STATE-SPACE DYNAMIC SYSTEMS (CONTINUOUS-TIME)

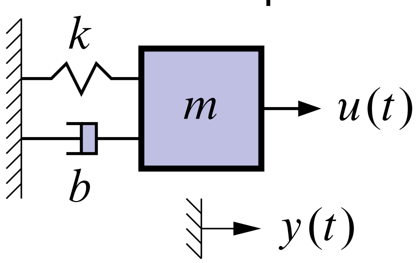
2.1: Introduction to LTI state-space models

1. What are they?
2. Why use them?
3. How are they related to the transfer functions we have used already?
4. How do we formulate them?

"Nice artwork kiddo. I have a feeling that a great many people will make a living off that third line someday!" (*Out of Control*, IEEE Control Systems Magazine)

What are they?

- Representation of dynamics of an nth-order system as a first-order differential equation in an n-vector called the <u>state</u> ➠ n first-order equations.
- Classic example: 2nd-order equation of motion (EOM).



$$m\ddot{y}(t) = u(t) - b\dot{y}(t) - ky(t)$$

$$\Rightarrow \quad \ddot{y}(t) = \frac{u(t) - b\dot{y}(t) - ky(t)}{m}.$$

- Define a state vector:

$$x(t) = \begin{bmatrix} y(t) \\ \dot{y}(t) \end{bmatrix}$$

then, $$\dot{x}(t) = \begin{bmatrix} \dot{y}(t) \\ \ddot{y}(t) \end{bmatrix} = \begin{bmatrix} \dot{y}(t) \\ -\frac{k}{m}y(t) - \frac{b}{m}\dot{y}(t) + \frac{1}{m}u(t) \end{bmatrix}.$$

- We can write this in the form $\dot{x}(t) = Ax(t) + Bu(t)$, where A and B are constant matrices:

$$\dot{x}(t) = \begin{bmatrix} \dot{y}(t) \\ \ddot{y}(t) \end{bmatrix} = \underbrace{\begin{bmatrix} \quad\quad\quad & \\ & \end{bmatrix}}_{A} \begin{bmatrix} y(t) \\ \dot{y}(t) \end{bmatrix} + \underbrace{\begin{bmatrix} \quad\quad \\ \end{bmatrix}}_{B} u(t).$$

- Complete picture by finding $y(t)$ as function of $x(t)$. General form is

$$y(t) = Cx(t) + Du(t),$$

where C and D are constant matrices:

$$C = \begin{bmatrix} \quad\quad\quad \end{bmatrix}, \qquad D = \begin{bmatrix} \quad\quad \end{bmatrix}.$$

- Fundamental form for LTI state-space model:

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t),$$

where $u(t)$ is the input, $x(t)$ is the "state", A, B, C, D are constant matrices.

DEFINITION: The state of a system at time t_0 is the minimum amount of information at t_0 that, together with the input $u(t)$, $t \geq t_0$, uniquely determines the behavior of the system for all $t \geq t_0$.

- Contrast with impulse-response (convolution) representation which requires all past history of $u(t)$

$$y(t) = \int_0^t h(\tau)u(t-\tau)\,\mathrm{d}\tau.$$

Why use them?

- Transfer functions provide input-output mapping: $u \to G(s) \to y$. State variables provide access to what is going on *inside* the system.

- Convenient way to express EOM. Matrix format great for computers.
- Allows new analysis and synthesis tools.
- *Great* for multi-input, multi-output systems. These are very hard to work with transfer functions.

Converting state-space to transfer function

- Start with the state equations

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t)\\ y(t) &= Cx(t) + Du(t).\end{aligned}$$

- Laplace transform

$$\begin{aligned}sX(s) - x(0) &= AX(s) + BU(s)\\ Y(s) &= CX(s) + DU(s),\end{aligned}$$

or

$$\begin{aligned}(sI - A)X(s) &= BU(s) + x(0)\\ X(s) &= (sI - A)^{-1}BU(s) + (sI - A)^{-1}x(0)\end{aligned}$$

and

$$Y(s) = \underbrace{\left[C(sI - A)^{-1}B + D\right]}_{\text{transfer function of system}} U(s) + \underbrace{C(sI - A)^{-1}x(0)}_{\text{response to initial conditions}}.$$

- So,

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

but

$$(sI - A)^{-1} = \frac{\text{adj}(sI - A)}{\det(sI - A)} = \frac{[\text{cofactor array of } (sI - A)]^T}{\det(sI - A)}.$$

- Slightly easier to compute (for SISO systems)

$$\frac{Y(s)}{U(s)} = C(sI-A)^{-1}B + D = \frac{\det\left[\begin{array}{c:c} sI-A & B \\ \hdashline -C & D \end{array}\right]}{\det(sI-A)}.$$

  - We will develop this result later in this chapter when defining transmission zeros of a state-space system.

EXAMPLE: Our mass-spring-damper example:

$$A = \begin{bmatrix} 0 & 1 \\ -\dfrac{k}{m} & -\dfrac{b}{m} \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ \dfrac{1}{m} \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

- The transfer function is:

$$G(s) = C(sI-A)^{-1}B + 0 = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} s & -1 \\ \dfrac{k}{m} & s+\dfrac{b}{m} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \dfrac{1}{m} \end{bmatrix}$$

$$= \frac{\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} s+\dfrac{b}{m} & 1 \\ -\dfrac{k}{m} & s \end{bmatrix} \begin{bmatrix} 0 \\ \dfrac{1}{m} \end{bmatrix}}{s^2 + (b/m)s + (k/m)} = \frac{\frac{1}{m}}{s^2 + \frac{b}{m}s + \frac{k}{m}} = \frac{1}{ms^2 + bs + k}.$$

- This is exactly what we expect from the first example in this section.

EXAMPLE: Using the special SISO formula,

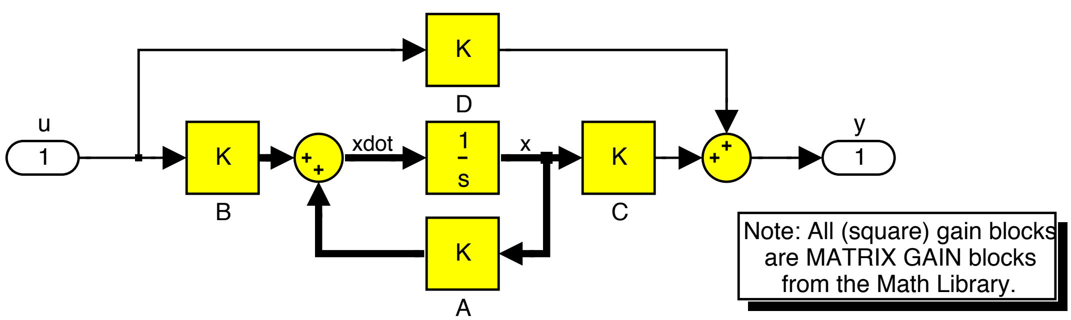
$$G(s) = \frac{\det\left[\begin{array}{cc:c} s & -1 & 0 \\ \dfrac{k}{m} & s+\dfrac{b}{m} & \dfrac{1}{m} \\ \hdashline -1 & 0 & 0 \end{array}\right]}{\det\begin{bmatrix} s & -1 \\ \dfrac{k}{m} & s+\dfrac{b}{m} \end{bmatrix}} = \frac{1/m}{s^2 + (b/m)s + (k/m)} = \frac{1}{ms^2 + bs + k}.$$

- Same result.
- Example shows that the characteristic equation for the system is

$$\chi(s) = \det(sI - A) = 0.$$

- Poles of system are roots of $\det(sI - A) = 0$ (eigenvalues).
- In transfer function matrix form, $G(s) = C(sI - A)^{-1}B + D$, a pole of *any* entry in $G(s)$ is a pole of the system.

SIMULATING SYSTEMS IN SIMULINK: To investigate how state-space systems work, we can simulate them in Simulink. We could use the "State Space" block from the "Continuous" library, or we can make our own. The following method has advantages because it gives us explicit access to the state and other internal signals. It is a direct implementation of the transfer function above, and the initial state may be set by setting the initial integrator values.



2.2: Four canonical forms for LTI state-space models

- We can make state-space forms from EOM, as we have seen.
- Also from transfer functions: there are four main standardized forms, plus a couple of other forms we will look at later.

Controller canonical form

- Three cases:

1] Transfer function is made up only of poles.

$$G(s) = \frac{1}{s^3 + a_1 s^2 + a_2 s + a_3} = \frac{Y(s)}{U(s)}$$

$$\Rightarrow \dddot{y}(t) + a_1 \ddot{y}(t) + a_2 \dot{y}(t) + a_3 y(t) = u(t).$$

- Choose output and derivatives as the state. $x(t) = \begin{bmatrix} \ddot{y}(t) & \dot{y}(t) & y(t) \end{bmatrix}^T$. Then

$$\dot{x}(t) = \begin{bmatrix} \dddot{y}(t) \\ \ddot{y}(t) \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} -a_1 & -a_2 & -a_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \ddot{y}(t) \\ \dot{y}(t) \\ y(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \end{bmatrix} u(t).$$

- Note the special form of A (<u>top-companion matrix</u>).

2] Transfer function has poles and zeros, but is strictly proper.

$$G(s) = \frac{b_1 s^2 + b_2 s + b_3}{s^3 + a_1 s^2 + a_2 s + a_3} = \frac{Y(s)}{U(s)}.$$

- Break up transfer function into two parts. $\dfrac{V(s)}{U(s)}$ contains all of the poles of $\dfrac{Y(s)}{U(s)}$. Then,

$$Y(s) = [b_1 s^2 + b_2 s + b_3]V(s).$$

- Or,

$$y(t) = b_1 \ddot{v}(t) + b_2 \dot{v}(t) + b_3 v(t).$$

- But,

$$V(s)[s^3 + a_1 s^2 + a_2 s + a_3] = U(s),$$

or,

$$\dddot{v}(t) + a_1 \ddot{v}(t) + a_2 \dot{v}(t) + a_3 v(t) = u(t).$$

- The representation for this is the same as in Case 1. Let

$$x(t) = \begin{bmatrix} \ddot{v}(t) & \dot{v}(t) & v(t) \end{bmatrix}^T.$$

- Then

$$\dot{x}(t) = \begin{bmatrix} \dddot{v}(t) \\ \ddot{v}(t) \\ \dot{v}(t) \end{bmatrix} = \begin{bmatrix} -a_1 & -a_2 & -a_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \ddot{v}(t) \\ \dot{v}(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u(t)$$

represents the dynamics of $v(t)$. All that remains is to couple in the zeros of the system.

