Notes adapted from MIT OpenCourseware
“Aeronautics and Astronautics: Feedback Control Systems”
Fall 2010 lecture series

SS Introduction

- State space model: a representation of the dynamics of an N^{th} order system as a first order differential equation in an N-vector, which is called the **state**.
  - Convert the N^{th} order differential equation that governs the dynamics into N first-order differential equations

- Classic example: second order mass-spring system

$$m\ddot{p} + c\dot{p} + kp = F$$

  - Let $x_1 = p$, then $x_2 = \dot{p} = \dot{x}_1$, and

$$\begin{aligned}\dot{x}_2 = \ddot{p} &= (F - c\dot{p} - kp)/m \\ &= (F - cx_2 - kx_1)/m\end{aligned}$$

$$\Rightarrow \begin{bmatrix} \dot{p} \\ \ddot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix} \begin{bmatrix} p \\ \dot{p} \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} u$$

  - Let $u = F$ and introduce the state

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} p \\ \dot{p} \end{bmatrix} \Rightarrow \dot{\mathbf{x}} = A\mathbf{x} + Bu$$

  - If the measured output of the system is the position, then we have that

$$y = p = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} p \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = C\mathbf{x}$$

- Most general continuous-time linear dynamical system has form

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

  where:

  - $t \in \mathbb{R}$ denotes time
  - $\mathbf{x}(t) \in \mathbb{R}^n$ is the state (vector)
  - $\mathbf{u}(t) \in \mathbb{R}^m$ is the input or control
  - $\mathbf{y}(t) \in \mathbb{R}^p$ is the output

  - $A(t) \in \mathbb{R}^{n\times n}$ is the dynamics matrix
  - $B(t) \in \mathbb{R}^{n\times m}$ is the input matrix
  - $C(t) \in \mathbb{R}^{p\times n}$ is the output or sensor matrix
  - $D(t) \in \mathbb{R}^{p\times m}$ is the feedthrough matrix

- Note that the plant dynamics can be time-varying.
- Also note that this is a multi-input / multi-output (MIMO) system.

- We will typically deal with the time-invariant case
  $\Rightarrow$ **Linear Time-Invariant (LTI)** state dynamics

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

  so that now A, B, C, D are constant and do not depend on t.

Basic Definitions

- **Linearity** – What is a linear dynamical system? A system G is linear with respect to its inputs and output

$$\mathbf{u}(t) \rightarrow \boxed{G(s)} \rightarrow \mathbf{y}(t)$$

iff superposition holds:

$$G(\alpha_1 \mathbf{u}_1 + \alpha_2 \mathbf{u}_2) = \alpha_1 G\mathbf{u}_1 + \alpha_2 G\mathbf{u}_2$$

So if $\mathbf{y}_1$ is the response of G to $\mathbf{u}_1$ ($\mathbf{y}_1 = G\mathbf{u}_1$), and $\mathbf{y}_2$ is the response of G to $\mathbf{u}_2$ ($\mathbf{y}_2 = G\mathbf{u}_2$), then the response to $\alpha_1\mathbf{u}_1 + \alpha_2\mathbf{u}_2$ is $\alpha_1\mathbf{y}_1 + \alpha_2\mathbf{y}_2$

- A system is said to be **time-invariant** if the relationship between the input and output is independent of time. So if the response to $\mathbf{u}(t)$ is $\mathbf{y}(t)$, then the response to $\mathbf{u}(t-t_0)$ is $\mathbf{y}(t-t_0)$

- Example: the system

$$\begin{aligned} \dot{x}(t) &= 3x(t) + u(t) \\ y(t) &= x(t) \end{aligned}$$

is LTI, but

$$\begin{aligned} \dot{x}(t) &= 3t\ x(t) + u(t) \\ y(t) &= x(t) \end{aligned}$$

is not.

  - A matrix of second system is a function of absolute time, so response to $u(t)$ will differ from response to $u(t-1)$.

- $\mathbf{x}(t)$ is called the **state of the system** at t because:
  - Future output depends only on current state and future input
  - Future output depends on past input only through current state
  - State summarizes effect of past inputs on future output – like the *memory of the system*

- **Example:** Rechargeable flashlight – the state is the *current state of charge* of the battery. If you know that state, then you do not need to know how that level of charge was achieved (assuming a perfect battery) to predict the future performance of the flashlight.
  - But to consider all nonlinear effects, you might also need to know how many cycles the battery has gone through
  - Key point is that you might expect a given linear model to accurately model the charge depletion behavior for a given number of cycles, but that model would typically change with the number cycles

Creating State-Space Models

- Most easily created from N^{th} order differential equations that describe the dynamics
  - This was the case done before.
  - Only issue is which set of states to use – there are many choices.

- Can be developed from transfer function model as well.
  - Much more on this later

- Problem is that we have restricted ourselves here to linear state space models, and almost all systems are nonlinear in real-life.
  - Can develop linear models from nonlinear system dynamics

Equilibrium Points

- Often have a nonlinear set of dynamics given by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$$

where $\mathbf{x}$ is once gain the state vector, $\mathbf{u}$ is the vector of inputs, and $\mathbf{f}(\cdot,\cdot)$ is a nonlinear vector function that describes the dynamics

- First step is to define the point about which the linearization will be performed.
  - Typically about **equilibrium points** – a point for which if the system starts there it will remain there for all future time.

- Characterized by setting the state derivative to zero:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = 0$$

  - Result is an algebraic set of equations that must be solved for both $\mathbf{x}_e$ and $\mathbf{u}_e$
  - Note that $\dot{\mathbf{x}}_e = 0$ and $\dot{\mathbf{u}}_e = 0$ by definition
  - Typically think of these nominal conditions $\mathbf{x}_e$, $\mathbf{u}_e$ as "set points" or "operating points" for the nonlinear system.

- Example – pendulum dynamics: $\ddot{\theta} + r\dot{\theta} + \frac{g}{l}\sin\theta = 0$ can be written in state space form as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -rx_2 - \frac{g}{l}\sin x_1 \end{bmatrix}$$

  - Setting $\mathbf{f}(\mathbf{x}, \mathbf{u}) = 0$ yields $x_2 = 0$ and $x_2 = -\frac{g}{rl}\sin x_1$, which implies that $x_1 = \theta = \{0, \pi\}$

Linearization

- Typically assume that the system is operating about some nominal state solution $\mathbf{x}_e$ (possibly requires a nominal input $\mathbf{u}_e$)
  - Then write the actual state as $\mathbf{x}(t) = \mathbf{x}_e + \delta\mathbf{x}(t)$ and the actual inputs as $\mathbf{u}(t) = \mathbf{u}_e + \delta\mathbf{u}(t)$
  - The "δ" is included to denote the fact that we expect the variations about the nominal to be "small"

- Can then develop the linearized equations by using the **Taylor series expansion** of $\mathbf{f}(\cdot,\cdot)$ about $\mathbf{x}_e$ and $\mathbf{u}_e$.

- Recall the vector equation $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$, each equation of which

$$\dot{x}_i = f_i(\mathbf{x}, \mathbf{u})$$

  can be expanded as

$$\begin{aligned} \frac{d}{dt}(x_{ei} + \delta x_i) &= f_i(\mathbf{x}_e + \delta\mathbf{x}, \mathbf{u}_e + \delta\mathbf{u}) \\ &\approx f_i(\mathbf{x}_e, \mathbf{u}_e) + \left.\frac{\partial f_i}{\partial \mathbf{x}}\right|_0 \delta\mathbf{x} + \left.\frac{\partial f_i}{\partial \mathbf{u}}\right|_0 \delta\mathbf{u} \end{aligned}$$

  where

$$\frac{\partial f_i}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_i}{\partial x_1} & \cdots & \frac{\partial f_i}{\partial x_n} \end{bmatrix}$$

  and $\cdot|_0$ means that we should evaluate the function at the nominal values of $\mathbf{x}_e$ and $\mathbf{u}_e$.

