AMME 3500: System Dynamics & Control

State Space Modelling and Control

Dr Ian R. Manchester
## Course Outline

<table>
<thead>
<tr>
<th>Week</th>
<th>Content</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Frequency Domain Modelling</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Transient Performance and the s-plane</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Block Diagrams</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Feedback System Characteristics</td>
<td>Assign 1 Due</td>
</tr>
<tr>
<td>6</td>
<td>Root Locus</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Root Locus 2</td>
<td>Assign 2 Due</td>
</tr>
<tr>
<td>8</td>
<td>Bode Plots</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Bode Plots 2</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>State Space Modeling</td>
<td>Assign 3 Due</td>
</tr>
<tr>
<td>11</td>
<td>State Space Design Techniques</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Advanced Control Topics</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Review</td>
<td>Assign 4 Due</td>
</tr>
</tbody>
</table>

Dr. Ian R. Manchester  
Amme 3500 : Introduction  
Slide 2
The Elephant and the Blind Men

It’s a Fan!

It’s a Spear!

It’s a Wall!

It’s a Rope!

It’s a Snake!

It’s a Tree!
Frequency Domain Techniques

• Until now we have been looking at classical, or Frequency Domain, techniques for feedback control analysis and design.
• We convert a systems DE to a transfer function.
• Root Locus and Bode techniques provide us with a means for modelling and designing controllers for LTI systems.
Limitations of Frequency Domain

- Frequency domain methods give a valuable insight and useful tools, **but:**
- They can be applied only to LTI systems or systems that can be approximated as such.
- It is generally only practical for handling systems with a single input and single output.
- It is difficult to **precisely** specify performance for high-order systems.
Limitations of Frequency Domain

- Even relatively simple seeming things like controlling both position and velocity simultaneously may be difficult using classical techniques.
Limitations of Frequency Domain

- Control of high performance aircraft, for example, is tightly coupled.
- Full 6DOF models are not easily accommodated using classical techniques.
State Space Theory

• Early development in 1960s driven by aerospace control problems.
• Current state of the art: optimized control of an entire petrochemical plant
• > 600 inputs and outputs.
State Space Modelling

• State space techniques provide us with an alternative means for modelling and control design.
  – They can handle non-linear, time-varying ODEs.
  – They can handle multiple input, multiple output systems.
  – They allow precise and “optimal” designs for higher-order systems.

• Sometimes this is referred to as “modern” control, as opposed to “classical” control (RL, Bode, …)

• In truth, a modern control engineer uses and understands both “modern” and “classical” theory.
State Space Modelling

• Using the state space approach, we represent a system by a set of $n$ first-order differential equations:

$$\dot{x} = Ax + Bu$$

• The output of the system is expressed as:

$$y = Cx + [Du]$$

- $x$ - state vector
- $y$ - output vector
- $u$ - input vector
- $A$ - state matrix
- $B$ - input matrix
- $C$ - output matrix
- $D$ - feedthrough or feedforward matrix (often zero)
What is the State?

• A state vector is a list of numbers is sufficient to *completely* determine the response of a system to a given input and/or initial condition.

• A system is *finite dimensional* if a finite number, usually denoted $n$, of such state variables suffices.

• E.g. a Newtonian system of point masses: the positions and velocities of every mass.
State Space Modelling

• We can draw a block diagram describing the general State Space Model

\[ \dot{x} = Ax + Bu \quad y = Cx + [Du] \]
In previous lectures, we considered this mechanical system.

We derived a differential equation defining its motion

\[ \sum F_z = M\ddot{z}(t) \]

\[ f(t) - Kz - K_d\dot{z} = M\ddot{z}(t) \]

\[ M\ddot{z}(t) + K_d\dot{z}(t) + K\dddot{z}(t) = f(t) \]
State Space Example

• For an \( n^{\text{th}} \) order differential equation, we will look for \( n \) state variables with which to describe the system

\[
M \ddot{z} + K_d \dot{z} + K z = f
\]

• We will select the state variables

\[
x = \begin{bmatrix} x_1 \\ x_2 \\ \dot{z} \end{bmatrix} = \begin{bmatrix} z \\ \dot{z} \end{bmatrix}
\]
State Space Example

- We want to write down a set of first order differential equations in terms of the state variables

\[
\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \ddot{z} \\ \ddot{z} \end{bmatrix}
\]

