
  - Output signal: y

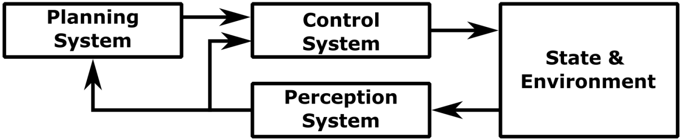


Planning, Control, and Perception

- In modern aerospace applications, design of **planning, control, and perception** systems, fundamental to operations of aerospace vehicles, a.k.a. flight vehicles

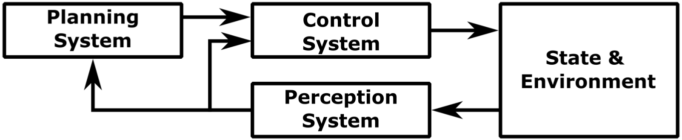
Planning, Control, and Perception

- In modern aerospace applications, design of **planning, control, and perception** systems, fundamental to operations of aerospace vehicles, a.k.a. flight vehicles
- Three systems understood as connected together through:



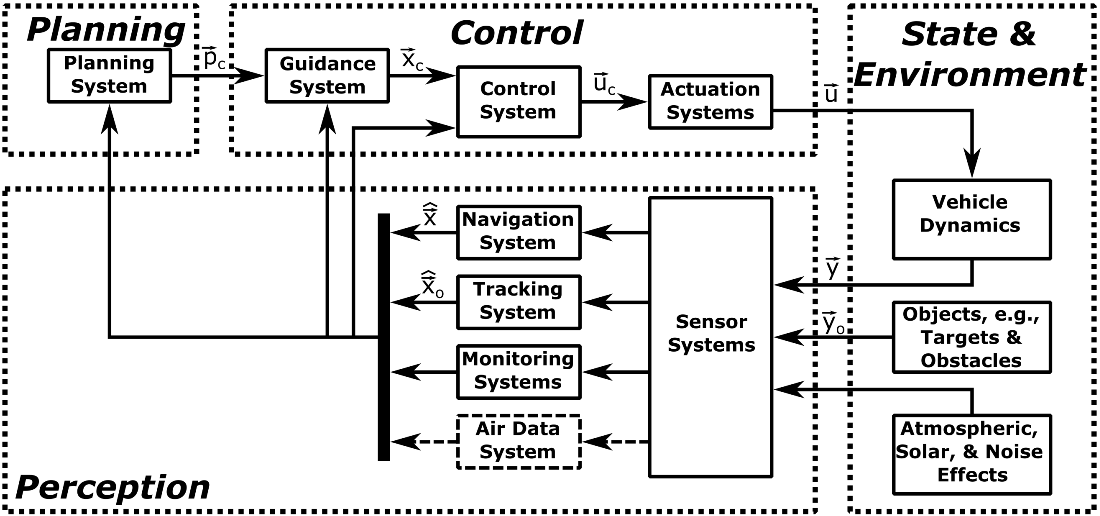
Planning, Control, and Perception

- In modern aerospace applications, design of **planning, control, and perception** systems, fundamental to operations of aerospace vehicles, a.k.a. flight vehicles
- Three systems understood as connected together through:



- Planning system: determines what mission/path/trajectory, $\vec{p}_c$, aerospace vehicle should fly
- Control system: executes path/trajectory for aerospace vehicle by actuating control inputs, $\vec{u}$
- Perception system(s): senses and estimates state of aerospace vehicle, $\hat{\vec{x}}$, state of object(s) in environment, $\hat{\vec{x}}_o$, and state of environment
  - For aerospace vehicles, perception systems provide feedback to planning system & control system

Aerospace Planning, Control, and Perception System



History of Control & Perception

- 1920s: control and perception systems first implemented in electric powerplant industries
  - Large process facilities used process controllers for regulating continuous variables, e.g. temperature, pressure, and flow rate, with electrical relays

History of Control & Perception

- 1920s: control and perception systems first implemented in electric powerplant industries
  - Large process facilities used process controllers for regulating continuous variables, e.g. temperature, pressure, and flow rate, with electrical relays
- 1930s and 1940s: control and perception systems began to see use in automotive and aviation industries

History of Control & Perception

- 1920s: control and perception systems first implemented in electric powerplant industries
  - Large process facilities used process controllers for regulating continuous variables, e.g. temperature, pressure, and flow rate, with electrical relays
- 1930s and 1940s: control and perception systems began to see use in automotive and aviation industries
- 1950s and 1960s: space race generated interest in embedded control and perception systems
  - Engineers constructed control and perception systems as part of end product, e.g. engine control units and flight simulators

History of Control & Perception

- 1920s: control and perception systems first implemented in electric powerplant industries
  - Large process facilities used process controllers for regulating continuous variables, e.g. temperature, pressure, and flow rate, with electrical relays
- 1930s and 1940s: control and perception systems began to see use in automotive and aviation industries
- 1950s and 1960s: space race generated interest in embedded control and perception systems
  - Engineers constructed control and perception systems as part of end product, e.g. engine control units and flight simulators
- 1970s, computer-based control and perception systems introduced and became standard
  - Drastic shift in control and perception system design
  - Led to modern use of model-based design for control and perception systems

4 Steps of Model-Based Design

1. **Plant modeling**: consists of identifying system model to be controlled, i.e. **plant** which comes from “powerplant” in early days of control and perception systems
    - Plant modeling can be based on first principles or machine learning
    - First principles modeling implements physics-derived mathematical model for plant dynamics, e.g. Newton-Euler EOMs
    - Machine learning modeling processes raw data from real-world system and uses learning algorithms to identify data-driven model for plant dynamics

4 Steps of Model-Based Design

1. **Plant modeling**: consists of identifying system model to be controlled, i.e. **plant** which comes from “powerplant” in early days of control and perception systems
   - Plant modeling can be based on first principles or machine learning
   - First principles modeling implements physics-derived mathematical model for plant dynamics, e.g. Newton-Euler EOMs
   - Machine learning modeling processes raw data from real-world system and uses learning algorithms to identify data-driven model for plant dynamics
2. **Control and perception design**: consists of **control and perception analysis** and **control and perception synthesis**
   - Mathematical plant model developed in first step used to identify suitable **design requirements** for controlled plant
   - Based on these identified requirements, control and perception strategies chosen
   - Control system and perception system synthesized: explicit **control law** and **observer law**, i.e. two mathematical algorithms for computing plant inputs automatically based on external commands and sensor data

4 Steps of Model-Based Design (continued)

3. **System simulation**: performed as multiple simulations with lower/higher fidelity plant models as well as offline/real-time simulations
   - Allow specification, requirements, and modeling errors to be found immediately, rather than later in overall system design effort
   - Real-time simulation: test control system, perception system, and plant on real-time modeling computer, a.k.a. **software-in-the-loop (SIL)** simulation
   - Control and perception systems implemented on embedded system hardware and simulated with real-time plant and sensor models, a.k.a. **hardware-in-the-loop (HIL)** simulation

