6.3100: Dynamic System Modeling and Control Design

State-Space Responses

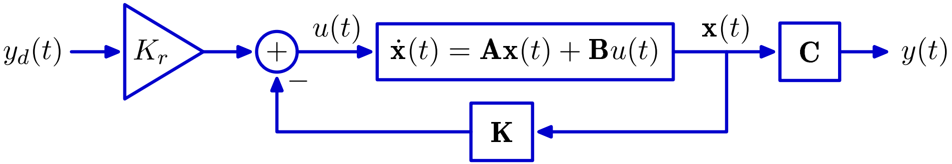
- Step Response
- Matrix Exponentials

November 04, 2024

State-Space Approach

Last week, we introduced the **State-Space** approach to control:

- Describe a system by its **states**.
- Describe dynamics of a system by **first-order** relations among states.
- Collect the states and relations in a single first-order **matrix** equation.



Plant: state matrix $\mathbf{A}$, input vector $\mathbf{B}$, and output vector $\mathbf{C}$:

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

$$y(t) = \mathbf{C}\mathbf{x}(t)$$

Feedback is characterized by a feedback vector $\mathbf{K}$ and input scaler K_r:

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

Combine to obtain **closed-loop** characterization:

$$\dot{\mathbf{x}}(t) = \Big(\mathbf{A} - \mathbf{B}\mathbf{K}\Big)\mathbf{x}(t) + \mathbf{B}\mathbf{K_r} y_d(t) \equiv \mathbf{A_c}\mathbf{x}(t) + \mathbf{B_c} y_d(t)$$

From State-Space to Transfer Function

Find the transfer function representation from the state-space description (similar to finding natural frequencies but $\mathbf{B_c} \neq 0$).

Start with the state equation:

$$\dot{\mathbf{x}}(t) = \mathbf{A_c}\mathbf{x}(t) + \mathbf{B_c}u(t)$$

Consider the input $u(t)$ and state $\mathbf{x}(t)$ at a particular complex frequency s:

$$u(t) = U(s)e^{st} \text{ and } \mathbf{x}(t) = \mathbf{X}(s)e^{st}$$

Find $H(s)$ at the same complex frequency.

$$s\mathbf{X}(s)e^{st} = \mathbf{A_c}\mathbf{X}(s)e^{st} + \mathbf{B_c}U(s)e^{st}$$

$$s\mathbf{X}(s) = \mathbf{A_c}\mathbf{X}(s) + \mathbf{B_c}U(s)$$

$$(s\mathbf{I} - \mathbf{A_c})\mathbf{X}(s) = \mathbf{B_c}U(s)$$

$$\mathbf{X}(s) = (s\mathbf{I} - \mathbf{A_c})^{-1}\mathbf{B_c}U(s)$$

$$Y(s) = \mathbf{C_c}\mathbf{X}(s) = \mathbf{C_c}(s\mathbf{I} - \mathbf{A_c})^{-1}\mathbf{B_c}U(s)$$

$$H(s) = \frac{Y(s)}{U(s)} = \mathbf{C_c}(s\mathbf{I} - \mathbf{A_c})^{-1}\mathbf{B_c}$$

From State-Space to Transfer Function

Example: find the open-loop transfer function

$$H(s) = \frac{Y(s)}{U(s)} = \mathbf{C}(s\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}$$

for mass-spring-dashpot system.

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

$$H(s) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} s & -1 \\ k/m & s+b/m \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ k/m \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \end{bmatrix} \frac{1}{s^2 + sb/m + k/m} \begin{bmatrix} s+b/m & 1 \\ -k/m & s \end{bmatrix} \begin{bmatrix} 0 \\ k/m \end{bmatrix}$$

$$= \frac{1}{s^2 + sb/m + k/m} \begin{bmatrix} s+b/m & 1 \end{bmatrix} \begin{bmatrix} 0 \\ k/m \end{bmatrix}$$

$$= \frac{k/m}{s^2 + sb/m + k/m} \qquad \surd$$

The denominator (and therefore poles) come from $\left| s\mathbf{I}-\mathbf{A} \right|$.