- Now rearrange the differential equation to yield

\[
\ddot{z} = -\frac{K_d}{M} \dot{z} - \frac{K}{M} z + \frac{f}{M}
\]
State Space Example

• Combining these equations yields

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\end{bmatrix} = \begin{bmatrix}
\ddot{z} \\
\ddot{z} \\
\end{bmatrix} = \begin{bmatrix}
0 & 1 \\
-\frac{K}{M} & -\frac{K_d}{M} \\
\end{bmatrix}\begin{bmatrix}
\dot{z} \\
\dot{z} \\
\end{bmatrix} + \begin{bmatrix}
0 \\
\frac{1}{M} \\
\end{bmatrix}f
\]

• Or

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\end{bmatrix} = \begin{bmatrix}
0 & 1 \\
-\frac{K}{M} & -\frac{K_d}{M} \\
\end{bmatrix}\begin{bmatrix}
x_1 \\
x_2 \\
\end{bmatrix} + \begin{bmatrix}
0 \\
\frac{1}{M} \\
\end{bmatrix}f
\]
State Space Example

• If we are interested in the position of the mass, then

\[ y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = z \]

• So for this system, we have

\[
A = \begin{bmatrix} 0 & 1 \\ \frac{K}{M} & -\frac{K_d}{M} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{1}{M} \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D = [0]
\]
Converting SS to Transfer Function

- Given standard SS formulation:
  \[ \dot{x} = Ax + Bu \]
  \[ y = Cx + Du \]
- Taking Laplace transforms and assuming zero initial conditions yields:
  \[ sX(s) = AX(s) + BU(s) \]
  \[ Y(s) = CX(s) + DU(s) \]
- Rearranging terms yields:
  \[ (sI - A)X(s) = BU(s) \]
  \[ X(s) = (sI - A)^{-1} BU(s) \]
Converting SS to Transfer Function

• Substituting into the output equation we find:

\[
Y(s) = C(sI - A)^{-1} BU(s) + DU(s)
\]

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

\[
\frac{Y(s)}{U(s)} = C(sI - A)^{-1} B + D
\]

• This is known as the transfer function matrix
The transfer function is

\[ \frac{Y(s)}{U(s)} = C(sI-A)^{-1}B + D \]

- **Poles** occur when \((sI-A)\) is not an invertible matrix.
- i.e. when \(\det(sI-A) = 0\)
- For what \(s\) does this occur?
Example SS to TF

- For our suspended mass we have

\[
\dot{x} = \begin{bmatrix}
0 & 1 \\
-\frac{K}{M} & -\frac{K_d}{M}
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix}
+ \begin{bmatrix}
0 \\
\frac{1}{M}
\end{bmatrix} f
\quad y = \begin{bmatrix} 1 & 0 \end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix}
\]

- So

\[
\frac{Y}{F} = C(sI - A)^{-1}B + D
\]

\[
= \begin{bmatrix} 1 & 0 \end{bmatrix}
\begin{bmatrix}
\frac{s}{K} & -\frac{1}{s + \frac{K_d}{M}} \\
\frac{1}{M} & \frac{1}{M}
\end{bmatrix}^{-1}
\begin{bmatrix} 0 \\
1
\end{bmatrix}
+ 0
\]

\[
= \frac{1}{M}
\]

\[
= \frac{K_d}{M} s + \frac{K}{M}
\]

\[
= \frac{1}{s^2 + \frac{K_d}{M} s + \frac{K}{M}}
\]
Converting Transfer Function to SS

- Consider a general system described by the differential equation

\[ \frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_1 \frac{dy}{dt} + a_0 y = b_0 u \]

- A common way to choose state variables is to choose the output, \( y(t) \), and its \( n-1 \) derivatives as the state variables

- This is called the phase-variable form
Converting SS to Transfer Function

• Using this assignment of states yield

\[
\begin{align*}
    x_1 &= y \\
    x_2 &= \frac{dy}{dt} \\
    \vdots \\
    x_n &= \frac{d^{n-1}y}{dt^{n-1}} \\
    \dot{x}_1 &= \frac{dy}{dt} \\
    \dot{x}_2 &= \frac{d^2y}{dt^2} \\
    \vdots \\
    \dot{x}_n &= \frac{d^ny}{dt^n}
\end{align*}
\]
Converting SS to Transfer Function