4 Steps of Model-Based Design (continued)

3. **System simulation**: performed as multiple simulations with lower/higher fidelity plant models as well as offline/real-time simulations
   - Allow specification, requirements, and modeling errors to be found immediately, rather than later in overall system design effort
   - Real-time simulation: test control system, perception system, and plant on real-time modeling computer, a.k.a. **software-in-the-loop (SIL)** simulation
   - Control and perception systems implemented on embedded system hardware and simulated with real-time plant and sensor models, a.k.a. **hardware-in-the-loop (HIL)** simulation
4. **Validation and verification (V&V)** of control and perception system
   - Control and perception systems implemented on embedded system hardware and tested with actual plant, e.g. aerospace vehicle, and sensors
   - Control and perception systems highly unlikely to work exactly same on actual plant and sensors as in simulation $\rightarrow$ iterative debugging process by analyzing test results and updating control and perception system designs untildesign requirements met in actuality

Systems and Signals

- Flight dynamics and control (FDC) formalized through signals and systems theory

Systems and Signals

- Flight dynamics and control (FDC) formalized through signals and systems theory
- **Signal**: mathematical description of how parameter varies with *time*
  - Can be continuous or discrete in time and continuous or discrete in values parameter may take

Systems and Signals

- Flight dynamics and control (FDC) formalized through signals and systems theory
- **Signal**: mathematical description of how parameter varies with *time*
  - Can be continuous or discrete in time and continuous or discrete in values parameter may take
  - **Analog signals**: continuous-time and continuously valued
  - **Digital signals**: discrete-time and discretely valued
  - **Discrete-time signals**: discrete-time and continuously valued
  - Signals continuous-time and discretely valued rarely occur, do not have particular name

Systems and Signals

- Flight dynamics and control (FDC) formalized through signals and systems theory
- **Signal**: mathematical description of how parameter varies with *time*
  - Can be continuous or discrete in time and continuous or discrete in values parameter may take
  - **Analog signals**: continuous-time and continuously valued
  - **Digital signals**: discrete-time and discretely valued
  - **Discrete-time signals**: discrete-time and continuously valued
  - Signals continuous-time and discretely valued rarely occur, do not have particular name
- **System**: process that produces output signals in response to input signals
  - Output signal from system, a.k.a. **system response**
  - System characterized by its different properties

Dynamical Systems

- Fundamental characterization: how output signal/system response depends on time
  - **Static system**: not time-dependent output signal

Dynamical Systems

- Fundamental characterization: how output signal/system response depends on time
  - **Static system**: not time-dependent output signal
  - **Dynamic system**: time-dependent output signal that evolves according to fixed mathematical rule, a.k.a. **dynamics equation**
    - Term “dynamical” corresponds to mathematical abstraction modeling some real-world dynamic process
    - For many physical dynamical systems, dynamics equation a.k.a. **equation of motion (EOM)**, e.g. FDC

Dynamical Systems

- Fundamental characterization: how output signal/system response depends on time
  - **Static system**: not time-dependent output signal
  - **Dynamic system**: time-dependent output signal that evolves according to fixed mathematical rule, a.k.a. **dynamics equation**
    - Term “dynamical” corresponds to mathematical abstraction modeling some real-world dynamic process
    - For many physical dynamical systems, dynamics equation a.k.a. **equation of motion (EOM)**, e.g. FDC

- Dynamical systems theory encompasses three broad topics:
  - Simulation
  - Control
  - **System identification (SID)**
  - Course: advanced simulation and control for aerospace vehicles

Dynamical System Characterization

- Characterize dynamical systems: signal types
  - **Analog systems** have only analog signals
  - **Digital systems** have only digital signals
  - Importantly, any electronic systems truly contains both
  - Analyzed separately along with analog-to-digital and/or digital-to-analog converters

Dynamical System Characterization

- Characterize dynamical systems: signal types
  - **Analog systems** have only analog signals
  - **Digital systems** have only digital signals
  - Importantly, any electronic systems truly contains both
  - Analyzed separately along with analog-to-digital and/or digital-to-analog converters
- Characterize dynamical systems: form of dynamics equation
  - **Linear system**: satisfies principle of superposition
    - Adding & scaling multiple input signals produces overall output equivalent to added & scaled outputs of each input
  - **Nonlinear system**: does not satisfy principle of superposition

Dynamical System Characterization

- Characterize dynamical systems: signal types
  - **Analog systems** have only analog signals
  - **Digital systems** have only digital signals
  - Importantly, any electronic systems truly contains both
  - Analyzed separately along with analog-to-digital and/or digital-to-analog converters
- Characterize dynamical systems: form of dynamics equation
  - **Linear system**: satisfies principle of superposition
    - Adding & scaling multiple input signals produces overall output equivalent to added & scaled outputs of each input
  - **Nonlinear system**: does not satisfy principle of superposition
  - **Time-invariant system**: mathematical rule that does not depend *explicitly* on time
  - **Time-varying system**: mathematical rule that depends *explicitly* on time

Dynamical System Characterization

- Characterize dynamical systems: signal types
  - **Analog systems** have only analog signals
  - **Digital systems** have only digital signals
  - Importantly, any electronic systems truly contains both
  - Analyzed separately along with analog-to-digital and/or digital-to-analog converters
- Characterize dynamical systems: form of dynamics equation
  - **Linear system**: satisfies principle of superposition
    - Adding & scaling multiple input signals produces overall output equivalent to added & scaled outputs of each input
  - **Nonlinear system**: does not satisfy principle of superposition
  - **Time-invariant system**: mathematical rule that does not depend *explicitly* on time
  - **Time-varying system**: mathematical rule that depends *explicitly* on time
  - **Deterministic system**: always produces same output signal for given input signal
  - **Stochastic system**: never produces same output signal for given input signal

Dynamical Systems Theory Scope

- Characterize system: number of input and output signals
  - **Single Input, Single Output (SISO) system**
  - **Single Input, Multiple Outputs (SIMO) system**
  - **Multiple Inputs, Single Output (MISO) system**
  - **Multiple Inputs, Multiple Outputs (MIMO) system**

Dynamical Systems Theory Scope

- Characterize system: number of input and output signals
  - **Single Input, Single Output (SISO) system**
  - **Single Input, Multiple Outputs (SIMO) system**
  - **Multiple Inputs, Single Output (MISO) system**
  - **Multiple Inputs, Multiple Outputs (MIMO) system**
- Focus: continuous-time, linear time-invariant (LTI), deterministic MIMO dynamical systems