- The meaning of "small" deviations now clear – the variations in $\delta\mathbf{x}$ and $\delta\mathbf{u}$ must be small enough that we can ignore the higher order terms in the Taylor expansion of $\mathbf{f}(\mathbf{x}, \mathbf{u})$.

- Since $\frac{d}{dt}x_{ei} = f_i(\mathbf{x}_e, \mathbf{u}_e)$, we thus have that

$$\frac{d}{dt}(\delta x_i) \approx \left.\frac{\partial f_i}{\partial \mathbf{x}}\right|_0 \delta\mathbf{x} + \left.\frac{\partial f_i}{\partial \mathbf{u}}\right|_0 \delta\mathbf{u}$$

- Combining for all n state equations, gives (note that we also set "$\approx$" $\rightarrow$ "$=$") that

$$\begin{aligned}\frac{d}{dt}\delta\mathbf{x} &= \begin{bmatrix} \left.\frac{\partial f_1}{\partial \mathbf{x}}\right|_0 \\ \left.\frac{\partial f_2}{\partial \mathbf{x}}\right|_0 \\ \vdots \\ \left.\frac{\partial f_n}{\partial \mathbf{x}}\right|_0 \end{bmatrix} \delta\mathbf{x} + \begin{bmatrix} \left.\frac{\partial f_1}{\partial \mathbf{u}}\right|_0 \\ \left.\frac{\partial f_2}{\partial \mathbf{u}}\right|_0 \\ \vdots \\ \left.\frac{\partial f_n}{\partial \mathbf{u}}\right|_0 \end{bmatrix} \delta\mathbf{u} \\ &= A(t)\delta\mathbf{x} + B(t)\delta\mathbf{u}\end{aligned}$$

where

$$A(t) \equiv \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ & & \vdots & \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_0 \quad \text{and} \quad B(t) \equiv \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \cdots & \frac{\partial f_2}{\partial u_m} \\ & & \vdots & \\ \frac{\partial f_n}{\partial u_1} & \frac{\partial f_n}{\partial u_2} & \cdots & \frac{\partial f_n}{\partial u_m} \end{bmatrix}_0$$

- Similarly, if the nonlinear measurement equation is $\mathbf{y} = \mathbf{g}(\mathbf{x}, \mathbf{u})$ and $\mathbf{y}(t) = \mathbf{y}_e + \delta\mathbf{y}$, then

$$\begin{aligned}
\delta\mathbf{y} &= \begin{bmatrix} \left.\frac{\partial g_1}{\partial \mathbf{x}}\right|_0 \\ \vdots \\ \left.\frac{\partial g_p}{\partial \mathbf{x}}\right|_0 \end{bmatrix} \delta\mathbf{x} + \begin{bmatrix} \left.\frac{\partial g_1}{\partial \mathbf{u}}\right|_0 \\ \vdots \\ \left.\frac{\partial g_p}{\partial \mathbf{u}}\right|_0 \end{bmatrix} \delta\mathbf{u} \\
&= C(t)\delta\mathbf{x} + D(t)\delta\mathbf{u}
\end{aligned}$$

- Typically drop the "δ" as they are rather cumbersome, and (abusing notation) we write the state equations as:

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

  which is of the same form as the previous linear models

- If the system is operating around just one set point then the partial fractions in the expressions for A–D are all constant $\rightarrow$ **LTI linearized model.**

Linearization Example

- **Example:** simple spring. With a mass at the end of a linear spring (rate k) we have the dynamics

$$m\ddot{x} = -kx$$

but with a "leaf spring" as is used on car suspensions, we have a nonlinear spring – the more it deflects, the stiffer it gets. Good model now is

$$m\ddot{x} = -k_1 x - k_2 x^3$$

which is a "cubic spring".

Fig. 1: Leaf spring from http://en.wikipedia.org/wiki/Image:Leafs1.jpg

- Restoring force depends on deflection x in a nonlinear way.

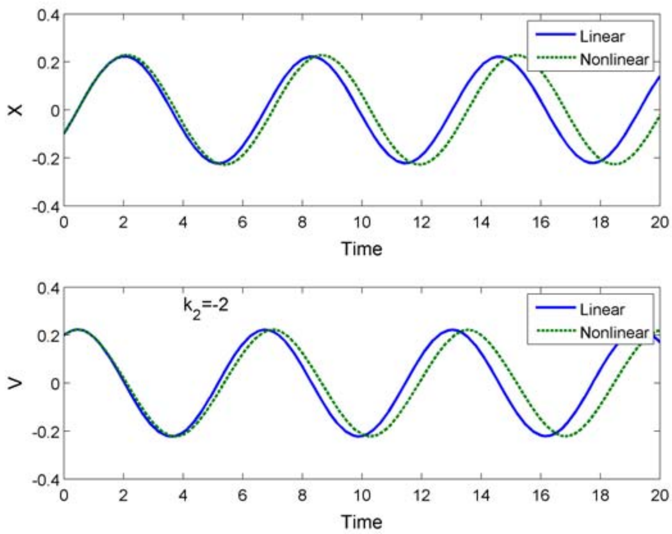


Fig. 2: Response to linear $k = 1$ and nonlinear $(k_1 = k, k_2 = -2)$ springs (code at the end)

- Consider the nonlinear spring with (set $m = 1$)

$$\ddot{y} = -k_1 y - k_2 y^3$$

gives us the nonlinear model ($x_1 = y$ and $x_2 = \dot{y}$)

$$\frac{d}{dt}\begin{bmatrix} y \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \dot{y} \\ -k_1 y - k_2 y^3 \end{bmatrix} \Rightarrow \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$

- Find the equilibrium points and then make a state space model

- For the equilibrium points, we must solve

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} \dot{y} \\ -k_1 y - k_2 y^3 \end{bmatrix} = 0$$

which gives

$$\dot{y}_e = 0 \quad \text{and} \quad k_1 y_e + k_2 (y_e)^3 = 0$$

  - Second condition corresponds to $y_e = 0$ or $y_e = \pm\sqrt{-k_1/k_2}$, which is only real if k_1 and k_2 are opposite signs.