- Equivalently, we can write

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= x_3 \\
&\vdots \\
\dot{x}_{n-1} &= x_n \\
\dot{x}_n &= -a_0 x_1 - a_1 x_2 -\cdots - a_{n-1} x_n - b_0 u
\end{align*}
\]
Converting SS to Transfer Function

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\vdots \\
\dot{x}_{n-1} \\
\dot{x}_n
\end{bmatrix} =
\begin{bmatrix}
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
\vdots \\
x_{n-1}
\end{bmatrix} +
\begin{bmatrix}
0 \\
0 \\
0 \\
\vdots \\
b_0
\end{bmatrix} u
\]

• with

\[
y = \begin{bmatrix}
1 & 0 & 0 & \cdots & 0 & 0
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
\vdots \\
x_{n-1} \\
x_n
\end{bmatrix}
\]
Example TF to SS

- Consider the system represented by

\[
\frac{20}{(s+1)(s+2)(s+10)}
\]

- Expanding terms we find

\[
(s^3 + 13s^2 + 32s + 20)C(s) = 20R(s)
\]

- Taking inverse LT yields

\[
\ddot{c} + 13\dot{c} + 32\dot{c} + 20c = 20r
\]
Example TF to SS

- We choose the state variables
  \[ x_1 = c \]
  \[ x_2 = \dot{c} \]
  \[ x_3 = \ddot{c} \]
- So
  \[
  \begin{bmatrix}
  \dot{x}_1 \\
  \dot{x}_2 \\
  \dot{x}_3 
  \end{bmatrix} =
  \begin{bmatrix}
  0 & 1 & 0 \\
  0 & 0 & 1 \\
  -20 & -32 & -13 
  \end{bmatrix}
  \begin{bmatrix}
  x_1 \\
  x_2 \\
  x_3 
  \end{bmatrix} +
  \begin{bmatrix}
  0 \\
  0 \\
  20 
  \end{bmatrix} r
  \]

  \[ y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x \]
• We can draw an equivalent block diagram describing this system.
Example TF to SS

• If we have zeros in the TF

\[
\begin{align*}
R(s) \quad \frac{s^2 + 5s + 20}{(s+1)(s+2)(s+10)} \quad C(s)
\end{align*}
\]

• We can rearrange the blocks to yield

\[
\begin{align*}
R(s) \quad \frac{1}{s^3 + 13s^2 + 32s + 20} \quad X_1(s) \quad \frac{s^2 + 5s + 20}{C(s)}
\end{align*}
\]
Example TF to SS

- The output of the first block, $X_1(s)$, is simply the first phase variable
- Examining the second block we find
  \[ Y(s) = C(s) = \left( s^2 + 5s + 20 \right) X_1(s) \]
- Taking inverse LT we find
  \[ y(t) = \ddot{x}_1 + 5 \dot{x}_1 + 20 x_1 \]
  or
  \[ y(t) = 20 x_1 + 5 x_2 + x_3 \]
Example TF to SS

So

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\dot{x}_3
\end{bmatrix} =
\begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
-20 & -32 & -13
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3
\end{bmatrix} +
\begin{bmatrix}
0 \\
0 \\
1
\end{bmatrix} r
\]

\[
y = \begin{bmatrix}
20 & 5 & 1
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3
\end{bmatrix}
\]
Example TF to SS

- The equivalent block diagram is now
Example Transfer Function

- Matlab provides tools for converting between State Space and Transfer Function representations for SISO systems
- `tf2ss` – converts from a transfer function to the state space
- `ss2tf` – converts from state space to a transfer function
State Space Example

• Recall this system from Assignment 1.
• We found the system was described by:
  \[ m_1 \ddot{y}_1 + k_d \dot{y}_1 + k_s y_1 = k_d \dot{y}_2 + k_s y_2 \]
  \[ m_2 \ddot{y}_2 + k_d \dot{y}_2 + (k_s + k_w) y_2 = k_d \dot{y}_1 + k_s y_1 + k_w u \]
• This yields the transfer function:
  \[
  \frac{Y_1(s)}{U(s)} = \frac{(k_d s + k_s) k_w}{m_1 m_2 s^4 + (m_1 + m_2) k_d s^3 + \left[ (k_s + k_w) m_1 + k_s m_2 \right] s^2 + k_d k_w s + k_s k_w}
  \]
State Space Example

- We will select the state variables:

\[
x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} y_1 \\ \dot{y}_1 \\ y_2 \\ \dot{y}_2 \end{bmatrix}
\]