Dynamical Systems Theory Scope

- Characterize system: number of input and output signals
  - **Single Input, Single Output (SISO) system**
  - **Single Input, Multiple Outputs (SIMO) system**
  - **Multiple Inputs, Single Output (MISO) system**
  - **Multiple Inputs, Multiple Outputs (MIMO) system**
- Focus: continuous-time, linear time-invariant (LTI), deterministic MIMO dynamical systems
- Unfortunately, no dynamical system perfectly linear, time-invariant, or deterministic
  1. Linearity implies that operation of system can be scaled to arbitrarily large magnitudes, not physically possible
  2. Time-invariance violated by aging effects that change outputs of systems over time
  3. Thermal noise/random phenomena cause some degree of stochastic behavior

Dynamical Systems Theory Scope

- Characterize system: number of input and output signals
  - **Single Input, Single Output (SISO) system**
  - **Single Input, Multiple Outputs (SIMO) system**
  - **Multiple Inputs, Single Output (MISO) system**
  - **Multiple Inputs, Multiple Outputs (MIMO) system**
- Focus: continuous-time, linear time-invariant (LTI), deterministic MIMO dynamical systems
- Unfortunately, no dynamical system perfectly linear, time-invariant, or deterministic
  1. Linearity implies that operation of system can be scaled to arbitrarily large magnitudes, not physically possible
  2. Time-invariance violated by aging effects that change outputs of systems over time
  3. Thermal noise/random phenomena cause some degree of stochastic behavior
- Assumptions simplify mathematical theory of dynamical systems & provide valuable insight into FDC

State-Space Representation

- Continuous-time MIMO dynamical systems: **continuous-time state-space model**

$$\begin{aligned} \dot{\vec{x}}(t) &= f(t, \vec{x}, \vec{u}) \\ \vec{y}(t) &= h(t, \vec{x}, \vec{u}) \end{aligned} \tag{1}$$

  - $\vec{u}(t) \in \mathbb{R}^{n_u}$: **input vector** of n_u input signals
  - $\vec{y}(t) \in \mathbb{R}^{n_y}$: **output vector**of n_y output signals
  - $\vec{x}(t) \in \mathbb{R}^{n_x}$: **state vector** of n_x^{th}-order dynamical system

State-Space Representation

- Continuous-time MIMO dynamical systems: **continuous-time state-space model**

$$\begin{aligned} \dot{\vec{x}}(t) &= f(t, \vec{x}, \vec{u}) \\ \vec{y}(t) &= h(t, \vec{x}, \vec{u}) \end{aligned} \tag{1}$$

  - $\vec{u}(t) \in \mathbb{R}^{n_u}$: **input vector** of n_u input signals
  - $\vec{y}(t) \in \mathbb{R}^{n_y}$: **output vector**of n_y output signals
  - $\vec{x}(t) \in \mathbb{R}^{n_x}$: **state vector** of n_x^{th}-order dynamical system
- First vector-valued differential equation: **dynamics equation** for state-space systems, a.k.a. **state equation**
- Second vector-valued algebraic equation: **output equation**
  - Relates state and input vectors to output vector

State-Space Representation

- Continuous-time MIMO dynamical systems: **continuous-time state-space model**

$$\begin{aligned} \dot{\vec{x}}(t) &= f(t, \vec{x}, \vec{u}) \\ \vec{y}(t) &= h(t, \vec{x}, \vec{u}) \end{aligned} \tag{1}$$

  - $\vec{u}(t) \in \mathbb{R}^{n_u}$: **input vector** of n_u input signals
  - $\vec{y}(t) \in \mathbb{R}^{n_y}$: **output vector**of n_y output signals
  - $\vec{x}(t) \in \mathbb{R}^{n_x}$: **state vector** of n_x^{th}-order dynamical system
- First vector-valued differential equation: **dynamics equation** for state-space systems, a.k.a. **state equation**
- Second vector-valued algebraic equation: **output equation**
  - Relates state and input vectors to output vector
- System state $\vec{x}$ dynamically *controlled* by input $\vec{u}$ and *observed* through output $\vec{y}$

State-Space by Component

- Vector-valued state-space system rewritten as n_x first order ODEs for state equation and n_y algebraic equations for output equation:

$$
\begin{aligned}
\dot{x}_1 &= f_1(t, x_1, \cdots, x_{n_x}, u_1, \cdots, u_{n_u}) \\
\vdots &= \vdots \\
\dot{x}_{n_x} &= f_{n_x}(t, x_1, \cdots, x_{n_x}, u_1, \cdots, u_{n_u}) \\
y_1 &= h_1(t, x_1, \cdots, x_{n_x}, u_1, \cdots, u_{n_u}) \\
\vdots &= \vdots \\
y_{n_y} &= h_{n_y}(t, x_1, \cdots, x_{n_x}, u_1, \cdots, u_{n_u})
\end{aligned}
\tag{2}
$$

  - x_i denotes i^{th} element of $\vec{x}$
  - u_j denotes j^{th} element of $\vec{u}$
  - y_k denotes k^{th} element of $\vec{y}$

Linear, Time-Invariant Systems

- Study of general continuous-time systems: advanced topic

Linear, Time-Invariant Systems

- Study of general continuous-time systems: advanced topic
- Course focus: analysis of MIMO time-invariant (LTI) systems with uncertainty
  - Use equilibrium and linearization analysis

Linear, Time-Invariant Systems

- Study of general continuous-time systems: advanced topic
- Course focus: analysis of MIMO time-invariant (LTI) systems with uncertainty
  - Use equilibrium and linearization analysis
- Linear systems satisfy **principle of superposition** described by two properties: scaling and additivity

Linear, Time-Invariant Systems

- Study of general continuous-time systems: advanced topic
- Course focus: analysis of MIMO time-invariant (LTI) systems with uncertainty
  - Use equilibrium and linearization analysis
- Linear systems satisfy **principle of superposition** described by two properties: scaling and additivity
  - Let output $y_1(t)$ be solution to linear ODE with input $u_1(t)$ and zero initial conditions
  - Scaling property: for any real number c, solution of linear ODE with input $u_s(t) = cu_1(t)$ and zero initial conditions is $y_s(t) = cy_1(t)$

Linear, Time-Invariant Systems

- Study of general continuous-time systems: advanced topic
- Course focus: analysis of MIMO time-invariant (LTI) systems with uncertainty
  - Use equilibrium and linearization analysis
- Linear systems satisfy **principle of superposition** described by two properties: scaling and additivity
  - Let output $y_1(t)$ be solution to linear ODE with input $u_1(t)$ and zero initial conditions
  - Scaling property: for any real number c, solution of linear ODE with input $u_s(t) = cu_1(t)$ and zero initial conditions is $y_s(t) = cy_1(t)$
  - Let $y_2(t)$ be solution with $u_2(t)$ and zero initial conditions
  - Additivity property: for solution of linear ODE with input $u_a(t) = u_1(t) + u_2(t)$ and zero initial conditions is $y_a(t) = y_1(t) + y_2(t)$

Continuous-Time Linear System Models

- **Continuous-time linear state-space model** general form:

$$\begin{aligned} \dot{\vec{x}}(t) &= A(t)\vec{x}(t) + B(t)\vec{u}(t) \\ \vec{y}(t) &= C(t)\vec{x}(t) + D(t)\vec{u}(t) \end{aligned} \tag{3}$$