- For the state space model,

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}_0 = \begin{bmatrix} 0 & 1 \\ -k_1 - 3k_2 (y)^2 & 0 \end{bmatrix}_0$$

$$= \begin{bmatrix} 0 & 1 \\ -k_1 - 3k_2 (y_e)^2 & 0 \end{bmatrix}$$

and the linearized model is $\dot{\delta\mathbf{x}} = A\delta\mathbf{x}$

- For the equilibrium point $y_e = 0$, $\dot{y}_e = 0$

$$A_0 = \begin{bmatrix} 0 & 1 \\ -k_1 & 0 \end{bmatrix}$$

which are the standard dynamics of a system with **just** a linear spring of stiffness k_1

  - Stable motion about $y = 0$ if $k_1 > 0$

- Assume that $k_1 = -1$, $k_2 = 1/2$, then we should get an equilibrium point at $\dot{y} = 0$, $y = \pm\sqrt{2}$, and since $k_1 + k_2(y_e)^2 = 0$ then

$$A_1 = \begin{bmatrix} 0 & 1 \\ -2 & 0 \end{bmatrix}$$

which are the dynamics of a stable oscillator about the equilibrium point

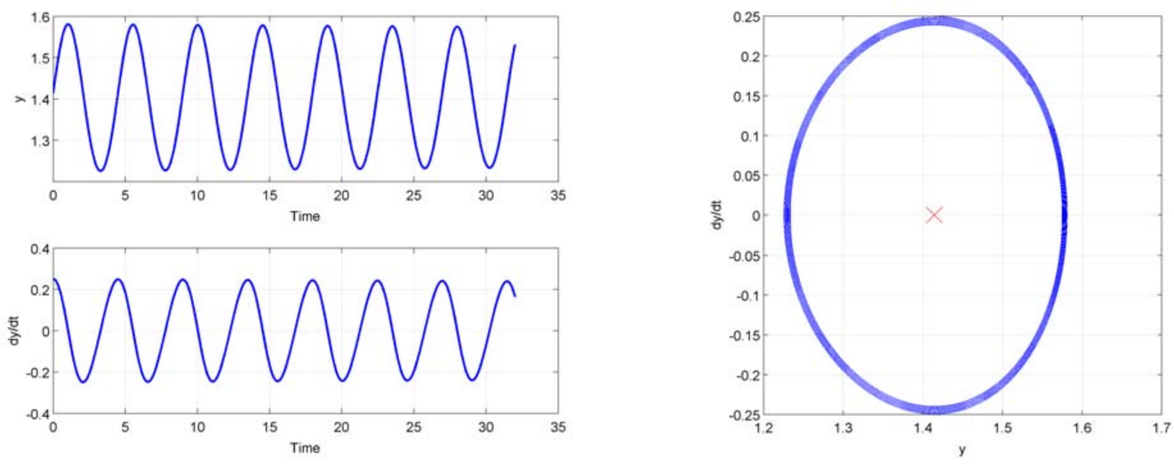


Fig. 3: Nonlinear response ($k_1 = -1, k_2 = 0.5$). The figure on the right shows the oscillation about the equilibrium point.

Time Response

- Can develop a lot of insight into the system response and how it is modeled by computing the time response $\mathbf{x}(t)$
  - Homogeneous part
  - Forced solution

- **Homogeneous Part**

$$\dot{\mathbf{x}} = A\mathbf{x}, \quad \mathbf{x}(0) \text{ known}$$

  - Take Laplace transform

$$X(s) = (sI - A)^{-1}\mathbf{x}(0)$$

  so that

$$\mathbf{x}(t) = \mathcal{L}^{-1}\left[(sI - A)^{-1}\right]\mathbf{x}(0)$$

  - But can show

$$\begin{aligned}(sI - A)^{-1} &= \frac{I}{s} + \frac{A}{s^2} + \frac{A^2}{s^3} + \dots \\ \text{so } \mathcal{L}^{-1}\left[(sI - A)^{-1}\right] &= I + At + \frac{1}{2!}(At)^2 + \dots \\ &= e^{At}\end{aligned}$$

$$\Rightarrow \mathbf{x}(t) = e^{At}\mathbf{x}(0)$$

- e^{At} is a special matrix that we will use many times in this course
  - *Transition matrix* or *Matrix Exponential*
  - Calculate in MATLAB using `expm.m` and not `exp.m` [1]
  - Note that $e^{(A+B)t} = e^{At}e^{Bt}$ iff $AB = BA$

[1] MATLAB is a trademark of the Mathworks Inc.

SS: Forced Solution

- **Forced Solution**
  - Consider a **scalar case:**

$$\begin{aligned}\dot{x} &= ax + bu, \quad x(0) \text{ given} \\ \Rightarrow x(t) &= e^{at}x(0) + \int_0^t e^{a(t-\tau)}bu(\tau)d\tau\end{aligned}$$

  where did this come from?

  1. $\dot{x} - ax = bu$
  2. $e^{-at}\left[\dot{x} - ax\right] = \frac{d}{dt}(e^{-at}x(t)) = e^{-at}bu(t)$
  3. $\int_0^t \frac{d}{d\tau}e^{-a\tau}x(\tau)d\tau = e^{-at}x(t) - x(0) = \int_0^t e^{-a\tau}bu(\tau)d\tau$

- **Forced Solution – Matrix case:**

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$$

where $\mathbf{x}$ is an n-vector and $\mathbf{u}$ is a m-vector

- Just follow the same steps as above to get

$$\mathbf{x}(t) = e^{At}\mathbf{x}(0) + \int_0^t e^{A(t-\tau)}B\mathbf{u}(\tau)d\tau$$

and if $\mathbf{y} = C\mathbf{x} + D\mathbf{u}$, then

$$\mathbf{y}(t) = Ce^{At}\mathbf{x}(0) + \int_0^t Ce^{A(t-\tau)}B\mathbf{u}(\tau)d\tau + D\mathbf{u}(t)$$

  - $Ce^{At}\mathbf{x}(0)$ is the initial response
  - $Ce^{A(t)}B$ is the impulse response of the system.

- Have seen the key role of e^{At} in the solution for $\mathbf{x}(t)$
  - Determines the system time response
  - But would like to get more insight!
- Consider what happens if the matrix A is diagonalizable, i.e. there exists a T such that

$$T^{-1}AT = \Lambda \text{ which is diagonal } \Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}$$

Then

$$e^{At} = Te^{\Lambda t}T^{-1}$$

where

$$e^{\Lambda t} = \begin{bmatrix} e^{\lambda_1 t} & & \\ & \ddots & \\ & & e^{\lambda_n t} \end{bmatrix}$$

- Follows since $e^{At} = I + At + \frac{1}{2!}(At)^2 + \ldots$ and that $A = T\Lambda T^{-1}$, so we can show that

$$\begin{aligned} e^{At} &= I + At + \frac{1}{2!}(At)^2 + \ldots \\ &= I + T\Lambda T^{-1}t + \frac{1}{2!}(T\Lambda T^{-1}t)^2 + \ldots \\ &= Te^{\Lambda t}T^{-1} \end{aligned}$$

- This is a simpler way to get the matrix exponential, but how find T and λ?
  - Eigenvalues and Eigenvectors

Dynamic Interpretation

- Since $A = T\Lambda T^{-1}$, then

$$e^{At} = Te^{\Lambda t}T^{-1} = \begin{bmatrix} | & & | \\ v_1 & \cdots & v_n \\ | & & | \end{bmatrix} \begin{bmatrix} e^{\lambda_1 t} & & \\ & \ddots & \\ & & e^{\lambda_n t} \end{bmatrix} \begin{bmatrix} - & w_1^T & - \\ & \vdots & \\ - & w_n^T & - \end{bmatrix}$$

where we have written

$$T^{-1} = \begin{bmatrix} - & w_1^T & - \\ & \vdots & \\ - & w_n^T & - \end{bmatrix}$$

which is a column of rows.