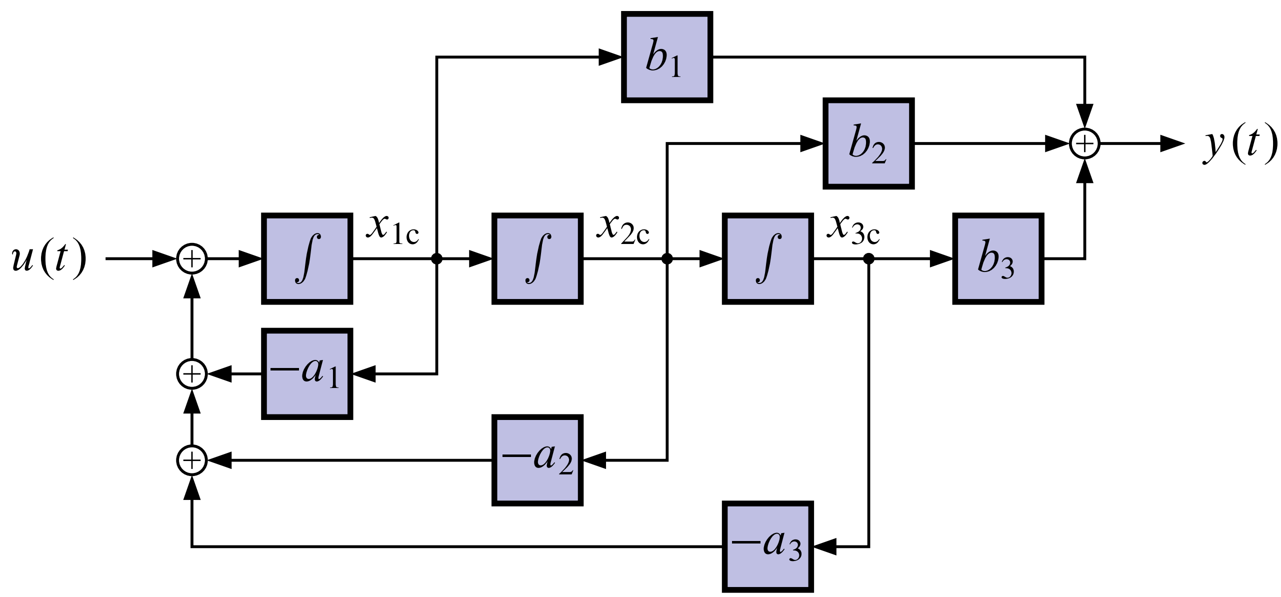
$$\begin{aligned} Y(s) &= [b_1 s^2 + b_2 s + b_3]V(s) \\ y(t) &= \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix} x(t) + \begin{bmatrix} 0 \end{bmatrix} u(t). \end{aligned}$$

3] Non-proper transfer function.

$$\begin{aligned} G(s) &= \frac{b_0 s^3 + b_1 s^2 + b_2 s + b_1}{s^3 + a_1 s^2 + a_2 s + a_3} \\ &= \frac{\beta_1 s^2 + \beta_2 s + \beta_3}{s^3 + a_1 s^2 + a_2 s + a_3} + D, \end{aligned}$$

where the β_i terms are computed via long division. The remainder D is the feedthrough term.

- This particular method of implementing a system in state-space form is called controller canonical form.
- MATLAB command `tf2ss(num,den)` converts a transfer-function form to state-space form.
- Analog computer implementation:



Observer canonical form

- Now, using the same transfer function,

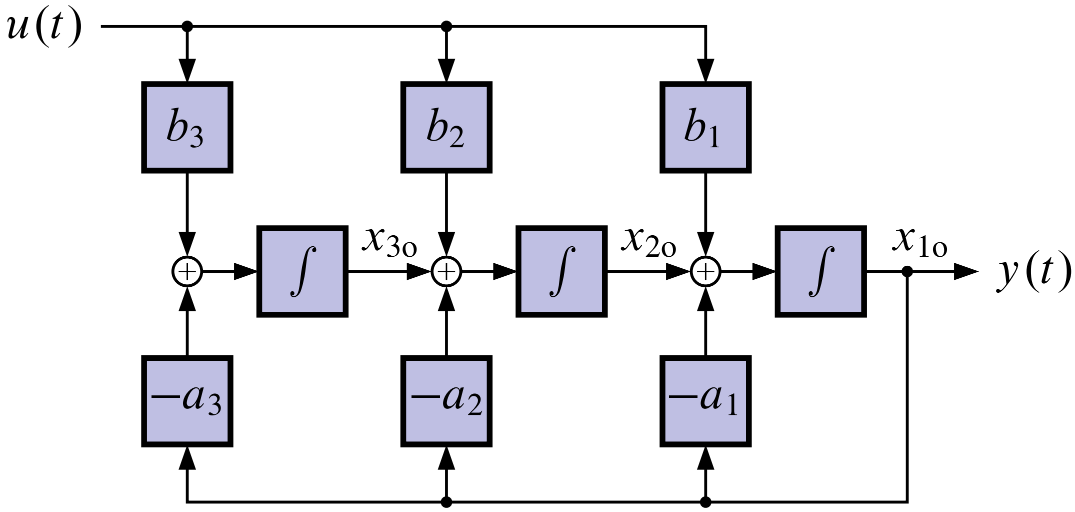
$$(s^3 + a_1 s^2 + a_2 s + a_3)Y(s) = (b_1 s^2 + b_2 s + b_3)U(s),$$

divide both sides by s^3

$$Y(s) = \left(-\frac{a_1}{s} - \frac{a_2}{s^2} - \frac{a_3}{s^3}\right)Y(s) + \left(\frac{b_1}{s} + \frac{b_2}{s^2} + \frac{b_3}{s^3}\right)U(s)$$

$$= \frac{1}{s}\left(b_1U(s) - a_1Y(s) + \frac{1}{s}\left(b_2U(s) - a_2Y(s) + \frac{1}{s}\left(b_3U(s) - a_3Y(s)\right)\right)\right).$$

- This has block-diagram:



- This is called observer canonical form:

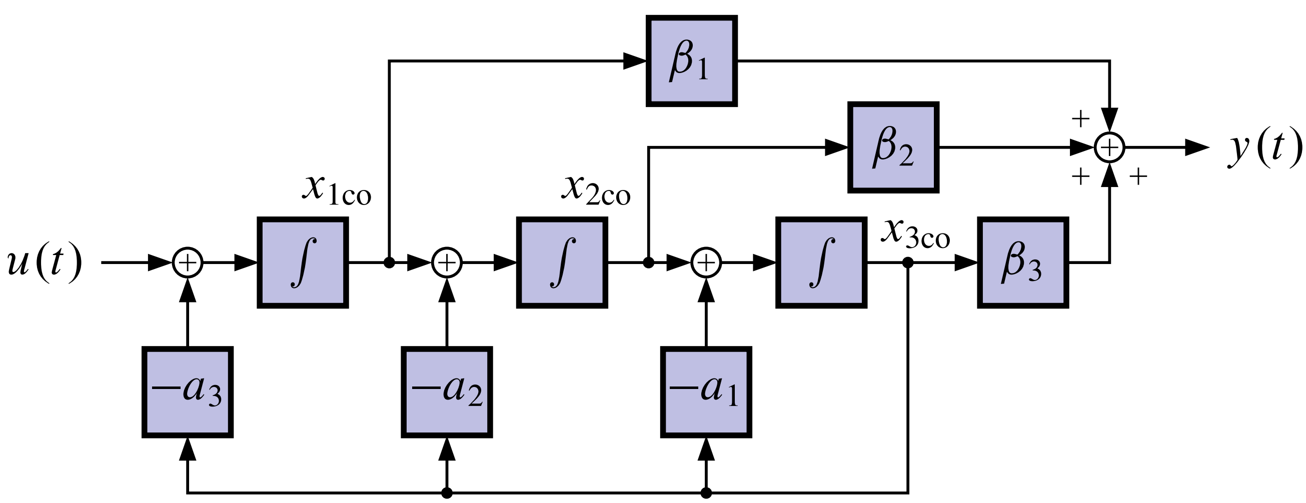
$$\dot{x}(t) = \begin{bmatrix} -a_1 & 1 & 0 \\ -a_2 & 0 & 1 \\ -a_3 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x(t).$$

- A is a left-companion matrix.

Controllability canonical form

- Third, consider the block diagram:



$$\begin{aligned} x_3 &= \frac{1}{s}(x_2 - a_1x_3) \\ x_2 &= \frac{1}{s}(x_1 - a_2x_3) \\ x_1 &= \frac{1}{s}(u - a_3x_3) \end{aligned} \quad \Rightarrow \quad \begin{aligned} X_3(s) &= \frac{1}{s^3 + a_1s^2 + a_2s + a_3}U(s) \\ X_2(s) &= \frac{s + a_1}{s^3 + a_1s^2 + a_2s + a_3}U(s) \\ X_1(s) &= \frac{s^2 + a_1s + a_2}{s^3 + a_1s^2 + a_2s + a_3}U(s). \end{aligned}$$

- Thus,

$$Y(s) = \frac{\beta_3 + \beta_2(s + a_1) + \beta_1(s^2 + a_1s + a_2)}{s^3 + a_1s^2 + a_2s + a_3}U(s).$$

- In order to get the correct transfer function, we must compute the $\{\beta_i\}$ values to get the desired numerator:

$$\begin{bmatrix} 1 & 0 & 0 \\ a_1 & 1 & 0 \\ a_2 & a_1 & 1 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$\begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ a_1 & 1 & 0 \\ a_2 & a_1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 - a_1b_1 \\ b_3 - a_1b_2 - a_2b_1 + a_1^2b_1 \end{bmatrix}.$$

- Or,

$$\dot{x}(t) = \begin{bmatrix} 0 & 0 & -a_3 \\ 1 & 0 & -a_2 \\ 0 & 1 & -a_1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} \beta_1 & \beta_2 & \beta_3 \end{bmatrix} x(t).$$

- A is a right-companion matrix.