  - $A(t) \in \mathbb{R}^{n_x \times n_x}$: **state matrix**
  - $B(t) \in \mathbb{R}^{n_x \times n_u}$: **input matrix**
  - $C(t) \in \mathbb{R}^{n_y \times n_x}$: **output matrix**
  - $D(t) \in \mathbb{R}^{n_y \times n_u}$: **feedthrough matrix**

Continuous-Time Time-Invariant System Models

- Time-invariant systems have dynamic equations that do not depend explicitly on time

Continuous-Time Time-Invariant System Models

- Time-invariant systems have dynamic equations that do not depend explicitly on time
- **Continuous-time time-invariant state-space model** general form:

$$\begin{aligned} \dot{\vec{x}}(t) &= f(\vec{x}, \vec{u}) \\ \vec{y}(t) &= h(\vec{x}, \vec{u}) \end{aligned} \tag{4}$$

Continuous-Time Time-Invariant System Models

- Time-invariant systems have dynamic equations that do not depend explicitly on time
- **Continuous-time time-invariant state-space model** general form:

$$\begin{aligned} \dot{\vec{x}}(t) &= f(\vec{x}, \vec{u}) \\ \vec{y}(t) &= h(\vec{x}, \vec{u}) \end{aligned} \tag{4}$$

- **Continuous-time linear time-invariant state-space models** general form

$$\begin{aligned} \dot{\vec{x}}(t) &= A\vec{x}(t) + B\vec{u}(t) \\ \vec{y}(t) &= C\vec{x}(t) + D\vec{u}(t) \end{aligned} \tag{5}$$

  - LTI systems theory borrows heavily from mathematical methods in differential equations and linear algebra, utilized throughout course

Notes on LTI Systems

- Particular LTI state-space model can be denoted by the quadruple (A, B, C, D)
  - Different choices for (A, B, C, D) that represent same dynamical system in terms of input-to-output relationship
  - Internal state different for each representation

Notes on LTI Systems

- Particular LTI state-space model can be denoted by the quadruple (A, B, C, D)
  - Different choices for (A, B, C, D) that represent same dynamical system in terms of input-to-output relationship
  - Internal state different for each representation
- To set $\vec{y} = \vec{x}$: set output matrix to **identity matrix**

$$C = I = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \vdots & 1 \end{bmatrix} \quad (6)$$

& set feedthrough matrix to **zero matrix**

$$D = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \quad (7)$$

Transfer Function

- $G(s)$, "transfers" transformed input $u(s)$ to transformed output $y(s)$:

$$y(s) = G(s)u(s) \tag{8}$$

Transfer Function

- $G(s)$, "transfers" transformed input $u(s)$ to transformed output $y(s)$:

$$y(s) = G(s)u(s) \tag{8}$$

- **Standard transform function form**:

$$G(s) = \frac{b_m s^m + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0} \tag{9}$$

Transfer Function

- $G(s)$, "transfers" transformed input $u(s)$ to transformed output $y(s)$:

$$y(s) = G(s)u(s) \tag{8}$$

- **Standard transform function form**:

$$G(s) = \frac{b_m s^m + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0} \tag{9}$$

- Numerator and denominator: polynomials $\rightarrow$ have real and/or complex roots
  - **Zero** of transfer function z: root of numerator
    - $|G(s)| \rightarrow 0$ for s equal to zero
  - **Pole** of transfer function p: root of denominator
    - $|G(s)| \rightarrow \infty$ for s equal to pole
    - Plotting $|G(s)|$ over complex plane, ∞ looks like tent pole in 3D

Transfer Function

- $G(s)$, "transfers" transformed input $u(s)$ to transformed output $y(s)$:

$$y(s) = G(s)u(s) \tag{8}$$

- **Standard transform function form**:

$$G(s) = \frac{b_m s^m + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0} \tag{9}$$

- Numerator and denominator: polynomials $\rightarrow$ have real and/or complex roots
  - **Zero** of transfer function z: root of numerator
    - $|G(s)| \rightarrow 0$ for s equal to zero
  - **Pole** of transfer function p: root of denominator
    - $|G(s)| \rightarrow \infty$ for s equal to pole
    - Plotting $|G(s)|$ over complex plane, ∞ looks like tent pole in 3D
- Can always factor transfer function as

$$G(s) = \frac{K(s - z_1)\cdots(s - z_m)}{(s - p_1)\cdots(s - p_n)} \tag{10}$$

Notes on Transfer Functions

- Can always factor transfer function as **partial-fraction decomposition**, i.e.

$$G(s) = \frac{r_1}{(s - p_1)} + \cdots + \frac{r_n}{(s - p_n)} \tag{11}$$

  - r_i: i^{th} constant factor known as **residue**, will be 0 if pole-zero cancellation exists for p_i

Notes on Transfer Functions

- Can always factor transfer function as **partial-fraction decomposition**, i.e.

$$G(s) = \frac{r_1}{(s - p_1)} + \cdots + \frac{r_n}{(s - p_n)} \tag{11}$$

  - r_i: i^{th} constant factor known as **residue**, will be 0 if pole-zero cancellation exists for p_i
- **Pole-zero cancellations**: pole and zero with same value
  - **Minimal realization** transfer function: no pole-zero cancellations

Notes on Transfer Functions

- Can always factor transfer function as **partial-fraction decomposition**, i.e.

$$G(s) = \frac{r_1}{(s - p_1)} + \cdots + \frac{r_n}{(s - p_n)} \tag{11}$$

  - r_i: i^{th} constant factor known as **residue**, will be 0 if pole-zero cancellation exists for p_i
- **Pole-zero cancellations**: pole and zero with same value
  - **Minimal realization** transfer function: no pole-zero cancellations
- **Final Value Theorem (FVT)**: if every pole of transfer function $F(s)$ not purely imaginary *except*, at most, single pole at origin, then:

$$\lim_{t \to \infty} f(t) = \lim_{s \to 0} sF(s) \tag{12}$$

  - $s \to 0$ denotes s approaching through positive numbers

Transfer Functions in Multivariate Domain

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

$$\begin{aligned}\dot{\vec{x}}(t) &= A\vec{x}(t) + B\vec{u}(t)\\ \vec{y}(t) &= C\vec{x}(t) + D\vec{u}(t)\end{aligned} \tag{13}$$

Transfer Functions in Multivariate Domain

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

$$\begin{aligned} \dot{\vec{x}}(t) &= A\vec{x}(t) + B\vec{u}(t) \\ \vec{y}(t) &= C\vec{x}(t) + D\vec{u}(t) \end{aligned} \tag{13}$$

- Transform to Laplace domain with zero initial conditions as

$$\begin{aligned} s\vec{x}(s) &= A\vec{x}(s) + B\vec{u}(s) \\ \vec{y}(s) &= C\vec{x}(s) + D\vec{u}(s) \end{aligned} \tag{14}$$