State-Space Analysis of Natural Frequencies

Are there frequencies s for which large outputs result when input $u(t)=0$?

$$H(s) = \frac{Y(s)}{X(s)} = \mathbf{C_c}(s\mathbf{I}-\mathbf{A})^{-1}\mathbf{B_c} = \mathbf{C_c}\,\frac{\text{adj}(s\mathbf{I}-\mathbf{A})}{|s\mathbf{I}-\mathbf{A}|}\,\mathbf{B_c}$$

If $|s\mathbf{I}-\mathbf{A}| = 0$, $H(s)$ is unbounded and therefore $|Y(s)| \to \infty$.

The natural frequencies are the solutions to the **characteristic equation**:

$$\left|s\,\mathbf{I}-\mathbf{A_c}\right| = 0$$

Example: mass-spring-dashpot system:

$$\mathbf{A_c} = \mathbf{A}-\mathbf{BK} = \begin{bmatrix} 0 & 1 \\ -k/m & -b/m \end{bmatrix} - \begin{bmatrix} 0 \\ k/m \end{bmatrix} \begin{bmatrix} K_1 & K_2 \end{bmatrix}$$

$$|s\,\mathbf{I}-\mathbf{A_c}| = \left|\begin{bmatrix} s & -1 \\ k(1+K_1)/m & s+(b+kK_2)/m \end{bmatrix}\right| = 0$$

Characteristic equation: $s^2 + (b+kK_2)s/m + k(1+K_1)/m = 0$ $\surd$

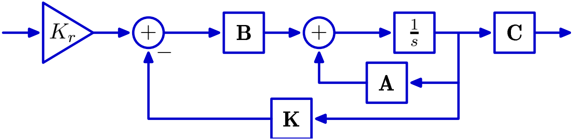
Check Yourself

A state-space controller is represented by the following equations:

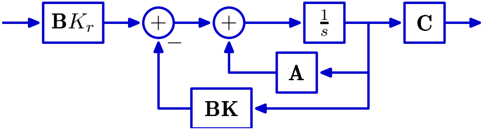
$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t)$$
$$u(t) = K_r y_d(t) - \mathbf{K}\mathbf{x}(t)$$
$$y(t) = \mathbf{C}\mathbf{x}(t)$$

Which block diagrams (below) correspond to the equations?

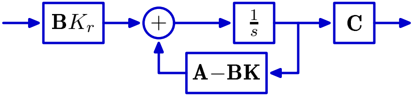
1.



2.



3.



Step Response

Find the step response $\mathbf{x_s}(t)$ of the following matrix system equation:

$$\dot{\mathbf{x}}(t) = \mathbf{P}\mathbf{x}(t) + \mathbf{Q}u(t)$$

Assume the initial value of the step response $\mathbf{x_s}(0) = \mathbf{0}$, and $u(t)=1$ for $t>0$.

Homogeneous equation: $\dot{\mathbf{x}}_\mathbf{h}(t) = \mathbf{P}\mathbf{x}_\mathbf{h}(t)$

If this were a scalar equation:

$$\dot{x}_h(t) = px_h(t)$$

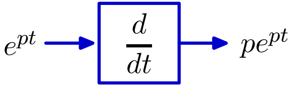
then the solution would be an exponential function of time:

$$x_h(t) = \alpha e^{pt}$$

Is there a matrix version of the exponential time function e^{pt}?

Scalar Exponential Function

Exponential functions are eigenfunctions of the derivative operator:



Express the exponential function as a power series:

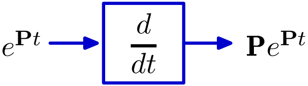
$$e^{pt} = 1 + \frac{pt}{1!} + \frac{p^2t^2}{2!} + \frac{p^3t^3}{3!} + \cdots$$

Differentiate term-by-term:

$$\begin{aligned}
\frac{d}{dt}e^{pt} &= 0 + \frac{p}{1!} + \frac{2p^2t}{2!} + \frac{3p^3t^2}{3!} + \cdots \\
&= p + p\frac{pt}{1!} + p\frac{p^2t^2}{2!} + p\frac{p^3t^3}{3!} + \cdots \\
&= p\left(1 + \frac{pt}{1!} + \frac{p^2t^2}{2!} + \frac{p^3t^3}{3!} + \cdots\right) \\
&= pe^{pt} \quad \checkmark
\end{aligned}$$

Matrix Exponential Function

Matrix exponentials are eigenfunctions of the matrix derivative operator:



The matrix exponential function can also be expanded as a power series:

$$e^{\mathbf{P}t} = \mathbf{I} + \frac{\mathbf{P}t}{1!} + \frac{\mathbf{P}^2t^2}{2!} + \frac{\mathbf{P}^3t^3}{3!} + \cdots$$