Observability canonical form

- Note that $H(s)$ is a scalar. So, $H^T(s) = H(s)$.

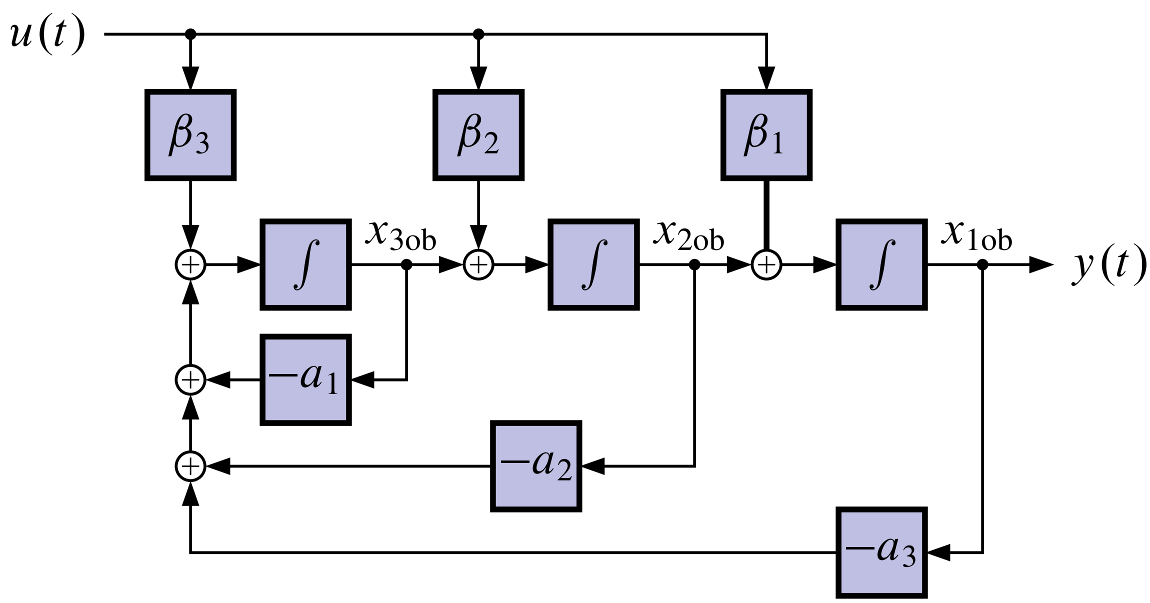
$$\begin{aligned} H(s) &= C(sI-A)^{-1}B+D \\ &= B^T(sI-A)^{-T}C^T+D^T \\ &= B^T(sI-A^T)^{-1}C^T+D^T. \end{aligned}$$

- So, $C \leftrightarrow B^T$, $A \leftrightarrow A^T$, $B \leftrightarrow C^T$ and $D \leftrightarrow D^T$ are dual forms.
- We have already seen this(!). Controller and observer are dual forms. Likewise, we can come up with

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_3 & -a_2 & -a_1 \end{bmatrix} x(t) + \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x(t)$$

as a dual form with the controllability form.



- A is a bottom-companion matrix.
- We will see that we have a lot of freedom when making our state-space models (*i.e.*, in choosing the components of $x(t)$).

2.3: One more canonical form, transformations

Modal (diagonal) form

- Yet another "canonical" form. Very useful...
- Assume $G(s) = \dfrac{N(s)}{D(s)}$, $\quad D(s)$ has distinct roots p_i (real).

$$\begin{aligned} G(s) &= \frac{N(s)}{(s-p_1)(s-p_2)\cdots(s-p_n)} \\ &= \frac{r_1}{s-p_1} + \frac{r_2}{s-p_2} + \cdots + \frac{r_n}{s-p_n}. \end{aligned}$$

- Now, let

$$\frac{X_1(s)}{U(s)} = \frac{r_1}{s-p_1} \quad ➠ \quad \dot{x}_1(t) = p_1 x_1(t) + r_1 u(t)$$

$$\vdots$$

$$\frac{X_n(s)}{U(s)} = \frac{r_n}{s-p_n} \quad ➠ \quad \dot{x}_n(t) = p_n x_n(t) + r_n u(t).$$

- Or,

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned}$$

$$A = \begin{bmatrix} p_1 & & & 0 \\ & p_2 & & \\ & & \ddots & \\ 0 & & & p_n \end{bmatrix}, \qquad B = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 \end{bmatrix}.$$

- Easily extends to handle complex poles $\lambda_i = \sigma_i + j\omega_i$.

- If A and B may have complex elements, no change is necessary.
- Otherwise, use "real modal form" which is made via partial-fraction expansion where complex pole-pairs are represented as

$$G_i(s) = \frac{\alpha_i s + \beta_i}{(s - \sigma_i)^2 + \omega_i^2}.$$

- The real-modal form has an A matrix which is block diagonal, and of the form

$$A = \text{diag}\left(\Lambda_r, \begin{bmatrix} \sigma_{r+1} & \omega_{r+1} \\ -\omega_{r+1} & \sigma_{r+1} \end{bmatrix}, \ldots, \begin{bmatrix} \sigma_n & \omega_n \\ -\omega_n & \sigma_n \end{bmatrix}\right)$$

where Λ_r is a diagonal matrix containing the real poles, and

$$\lambda_i = \sigma_i + j\omega_i, \qquad i = r+1, \ldots, n$$

are the complex poles.

- The B matrix has corresponding entries:

$$\begin{bmatrix} b_{i,1} \\ b_{i,2} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -\omega_i - \sigma_i & \omega_i - \sigma_i \end{bmatrix}^{-1} \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix}.$$

- Modal form is convenient for keeping track of system poles… they are right on the diagonal!
- Good representation to use… numerical robustness.
- *All canonical forms related by linear algebra—change of basis*.
- Diagonal is very useful, but we cannot always put a system in diagonal form. (What was our assumption above?)
- We will see one more canonical (Jordan) form in a little while that is very similar to diagonal. *All* systems can be put in Jordan form.

Transformations

- We have seen that state-space representations are not unique. Selection of vector components in x is quite arbitrary.
- Can we convert from one representation to another and get equivalent systems?
- Analyze the transformation of

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t).$$

- Let $x(t) = Tz(t)$, where T is an invertible (similarity) transformation matrix.

$$\begin{aligned}\dot{z}(t) &= T^{-1}\dot{x}(t)\\ &= T^{-1}[Ax(t) + Bu(t)]\\ &= T^{-1}[ATz(t) + Bu(t)]\\ &= \underbrace{T^{-1}AT}_{\bar{A}}z(t) + \underbrace{T^{-1}B}_{\bar{B}}u(t)\\ y(t) &= \underbrace{CT}_{\bar{C}}z(t) + \underbrace{D}_{\bar{D}}u(t)\end{aligned}$$

$$\text{so } \dot{z}(t) = \bar{A}z(t) + \bar{B}u(t)$$
$$y(t) = \bar{C}z(t) + \bar{D}u(t).$$

- Argue that we should be able to use either model.
- Are they going to give the same transfer function?

$$H_1(s) = C(sI - A)^{-1}B + D$$
$$H_2(s) = \bar{C}(sI - \bar{A})^{-1}\bar{B} + \bar{D}.$$

- Need $H_1(s) = H_2(s)$.

$$\begin{aligned} H_1(s) &= C(sI - A)^{-1}B + D \\ &= CTT^{-1}(sI - A)^{-1}TT^{-1}B + D \\ &= (CT)[T^{-1}(sI - A)T]^{-1}(T^{-1}B) + D \\ &= \bar{C}(sI - \bar{A})^{-1}\bar{B} + \bar{D} = H_2(s). \end{aligned}$$

Transfer function not changed by similarity transform.

OBSERVATION: Consider

$$H(s) = \frac{b_1 s^2 + b_2 s + b_3}{s^3 + a_1 s^2 + a_2 s + a_3}.$$

- Only six parameters in transfer function. But, A has 3×3, B has 3×1, C has 1×3: a total of 15 parameters.
- Appears that we have 9 degrees of freedom in state-space model. Contradiction?

$$9 = \text{size}\Big[\quad \Big].$$

- We will see (Chapter 5) how to design T to put a system into the various canonical forms.

EXAMPLE: Controller canonical form for

$$\frac{2s + 3}{(s+1)(s+2)} = \frac{2s + 3}{s^2 + 3s + 2}$$

is

$$\dot{x}_c(t) = \begin{bmatrix} -3 & -2 \\ 1 & 0 \end{bmatrix} x_c(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 2 & 3 \end{bmatrix} x_c(t).$$

- Suppose we have the transformation matrix (note: $\det(T) = 1$ so T is invertible).

$$T = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}.$$

- Let $x_c = T\bar{x}$, where $\bar{x}$ is a new state. Then,

$$\begin{aligned} \dot{\bar{x}}(t) &= (T^{-1}AT)\bar{x}(t) + (T^{-1}B)u(t) \\ y(t) &= (CT)\bar{x}(t). \end{aligned}$$

- Plugging in A, B, C and T:

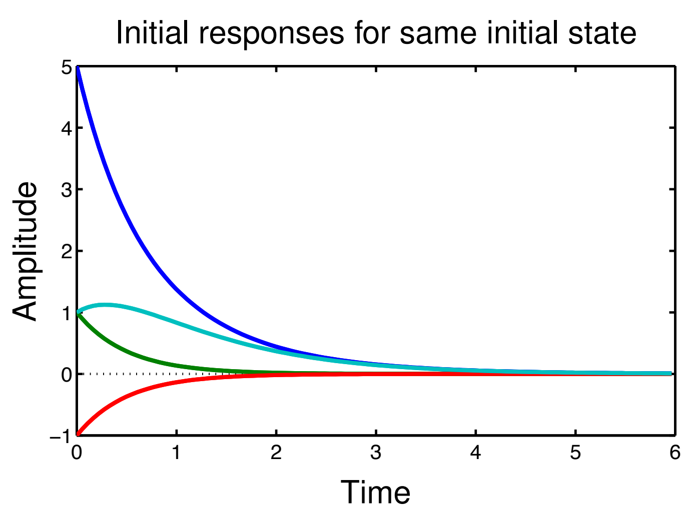
$$\begin{aligned} \dot{\bar{x}}(t) &= \begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix}\bar{x}(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix}u(t) \\ y(t) &= \begin{bmatrix} 1 & 1 \end{bmatrix}\bar{x}(t), \end{aligned}$$

  which gives the diagonal realization of the transfer function!