- Multiply this expression out and we get that

$$e^{At} = \sum_{i=1}^{n} e^{\lambda_i t} v_i w_i^T$$

- Assume A diagonalizable, then $\dot{\mathbf{x}} = A\mathbf{x}$, $\mathbf{x}(0)$ given, has solution

$$\begin{aligned} \mathbf{x}(t) &= e^{At}\mathbf{x}(0) = Te^{\Lambda t}T^{-1}\mathbf{x}(0) \\ &= \sum_{i=1}^{n} e^{\lambda_i t} v_i \{w_i^T \mathbf{x}(0)\} \\ &= \sum_{i=1}^{n} e^{\lambda_i t} v_i \beta_i \end{aligned}$$

- State solution is **linear combination** of the system **modes** $v_i e^{\lambda_i t}$

$e^{\lambda_i t}$ – Determines **nature** of the time response

v_i – Determines how each state **contributes** to that mode

β_i – Determines extent to which initial condition **excites** the mode

- Note that the v_i give the relative sizing of the response of each part of the state vector to the response.

$$v_1(t) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} e^{-t} \quad \text{mode 1}$$

$$v_2(t) = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} e^{-3t} \quad \text{mode 2}$$

- Clearly $e^{\lambda_i t}$ gives the time modulation
  - λ_i real – growing/decaying exponential response
  - λ_i complex – growing/decaying exponential damped sinusoidal

- **Bottom line:** The locations of the eigenvalues determine the pole locations for the system, thus:
  - They determine the stability and/or performance & transient behavior of the system.
  - It is their locations that we will want to modify when we start the control work

EV Mechanics

- Consider $A = \begin{bmatrix} -1 & 1 \\ -8 & 5 \end{bmatrix}$

$$\begin{aligned} (sI - A) &= \begin{bmatrix} s+1 & -1 \\ 8 & s-5 \end{bmatrix} \\ \det(sI - A) &= (s+1)(s-5) + 8 = s^2 - 4s + 3 = 0 \end{aligned}$$

so the eigenvalues are $s_1 = 1$ and $s_2 = 3$

- Eigenvectors $(sI - A)v = 0$

$$(s_1 I - A)v_1 = \begin{bmatrix} s+1 & -1 \\ 8 & s-5 \end{bmatrix}_{s=1} \begin{bmatrix} v_{11} \\ v_{21} \end{bmatrix} = 0$$

$$\begin{bmatrix} 2 & -1 \\ 8 & -4 \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{21} \end{bmatrix} = 0 \qquad 2v_{11} - v_{21} = 0, \Rightarrow v_{21} = 2v_{11}$$

v_{11} is then arbitrary ($\neq 0$), so set $v_{11} = 1$

$$v_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$(s_2 I - A)v_2 = \begin{bmatrix} 4 & -1 \\ 8 & -2 \end{bmatrix} \begin{bmatrix} v_{12} \\ v_{22} \end{bmatrix} = 0 \qquad 4v_{12} - v_{22} = 0, \Rightarrow v_{22} = 4v_{12}$$

$$v_2 = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$$

- Confirm that $Av_i = \lambda_i v_i$

Stability of LTI Systems

- Consider a solution $\mathbf{x}_s(t)$ to a differential equation for a given initial condition $\mathbf{x}_s(t_0)$.
    - Solution is **stable** if other solutions $\mathbf{x}_b(t_0)$ that start near $\mathbf{x}_s(t_0)$ stay close to $\mathbf{x}_s(t)$ $\forall\, t \Rightarrow$ **stable in sense of Lyapunov** (SSL).
    - If other solutions are SSL, but the $\mathbf{x}_b(t)$ do not converge to $\mathbf{x}_s(t)$ $\Rightarrow$ solution is **neutrally stable**.
    - If other solutions are SSL and $\mathbf{x}_b(t) \to \mathbf{x}(t)$ as $t \to \infty \Rightarrow$ solution is **asymptotically stable**.
    - A solution $\mathbf{x}_s(t)$ is **unstable** if it is not stable.

- Note that a linear (autonomous) system $\dot{\mathbf{x}} = A\mathbf{x}$ has an equilibrium point at $\mathbf{x}_e = 0$
    - This equilibrium point is **stable** if and only if all of the eigenvalues of A satisfy $\mathbb{R}\lambda_i(A) \leq 0$ and every eigenvalue with $\mathbb{R}\lambda_i(A) = 0$ has a Jordan block of order one.[1]
    - Thus the stability test for a linear system is the familiar one of determining if $\mathbb{R}\lambda_i(A) \leq 0$

- Somewhat surprisingly perhaps, we can also infer stability of the original nonlinear from the analysis of the linearized system model

[1] more on Jordan blocks on 6–??, but this basically means that these eigenvalues are not repeated.

- **Lyapunov's indirect method**[2] Let $\mathbf{x}_e = 0$ be an equilibrium point for the nonlinear autonomous system

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t))$$

where $\mathbf{f}$ is continuously differentiable in a neighborhood of $\mathbf{x}_e$. Assume

$$A = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}_e}$$

Then:

  - The origin is an asymptotically stable equilibrium point for the nonlinear system if $\mathbb{R}\lambda_i(A) < 0 \;\forall\; i$
  - The origin is unstable if $\mathbb{R}\lambda_i(A) > 0$ for any i

- Note that this doesn't say anything about the stability of the nonlinear system if the linear system is neutrally stable.

- A very powerful result that is the basis of all linear control theory.

[2] Much more on Lyapunov methods later too.

State-Space Model Features

- There are some key characteristics of a state-space model that we need to identify.
  - Will see that these are very closely associated with the concepts of pole/zero cancelation in transfer functions.

- **Example:** Consider a simple system

$$G(s) = \frac{6}{s+2}$$

for which we develop the state-space model

$$\text{Model } \# \ 1 \qquad \begin{aligned} \dot{x} &= -2x + 2u \\ y &= 3x \end{aligned}$$

- But now consider the new state space model $\bar{\mathbf{x}} = [x \ \ x_2]^T$

$$\text{Model } \# \ 2 \qquad \begin{aligned} \dot{\bar{\mathbf{x}}} &= \begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix} \bar{\mathbf{x}} + \begin{bmatrix} 2 \\ 1 \end{bmatrix} u \\ y &= \begin{bmatrix} 3 & 0 \end{bmatrix} \bar{\mathbf{x}} \end{aligned}$$

which is clearly different than the first model, and larger.

- But let's looks at the transfer function of the new model:

$$\begin{aligned} \bar{G}(s) &= C(sI - A)^{-1}B + D \\ &= \begin{bmatrix} 3 & 0 \end{bmatrix} \left(sI - \begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 2 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 3 & 0 \end{bmatrix} \begin{bmatrix} \frac{2}{s+2} \\ \frac{1}{s+1} \end{bmatrix} = \frac{6}{s+2} \quad !! \end{aligned}$$

- This is a bit strange, because previously our figure of merit when comparing one state-space model to another (page 6–??) was whether they reproduced the same same transfer function
  - Now we have two very different models that result in the same transfer function
  - Note that I showed the second model as having 1 extra state, but I could easily have done it with 99 extra states!!

- So what is going on?
  - A clue is that the dynamics associated with the second state of the model x_2 were eliminated when we formed the product

$$\bar{G}(s) = \begin{bmatrix} 3 & 0 \end{bmatrix} \begin{bmatrix} \frac{2}{s+2} \\ \frac{1}{s+1} \end{bmatrix}$$

  because the A is decoupled and there is a zero in the C matrix
  - Which is exactly the same as saying that there is a **pole-zero cancelation** in the transfer function $\tilde{G}(s)$

$$\frac{6}{s+2} = \frac{6(s+1)}{(s+2)(s+1)} \triangleq \tilde{G}(s)$$

  - Note that model #2 is one possible state-space model of $\tilde{G}(s)$ (has 2 poles)

- For this system we say that the dynamics associated with the second state are **unobservable** using this sensor (defines C matrix).
  - There could be a lot "motion" associated with x_2, but we would be unaware of it using this sensor.