Differentiate term-by-term:

$$\begin{aligned} \frac{d}{dt}e^{\mathbf{P}t} &= 0 + \frac{\mathbf{P}}{1!} + \frac{2\mathbf{P}^2t}{2!} + \frac{3\mathbf{P}^3t^2}{3!} + \cdots \\ &= \mathbf{P} + \mathbf{P}\frac{\mathbf{P}t}{1!} + \mathbf{P}\frac{\mathbf{P}^2t^2}{2!} + \mathbf{P}\frac{\mathbf{P}^3t^3}{3!} + \cdots \\ &= \mathbf{P}\left(\mathbf{I} + \frac{\mathbf{P}t}{1!} + \frac{\mathbf{P}^2t^2}{2!} + \frac{\mathbf{P}^3t^3}{3!} + \cdots\right) \\ &= \mathbf{P}e^{\mathbf{P}t} = e^{\mathbf{P}t}\mathbf{P} \quad \surd \end{aligned}$$

Step Response

Find the step response $\mathbf{x_s}(t)$ of the following matrix system equation:

$$\dot{\mathbf{x}}(t) = \mathbf{P}\mathbf{x}(t) + \mathbf{Q}u(t)$$

Assume the initial value of the step response $\mathbf{x_s}(0) = \mathbf{0}$, and $u(t)=1$ for $t>0$.

Homogeneous equation: $\dot{\mathbf{x}}_\mathbf{h}(t) = \mathbf{P}\mathbf{x_h}(t)$

$$\mathbf{x_h}(t) = e^{\mathbf{P}t}\boldsymbol{\Psi}$$

Particular solution: $\mathbf{x_p}(t) = \boldsymbol{\Phi}$

$$\dot{\mathbf{x}}_\mathbf{p}(t) = \mathbf{0} = \mathbf{P}\boldsymbol{\Phi} + \mathbf{Q}$$

$$\boldsymbol{\Phi} = -\mathbf{P}^{-1}\mathbf{Q} \quad \text{(provided that } \mathbf{P} \text{ is not singular)}$$

Initial condition: $\mathbf{x}(0) = \boldsymbol{\Psi} - \mathbf{P}^{-1}\mathbf{Q} = \mathbf{0}$

$$\boldsymbol{\Psi} = \mathbf{P}^{-1}\mathbf{Q}$$

Step response:

$$\begin{aligned}\mathbf{x_s}(t) &= (e^{\mathbf{P}t} - \mathbf{I})\mathbf{P}^{-1}\mathbf{Q} \\ &= \mathbf{P}^{-1}(e^{\mathbf{P}t} - \mathbf{I})\mathbf{Q}\end{aligned}$$

Exponential functions play important role in solving matrix diff eq's.

Computing Matrix Exponentials

Finding the series expansion of a matrix exponential

$$e^{\mathbf{P}} = \mathbf{I} + \mathbf{P} + \mathbf{P^2}/2! + \mathbf{P^3}/3! + \mathbf{P^4}/4! + \cdots$$

is easy when $\mathbf{P}$ is **diagonal**:

$$\mathbf{P} = \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \ddots \end{bmatrix}$$

$$e^{\mathbf{P}} = \begin{bmatrix} 1 & & \\ & 1 & \\ & & \ddots \end{bmatrix} + \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \ddots \end{bmatrix} + \frac{1}{2!}\begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \ddots \end{bmatrix}^2 + \cdots$$

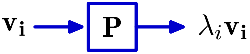
$$= \begin{bmatrix} 1 & & \\ & 1 & \\ & & \ddots \end{bmatrix} + \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \ddots \end{bmatrix} + \frac{1}{2!}\begin{bmatrix} \lambda_1^2 & & \\ & \lambda_2^2 & \\ & & \ddots \end{bmatrix} + \cdots$$

$$= \begin{bmatrix} 1 + \lambda_1 + \lambda_1^2/2! + \cdots & & \\ & 1 + \lambda_2 + \lambda_2^2/2! + \cdots & \\ & & \ddots \end{bmatrix} = \begin{bmatrix} e^{\lambda_1} & & \\ & e^{\lambda_2} & \\ & & \ddots \end{bmatrix}$$