- We'll often change coordinates in a system, for example to solve a particular problem more easily.

EXAMPLE: Consider the system in the above example, implemented in the four main canonical forms. Let the initial state for each form be $x(0) = [1\ 1]^T$. Simulate response of each system.

- The systems have the same transfer function, but different responses to initial states since the states have different interpretations.



2.4: Time (dynamic) response

- Develop more insight into the system response by looking at time-domain solution for $x(t)$.
- Scalar case first, then many states and MIMO.

Homogeneous part (scalar)

- $\dot{x}(t) = ax(t), \quad x(0)$.
- Take Laplace. $X(s) = (s-a)^{-1}x(0)$.
- Inverse Laplace. $x(t) = e^{at}x(0)$.

Homogeneous part (full solution)

- $\dot{x}(t) = Ax(t), \qquad x(0)$.
- Take Laplace. $X(s) = (sI - A)^{-1}x(0)$.
- $x(t) = \mathcal{L}^{-1}[(sI-A)^{-1}]x(0)$.
- But,
$$(sI - A)^{-1} = \frac{I}{s} + \frac{A}{s^2} + \frac{A^2}{s^3} + \cdots$$
so,
$$\begin{aligned}\mathcal{L}^{-1}[(sI-A)^{-1}] &= I + At + \frac{A^2t^2}{2!} + \frac{A^3t^3}{3!} + \cdots \\ &\triangleq e^{At} \qquad \text{matrix exponential}\end{aligned}$$
$$x(t) = e^{At}x(0).$$
- e^{At} is called the <u>transition matrix</u> or <u>state-transition matrix</u>.
- Matrix exponential `expm.m`

- $e^{(A+B)t} = e^{At}e^{Bt}$ iff $AB = BA$. (*i.e.,* not in general).
- Will say more about e^{At} when we discuss the structure of A.
- Computation of $e^{At} = \mathcal{L}^{-1}[(sI - A)^{-1}]$ straightforward for 2×2.

EXAMPLE:

$$\dot{x} = Ax, \qquad A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}$$

$$(sI - A)^{-1} = \begin{bmatrix} s & -1 \\ 2 & s+3 \end{bmatrix}^{-1} = \begin{bmatrix} s+3 & 1 \\ -2 & s \end{bmatrix} \frac{1}{(s+2)(s+1)}$$

$$= \begin{bmatrix} \dfrac{2}{s+1} - \dfrac{1}{s+2} & \dfrac{1}{s+1} - \dfrac{1}{s+2} \\ \dfrac{-2}{s+1} + \dfrac{2}{s+2} & \dfrac{-1}{s+1} + \dfrac{2}{s+2} \end{bmatrix}$$

$$e^{At} = \begin{bmatrix} 2e^{-t} - e^{-2t} & e^{-t} - e^{-2t} \\ -2e^{-t} + 2e^{-2t} & -e^{-t} + 2e^{-2t} \end{bmatrix} 1(t)$$

- This is the best way to find e^{At} if A 2×2.

Forced solution (scalar)

$$\dot{x}(t) = ax(t) + bu(t), \quad x(0)$$

$$x(t) = e^{at}x(0) + \underbrace{\int_0^t e^{a(t-\tau)}bu(\tau)\,\mathrm{d}\tau}_{\text{convolution}}.$$

- Where did this come from?

1. $\dot{x}(t) - ax(t) = bu(t)$
2. $e^{-at}[\dot{x}(t) - ax(t)] = \dfrac{\mathrm{d}}{\mathrm{d}t}[e^{-at}x(t)] = e^{-at}bu(t)$.

3. $\displaystyle\int_0^t \frac{\mathrm{d}}{\mathrm{d}\tau}[e^{-a\tau}x(\tau)]\,\mathrm{d}\tau = e^{-at}x(t) - x(0) = \int_0^t e^{-a\tau}bu(\tau)\,\mathrm{d}\tau.$

Forced solution (full solution)

- Now, let $\dot{x}(t) = Ax(t) + Bu(t), \qquad x \in \mathbb{R}^{n\times 1}, \quad u \in \mathbb{R}^{m\times 1}.$
- Follow three steps above to get

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}Bu(\tau)\,\mathrm{d}\tau$$

- Clearly, if $y(t) = Cx(t) + Du(t)$,

$$y(t) = \underbrace{Ce^{At}x(0)}_{\text{initial resp.}} + \underbrace{\int_0^t Ce^{A(t-\tau)}Bu(\tau)\,\mathrm{d}\tau}_{\text{convolution}} + \underbrace{Du(t)}_{\text{feedthrough}}.$$

Thought: Maybe we can simplify the matrix exponential. . .

- Have seen the key role of e^{At} in the solution for $x(t)$. Impacts the system response, but need more insight.
- Consider what happens if the matrix A is *diagonalizable,* that is, if there exists a matrix T such that $T^{-1}AT = \Lambda =$diagonal.
- Then, $e^{At} = Te^{\Lambda t}T^{-1}$, and

$$e^{\Lambda t} = \begin{bmatrix} e^{\lambda_1 t} & & & 0 \\ & e^{\lambda_2 t} & & \\ & & \ddots & \\ 0 & & & e^{\lambda_n t} \end{bmatrix}.$$

- Much simpler form for the exponential, but how to find T, Λ?
- Eigenvalues/eigenvectors.

2.5: Diagonalizing the A matrix

- "Eigen" is a German word meaning "self" or "characteristic".
- The eigenvectors and eigenvalues of A characterize its behavior.
- An eigenvector is a vector satisfying

$$Av = \lambda v,$$

  where λ is a (possibly complex) constant, and $v \neq 0$.
- Multiplying by A does nothing to the vector except change its length!
- This is a very unusual vector. There are usually only n of them if A has size $n \times n$.
  - Note that if v is an eigenvector, kv is also an eigenvector—so eigenvectors are often normalized to have unit length: $\|v\|^2 = 1$.
- The constant λ is an eigenvalue. Specifically, it is the eigenvalue associated with eigenvector v.
- Since there are (usually) n eigenvectors with n corresponding eigenvalues, we label the eigenvectors and eigenvalues v_i and λ_i where $1 \leq i \leq n$.
- To find eigenvalues, consider that $(\lambda I - A)v = 0$.
- Since $v \neq 0$, $\lambda I - A$ must drop rank for some value of λ associated with v. A matrix which is not full rank has zero determinant. So, we can solve for the eigenvalues by solving

$$\det(\lambda I - A) = 0.$$

- We have seen already that this is how we solve for the poles of a state-space system, so the eigenvalues of the A matrix are the poles of the dynamic system.

- There are very efficient and numerically robust methods of finding eigenvectors/values. These methods do not use the determinant rule, above. The determinant rule is useful for mathematical analysis.
- In MATLAB, `[V,Lambda]=eig(A);`

Example: Finding eigenvalues/eigenvectors of 2×2 matrix

- Let
$$A = \begin{bmatrix} 3 & 3 \\ -5 & -5 \end{bmatrix}.$$
- To find eigenvalues, we solve:
$$\begin{aligned} \det(\lambda I - A) &= 0 \\ \det\left(\begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} - \begin{bmatrix} 3 & 3 \\ -5 & -5 \end{bmatrix}\right) &= 0 \\ \det\left(\begin{bmatrix} \lambda - 3 & -3 \\ 5 & \lambda + 5 \end{bmatrix}\right) &= 0 \\ (\lambda - 3)(\lambda + 5) + 15 &= 0 \\ \lambda^2 + 2\lambda &= 0. \end{aligned}$$
- We see that there are eigenvalues at $\lambda_1 = 0$ and $\lambda_2 = -2$.
- To find eigenvector corresponding to $\lambda_1 = 0$, we solve
$$\begin{aligned} Av_1 &= \lambda_1 v_1 \\ \begin{bmatrix} 3 & 3 \\ -5 & -5 \end{bmatrix} \begin{bmatrix} v_{1a} \\ v_{1b} \end{bmatrix} &= 0 \begin{bmatrix} v_{1a} \\ v_{1b} \end{bmatrix}. \end{aligned}$$
- This gives us the equation that $v_{1a} = -v_{1b}$. We can arbitrarily choose one component, so let $v_{1a} = 1$ giving $v_1 = \begin{bmatrix} 1 & -1 \end{bmatrix}^T$.