- First, recast the system equations:

\[
\ddot{y}_1 = \frac{k_d}{m_1} (\dot{y}_2 - \dot{y}_1) + \frac{k_s}{m_1} (y_2 - y_1)
\]

\[
\ddot{y}_2 = \frac{k_d}{m_2} (\dot{y}_1 - \dot{y}_2) + \frac{k_s}{m_2} (y_1 - y_2) + \frac{k_w}{m_2} (u - y_2)
\]
State Space Example

- In matrix form:

$$\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\dot{x}_3 \\
\dot{x}_4
\end{bmatrix} =
\begin{bmatrix}
\ddot{y}_1 \\
\ddot{y}_1 \\
\ddot{y}_2 \\
\ddot{y}_2
\end{bmatrix} =
\begin{bmatrix}
0 & 1 & 0 & 0 \\
-k_s & -k_d & k_s & k_d \\
m_1 & m_1 & m_1 & m_1 \\
0 & 0 & 0 & 1 \\
k_s & k_d & -(k_s + k_w) & k_d \\
m_2 & m_2 & m_2 & m_2
\end{bmatrix}
\begin{bmatrix}
y_1 \\
y_1 \\
y_2 \\
y_2
\end{bmatrix} +
\begin{bmatrix}
0 \\
0 \\
k_w \\
m_2
\end{bmatrix}u$$

- This is a system of 4 first order differential equations.
State Space Example

- If we’re interested in the position of car over time:

\[
\gamma = \begin{bmatrix}
1 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\gamma_1 \\
\dot{\gamma}_1 \\
\gamma_2 \\
\dot{\gamma}_2
\end{bmatrix}
\]

- We might also want to find the motion of both masses:

\[
\begin{bmatrix}
\gamma_1 \\
\gamma_2
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
\gamma_1 \\
\dot{\gamma}_1 \\
\gamma_2 \\
\dot{\gamma}_2
\end{bmatrix}
\]
Higher Order Poles

• Classical controller designs are based on selecting the location of the dominant second order pole locations

• Higher order poles are difficult to set explicitly
Higher Order Poles: Example

• For the following plant

\[ G(s) = \frac{1}{s(s + 2)(s + 10)} \]

• Design a PD controller to yield an overshoot of 5% and a settling time of 1s
Higher Order Poles: Example

• Consider the design of a PD controller to achieve the required specifications

• Desired root locations are at

\[ s = -4 \pm 4.08j \]

\[ \theta_z - \theta_1 - \theta_2 - \theta_3 = (2k + 1)180^\circ \]

\[ \theta_z = 134.4 + 116.1 + 34.2 - 180 \]

\[ \theta_z = 104.7^\circ \]

\[ \therefore z_c = 2.93 \quad \text{With } K=44.7 \]
Higher Order Poles: Example

Root Locus

Step Response

Dr Ian R. Manchester
Amme 3500 : State Space
Slide 42
Higher Order Poles

• As can be seen, the position of the higher order poles can have a significant impact on the actual system behaviour

• Classical techniques do not easily afford us with a means of choosing the position of higher order poles
State Space Modelling

• Recall that for the state space approach, we represent a system by a set of $n$ first-order differential equations:

$$\dot{x} = Ax + Bu$$

• The output of the system is expressed as:

$$y = Cx + [Du]$$

- $x$ - state vector
- $y$ - output vector
- $u$ - input vector
- $A$ - state matrix
- $B$ - input matrix
- $C$ - output matrix
- $D$ - feedthrough or feedforward matrix (often zero)
State Feedback Control

- We can represent a general state space system as a Block Diagram.
- If we feedback the state variables, we end up with $n$ controllable parameters.
- State feedback with the control input

$$u = -Kx + r$$

State Space Control

- Setting \( u = -Kx + r \) yields
  \[
  \dot{x}(t) = Ax(t) + B(-Kx(t) + r)
  \]
  \[
y(t) = Cx(t)
  \]

- Rearranging the state equation and taking LT yields
  \[
  sX(s) = (A - BK)X(s) + BR(s)
  \]
  \[
  (sI - (A - BK))X(s) = BR(s)
  \]
  \[
  \frac{X(s)}{R(s)} = \left(sI - (A - BK)\right)^{-1} B
  \]

- Essentially we select values of \( K \) so that the eigenvalues (root locations) of \((A - BK)\) are at a particular location. How?
THEOREM: The following statements are equivalent:

1. For any time $T > 0$ there exists a control signal $u(t)$ driving the state from any initial state $x(0)$ to any final state $x(T)$.