Transfer Function Matrix

- To find **transfer function matrix**, $[G(s)]$, maps each input to each output through matrix of transfer functions, i.e.

$$\vec{y}(s) = [G(s)]\vec{u}(s) \tag{15}$$

Transfer Function Matrix

- To find **transfer function matrix**, $[G(s)]$, maps each input to each output through matrix of transfer functions, i.e.

$$\vec{y}(s) = [G(s)]\vec{u}(s) \tag{15}$$

- Requires finding $\vec{x}(s)$ in terms of $\vec{u}(s)$ and substituting into output equation

$$s\vec{x}(s) - A\vec{x}(s) = B\vec{u}(s) \tag{16}$$

Transfer Function Matrix

- To find **transfer function matrix**, $[G(s)]$, maps each input to each output through matrix of transfer functions, i.e.

$$\vec{y}(s) = [G(s)]\vec{u}(s) \tag{15}$$

- Requires finding $\vec{x}(s)$ in terms of $\vec{u}(s)$ and substituting into output equation

$$s\vec{x}(s) - A\vec{x}(s) = B\vec{u}(s) \tag{16}$$

$$(sI - A)\,\vec{x}(s) = B\vec{u}(s) \tag{17}$$

Transfer Function Matrix

- To find **transfer function matrix**, $[G(s)]$, maps each input to each output through matrix of transfer functions, i.e.

$$\vec{y}(s) = [G(s)]\vec{u}(s) \tag{15}$$

- Requires finding $\vec{x}(s)$ in terms of $\vec{u}(s)$ and substituting into output equation

$$s\vec{x}(s) - A\vec{x}(s) = B\vec{u}(s) \tag{16}$$

$$(sI - A)\,\vec{x}(s) = B\vec{u}(s) \tag{17}$$

$$\vec{x}(s) = (sI - A)^{-1}\,B\vec{u}(s) \tag{18}$$

  - $(sI - A)^{-1}$: **resolvent** of A

Standard Transfer Function Matrix

- By substitution

$$\vec{y}(s) = C\left(sI - A\right)^{-1} B\vec{u}(s) + D\vec{u}(s) \tag{19}$$

Standard Transfer Function Matrix

- By substitution

$$\vec{y}(s) = C(sI - A)^{-1} B\vec{u}(s) + D\vec{u}(s) \tag{19}$$

$$\vec{y}(s) = \left(C(sI - A)^{-1} B + D\right) \vec{u}(s) \tag{20}$$

Standard Transfer Function Matrix

- By substitution

$$\vec{y}(s) = C\left(sI - A\right)^{-1} B\vec{u}(s) + D\vec{u}(s) \tag{19}$$

$$\vec{y}(s) = \left(C\left(sI - A\right)^{-1} B + D\right)\vec{u}(s) \tag{20}$$

- By definition of transfer function matrix, **standard transfer function matrix**:

$$[G(s)] = C\left(sI - A\right)^{-1} B + D \tag{21}$$

- $n \times m$ matrix of transfer function elements, $G_{ij}(s)$, i.e.

$$[G(s)] = \begin{bmatrix} G_{11}(s) & \cdots & G_{1m}(s) \\ \vdots & \ddots & \vdots \\ G_{n1} & \cdots & G_{nm}(s) \end{bmatrix} \tag{22}$$

Equilibrium Points for State-Space Models

- State-input pair $(\vec{x}, \vec{u})$ **equilibrium point**: if $\dot{\vec{x}}$ zero for all $t > 0$, i.e. if

$$f(t, \vec{x}, \vec{u}) = 0 \tag{23}$$

valid solution for all $t \geq 0$

Equilibrium Points for State-Space Models

- State-input pair $(\bar{\vec{x}}, \bar{\vec{u}})$ **equilibrium point**: if $\dot{\vec{x}}$ zero for all $t > 0$, i.e. if

$$f(t, \bar{\vec{x}}, \bar{\vec{u}}) = 0 \tag{23}$$

  valid solution for all $t \geq 0$
- If initialize $\vec{x}(t) = \bar{\vec{x}}$ at $t = 0$ *and* set $\vec{u}(t) = \bar{\vec{u}}$ for $t \geq 0$ then $\vec{x}(t) = \bar{\vec{x}}$ and $\bar{\vec{y}} = h(\bar{\vec{x}}, \bar{\vec{u}})$ for all $t \geq 0$
  - I.e. $\vec{x}$ “steady”

Equilibrium Points for State-Space Models

- State-input pair $(\bar{\vec{x}}, \bar{\vec{u}})$ **equilibrium point**: if $\dot{\vec{x}}$ zero for all $t > 0$, i.e. if

$$f(t, \bar{\vec{x}}, \bar{\vec{u}}) = 0 \tag{23}$$

  valid solution for all $t \geq 0$
- If initialize $\vec{x}(t) = \bar{\vec{x}}$ at $t = 0$ *and* set $\vec{u}(t) = \bar{\vec{u}}$ for $t \geq 0$

  then $\vec{x}(t) = \bar{\vec{x}}$ and $\bar{\vec{y}} = h(\bar{\vec{x}}, \bar{\vec{u}})$ for all $t \geq 0$
  - I.e. $\vec{x}$ "steady"
- Also may be multiple, even infinite, solutions for particular **trim state**, $\bar{\vec{x}}$
  - Fewer equations than free variables for vector-valued state equation
  - I.e. n_x elements and $n_x + n_u$ free variables

Lyapunov Stability Definitions

- Important characterization of system equilibrium points: stability
  - First studied by Lyapunov in dissertation: *The General Problem of Stability of Motion*

Lyapunov Stability Definitions

- Important characterization of system equilibrium points: stability
  - First studied by Lyapunov in dissertation: *The General Problem of Stability of Motion*

- With condition $u(t) = \bar{u}$, equilibrium point $\bar{y}$ **stable in sense of Lyapunov (SISL)**: if for some $|\epsilon| > 0$ with $y(0) = \bar{y} + \epsilon$, there exists some δ for which $|y(t)| < |\bar{y} + \delta|$ as $t \to \infty$
  - I.e. if initialize dynamical system “near enough” to equilibrium, system will “remain close” to equilibrium

Lyapunov Stability Definitions

- Important characterization of system equilibrium points: stability
  - First studied by Lyapunov in dissertation: *The General Problem of Stability of Motion*
- With condition $u(t) = \bar{u}$, equilibrium point $\bar{y}$ **stable in sense of Lyapunov (SISL)**: if for some $|\epsilon| > 0$ with $y(0) = \bar{y} + \epsilon$, there exists some δ for which $|y(t)| < |\bar{y} + \delta|$ as $t \to \infty$
  - I.e. if initialize dynamical system “near enough” to equilibrium, system will “remain close” to equilibrium
  - If δ exists for all $\epsilon \in \mathbb{R}$, then equilibrium point **globally stable**