- We should further normalize this to have unit length by dividing by $\sqrt{(1)^2+(-1)^2}$. This gives

$$v_1^{\text{normalized}} = \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}.$$

- We find the second eigenvector similarly:

$$\begin{bmatrix} 3 & 3 \\ -5 & -5 \end{bmatrix} \begin{bmatrix} v_{2a} \\ v_{2b} \end{bmatrix} = -2 \begin{bmatrix} v_{2a} \\ v_{2b} \end{bmatrix}.$$

- Again, arbitrarily choose $v_{2a} = 1$. Then, we find that $v_{2b} = -5/3$.
- We can normalize by dividing by $\sqrt{(1)^2+(-5/3)^2}$. We get the answer

$$v_2^{\text{normalized}} = \begin{bmatrix} 0.5145 \\ 0.8575 \end{bmatrix}.$$

The diagonal form

- Assume that eigenvectors $v_1,\ v_2,\ \ldots\ v_n$ are linearly independent. Stack together in a matrix equation:

$$Av_i = \lambda_i v_i \qquad i = 1, 2, \ldots, n$$

$$A \underbrace{\begin{bmatrix} v_1 & v_2 & \ldots & v_n \end{bmatrix}}_{T} = \begin{bmatrix} v_1 & v_2 & \ldots & v_n \end{bmatrix} \underbrace{\begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix}}_{\Lambda}$$

- $AT = T\Lambda \quad \Rightarrow \quad T^{-1}AT = V^{-1}AV = \Lambda.$
- We have found the transformation matrix T that diagonalizes A!
- However, not all matrices are diagonalizable. For example,

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad \det(\lambda I - A) = \lambda^2.$$

- One eigenvalue $\lambda = 0$. Solve for the eigenvectors

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} v_a \\ v_b \end{bmatrix} = 0 \quad \Longrightarrow \quad \text{all vectors of the form } \begin{bmatrix} v_a \\ 0 \end{bmatrix} \neq 0.$$

Dynamic interpretation using diagonal form

- Diagonal form makes it easier to understand what is happening in a state-space dynamic system.
  - Assume A is diagonalizable by $T = V$.
  - Define new coordinates by $x(t) = T\tilde{x}(t)$ so

$$\dot{\tilde{x}}(t) = T^{-1}Ax(t) = T^{-1}AT\tilde{x}(t) = \Lambda\tilde{x}(t).$$

  - In new coordinate system, system is diagonal (decoupled).
- Trajectories consist of n independent modes; that is,

$$\tilde{x}_i(t) = e^{\lambda_i t}\tilde{x}_i(0)$$

hence the name, modal form.

- To understand in original coordinate system, write $T^{-1}AT = \Lambda$ as $T^{-1}A = \Lambda T^{-1}$ with

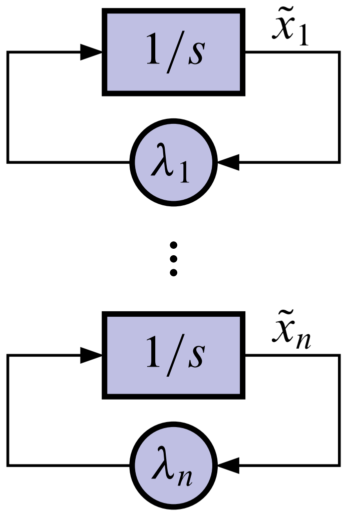
$$T^{-1} = \begin{bmatrix} w_1^T \\ w_2^T \\ \vdots \\ w_n^T \end{bmatrix}, \qquad i.e., \text{ rows of } T^{-1}.$$

$w_i^T A = \lambda_i w_i^T$, so w_i is a left eigenvector of A and note that $w_i^T v_j = \delta_{i,j}$.

- How does this help?

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

$$= \begin{bmatrix} v_1 & v_2 & \dots & v_n \end{bmatrix} \begin{bmatrix} e^{\lambda_1 t} & & & 0 \\ & e^{\lambda_2 t} & & \\ & & \ddots & \\ 0 & & & e^{\lambda_n t} \end{bmatrix} \begin{bmatrix} w_1^T \\ w_2^T \\ \vdots \\ w_n{}^T \end{bmatrix} = \sum_{i=1}^{n} e^{\lambda_i t} v_i w_i^T.$$



- Can write

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

Thus, trajectory can be expressed as linear combination of modes.

Interpretation.

- Left eigenvectors decompose initial state $x(0)$ into modal components $w_i^T x(0)$.
- $e^{\lambda_i t}$ term propagates ith mode forward t seconds. Stability?
- Reconstruct state as linear combination of right eigenvectors, where v_i corresponds to "relative phasing" of state contribution to the modal response.

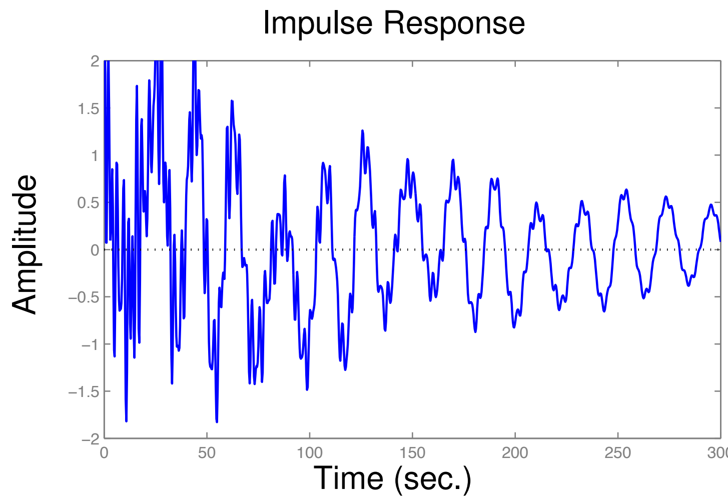
EXAMPLE: Let's consider a specific system

$$\dot{x}(t) = Ax(t)$$

$$y(t) = Cx(t)$$

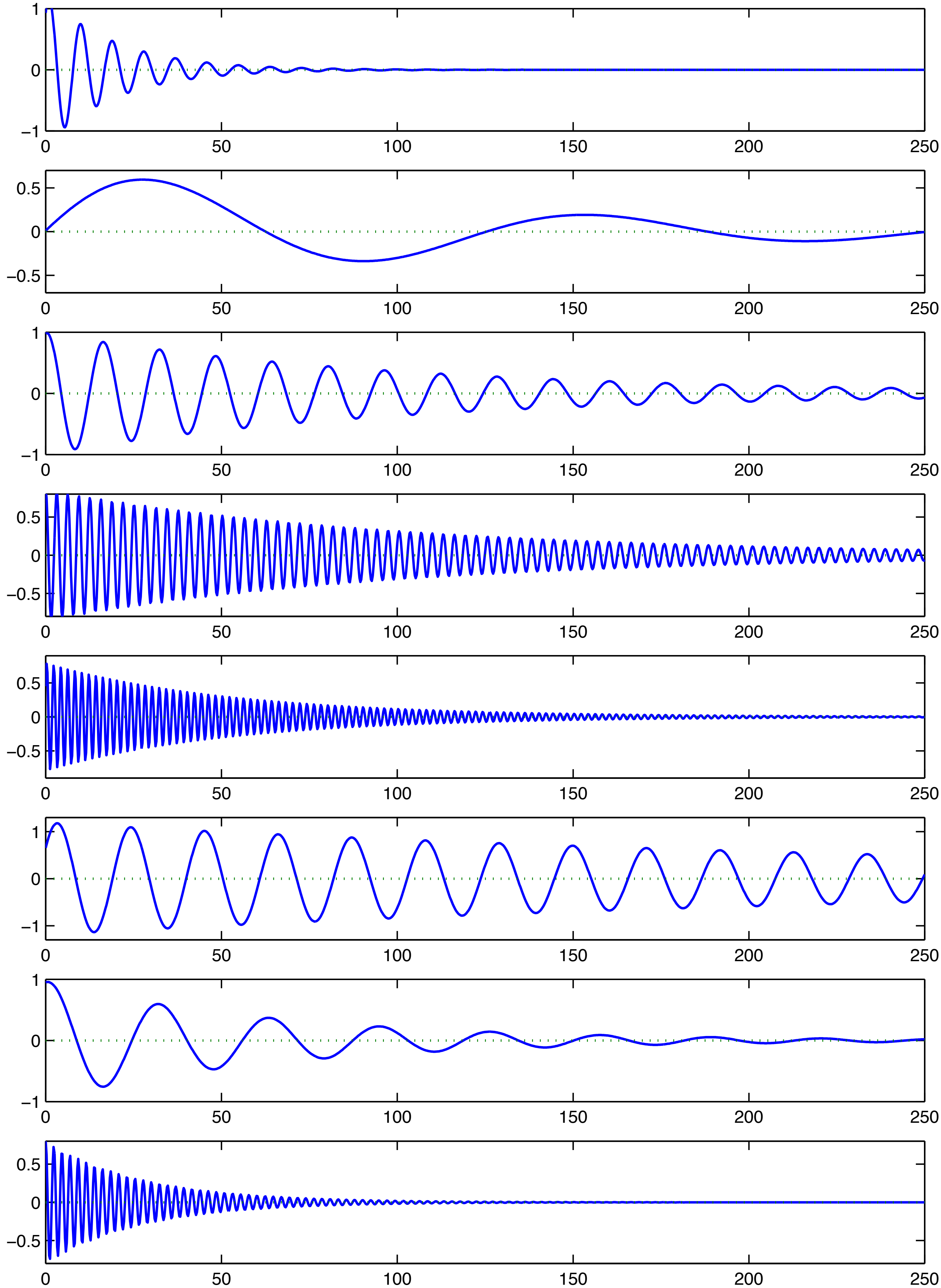
with $x(t) \in \mathbb{R}^{16\times 1}$, $y(t) \in \mathbb{R}$. (16-state, single output).

- A lightly damped system.
- Typical output to initial conditions shown.
- Output waveform is very complicated. Looks almost random.



- However, such a solution can be decomposed into much simpler modal components (see next page).

BOTTOM LINE: While responses of state-space systems can appear very complicated, they are fundamentally simply summations of very simple responses. Poles (eigenvalues of) A determine character of the response.


1
0
−1
0 50 100 150 200 250
0.5
0
−0.5
0 50 100 150 200 250
1
0
−1
0 50 100 150 200 250
0.5
0
−0.5
0 50 100 150 200 250
0.5
0
−0.5
0 50 100 150 200 250
1
0
−1
0 50 100 150 200 250
1
0
−1
0 50 100 150 200 250
0.5
0
−0.5
0 50 100 150 200 250

2.6: The Jordan canonical form; MIMO canonical forms

- What if A cannot be diagonalized?
- Any matrix $A \in \mathbb{R}^{n\times n}$ can be put in Jordan canonical form by a similarity transformation.
- That is, we can find a transformation matrix T such that $T^{-1}AT = J$, where J is a matrix in Jordan form.
- The Jordan form of a matrix is block-diagonal and looks like

$$J = \begin{bmatrix} J_1 & & 0 \\ & \ddots & \\ 0 & & J_q \end{bmatrix}$$

where

$$J_i = \begin{bmatrix} \lambda_i & 1 & & 0 \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ 0 & & & \lambda_i \end{bmatrix} \in \mathbb{R}^{n_i \times n_i}$$

is called a *Jordan block* of size n_i with eigenvalue λ_i (so $n = \sum_{i=1}^{q} n_i$).