- There is an analogous problem on the input side as well. Consider:

$$\text{Model \# 1} \qquad \begin{aligned} \dot{x} &= -2x + 2u \\ y &= 3x \end{aligned}$$

with $\bar{\mathbf{x}} = \left[\, x \quad x_2 \,\right]^T$

$$\text{Model \# 3} \qquad \begin{aligned} \dot{\bar{\mathbf{x}}} &= \begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix} \bar{\mathbf{x}} + \begin{bmatrix} 2 \\ 0 \end{bmatrix} u \\ y &= \begin{bmatrix} 3 & 2 \end{bmatrix} \bar{\mathbf{x}} \end{aligned}$$

which is also **clearly different** than model #1, and has a different form from the second model.

$$\begin{aligned} \hat{G}(s) &= \begin{bmatrix} 3 & 2 \end{bmatrix} \left(sI - \begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 2 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \frac{3}{s+2} & \frac{2}{s+1} \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \frac{6}{s+2} \quad !! \end{aligned}$$

- Once again the dynamics associated with the pole at $s = -1$ are canceled out of the transfer function.
  - But in this case it occurred because there is a 0 in the B matrix

- So in this case we can "see" the state x_2 in the output $C = \begin{bmatrix} 3 & 2 \end{bmatrix}$, but we cannot "influence" that state with the input since

$$B = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

- So we say that the dynamics associated with the second state are **uncontrollable** using this actuator (defines the B matrix).

- Of course it can get even worse because we could have

$$\begin{aligned}\dot{\bar{\mathbf{x}}} &= \begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix} \bar{\mathbf{x}} + \begin{bmatrix} 2 \\ 0 \end{bmatrix} u \\ y &= \begin{bmatrix} 3 & 0 \end{bmatrix} \bar{\mathbf{x}}\end{aligned}$$

- So now we have

$$\begin{aligned}\widetilde{G(s)} &= \begin{bmatrix} 3 & 0 \end{bmatrix} \left(sI - \begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 2 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \frac{3}{s+2} & \frac{0}{s+1} \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \frac{6}{s+2} \quad !!\end{aligned}$$

- Get same result for the transfer function, but now the dynamics associated with x_2 are both unobservable and uncontrollable.

- **Summary:** Dynamics in the state-space model that are **uncontrollable**, **unobservable**, or **both** do not show up in the transfer function.

- Would like to develop models that **only have** dynamics that are both **controllable** and **observable**
  $\Rrightarrow$ called a **minimal realization**
  - A state space model that has the lowest possible order for the given transfer function.

- But first need to develop tests to determine if the models are observable and/or controllable

Observability

- **Definition:** An LTI **system** is **observable** if the initial state $\mathbf{x}(0)$ can be **uniquely deduced** from the knowledge of the input $\mathbf{u}(t)$ and output $\mathbf{y}(t)$ for all t between 0 and any finite $T > 0$.
  - If $\mathbf{x}(0)$ can be deduced, then we can reconstruct $\mathbf{x}(t)$ exactly because we know $\mathbf{u}(t) \Rrightarrow$ we can find $\mathbf{x}(t)\ \forall\ t$.
  - Thus we need only consider the zero-input (homogeneous) solution to study observability.

$$\mathbf{y}(t) = Ce^{At}\mathbf{x}(0)$$

- This definition of observability is consistent with the notion we used before of being able to "see" all the states in the output of the decoupled examples
  - ROT: For those decoupled examples, if part of the state cannot be "seen" in $\mathbf{y}(t)$, then it would be impossible to deduce that part of $\mathbf{x}(0)$ from the outputs $\mathbf{y}(t)$.

- **Definition:** A **state** $\mathbf{x}^\star \neq 0$ is said to be **unobservable** if the zero-input solution $\mathbf{y}(t)$, with $\mathbf{x}(0) = \mathbf{x}^\star$, is zero for all $t \geq 0$
  - Equivalent to saying that $\mathbf{x}^\star$ is an unobservable state if

$$Ce^{At}\mathbf{x}^\star = 0 \;\forall\; t \;\geq 0$$

- For the problem we were just looking at, consider Model #2 with $\mathbf{x}^\star = [\begin{array}{cc} 0 & 1 \end{array}]^T \neq 0$, then

$$\begin{aligned} \dot{\bar{\mathbf{x}}} &= \begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix} \bar{\mathbf{x}} + \begin{bmatrix} 2 \\ 1 \end{bmatrix} u \\ y &= \begin{bmatrix} 3 & 0 \end{bmatrix} \bar{\mathbf{x}} \end{aligned}$$

so

$$\begin{aligned} Ce^{At}\mathbf{x}^\star &= \begin{bmatrix} 3 & 0 \end{bmatrix} \begin{bmatrix} e^{-2t} & 0 \\ 0 & e^{-t} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 3e^{-2t} & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0 \;\forall\; t \end{aligned}$$

So, $\mathbf{x}^\star = [\begin{array}{cc} 0 & 1 \end{array}]^T$ is an unobservable state for this system.

- But that is as expected, because we knew there was a problem with the state x_2 from the previous analysis

- **Theorem: An LTI system is observable iff it has no unobservable states.**
  - We normally just say that the **pair (A,C) is observable.**

- **Pseudo-Proof:** Let $\mathbf{x}^\star \neq 0$ be an unobservable state and compute the outputs from the initial conditions $\mathbf{x}_1(0)$ and $\mathbf{x}_2(0) = \mathbf{x}_1(0) + \mathbf{x}^\star$
  - Then
$$\mathbf{y}_1(t) = Ce^{At}\mathbf{x}_1(0) \quad \text{and} \quad \mathbf{y}_2(t) = Ce^{At}\mathbf{x}_2(0)$$
but
$$\begin{aligned} \mathbf{y}_2(t) &= Ce^{At}(\mathbf{x}_1(0) + \mathbf{x}^\star) = Ce^{At}\mathbf{x}_1(0) + Ce^{At}\mathbf{x}^\star \\ &= Ce^{At}\mathbf{x}_1(0) = \mathbf{y}_1(t) \end{aligned}$$
  - Thus 2 different initial conditions give the same output $\mathbf{y}(t)$, so it would be impossible for us to deduce the actual initial condition of the system $\mathbf{x}_1(t)$ or $\mathbf{x}_2(t)$ given $\mathbf{y}_1(t)$

- Testing system observability by searching for a vector $\mathbf{x}(0)$ such that $Ce^{At}\mathbf{x}(0) = 0 \ \forall \ t$ is feasible, but very hard in general.
  - Better tests are available.

- **Theorem: The vector $\mathrm{x}^\star$ is an unobservable state iff**

$$\begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \mathbf{x}^\star = 0$$

- **Pseudo-Proof:** If $\mathbf{x}^\star$ is an unobservable state, then by definition,

$$Ce^{At}\mathbf{x}^\star = 0 \quad \forall\, t \geq 0$$

But all the derivatives of Ce^{At} exist and for this condition to hold, all derivatives must be zero at $t = 0$. Then

$$Ce^{At}\mathbf{x}^\star\big|_{t=0} = 0 \;\Rightarrow\; C\mathbf{x}^\star = 0$$

$$\frac{d}{dt}Ce^{At}\mathbf{x}^\star\bigg|_{t=0} = 0 \;\Rightarrow\; CAe^{At}\mathbf{x}^\star\big|_{t=0} = CA\mathbf{x}^\star = 0$$

$$\frac{d^2}{dt^2}Ce^{At}\mathbf{x}^\star\bigg|_{t=0} = 0 \;\Rightarrow\; CA^2e^{At}\mathbf{x}^\star\big|_{t=0} = CA^2\mathbf{x}^\star = 0$$

$$\vdots$$

$$\frac{d^k}{dt^k}Ce^{At}\mathbf{x}^\star\bigg|_{t=0} = 0 \;\Rightarrow\; CA^ke^{At}\mathbf{x}^\star\big|_{t=0} = CA^k\mathbf{x}^\star = 0$$

- We only need retain up to the $n-1^{\text{th}}$ derivative because of the *Cayley-Hamilton* theorem.