Lyapunov Stability Definitions

- Important characterization of system equilibrium points: stability
  - First studied by Lyapunov in dissertation: *The General Problem of Stability of Motion*
- With condition $u(t) = \bar{u}$, equilibrium point $\bar{y}$ **stable in sense of Lyapunov (SISL)**: if for some $|\epsilon| > 0$ with $y(0) = \bar{y} + \epsilon$, there exists some δ for which $|y(t)| < |\bar{y} + \delta|$ as $t \to \infty$
  - I.e. if initialize dynamical system “near enough” to equilibrium, system will “remain close” to equilibrium
  - If δ exists for all $\epsilon \in \mathbb{R}$, then equilibrium point **globally stable**
  - If $y \to \bar{y}$ as $t \to \infty$, then equilibrium point **asymptotically stable**

Lyapunov Stability Definitions

- Important characterization of system equilibrium points: stability
  - First studied by Lyapunov in dissertation: *The General Problem of Stability of Motion*
- With condition $u(t) = \bar{u}$, equilibrium point $\bar{y}$ **stable in sense of Lyapunov (SISL)**: if for some $|\epsilon| > 0$ with $y(0) = \bar{y} + \epsilon$, there exists some δ for which $|y(t)| < |\bar{y} + \delta|$ as $t \to \infty$
  - I.e. if initialize dynamical system “near enough” to equilibrium, system will “remain close” to equilibrium
  - If δ exists for all $\epsilon \in \mathbb{R}$, then equilibrium point **globally stable**
  - If $y \to \bar{y}$ as $t \to \infty$, then equilibrium point **asymptotically stable**
  - If $y \to \bar{y}$ for all $\epsilon \in \mathbb{R}$, then equilibrium point **globally asymptotically stable (GAS)**

Lyapunov's Methods

- To qualitatively assess stability of equilibrium points: Lyapunov developed two methods

Lyapunov's Methods

- To qualitatively assess stability of equilibrium points: Lyapunov developed two methods
  1. If *linearized dynamical system* about equilibrium point stable, then original dynamical system also stable for some "stability neighborhood" about equilibrium point
  - Size of "stability neighborhood" directly related to effects of neglected higher-order terms (HOT) in linearization

Lyapunov's Methods

- To qualitatively assess stability of equilibrium points: Lyapunov developed two methods
  1. If *linearized dynamical system* about equilibrium point stable, then original dynamical system also stable for some "stability neighborhood" about equilibrium point
     - Size of "stability neighborhood" directly related to effects of neglected higher-order terms (HOT) in linearization
  2. For highly nonlinear ODEs one typically analyzes system's stability using Lyapunov's second method as linearized stability neighborhood too small to be practically useful

Lyapunov's Methods

- To qualitatively assess stability of equilibrium points: Lyapunov developed two methods
  1. If *linearized dynamical system* about equilibrium point stable, then original dynamical system also stable for some "stability neighborhood" about equilibrium point
     - Size of "stability neighborhood" directly related to effects of neglected higher-order terms (HOT) in linearization
  2. For highly nonlinear ODEs one typically analyzes system's stability using Lyapunov's second method as linearized stability neighborhood too small to be practically useful

- LTI FDC considers Lyapunov's first method for time-invariant systems
  - Provide methods for assessing flight vehicle's stability and designing control systems
  - Nonlinear and time-varying control: discusses Lyapunov's second method

Multivariate Linearization

- **Taylor Series** for multivariate function $f(\vec{x}, \vec{u})$ about vector pair $(\bar{\vec{x}}, \bar{\vec{u}})$:

$$\dot{\vec{x}}(t) = f(\vec{x}, \vec{u}) = f(\bar{\vec{x}}, \bar{\vec{u}}) + \left[\frac{\partial f}{\partial \vec{x}}(\bar{\vec{x}}, \bar{\vec{u}})\right](\vec{x} - \bar{\vec{x}}) + \left[\frac{\partial f}{\partial \vec{u}}(\bar{\vec{x}}, \bar{\vec{u}})\right](\vec{u} - \bar{\vec{u}}) + \text{HOT} \quad (24)$$

&

$$\vec{y}(t) = h(\vec{x}, \vec{u}) = h(\bar{\vec{x}}, \bar{\vec{u}}) + \left[\frac{\partial h}{\partial \vec{x}}(\bar{\vec{x}}, \bar{\vec{u}})\right](\vec{x} - \bar{\vec{x}}) + \left[\frac{\partial h}{\partial \vec{u}}(\bar{\vec{x}}, \bar{\vec{u}})\right](\vec{u} - \bar{\vec{u}}) + \text{HOT} \quad (25)$$

Multivariate Linearization

- **Taylor Series** for multivariate function $f(\vec{x}, \vec{u})$ about vector pair $(\bar{\vec{x}}, \bar{\vec{u}})$:

$$\dot{\vec{x}}(t) = f(\vec{x}, \vec{u}) = f(\bar{\vec{x}}, \bar{\vec{u}}) + \left[\frac{\partial f}{\partial \vec{x}}(\bar{\vec{x}}, \bar{\vec{u}})\right](\vec{x} - \bar{\vec{x}}) + \left[\frac{\partial f}{\partial \vec{u}}(\bar{\vec{x}}, \bar{\vec{u}})\right](\vec{u} - \bar{\vec{u}}) + \text{HOT} \quad (24)$$

&

$$\vec{y}(t) = h(\vec{x}, \vec{u}) = h(\bar{\vec{x}}, \bar{\vec{u}}) + \left[\frac{\partial h}{\partial \vec{x}}(\bar{\vec{x}}, \bar{\vec{u}})\right](\vec{x} - \bar{\vec{x}}) + \left[\frac{\partial h}{\partial \vec{u}}(\bar{\vec{x}}, \bar{\vec{u}})\right](\vec{u} - \bar{\vec{u}}) + \text{HOT} \quad (25)$$

- $\left[\frac{\partial f}{\partial \vec{x}}(\bar{\vec{x}}, \bar{\vec{u}})\right]$: **Jacobian** of $f()$
- Multivariate linearization a.k.a. **Jacobian linearization**