- J is block-diagonal and upper bidiagonal.
- J is diagonal is the special case of n Jordan blocks of size $n_i = 1$.
- Jordan form is unique (up to permutations of the blocks).
- Can have multiple blocks with the same eigenvalue.

NOTE: The Jordan form is a *conceptual tool,* never used in numerical computations!

- $\chi(\lambda) = \det(\lambda I - A) = \det(\lambda I - J) = (\lambda - \lambda_1)^{n_1} \cdots (\lambda - \lambda_q)^{n_q} = 0$ hence distinct eigenvalues ➠ $n_i = 1$ ➠ A diagonalizable.
- $\dim \mathcal{N}(\lambda_i I - A) = \dim \mathcal{N}(\lambda_i I - J)$ is the number of Jordan blocks with eigenvalue λ_i.
- The sizes of each Jordan block may also be computed, but this is complicated. *i.e.,* leave it to MATLAB! ➠ `jordan(A)`

EXAMPLE: Consider

$$A = \begin{bmatrix} 3 & -1 & 1 & 1 & 0 & 0 \\ 1 & 1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 1 & 1 \\ 0 & 0 & 0 & 2 & -1 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

- From MATLAB, we find that A has eigenvalue 2 with multiplicity 5 and eigenvalue 0 with multiplicity 1:

$$\det(\lambda I - A) = (\lambda - 2)^5 \lambda.$$

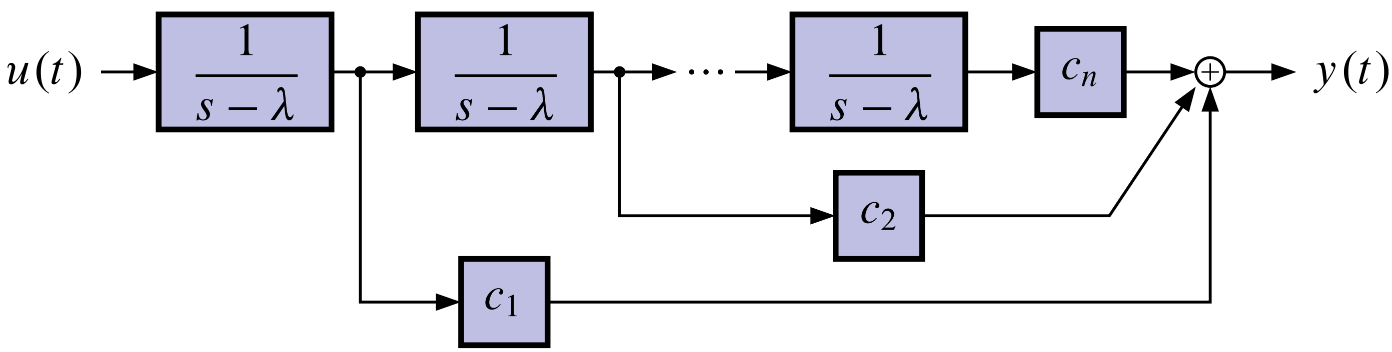
- $\text{rank}(2I - A) = 4$, so $\dim(\mathcal{N}(2I - A)) = 6 - 4 = 2$ so there are two Jordan blocks with eigenvalue 2.
- We can check this in MATLAB: `jordan(A)`

$$J = \left[\begin{array}{c:ccc:cc} 0 & 0 & 0 & 0 & 0 & 0 \\ \hdashline 0 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{array}\right]$$

- Note that without further information (computation) the following form might also be the Jordan form for A (but it isn't)

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

- System written in Jordan form is decomposed into independent Jordan block systems $\dot{\tilde{x}}_i(t) = J_i \tilde{x}_i(t)$



- Jordan blocks sometimes called Jordan chains (diagram shows why).
- What does this mean in the time domain?

$$(sI - J_\lambda)^{-1} = \begin{bmatrix} s-\lambda & -1 & & 0 \\ & s-\lambda & \ddots & \\ & & \ddots & -1 \\ 0 & & & s-\lambda \end{bmatrix}^{-1}$$

$$= \begin{bmatrix} (s-\lambda)^{-1} & (s-\lambda)^{-2} & \cdots & (s-\lambda)^{-k} \\ & (s-\lambda)^{-1} & \cdots & (s-\lambda)^{-k+1} \\ & & \ddots & \vdots \\ 0 & & & (s-\lambda)^{-1} \end{bmatrix}$$

$$= (s-\lambda)^{-1} I + (s-\lambda)^{-2} F_1 + \cdots + (s-\lambda)^{-k} F_k$$

where F_k is the matrix with ones on the kth upper diagonal.

- Hence, the matrix exponential is

$$e^{J_\lambda t} = e^{\lambda t} \begin{bmatrix} 1 & t & \cdots & t^{k-1}/(k-1)! \\ & 1 & \cdots & t^{k-2}/(k-2)! \\ & & \ddots & \vdots \\ 0 & & & 1 \end{bmatrix}$$

$$= e^{\lambda t}(I + tF_1 + \cdots + t^{k-1}/(k-1)!F_k).$$

- Thus, Jordan blocks expanded in e^{At} yield repeated poles and terms of the form $t^p e^{\lambda t}$.

Canonical forms for MIMO systems

- Consider

$$\begin{aligned} [G(s)] &= C(sI-A)^{-1}B + D \\ &= \frac{C[\mathrm{adj}(sI-A)]B}{\det(sI-A)} + D. \end{aligned}$$

- Now,

$$\det(sI-A) = s^n + \alpha_1 s^{n-1} + \alpha_2 s^{n-2} + \cdots + \alpha_n$$

and

$$C\,\mathrm{adj}(sI-A)B = N(s) = [N_1 s^{n-1} + N_2 s^{n-2} + \cdots + N_n].$$

- We can then write

$$\dot{x}(t) = \begin{bmatrix} -\alpha_1 I & -\alpha_2 I & \cdots & -\alpha_{n-1} I & -\alpha_n I \\ I & 0 & & 0 & 0 \\ 0 & I & & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & I & 0 \end{bmatrix} x(t) + \begin{bmatrix} I \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} N_1 & N_2 & \cdots & N_{n-1} & N_n \end{bmatrix} x(t) + G(\infty)u(t),$$

which is multivariable controller canonical form.

- We can also write

$$\dot{x}(t) = \begin{bmatrix} -\alpha_1 I & I & 0 & \cdots & 0 \\ -\alpha_2 I & 0 & I & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ -\alpha_{n-1} I & 0 & 0 & \cdots & I \\ -\alpha_n I & 0 & 0 & \cdots & 0 \end{bmatrix} x(t) + \begin{bmatrix} N_1 \\ N_2 \\ \vdots \\ N_{n-1} \\ N_n \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} I & 0 & 0 & \cdots & 0 \end{bmatrix} x(t) + G(\infty)u(t),$$

which is multivariable observer canonical form.

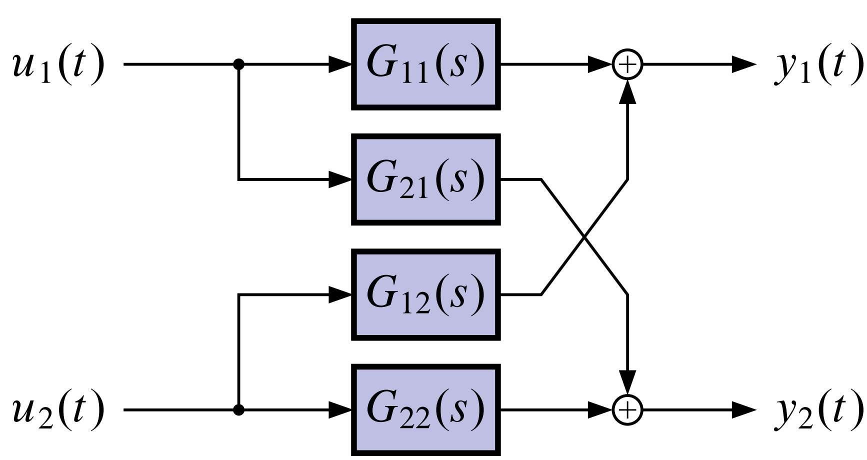
- We generally find that SISO canonical forms are more useful than MIMO canonical forms.

2.7: Zeros of a state-space system

- Have seen eigenvalues of A, or poles of the entries of $G(s)$ are the poles. Zeros of transfer function?
- What is a zero? *Two* types of zero in a MIMO system: Blocking zeros and transmission zeros.
- Consider a system with input $U(s) = \dfrac{K}{s-\lambda}$, $K \neq 0$ a constant vector.
- If λ is a blocking zero, $e^{\lambda t}$ will not appear at the output for any K, $x(0)$. (Not considered a very useful definition for MIMO zero).
- If λ is a transmission zero, $e^{\lambda t}$ will not appear at the output for some specific K, $x(0)$.

$$\{\text{blocking zeros}\} \subseteq \{\text{transmission zeros}\}.$$

IDEA: (but not the entire story) Consider a two-input two-output system



- A blocking zero will show up in $G_{11}(s)$, $G_{12}(s)$, $G_{21}(s)$ and $G_{22}(s)$. No matter what K is, if $U(s) = K/(s-\lambda)$, and λ is a blocking zero, the output does not have an $e^{\lambda t}$ term.