- **Simple test:** Necessary and sufficient condition for observability is that

$$\texttt{rank}\ \mathcal{M}_o \triangleq \texttt{rank} \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} = n$$

- Why does this make sense?
  - The requirement for an unobservable state is that for $\mathbf{x}^\star \neq 0$

$$\mathcal{M}_o \mathbf{x}^\star = 0$$

  - Which is equivalent to saying that $\mathbf{x}^\star$ is orthogonal to each row of $\mathcal{M}_o$.
  - But if the rows of $\mathcal{M}_o$ are considered to be vectors and these **span the full n-dimensional space**, then it is not possible to find an n-vector $\mathbf{x}^\star$ that is orthogonal to each of these.
  - To determine if the n rows of $\mathcal{M}_o$ span the full n-dimensional space, we need to test their **linear independence**, which is equivalent to the `rank` test[1]

[1]Let M be a $m \times p$ matrix, then the `rank` of M satisfies:

1. `rank` $M \equiv$ number of linearly independent columns of M
2. `rank` $M \equiv$ number of linearly independent rows of M
3. `rank` $M \leq \min\{m, p\}$

Controllability

- **Definition:** An LTI **system** is **controllable** if, for every $\mathbf{x}^\star(t)$ and every finite $T > 0$, there exists an input function $\mathbf{u}(t)$, $0 < t \leq T$, such that the system state goes from $\mathbf{x}(0) = 0$ to $\mathbf{x}(T) = \mathbf{x}^\star$.

  - Starting at 0 is not a special case – if we can get to any state in finite time from the origin, then we can get from any initial condition to that state in finite time as well. [1]

- This definition of controllability is consistent with the notion we used before of being able to "influence" all the states in the system in the decoupled examples (page 9–??).

  - ROT: For those decoupled examples, if part of the state cannot be "influenced" by $\mathbf{u}(t)$, then it would be impossible to move that part of the state from 0 to $\mathbf{x}^\star$

- Need only consider the forced solution to study controllability.

$$\mathbf{x}_f(t) = \int_0^t e^{A(t-\tau)} B\mathbf{u}(\tau) d\tau$$

  - Change of variables $\tau_2 = t - \tau$, $d\tau = -d\tau_2$ gives a form that is a little easier to work with:

$$\mathbf{x}_f(t) = \int_0^t e^{A\tau_2} B\mathbf{u}(t - \tau_2) d\tau_2$$

  - Assume system has m inputs.

[1] This controllability from the origin is often called **reachability**.

- Note that, regardless of the eigenstructure of A, the Cayley-Hamilton theorem gives

$$e^{At} = \sum_{i=0}^{n-1} A^i \alpha_i(t)$$

for some computable scalars $\alpha_i(t)$, so that

$$\mathbf{x}_f(t) = \sum_{i=0}^{n-1}(A^iB)\int_0^t \alpha_i(\tau_2)\mathbf{u}(t-\tau_2)d\tau_2 = \sum_{i=0}^{n-1}(A^iB)\boldsymbol{\beta}_i(t)$$

for coefficients $\boldsymbol{\beta}_i(t)$ that depend on the input $\mathbf{u}(\tau)$, $0 < \tau \leq t$.

- Result can be interpreted as meaning that the state $\mathbf{x}_f(t)$ is a linear combination of the nm vectors A^iB (with m inputs).
  - All linear combinations of these nm vectors is the *range space* of the matrix formed from the A^iB column vectors:

$$\mathcal{M}_c = \left[\, B \;\; AB \;\; A^2B \;\; \cdots \;\; A^{n-1}B \,\right]$$

- **Definition:** Range space of M_c is **controllable subspace** of the system
  - If a state $\mathbf{x}_c(t)$ is not in the range space of M_c, it is not a linear combination of these columns $\Rightarrow$ it is impossible for $\mathbf{x}_f(t)$ to ever equal $\mathbf{x}_c(t)$ – called **uncontrollable state.**

- **Theorem: LTI system is controllable iff it has no uncontrollable states.**
  - Necessary and sufficient condition for controllability is that

$$\texttt{rank}\;\mathcal{M}_c \triangleq \texttt{rank}\left[\, B \;\; AB \;\; A^2B \;\; \cdots \;\; A^{n-1}B \,\right] = n$$

Further Examples

- With Model # 2:

$$
\begin{aligned}
\dot{\bar{\mathbf{x}}} &= \begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix} \bar{\mathbf{x}} + \begin{bmatrix} 2 \\ 1 \end{bmatrix} u \\
y &= \begin{bmatrix} 3 & 0 \end{bmatrix} \bar{\mathbf{x}} \\
\mathcal{M}_0 &= \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ -6 & 0 \end{bmatrix} \\
\mathcal{M}_c &= \begin{bmatrix} B & AB \end{bmatrix} = \begin{bmatrix} 2 & -4 \\ 1 & -1 \end{bmatrix}
\end{aligned}
$$

  - rank $\mathcal{M}_0 = 1$ and rank $\mathcal{M}_c = 2$
  - So this model of the system is controllable, but not observable.

- With Model # 3:

$$
\begin{aligned}
\dot{\bar{\mathbf{x}}} &= \begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix} \bar{\mathbf{x}} + \begin{bmatrix} 2 \\ 0 \end{bmatrix} u \\
y &= \begin{bmatrix} 3 & 2 \end{bmatrix} \bar{\mathbf{x}} \\
\mathcal{M}_0 &= \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ -6 & -2 \end{bmatrix} \\
\mathcal{M}_c &= \begin{bmatrix} B & AB \end{bmatrix} = \begin{bmatrix} 2 & -4 \\ 0 & 0 \end{bmatrix}
\end{aligned}
$$

  - rank $\mathcal{M}_0 = 2$ and rank $\mathcal{M}_c = 1$
  - So this model of the system is observable, but not controllable.

- Note that controllability/observability are **not** intrinsic properties of a system. Whether the model has them or not depends on the representation that you choose.
  - But they indicate that something else more fundamental is wrong. . .

Weaker Conditions

- Often it is too much to assume that we will have full observability and controllability. Often have to make do with the following. System called:
  - **Detectable** if all unstable modes are **observable**
  - **Stabilizable** if all unstable modes are **controllable**

- So if you had a stabilizable and detectable system, there could be dynamics that you are not aware of and cannot influence, but you know that they are at least stable.

- That is enough information on the system model for now – will assume minimal models from here on and start looking at the control issues.

Full-state Feedback Controller

- Assume that the single-input system dynamics are given by

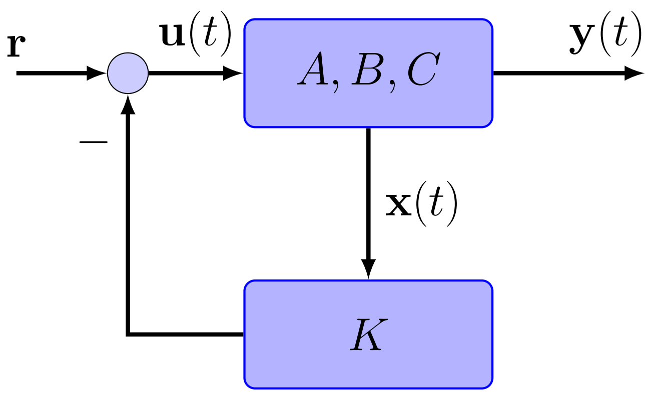
$$\begin{aligned}\dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + B\mathbf{u}(t)\\ \mathbf{y}(t) &= C\mathbf{x}(t)\end{aligned}$$

  so that $D = 0$.