2. For any choice of closed loop pole locations, there exists a constant gain matrix $K$ such that $u = -Kx$ has the desired poles.

3. The matrix $[B \ AB A^2B \ldots A^{n-1}B]$ is full rank
Controllability

- In State Space systems, we can check for controllability using the controllability matrix

\[ R = \begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix} \]

- If \( R \) has rank less than \( n \) then the system is not controlllable

- Why up to \( n-1 \)?
Controllability Example

• Returning to our suspended mass example, we find

\[
R = \begin{bmatrix} B & AB \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{M} \\ \frac{1}{M} & -\frac{K_d}{M^2} \end{bmatrix}
\]

• This has rank 2 (the rows are linearly independent) so the system is controllable

• This means that for any given position and velocity, we can move the system to another position and velocity using the input force \( f \)
Controllability Example II

• Consider a simple system that consists of a point mass moving on a frictionless surface.

\[ \sum F_x = M\dot{x}(t) \]
\[ \sum F_y = M\dot{y}(t) \]
Controllability Example II

• For this system, the state equations can be described by

\[
\begin{bmatrix}
\dot{x} \\
\ddot{x} \\
\dot{y} \\
\ddot{y}
\end{bmatrix} =
\begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
x \\
\dot{x} \\
y \\
\dot{y}
\end{bmatrix} +
\begin{bmatrix}
0 & 0 \\
1 & 0 \\
m & 0 \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
F_x \\
F_y
\end{bmatrix}
\]

• The rank of the controllability matrix is 4 so this system is controllable
Controllability Example II

- If we remove $F_y$ we find

$$\mathbf{x}' = \begin{bmatrix} \dot{x} \\ \dot{\dot{x}} \\ \dot{y} \\ \dot{\dot{y}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{m} \\ 0 \end{bmatrix}$$

- The rank of the controllability matrix is now 2 so this system is no longer controllable.
Pole Placement

• Given a system of the form

\[ A = \begin{bmatrix}
0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
-a_0 & -a_1 & -a_2 & \cdots & -a_{n-1}
\end{bmatrix}, \quad B = \begin{bmatrix}
0 \\
0 \\
\vdots \\
1
\end{bmatrix}, \quad C = \begin{bmatrix}
c_1 \\
c_2 \\
\vdots \\
c_n
\end{bmatrix} \]

• We can find the characteristic equation by solving for \( \det(sI-(A-BK)) \)
Pole Placement

• Using $u=-Kx$, we can find the system matrix $A-BK$

$$A - BK = \begin{bmatrix}
0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
-(a_0 + k_1) & -(a_1 + k_2) & -(a_2 + k_3) & \cdots & -(a_{n-1} + k_n)
\end{bmatrix}$$

• With characteristic equation of the form

$$\det(sI - (A - BK)) = s^n + (a_{n-1} + k_n)s^{n-1} + \cdots + (a_1 + k_2)s + (a_0 + k_1) = 0$$
Pole Placement

- Given a desired pole placement of
  \[ s^n + d_{n-1}s^{n-1} + \cdots + d_1s + d_0 = 0 \]
- and equating terms, we find
  \[ d_i = a_i + k_{i+1} \quad i = 0, 1, 2, \ldots, n - 1 \]
- or
  \[ k_{i+1} = d_i - a_i \]
- This will allow us to place all of the closed loop poles where we want them
Pole Placement

• Represent the plant in state space via the phase-variable form
• Feed back each phase variable to the input of the plant through a gain, $k_i$
• Find the characteristic equation of the closed-loop system
• Decide upon closed-loop pole locations and determine an equivalent characteristic equation
• Equate like coefficients of the characteristic equations and solve for $k_i$
Pole Placement Example

• For the following plant

\[ G(s) = \frac{1}{s(s+2)(s+10)} \]

• Design a full state feedback controller to yield an overshoot of 5% and a settling time of 1s
Pole Placement Example

• The OL transfer function for the system is

\[ G(s) = \frac{1}{s^3 + 12s^2 + 20s} \]