- A transmission zero may not show up as a zero in any of the individual transfer functions, but will in combinations thereof (with specific initial states).
- To find transmission zeros, put in $u(t) = u_0 e^{z_i t}$ and you get a zero output *at "frequency"* $e^{z_i t}$.
- State space: Have input and state contributions (consider first the SISO case)

$$u(t) = u_0 e^{z_i t}, \qquad x(t) = x_0 e^{z_i t} \quad \ldots \quad y(t) = 0.$$

$$\begin{aligned}
\dot{x}(t) = Ax(t) + Bu(t) &\Rightarrow z_i e^{z_i t} x_0 = A x_0 e^{z_i t} + B u_0 e^{z_i t} \\
&\Rightarrow \left[\begin{array}{c:c} z_i I - A & B \end{array}\right] \left[\begin{array}{c} x_0 \\ \hdashline -u_0 \end{array}\right] = 0 \\
y(t) = Cx(t) + Du(t) &\Rightarrow C x_0 e^{z_i t} + D u_0 e^{z_i t} = 0 \\
&\Rightarrow \left[\begin{array}{c:c} -C & D \end{array}\right] \left[\begin{array}{c} x_0 \\ \hdashline -u_0 \end{array}\right] = 0.
\end{aligned}$$

- Put the two together

$$\left[\begin{array}{c:c} z_i I - A & B \\ \hdashline -C & D \end{array}\right] \left[\begin{array}{c} x_0 \\ \hdashline -u_0 \end{array}\right] = 0.$$

- Zero at frequency z_i if there exists a nontrivial solution of

$$\det \left[\begin{array}{c:c} z_i I - A & B \\ \hdashline -C & D \end{array}\right] = 0.$$

- Recall

$$G(s) = \frac{\det\left[\begin{array}{c:c} sI - A & B \\ \hdashline -C & D \end{array}\right]}{\det(sI - A)}.$$

Ahah! (The $-u_0$ before gave us the correct sign in $G(s)$).

- In the MIMO case, with n state variables, p inputs and q outputs, a transmission zero is any value z_i for which

$$\operatorname{rank}\left[\begin{array}{c:c} z_i I - A & B \\ \hdashline -C & D \end{array}\right] < n + \min\{p, q\}$$

EXAMPLE:

$$G(s) = \begin{bmatrix} \dfrac{s(s+1)}{s^2+1} & \dfrac{s+1}{s+2} \\ 0 & \dfrac{(s+2)(s+1)}{s^2+2s+2} \end{bmatrix}.$$

- We can find that $G(s)$ has a blocking zero at $s = -1$ and has transmission zeros at $s = 0$, $s = -1$ and $s = -2$.

$$Y_1(s) = \frac{s(s+1)}{s^2+1}U_1(s) + \frac{s+1}{s+2}U_2(s)$$

$$Y_2(s) = \frac{(s+2)(s+1)}{s^2+2s+2}U_2(s).$$

Let $U = K/(s - \lambda)$

$$Y_1(s) = \frac{s+1}{s-\lambda}\left[\frac{s(s+2)k_1 + (s^2+1)k_2}{(s^2+1)(s+2)}\right]$$

$$Y_2(s) = \frac{s+1}{s-\lambda}\left[\frac{(s+2)k_2}{s^2+2s+2}\right].$$

- For all k_1, k_2, $s = -1$ is a zero. Therefore, both blocking and transmission.
- For $k_2 = 0$, $k_1 \in \mathbb{R}$, $s = 0$ is a zero. Therefore, transmission.
- Not so obvious, but $s = -2$ is also a zero. Therefore, transmission. [In a MIMO system, we can have a zero and pole at the same frequency!]
- Recall from before,

$$\left[\begin{array}{c|c} z_i I - A & B \\ \hline -C & D \end{array}\right]\left[\begin{array}{c} x_0 \\ \hline -u_0 \end{array}\right] = 0.$$

gives the initial state x_0 and $K = u_0$ if z_i is a transmission zero.

- In MATLAB: `tzero(sys)`

2.8: Linear time-varying systems

- With relatively small changes in what we've seen, we can also model linear time-varying (LTV) systems

$$\dot{x}(t) = A(t)x(t) + B(t)u(t)$$

$$y(t) = C(t)x(t) + D(t)u(t).$$

- LTV systems do not have transfer functions (so we cannot analyze in the Laplace domain).
- However, they do have (time varying) impulse responses.
- Let $g(t,\tau)$ be the output signal from the system corresponding to an input signal $\delta(t-\tau)$.
- Then, the general output from a causal LTV system can be written as

$$y(t) = \int_0^\infty g(t,\tau)u(\tau)\,\mathrm{d}\tau.$$

Homogeneous response

- To see responses of LTV state-space systems, consider first the homogeneous case

$$\dot{x}(t) = A(t)x(t), \qquad x(t_0) = x_0, \qquad t \ge 0.$$

FACT: The unique solution to this equation can be given by

$$x(t) = \Phi(t,t_0)x_0, \qquad t \ge 0,$$

where the $n \times n$ state-transition matrix $\Phi(t,t_0)$ is given by

$$\Phi(t,t_0) = I + \int_{t_0}^{t} A(s_1)\,\mathrm{d}s_1 + \int_{t_0}^{t} A(s_1)\int_{t_0}^{s_1} A(s_2)\,\mathrm{d}s_2\,\mathrm{d}s_1$$
$$+ \int_{t_0}^{t} A(s_1)\int_{t_0}^{s_1} A(s_2)\int_{t_0}^{s_2} A(s_3)\,\mathrm{d}s_3\,\mathrm{d}s_2\,\mathrm{d}s_1 + \cdots$$

- Some important properties of the state-transition matrix are:

  1. $\dot{\Phi}(t,t_0) = A(t)\Phi(t,t_0), \qquad \Phi(t_0,t_0) = I, \qquad t \geq 0.$
  2. $\Phi(t,s)\Phi(s,\tau) = \Phi(t,\tau).$
  3. $\Phi(t,\tau)^{-1} = \Phi(\tau,t).$

- To prove the first property, we start with

$$\begin{aligned} x(t) &= \Phi(t,t_0)x_0 \\ \dot{x}(t) &= \dot{\Phi}(t,t_0)x_0 + \underbrace{\Phi(t,t_0)\dot{x}_0}_{0} = A(t)x(t). \end{aligned}$$

- So, we have

$$\begin{aligned} A(t)x(t) &= \dot{\Phi}(t,t_0)x_0 \\ A(t)\Phi(t,t_0)x_0 &= \dot{\Phi}(t,t_0)x_0. \end{aligned}$$

- Since this must be true for arbitrary x_0, we have shown that

$$\dot{\Phi}(t,t_0) = A(t)\Phi(t,t_0).$$

- The second property is easily shown by breaking up the time line into the intervals from τ to s, and then from s to t.

$$\begin{aligned} x(s) &= \Phi(s,\tau)x(\tau) \\ x(t) &= \Phi(t,s)x(s) \\ x(t) &= \underbrace{\Phi(t,s)\Phi(s,\tau)}_{\Phi(t,\tau)}x(\tau). \end{aligned}$$

- The third property relates to going backward in time from a final state to an initial state.

$$\begin{aligned} x(t) &= \Phi(t,t_0)x(t_0) \\ x(t_0) &= \Phi^{-1}(t,t_0)x(t). \end{aligned}$$

- For an LTI system, we have $\Phi(t,t_0) = e^{A(t-t_0)}$. So, this new state-transition matrix is simply a generalization of what we have already seen.

Forced response

- Generalizing the prior result, we can find that the nonhomogeneous state solution is

$$x(t) = \Phi(t,t_0)x_0 + \int_{t_0}^{t} \Phi(t,\tau)B(\tau)u(\tau)\,\mathrm{d}\tau.$$

- To show this, start with the assumed answer and differentiate with respect to time t

$$\dot{x}(t) = \dot{\Phi}(t,t_0)x_0 + \int_{t_0}^{t} \dot{\Phi}(t,\tau)B(\tau)u(\tau)\,\mathrm{d}\tau + \Phi(t,t)B(t)u(t),$$

where we have needed Liebnitz' rule for differentiating an integral, which is

$$\frac{\mathrm{d}}{\mathrm{d}\theta}\int_{a(\theta)}^{b(\theta)} f(x,\theta)\,\mathrm{d}x = \int_{a(\theta)}^{b(\theta)} \frac{\partial f(x,\theta)}{\partial\theta}\,\mathrm{d}x + f(b(\theta),\theta)\frac{\partial b(\theta)}{\partial\theta} - f(a(\theta),\theta)\frac{\partial a(\theta)}{\partial\theta},$$

assuming that $f(x,\theta)$ is continuous and that the integral exists.

- Using our known result for $\dot{\Phi}(t,t_0)$ and the fact that $\Phi(t,t) = I$,

$$\dot{x}(t) = A(t)\Phi(t,t_0)x_0 + \int_{t_0}^{t} A(t)\Phi(t,\tau)B(\tau)u(\tau)\,\mathrm{d}\tau + B(t)u(t).$$

- Since the integral is with respect to τ, we can factor out $A(t)$

$$\begin{aligned}\dot{x}(t) &= A(t)\left(\Phi(t,t_0)x_0 + \int_{t_0}^{t} \Phi(t,\tau)B(\tau)u(\tau)\,\mathrm{d}\tau\right) + B(t)u(t)\\ &= A(t)x(t) + B(t)u(t).\end{aligned}$$

- Substituting into $y(t) = C(t)x(t) + D(t)u(t)$, we get

$$y(t) = C(t)\Phi(t,t_0)x_0 + \int_{t_0}^{t} C(t)\Phi(t,\tau)B(\tau)u(\tau)\,\mathrm{d}\tau + D(t)u(t).$$

- We can rewrite this as

$$\begin{aligned} y(t) &= C(t)\Phi(t,t_0)x_0 + \int_{t_0}^{t} \left[C(t)\Phi(t,\tau)B(\tau) + D(\tau)\delta(t-\tau)\right]u(\tau)\,\mathrm{d}\tau \\ &= C(t)\Phi(t,t_0)x_0 + \int_{t_0}^{t} h(t,\tau)u(\tau)\,\mathrm{d}\tau \end{aligned}$$

which allows us to recognize the impulse response of a LTV system to be

$$h(t,\tau) = C(t)\Phi(t,\tau)B(\tau) + D(\tau)\delta(t-\tau).$$

Equivalent transformations for LTV systems

- Transformations of LTV systems are usually time-varying.
- Let $T(t)$ be nonsingular for all t and define

$$x(t) = T(t)z(t)$$

such that

$$\dot{x}(t) = \dot{T}(t)z(t) + T(t)\dot{z}(t) = A(t)T(t)z(t) + B(t)u(t).$$

- Then, we have

$$T(t)\dot{z}(t) = \left[A(t)T(t) - \dot{T}(t)\right]z(t) + B(t)u(t),$$

which gives the equivalent LTV system

$$\begin{aligned} \dot{z}(t) &= T^{-1}(t)\left[A(t)T(t) - \dot{T}(t)\right]z(t) + T^{-1}(t)B(t)u(t) \\ y(t) &= C(t)T(t)z(t) + D(t)u(t). \end{aligned}$$