  - The multi-actuator case is quite a bit more complicated as we would have many extra degrees of freedom.

- Recall that the system poles are given by the eigenvalues of A.

  - Want to use the input $\mathbf{u}(t)$ to modify the eigenvalues of A to change the system dynamics.



- Assume a full-state feedback of the form:

$$\mathbf{u}(t) = \mathbf{r} - K\mathbf{x}(t)$$

  where $\mathbf{r}$ is some **reference input** and the **gain** K is $\mathbb{R}^{1\times n}$

  - If $\mathbf{r} = 0$, we call this controller a **regulator**

- Find the closed-loop dynamics:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + B(\mathbf{r} - K\mathbf{x}(t))\\ &= (A - BK)\mathbf{x}(t) + B\mathbf{r}\\ &= A_{cl}\mathbf{x}(t) + B\mathbf{r}\\ \mathbf{y}(t) &= C\mathbf{x}(t)\end{aligned}$$

- **Objective:** Pick K so that A_{cl} has the desired properties, *e.g.*,
  - A unstable, want A_{cl} stable
  - Put 2 poles at $-2 \pm 2\mathbf{i}$
- Note that there are n parameters in K and n eigenvalues in A, so it looks promising, but what can we achieve?

- **Example #1:** Consider:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u$$

  - Then $\det(sI - A) = (s-1)(s-2) - 1 = s^2 - 3s + 1 = 0$ so the system is unstable.

  - Define $u = -\begin{bmatrix} k_1 & k_2 \end{bmatrix} \mathbf{x}(t) = -K\mathbf{x}(t)$, then

$$\begin{aligned} A_{cl} = A - BK &= \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \end{bmatrix} \\ &= \begin{bmatrix} 1-k_1 & 1-k_2 \\ 1 & 2 \end{bmatrix} \end{aligned}$$

    which gives

$$\det(sI - A_{cl}) = s^2 + (k_1 - 3)s + (1 - 2k_1 + k_2) = 0$$

  - Thus, by choosing k_1 and k_2, we can put $\lambda_i(A_{cl})$ anywhere in the complex plane (assuming complex conjugate pairs of poles).

- To put the poles at $s = -5, \ -6$, compare the *desired characteristic equation*

$$(s+5)(s+6) = s^2 + 11s + 30 = 0$$

with the closed-loop one

$$s^2 + (k_1 - 3)s + (1 - 2k_1 + k_2) = 0$$

to conclude that

$$\left.\begin{array}{r} k_1 - 3 = 11 \\ 1 - 2k_1 + k_2 = 30 \end{array}\right\} \quad \begin{array}{l} k_1 = 14 \\ k_2 = 57 \end{array}$$

so that $K = \left[\ 14 \quad 57\ \right]$, which is called **Pole Placement**.

- Of course, it is not always this easy, as lack of **controllability** might be an issue.

- **Example #2:** Consider this system:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u$$

with the same control approach

$$A_{cl} = A - BK = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix} \left[\ k_1 \quad k_2\ \right] = \begin{bmatrix} 1 - k_1 & 1 - k_2 \\ 0 & 2 \end{bmatrix}$$

so that

$$\det(sI - A_{cl}) = (s - 1 + k_1)(s - 2) = 0$$

So the feedback control can modify the pole at $s = 1$, but it cannot move the pole at $s = 2$.

- **System cannot be stabilized with full-state feedback.**
- Problem caused by a lack of controllability of the e^{2t} mode.

- Consider the basic controllability test:

$$\mathcal{M}_c = \left[\, B \,\middle|\, AB \,\right] = \left[\begin{bmatrix} 1 \\ 0 \end{bmatrix} \middle| \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right]$$

So that $\texttt{rank}\ \mathcal{M}_c = 1 < 2$.

- **Modal analysis** of controllability to develop a little more insight

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix}, \quad \text{decompose as} \quad AV = V\Lambda \quad \Rightarrow \Lambda = V^{-1}AV$$

where

$$\Lambda = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \qquad V = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad V^{-1} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}$$

Convert

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + Bu \quad \xrightarrow{z=V^{-1}\mathbf{x}(t)} \quad \dot{z} = \Lambda z + V^{-1}Bu$$

where $z = \left[\, z_1 \ \ z_2 \,\right]^T$. But:

$$V^{-1}B = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix}$$

so that the dynamics in modal form are:

$$\dot{z} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} z + \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} u$$

- With this zero in the modal B-matrix, can easily see that the mode associated with the z_2 state is **uncontrollable.**
  - **Must assume that the pair $(A,\ B)$ are controllable.**

Ackermann's Formula

- The previous outlined a design procedure and showed how to do it by hand for second-order systems.
  - Extends to higher order (controllable) systems, but tedious.

- **Ackermann's Formula** gives us a method of doing this entire design process is one easy step.

$$K = \left[\begin{array}{cccc} 0 & \ldots & 0 & 1 \end{array}\right] \mathcal{M}_c^{-1} \Phi_d(A)$$

  - $\mathcal{M}_c = \left[\begin{array}{cccc} B & AB & \ldots & A^{n-1}B \end{array}\right]$ as before
  - $\Phi_d(s)$ is the characteristic equation for the closed-loop poles, which we then evaluate for $s = A$.
  - Note: is explicit that the **system must be controllable** because we are inverting the controllability matrix.

- Revisit **Example # 1:** $\Phi_d(s) = s^2 + 11s + 30$

$$\mathcal{M}_c = \left[\begin{array}{c|c} B & AB \end{array}\right] = \left[\begin{array}{c|c} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{array}\right] = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

So

$$\begin{aligned} K &= \left[\begin{array}{cc} 0 & 1 \end{array}\right] \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}^{-1} \left(\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}^2 + 11 \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} + 30I \right) \\ &= \left[\begin{array}{cc} 0 & 1 \end{array}\right] \left(\begin{bmatrix} 43 & 14 \\ 14 & 57 \end{bmatrix} \right) = \left[\begin{array}{cc} 14 & 57 \end{array}\right] \end{aligned}$$

- Automated in Matlab: `place.m` & `acker.m` (see `polyvalm.m` too)

Reference Inputs

- So far we have looked at how to pick K to get the dynamics to have some nice properties (*i.e.* stabilize A)

- The question remains as to how well this controller allows us to track a reference command?
  - Performance issue rather than just stability.

- Started with

$$\begin{aligned}\dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + Bu \qquad y = C\mathbf{x}(t)\\ u &= r - K\mathbf{x}(t)\end{aligned}$$

- For **good tracking performance** we want

$$y(t) \approx r(t) \text{ as } t \rightarrow \infty$$

- Consider this performance issue in the frequency domain. Use the final value theorem:

$$\lim_{t\rightarrow\infty} y(t) = \lim_{s\rightarrow 0} sY(s)$$

  Thus, for good performance, we want

$$sY(s) \approx sR(s) \text{ as } s \rightarrow 0 \quad \Rightarrow \quad \left.\frac{Y(s)}{R(s)}\right|_{s=0} = 1$$

- So, for good performance, the transfer function from $R(s)$ to $Y(s)$ should be approximately 1 at DC.