Perturbation Vectors

- State **perturbation vector** about constant $\bar{\vec{x}}$:

$$\Delta \vec{x}(t) = \vec{x}(t) - \bar{\vec{x}} \tag{26}$$

Perturbation Vectors

- State **perturbation vector** about constant $\bar{\vec{x}}$:

$$\Delta \vec{x}(t) = \vec{x}(t) - \bar{\vec{x}} \tag{26}$$

- Input **perturbation vector** about constant $\bar{\vec{u}}$:

$$\Delta \vec{u}(t) = \vec{u}(t) - \bar{\vec{u}} \tag{27}$$

Perturbation Vectors

- State **perturbation vector** about constant $\bar{\vec{x}}$:

$$\Delta\vec{x}(t) = \vec{x}(t) - \bar{\vec{x}} \tag{26}$$

- Input **perturbation vector** about constant $\bar{\vec{u}}$:

$$\Delta\vec{u}(t) = \vec{u}(t) - \bar{\vec{u}} \tag{27}$$

- Output **perturbation vector** about constant $\bar{\vec{y}}$:

$$\Delta\vec{y}(t) = \vec{y}(t) - \bar{\vec{y}} \tag{28}$$

Perturbation Vectors

- State **perturbation vector** about constant $\bar{\vec{x}}$:

$$\Delta\vec{x}(t) = \vec{x}(t) - \bar{\vec{x}} \tag{26}$$

- Input **perturbation vector** about constant $\bar{\vec{u}}$:

$$\Delta\vec{u}(t) = \vec{u}(t) - \bar{\vec{u}} \tag{27}$$

- Output **perturbation vector** about constant $\bar{\vec{y}}$:

$$\Delta\vec{y}(t) = \vec{y}(t) - \bar{\vec{y}} \tag{28}$$

- Recognize for trim, $f(\bar{\vec{x}}, \bar{\vec{u}}) = 0$ and $h(\bar{\vec{x}}, \bar{\vec{u}}) = \bar{\vec{y}}$:

$$\Delta\dot{\vec{x}}(t) = \dot{\vec{x}}(t) = f(\vec{x}, \vec{u}) = f(\bar{\vec{x}}, \bar{\vec{u}}) + \left[\frac{\partial f}{\partial \vec{x}}(\bar{\vec{x}}, \bar{\vec{u}})\right]\Delta\vec{x}(t) + \left[\frac{\partial f}{\partial \vec{u}}(\bar{\vec{x}}, \bar{\vec{u}})\right]\Delta\vec{u}(t) + \text{HOT} \tag{29}$$

Time-Invariant State-Space Model

- Recall general form for time-invariant state-space model:

$$\begin{aligned} \dot{\vec{x}} &= f(\vec{x}, \vec{u}) \\ \vec{y} &= h(\vec{x}, \vec{u}) \end{aligned} \tag{31}$$

Time-Invariant State-Space Model

- Recall general form for time-invariant state-space model:

$$\begin{aligned} \dot{\vec{x}} &= f(\vec{x}, \vec{u}) \\ \vec{y} &= h(\vec{x}, \vec{u}) \end{aligned} \tag{31}$$

- Set

$$A = \left[\frac{\partial f}{\partial \vec{x}}(\bar{\vec{x}}, \bar{\vec{u}})\right] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial f_1}{\partial x_{n_x}}(\bar{\vec{x}}, \bar{\vec{u}}) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{n_x}}{\partial x_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial f_{n_x}}{\partial x_{n_x}}(\bar{\vec{x}}, \bar{\vec{u}}) \end{bmatrix} \tag{32}$$

$$B = \left[\frac{\partial f}{\partial \vec{u}}(\bar{\vec{x}}, \bar{\vec{u}})\right] = \begin{bmatrix} \frac{\partial f_1}{\partial u_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial f_1}{\partial u_{n_u}}(\bar{\vec{x}}, \bar{\vec{u}}) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{n_x}}{\partial u_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial f_{n_x}}{\partial u_{n_u}}(\bar{\vec{x}}, \bar{\vec{u}}) \end{bmatrix} \tag{33}$$

Time-Invariant State-Space Model (continued)

$$C = \left[\frac{\partial h}{\partial \vec{x}}(\bar{\vec{x}}, \bar{\vec{u}})\right] = \begin{bmatrix} \frac{\partial h_1}{\partial x_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial h_1}{\partial x_{n_x}}(\bar{\vec{x}}, \bar{\vec{u}}) \\ \vdots & \ddots & \vdots \\ \frac{\partial h_{n_y}}{\partial x_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial h_{n_y}}{\partial x_{n_x}}(\bar{\vec{x}}, \bar{\vec{u}}) \end{bmatrix} \tag{34}$$

$$D = \left[\frac{\partial h}{\partial \vec{u}}(\bar{\vec{x}}, \bar{\vec{u}})\right] = \begin{bmatrix} \frac{\partial h_1}{\partial u_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial h_1}{\partial u_{n_u}}(\bar{\vec{x}}, \bar{\vec{u}}) \\ \vdots & \ddots & \vdots \\ \frac{\partial h_{n_y}}{\partial u_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial h_{n_y}}{\partial u_{n_u}}(\bar{\vec{x}}, \bar{\vec{u}}) \end{bmatrix} \tag{35}$$

Time-Invariant State-Space Model (continued)

$$C = \left[\frac{\partial h}{\partial \vec{x}}(\bar{\vec{x}}, \bar{\vec{u}})\right] = \begin{bmatrix} \frac{\partial h_1}{\partial x_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial h_1}{\partial x_{n_x}}(\bar{\vec{x}}, \bar{\vec{u}}) \\ \vdots & \ddots & \vdots \\ \frac{\partial h_{n_y}}{\partial x_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial h_{n_y}}{\partial x_{n_x}}(\bar{\vec{x}}, \bar{\vec{u}}) \end{bmatrix} \tag{34}$$

$$D = \left[\frac{\partial h}{\partial \vec{u}}(\bar{\vec{x}}, \bar{\vec{u}})\right] = \begin{bmatrix} \frac{\partial h_1}{\partial u_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial h_1}{\partial u_{n_u}}(\bar{\vec{x}}, \bar{\vec{u}}) \\ \vdots & \ddots & \vdots \\ \frac{\partial h_{n_y}}{\partial u_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial h_{n_y}}{\partial u_{n_u}}(\bar{\vec{x}}, \bar{\vec{u}}) \end{bmatrix} \tag{35}$$

- Yields approximate LTI state-space model about $\bar{\vec{x}}$ and $\bar{\vec{u}}$:

$$\begin{aligned} \Delta\dot{\vec{x}}(t) &\approx A\Delta\vec{x}(t) + B\Delta\vec{u}(t) \\ \Delta\vec{y}(t) &\approx C\Delta\vec{x}(t) + D\Delta\vec{u}(t) \end{aligned} \tag{36}$$

General Solution for LTI Systems

- General LTI state-space form

$$\begin{aligned}\dot{\vec{x}}(t) &= A\vec{x}(t) + B\vec{u}(t)\\ \vec{y}(t) &= C\vec{x}(t) + D\vec{u}(t)\end{aligned} \tag{37}$$

  - Boundary conditions: $\vec{x}(t_0)$

General Solution for LTI Systems

- General LTI state-space form

$$\begin{aligned}\dot{\vec{x}}(t) &= A\vec{x}(t) + B\vec{u}(t) \\ \vec{y}(t) &= C\vec{x}(t) + D\vec{u}(t)\end{aligned} \tag{37}$$