- A transformation matrix is called a fundamental matrix when it annihilates the term in square brackets

$$\dot{T}(t) = A(t)T(t), \qquad T(t_0) \neq 0.$$

- In this case, then

$$\dot{z}(t) = T^{-1}(t)B(t)u(t)$$

or

$$z(t) = z(t_0) + \int_{t_0}^{t} T^{-1}(\tau)B(\tau)u(\tau)\,\mathrm{d}\tau$$

and

$$\begin{aligned} x(t) &= T(t)z(t_0) + \int_{t_0}^{t} T(t)T^{-1}(\tau)B(\tau)u(\tau)\,\mathrm{d}\tau \\ &= T(t)T^{-1}(t_0)x_0 + \int_{t_0}^{t} T(t)T^{-1}(\tau)B(\tau)u(\tau)\,\mathrm{d}\tau \\ &= \Phi(t,t_0)x_0 + \int_{t_0}^{t} \Phi(t,\tau)B(\tau)u(\tau)\,\mathrm{d}\tau. \end{aligned}$$

- This gives us a practical way to compute the state-transition matrix

$$\Phi(t,\tau) = T(t)T^{-1}(\tau).$$

EXAMPLE: Consider the LTV system

$$\dot{x}(t) = \begin{bmatrix} 0 & 0 \\ t & 0 \end{bmatrix} x.$$

- This is equivalent to stating

$$\begin{aligned} \dot{x}_1(t) &= 0 \\ \dot{x}_2(t) &= t x_1(t) \end{aligned} \qquad \text{or} \qquad \begin{aligned} x_1(t) &= x_1(t_0) \\ x_2(t) &= x_2(t_0) + \frac{1}{2}(t^2 - t_0^2)x_1(t_0). \end{aligned}$$

- We wish to find a valid fundamental matrix $T(t)$, and from it find the state-transition matrix $\Phi(t, t_0)$.
- So, we need to find any matrix $T(t)$ that satisfies

$$\dot{T}(t) = A(t)T(t), \qquad T(t_0) \neq 0.$$

- We choose to initialize the fundamental matrix $T(0) = I$. This then implies that if

$$\begin{bmatrix} T_{11}(0) \\ T_{21}(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad \text{then} \qquad \begin{bmatrix} T_{11}(t) \\ T_{21}(t) \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{1}{2}t^2 \end{bmatrix}$$

and if

$$\begin{bmatrix} T_{12}(0) \\ T_{22}(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad \text{then} \qquad \begin{bmatrix} T_{12}(t) \\ T_{22}(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

- Combining these results as columns of T, we have

$$T(t) = \begin{bmatrix} 1 & 0 \\ \frac{1}{2}t^2 & 1 \end{bmatrix}.$$

- You can verify that this matrix satisfies $\dot{T}(t) = A(t)T(t)$.
- Using this to find the state-transition matrix,

$$\begin{aligned} \Phi(t, t_0) &= T(t)T^{-1}(t_0) \\ &= \begin{bmatrix} 1 & 0 \\ \frac{1}{2}t^2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{1}{2}t_0^2 & 1 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} 1 & 0 \\ \frac{1}{2}t^2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{1}{2}t_0^2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2}(t^2 - t_0^2) & 1 \end{bmatrix}. \end{aligned}$$

2.9: What about nonlinear systems?

- Nonlinear continuous-time state-space systems are modeled as

$$\dot{x}(t) = f(x(t), u(t))$$
$$y(t) = g(x(t), u(t)).$$

Linearizing around an equilibrium point

- To work with nonlinear systems, *using small input signals*, we can linearize these equations.
- Suppose that there is an equilibrium constant solution

$$u(t) = u^{\text{eq}}, \qquad x(t) = x^{\text{eq}}, \qquad \text{and} \qquad y(t) = y^{\text{eq}} = g(x^{\text{eq}}, u^{\text{eq}}),$$

that satisfies the nonlinear form.
- Then, let the actual input signal and initial state be written as

$$u(t) = u^{\text{eq}} + \delta u(t), \qquad x(t_0) = x^{\text{eq}} + \delta x^{\text{eq}}(t_0)$$

for *small* $\delta u(t)$.
- Then, we can write actual state and output as

$$x(t) = x^{\text{eq}} + \delta x(t), \qquad y(t) = y^{\text{eq}} + \delta y(t).$$

- Using the known form for $y(t)$, we have

$$\begin{aligned}\delta y(t) &= g(x(t), u(t)) - y^{\text{eq}} \\ &= g(x^{\text{eq}} + \delta x(t), u^{\text{eq}} + \delta u(t)) - g(x^{\text{eq}}, u^{\text{eq}}).\end{aligned}$$

- We expand the function $g(\cdot)$ using Taylor-series around the equilibrium point $(x^{\text{eq}}, u^{\text{eq}})$

$$g(x^{\text{eq}}+\delta x(t),u^{\text{eq}}+\delta u(t)) = g(x^{\text{eq}},u^{\text{eq}}) + \left(\frac{\mathrm{d}g(x^{\text{eq}},u^{\text{eq}})}{\mathrm{d}x}\right)\delta x(t) + \left(\frac{\mathrm{d}g(x^{\text{eq}},u^{\text{eq}})}{\mathrm{d}u}\right)\delta u(t) + h.o.t.$$

- Substituting into our result for $\delta y(t)$,

$$\delta y(t) \approx \underbrace{\left(\frac{\mathrm{d}g(x^{\text{eq}},u^{\text{eq}})}{\mathrm{d}x}\right)}_{C}\delta x(t) + \underbrace{\left(\frac{\mathrm{d}g(x^{\text{eq}},u^{\text{eq}})}{\mathrm{d}u}\right)}_{D}\delta u(t).$$

- To determine the evolution of δx, we first take its derivative

$$\delta\dot{x}(t) = \dot{x}(t) = f(x^{\text{eq}}+\delta x, u^{\text{eq}}+\delta u).$$

- By Taylor series,

$$\delta\dot{x}(t) = \underbrace{\left(\frac{\mathrm{d}f(x^{\text{eq}},u^{\text{eq}})}{\mathrm{d}x}\right)}_{A}\delta x(t) + \underbrace{\left(\frac{\mathrm{d}f(x^{\text{eq}},u^{\text{eq}})}{\mathrm{d}u}\right)}_{B}\delta u(t).$$

- To summarize, the linearized perturbation system is

$$\delta\dot{x}(t) = A\delta x(t) + B\delta u(t)$$

$$\delta y(t) = C\delta x(t) + D\delta u(t)$$

and the overall state and output can be computed as

$$x(t) = x^{\text{eq}} + \delta x(t) \qquad \text{and} \qquad y(t) = y^{\text{eq}} + \delta y(t).$$

Linearizing around a trajectory

- Instead of linearizing around an equilibrium point, which requires small $\delta u(t)$, we can also choose to linearize around a known solution trajectory.

- Suppose it is known that

$$u^{\text{sol}}(t), \qquad x^{\text{sol}}(t), \qquad y^{\text{sol}}(t)$$

form a time-varying solution to the nonlinear dynamics of the system.

- This means that

$$\begin{aligned}\dot{x}^{\text{sol}}(t) &= f(x^{\text{sol}}(t), u^{\text{sol}}(t))\\ y^{\text{sol}}(t) &= g(x^{\text{sol}}(t), u^{\text{sol}}(t)).\end{aligned}$$

- Then, let general input, state, and output be

$$\begin{aligned}u(t) &= u^{\text{sol}}(t) + \delta u(t)\\ x(t) &= x^{\text{sol}}(t) + \delta x(t)\\ y(t) &= y^{\text{sol}}(t) + \delta y(t).\end{aligned}$$

- Proceeding as before,

$$\begin{aligned}\dot{x}(t) &= \dot{x}^{\text{sol}}(t) + \delta\dot{x}(t)\\ \delta\dot{x}(t) &= \dot{x}(t) - \dot{x}^{\text{sol}}(t)\\ &= f(x^{\text{sol}}(t) + \delta x(t), u^{\text{sol}}(t) + \delta u(t)) - f(x^{\text{sol}}(t), u^{\text{sol}}(t)).\end{aligned}$$

- Using Taylor series,

$$\begin{aligned}\delta\dot{x}(t) &\approx \underbrace{\left(\frac{\mathrm{d}f(x^{\text{sol}}, u^{\text{sol}})}{\mathrm{d}x(t)}\right)}_{A(t)}\delta x(t) + \underbrace{\left(\frac{\mathrm{d}f(x^{\text{sol}}, u^{\text{sol}})}{\mathrm{d}u(t)}\right)}_{B(t)}\delta u(t)\\ \delta y(t) &\approx \underbrace{\left(\frac{\mathrm{d}g(x^{\text{sol}}, u^{\text{sol}})}{\mathrm{d}x(t)}\right)}_{C(t)}\delta x(t) + \underbrace{\left(\frac{\mathrm{d}g(x^{\text{sol}}, u^{\text{sol}})}{\mathrm{d}u(t)}\right)}_{D(t)}\delta u(t).\end{aligned}$$

- To summarize, the linearized perturbation system is

$$\delta\dot{x}(t) = A(t)\delta x(t) + B(t)\delta u(t)$$

$$\delta y(t) = C(t)\delta x(t) + D(t)\delta u(t)$$

and the overall state and output can be computed as

$$x(t) = x^{\text{sol}} + \delta x(t) \qquad \text{and} \qquad y(t) = y^{\text{sol}} + \delta y(t).$$

- Notice that even if the nonlinear system is time-invariant, the linearized system will be time-varying, in general.

Where to from here?

- We have now seen how to model continuous-time LTI, LTV, and nonlinear systems in state-space form.
- We have seen some ways to analyze and predict system behaviors.
- Our next step is to repeat this approach with discrete-time systems.
- We will see that the results are very similar, which will allow us to use common frameworks to develop deeper insights into state-space systems and to synthesize controllers.