- **Example #1 continued:** For the system

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}(t)\end{aligned}$$

  - Already designed $K = \begin{bmatrix} 14 & 57 \end{bmatrix}$ so the closed-loop system is

$$\begin{aligned}\dot{\mathbf{x}}(t) &= (A - BK)\mathbf{x}(t) + Br \\ y &= C\mathbf{x}(t)\end{aligned}$$

  which gives the transfer function

$$\begin{aligned}\frac{Y(s)}{R(s)} &= C\left(sI - (A - BK)\right)^{-1} B \\ &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} s+13 & 56 \\ -1 & s-2 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ &= \frac{s-2}{s^2 + 11s + 30}\end{aligned}$$

- Assume that $r(t)$ is a step, then by the FVT

$$\left.\frac{Y(s)}{R(s)}\right|_{s=0} = -\frac{2}{30} \neq 1 \text{ !!}$$

  - So our step response is quite poor!

- One solution is to scale the reference input $r(t)$ so that

$$u = \overline{N}r - K\mathbf{x}(t)$$

  - $\overline{N}$ extra gain used to scale the closed-loop transfer function

- Now we have

$$\begin{aligned} \dot{\mathbf{x}}(t) &= (A - BK)\mathbf{x}(t) + B\overline{N}r \\ y &= C\mathbf{x}(t) \end{aligned}$$

so that

$$\frac{Y(s)}{R(s)} = C\left(sI - (A - BK)\right)^{-1} B\overline{N} = G_{cl}(s)\overline{N}$$

If we had made $\overline{N} = -15$, then

$$\frac{Y(s)}{R(s)} = \frac{-15(s-2)}{s^2 + 11s + 30}$$

so with a step input, $y(t) \to 1$ as $t \to \infty$.

- Clearly can compute

$$\overline{N} = G_{cl}(0)^{-1} = -\left(C(A - BK)^{-1}B\right)^{-1}$$

- Note that this development assumed that r was constant, but it could also be used if r is a slowly time-varying command.

Pole Placement Approach

- So far we have looked at how to pick K to get the dynamics to have some nice properties (*i.e.* stabilize A)

$$\lambda_i(A) \leadsto \lambda_i(A - BK)$$

- **Question:** where should we put the closed-loop poles?

- **Approach #1:** use time-domain specifications to locate dominant poles – roots of:

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$$

  - Then place rest of the poles so they are "much faster" than the dominant 2nd order behavior.
  - Example: could keep the same damped frequency w_d and then move the real part to be 2–3 times faster than the real part of dominant poles $\zeta\omega_n$
    - Just be careful moving the poles too far to the left because it takes a lot of control effort

- Recall ROT for 2nd order response (4–??):

10-90% rise time	$t_r = \dfrac{1 + 1.1\zeta + 1.4\zeta^2}{\omega_n}$
Settling time (5%)	$t_s = \dfrac{3}{\zeta\omega_n}$
Time to peak amplitude	$t_p = \dfrac{\pi}{\omega_n\sqrt{1-\zeta^2}}$
Peak overshoot	$M_p = e^{-\zeta\omega_n t_p}$

- **Key difference** in this case: since all poles are being placed, the assumption of dominant 2nd order behavior is pretty much guaranteed to be valid.

Linear Quadratic Regulator

- **Approach #2:** is to place the pole locations so that the closed-loop system optimizes the cost function

$$J_{LQR} = \int_0^\infty \left[\mathbf{x}(t)^T Q \mathbf{x}(t) + \mathbf{u}(t)^T R \mathbf{u}(t)\right] dt$$

where:

  - $\mathbf{x}^T Q \mathbf{x}$ is the **State Cost** with weight Q
  - $\mathbf{u}^T R \mathbf{u}$ is called the **Control Cost** with weight R
  - Basic form of **Linear Quadratic Regulator** problem.

- MIMO optimal control is a time invariant linear state feedback

$$\mathbf{u}(t) = -K_{\text{lqr}} \mathbf{x}(t)$$

and K_{lqr} found by solving **Algebraic Riccati Equation** (ARE)

$$\begin{aligned} 0 &= A^T P + PA + Q - PBR^{-1}B^T P \\ K_{\text{lqr}} &= R^{-1} B^T P \end{aligned}$$

  - Some details to follow, but discussed at length in 16.323

- **Note:** state cost written using output $\mathbf{x}^T Q \mathbf{x}$, but could define a system output of interest $\mathbf{z} = C_z \mathbf{x}$ that is not based on a physical sensor measurement and use cost ftn:

$$\Rightarrow \quad J_{LQR} = \int_0^\infty \left[\mathbf{x}^T(t) C_z^T \tilde{Q} C_z \mathbf{x}(t) + \rho\, \mathbf{u}(t)^T \mathbf{u}(t)\right] dt$$

  - Then effectively have state penalty $Q = (C_z^T \tilde{Q} C_z)$
  - Selection of $\mathbf{z}$ used to isolate system states of particular interest that you would like to be regulated to "zero".
  - $R = \rho I$ effectively sets the controller bandwidth

LQR Weight Matrix Selection

- Good ROT (typically called Bryson's Rules) when selecting the weighting matrices Q and R is to normalize the signals:

$$Q = \begin{bmatrix} \frac{\alpha_1^2}{(x_1)^2_{\max}} & & & \\ & \frac{\alpha_2^2}{(x_2)^2_{\max}} & & \\ & & \ddots & \\ & & & \frac{\alpha_n^2}{(x_n)^2_{\max}} \end{bmatrix}$$

$$R = \rho \begin{bmatrix} \frac{\beta_1^2}{(u_1)^2_{\max}} & & & \\ & \frac{\beta_2^2}{(u_2)^2_{\max}} & & \\ & & \ddots & \\ & & & \frac{\beta_m^2}{(u_m)^2_{\max}} \end{bmatrix}$$

- The $(x_i)_{\max}$ and $(u_i)_{\max}$ represent the largest desired response or control input for that component of the state/actuator signal.

- $\sum_i \alpha_i^2 = 1$ and $\sum_i \beta_i^2 = 1$ are used to add an additional relative weighting on the various components of the state/control

- ρ is used as the last relative weighting between the control and state penalties $\Rightarrow$ gives a relatively concrete way to discuss the relative size of Q and R and their ratio Q/R

Regulator Summary

- Dominant second order approach places the closed-loop pole locations **with no regard to the amount of control effort required.**
  - Designer must iterate on the selected bandwidth (ω_n) to ensure that the control effort is reasonable.

- LQR selects closed-loop poles that **balance** between state errors and control effort.
  - Easy design iteration using R
  - Sometimes difficult to relate the desired transient response to the LQR cost function.
  - Key thing is that the designer is focused on system performance issues rather than the mechanics of the design process

Regulator/Estimator Comparison

- **Regulator Problem:**
  - Concerned with controllability of (A, B)

    For a controllable system we can place the eigenvalues of $A - BK$ arbitrarily.
  - Choose $K \in \mathbb{R}^{1\times n}$ (SISO) such that the closed-loop poles

    $$\det(sI - A + BK) = \Phi_c(s)$$

    are in the desired locations.

- **Estimator Problem:**
  - For estimation, were concerned with observability of pair (A, C).

    For a observable system we can place the eigenvalues of $A - LC$ arbitrarily.
  - Choose $L \in \mathbb{R}^{n\times 1}$ (SISO) such that the closed-loop poles

    $$\det(sI - A + LC) = \Phi_o(s)$$

    are in the desired locations.

- These problems are obviously very similar – in fact they are called **dual problems**.