• Convert the transfer function to SS

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\dot{x}_3
\end{bmatrix} =
\begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
0 & -20 & -12
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3
\end{bmatrix}
+ \begin{bmatrix}
0 \\
0 \\
1
\end{bmatrix} u
\]

\[ y = \begin{bmatrix}
1 & 0 & 0
\end{bmatrix} x \]
Pole Placement Example

- Decide on desired root locations
- From specifications, we require
  - 5% OS and settling time of 1s
- Select dominant second order roots to be at
  \[ s = -4 \pm 4j \]
- Select third root 5 times to left at \[ s = -20 \]
- Desired characteristic equation:
  \[ s^3 + 28s^2 + 192s + 640 \]
Pole Placement Example

- Feeding back states we find

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\dot{x}_3
\end{bmatrix} =
\begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
0 & -20 & -12
\end{bmatrix}
- \begin{bmatrix}
0 \\
0 \\
K_1 & K_2 & K_3
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3
\end{bmatrix} + \begin{bmatrix}
0 \\
0 \\
1
\end{bmatrix}^r
\]
Pole Placement Example

- Solve for the characteristic equation

\[
\begin{align*}
  s^3 + (12 + K_3)s^2 + (20 + K_2)s + K_1 &= 0 \\
  s &\quad -1 &\quad 0 \\
  0 &\quad s &\quad -1 \\
  K_1 &\quad (20 + K_2) &\quad s + (12 + K_3)
\end{align*}
\]

\[
s^3 + (12 + K_3)s^2 + (20 + K_2)s + K_1 = 0
\]
Pole Placement Example

- Equating like terms we find
  \[ s^3 + 28s^2 + 192s + 640 \]
  \[ s^3 + (12 + K_3)s^2 + (20 + K_2)s + K_1 \]

- So

  \[ K_1 = 640 \]
  \[ K_2 = 172 \]
  \[ K_3 = 16 \]
Pole Placement Example

- Looks like we have met the overshoot and settling time requirements
- However, steady state error is not addressed here
State Space Example

• Consider for example the lunar lander

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass</td>
<td>15,000kg</td>
</tr>
<tr>
<td>Mass of Fuel</td>
<td>8,000kg</td>
</tr>
<tr>
<td>Moment of Inertia</td>
<td>100,000 kg m$^2$</td>
</tr>
<tr>
<td>Max. Ft propulsion</td>
<td>44kN</td>
</tr>
<tr>
<td>Max. Fl propulsion</td>
<td>0.5kN</td>
</tr>
<tr>
<td>Rocket Thruster Specific impulse</td>
<td>3.0kN s/kg</td>
</tr>
<tr>
<td>Gravitational Constant</td>
<td>1.6m/s$^2$</td>
</tr>
</tbody>
</table>
• From the FBD we can derive the system equations

\[ m\ddot{x} = F_l \cos \theta - F_l \sin \theta \]
\[ m\ddot{y} = F_l \sin \theta + F_l \cos \theta - mg \]
\[ J\ddot{\theta} = 4F_l \]
State Space Example

\[
\begin{bmatrix}
\dot{x} \\
\ddot{x} \\
\dot{y} \\
\ddot{y} \\
\dot{\theta} \\
\ddot{\theta}
\end{bmatrix} = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \begin{bmatrix}
x \\
\dot{x} \\
y \\
\dot{y} \\
\theta \\
\dot{\theta}
\end{bmatrix} + \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
\frac{m}{\cos \theta} & -\frac{m}{\sin \theta} & 0 & 0 & 0 & 0 \\
0 & \frac{m}{\cos \theta} & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{4}{J} & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{m}{\cos \theta} & -\frac{m}{\sin \theta} & 0 \\
0 & 0 & 0 & 0 & 0 & -1
\end{bmatrix} \begin{bmatrix}
F_x' \\
F_y' \\
g
\end{bmatrix}
\]

\[
y = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
x \\
\dot{x} \\
y \\
\dot{y} \\
\theta \\
\dot{\theta}
\end{bmatrix}
\]
Advantages of State Space Models

• The examples given here have all related to Linear Time-Invariant (LTI) systems
• These can be analysed and understood using our classical control techniques
• However, state space methods are also applicable for non-linear and time-varying systems
Conclusions

• We have looked at techniques for modelling and controlling systems using State Space techniques.

• **Controllability** is a key property

• Next week we will consider the design of more advanced control systems using the state space methods
Further Reading

• Nise
  – Sections 3.1-3.7 and 12.1-12.8

• Franklin & Powell
  – Section 2.2 and 7.1-7.6