  - Boundary conditions: $\vec{x}(t_0)$

- General solution for vector-valued first-order state equation:

$$\begin{aligned}\vec{x}(t) &= e^{A(t-t_0)}\vec{x}(t_0) + \int_{t_0}^{t} e^{A(t-\tau)}B\vec{u}(\tau)d\tau \\ \vec{y}(t) &= C\left[e^{A(t-t_0)}\vec{x}(t_0) + \int_{t_0}^{t} e^{A(t-\tau)}B\vec{u}(\tau)d\tau\right] + D\vec{u}(t)\end{aligned} \tag{38}$$

State-Space Characteristic Equation

- **State-space characteristic equation** for general LTI systems:

$$\det\left[\lambda I - A\right] = 0 \tag{39}$$

  - Solutions to equation: n_x roots of state-space characteristic polynomial, may be repeated and will either be real numbers or complex-conjugate pairs

State-Space Characteristic Equation

- **State-space characteristic equation** for general LTI systems:

$$\det\left[\lambda I - A\right] = 0 \tag{39}$$

  - Solutions to equation: n_x roots of state-space characteristic polynomial, may be repeated and will either be real numbers or complex-conjugate pairs

- Roots of characteristic polynomials a.k.a. **system poles**
  - If no pole-zero cancellations, then poles of any $G_{ij}(s)$ also system poles

State-Space Characteristic Equation

- **State-space characteristic equation** for general LTI systems:

$$\det\left[\lambda I - A\right] = 0 \tag{39}$$

  - Solutions to equation: n_x roots of state-space characteristic polynomial, may be repeated and will either be real numbers or complex-conjugate pairs
- Roots of characteristic polynomials a.k.a. **system poles**
  - If no pole-zero cancellations, then poles of any $G_{ij}(s)$ also system poles
- Each distinct real pole and each distinct complex-conjugate pair: **system mode**
  - Poles in LHP: stable

Stability

- Real part of each individual mode: characterize stability of each mode
  - **Stable** mode: real part of $\lambda < 0$
  - **Marginally stable** mode: real part of $\lambda = 0$
  - **Unstable** mode: real part of $\lambda > 0$

Stability

- Real part of each individual mode: characterize stability of each mode
  - **Stable** mode: real part of $\lambda < 0$
  - **Marginally stable** mode: real part of $\lambda = 0$
  - **Unstable** mode: real part of $\lambda > 0$

- Stable LTI system: *all* modes stable
  - If *any* mode unstable $\rightarrow$ unstable LTI system

Stability

- Real part of each individual mode: characterize stability of each mode
  - **Stable** mode: real part of $\lambda < 0$
  - **Marginally stable** mode: real part of $\lambda = 0$
  - **Unstable** mode: real part of $\lambda > 0$
- Stable LTI system: *all* modes stable
  - If *any* mode unstable $\rightarrow$ unstable LTI system
- Eigenvalues may be complex-valued $\rightarrow$ system poles located in left half of complex plane, i.e. **left half plane (LHP)**, correspond to stable modes

Modal Characteristics

- Each mode corresponds to unique **modal time constant**
  - Describes exponential decay/growth rate

$$\tau = \frac{1}{\text{Real}(\lambda)} \tag{40}$$

  - Real(λ) represents the real part of λ, i.e. if $\tau < 0$, then stable mode
  - Modes with faster decay on output $=$ modes with higher eigenvalues

Transfer Function Modal Analysis

- For complex-conjugate pairs, $\lambda = \frac{1}{\tau} \pm j\omega_d$, define **modal damped frequency**: ω_d
  - Related to **modal undamped natural frequency**: ω_n
  - **Modal damping ratio**, $0 < \zeta_i < 1$

Transfer Function Modal Analysis

- For complex-conjugate pairs, $\lambda = \frac{1}{\tau} \pm j\omega_d$, define **modal damped frequency**: ω_d
  - Related to **modal undamped natural frequency**: ω_n
  - **Modal damping ratio**, $0 < \zeta_i < 1$

$$\omega_d = \omega_n \sqrt{1 - \zeta^2} \tag{41}$$

$$\tau = -\frac{1}{\zeta \omega_n} \tag{42}$$

Transfer Function Modal Analysis

- For complex-conjugate pairs, $\lambda = \frac{1}{\tau} \pm j\omega_d$, define **modal damped frequency**: ω_d
  - Related to **modal undamped natural frequency**: ω_n
  - **Modal damping ratio**, $0 < \zeta_i < 1$

$$\omega_d = \omega_n \sqrt{1 - \zeta^2} \tag{41}$$

$$\tau = -\frac{1}{\zeta \omega_n} \tag{42}$$

- Note: one can compare:

$$(\lambda - \frac{1}{\tau} + j\omega_d)(\lambda - \frac{1}{\tau} - j\omega_d) \tag{43}$$

$$\lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2 \tag{44}$$

Summary

- Model-based design: provides hierarchical and procedural approach to overall system design for complex systems, including aerospace vehicles

Summary

- Model-based design: provides hierarchical and procedural approach to overall system design for complex systems, including aerospace vehicles
- Aerospace planning, control, and perception systems differentiate aspects of operation of vehicle
  - Plan flight trajectory, actuate vehicle to follow trajectory, sense and estimate state of vehicle, target(s), & environment

Summary

- Model-based design: provides hierarchical and procedural approach to overall system design for complex systems, including aerospace vehicles
- Aerospace planning, control, and perception systems differentiate aspects of operation of vehicle
  - Plan flight trajectory, actuate vehicle to follow trajectory, sense and estimate state of vehicle, target(s), & environment
- Model-based design of control and perception systems uses 4 steps
  - Plant modeling, control and perception design, system simulation, V&V

Summary

- Model-based design: provides hierarchical and procedural approach to overall system design for complex systems, including aerospace vehicles
- Aerospace planning, control, and perception systems differentiate aspects of operation of vehicle
  - Plan flight trajectory, actuate vehicle to follow trajectory, sense and estimate state of vehicle, target(s), & environment
- Model-based design of control and perception systems uses 4 steps
  - Plant modeling, control and perception design, system simulation, V&V
- Systems and signals key to understanding dynamics and controls theory
- Continuous-time dynamical systems theory use differential equations in standard state-space form
  - Simplify mathematical theory for tractable analysis and design

Summary

- Model-based design: provides hierarchical and procedural approach to overall system design for complex systems, including aerospace vehicles
- Aerospace planning, control, and perception systems differentiate aspects of operation of vehicle
  - Plan flight trajectory, actuate vehicle to follow trajectory, sense and estimate state of vehicle, target(s), & environment
- Model-based design of control and perception systems uses 4 steps
  - Plant modeling, control and perception design, system simulation, V&V
- Systems and signals key to understanding dynamics and controls theory
- Continuous-time dynamical systems theory use differential equations in standard state-space form
  - Simplify mathematical theory for tractable analysis and design
- Course focus: MIMO, continuous-time dynamical systems
  - Advanced dynamics modeling
  - Modern linear control design for aerospace vehicles using trim & linearization