Computing Matrix Exponentials

Fortunately it's easy to **diagonalize** a matrix that is full-rank and has distinct eigenvalues. Start with the eigenvector/eigenvalue property:

$$\mathbf{P}\mathbf{v_i} = \lambda_i \mathbf{v_i}$$



where λ_i is the i^{th} eigenvalue and $\mathbf{v_i}$ is the i^{th} eigenvector (a column vector). If $\mathbf{P}$ is full rank and if none of the eigenvalues are repeated

$$\begin{aligned}
\mathbf{P}\left[\mathbf{v_1}|\mathbf{v_2}|\mathbf{v_3}|\cdots|\mathbf{v_n}\right] &= \left[\mathbf{P}\mathbf{v_1}|\mathbf{P}\mathbf{v_2}|\mathbf{P}\mathbf{v_3}|\cdots|\mathbf{P}\mathbf{v_n}\right] \\
&= \left[\lambda_1\mathbf{v_1}|\lambda_2\mathbf{v_2}|\lambda_3\mathbf{v_3}|\cdots|\lambda_n\mathbf{v_n}\right] \\
&= \left[\mathbf{v_1}\middle|\mathbf{v_2}\middle|\mathbf{v_3}\middle|\cdots\middle|\mathbf{v_n}\right] \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \ddots \end{bmatrix} \\
\mathbf{P}\mathbf{V} &= \mathbf{V}\mathbf{\Lambda} \\
\mathbf{P} &= \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}
\end{aligned}$$

where $\mathbf{V} = \left[\mathbf{v_1}\middle|\mathbf{v_2}\middle|\mathbf{v_3}\middle|\cdots\middle|\mathbf{v_n}\right]$ and $\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \ddots \end{bmatrix}$.

Computing Matrix Exponentials

Substitute the diagonal expansion of $\mathbf{P}$:

$$\mathbf{P} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$$

into the series expansion of $e^{\mathbf{P}}$:

$$e^{\mathbf{P}} = \mathbf{I} + \mathbf{P} + \mathbf{P}^2/2! + \mathbf{P}^3/3! + \mathbf{P}^4/4! + \cdots$$

$$= \mathbf{I} + \; e^{\mathbf{P}t} \rightarrow \boxed{\frac{d}{dt}} \rightarrow \mathbf{P}e^{\mathbf{P}t} + \frac{1}{2!}\, e^{\mathbf{P}t} \rightarrow \boxed{\frac{d}{dt}} \rightarrow \mathbf{P}e^{\mathbf{P}t}e^{\mathbf{P}t} \rightarrow \boxed{\frac{d}{dt}} \rightarrow$$

$$= \mathbf{I} + \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1} + \frac{1}{2!}\mathbf{V}\mathbf{\Lambda}\mathbf{v_i} \rightarrow \boxed{\mathbf{P}} \rightarrow \lambda_i \mathbf{v_i}\mathbf{\Lambda}\mathbf{V}^{-1} + \frac{1}{3!}\mathbf{V}\mathbf{\Lambda}\mathbf{v_i} \rightarrow \boxed{\mathbf{P}} \rightarrow \lambda_i \mathbf{v}$$

$$= \mathbf{I} + \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1} + \frac{1}{2!}\mathbf{V}\mathbf{\Lambda}^2\mathbf{V}^{-1} + \frac{1}{3!}\mathbf{V}\mathbf{\Lambda}^3\mathbf{V}^{-1} + \cdots$$

$$= \mathbf{V}\mathbf{V}^{-1} + \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1} + \frac{1}{2!}\mathbf{V}\mathbf{\Lambda}^2\mathbf{V}^{-1} + \frac{1}{3!}\mathbf{V}\mathbf{\Lambda}^3\mathbf{V}^{-1} + \cdots$$

$$= \mathbf{V}\left(\mathbf{I} + \mathbf{\Lambda} + \frac{1}{2!}\mathbf{\Lambda}^2 + \frac{1}{3!}\mathbf{\Lambda}^3 + \cdots\right)\mathbf{V}^{-1}$$

$$= \mathbf{V}e^{\mathbf{\Lambda}}\mathbf{V}^{-1}$$

The matrix exponential of $\mathbf{P}$ can be directly computed from the eigenvalues and eigenvectors of $\mathbf{P}$.

Computing Matrix Exponentials: Example

Determine $e^{\mathbf{P}t}$ for $\mathbf{P} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}$.

Step 1: Find the eigenvalues of $\mathbf{P}$: -1 and -2.

Step 2: Find the eigenvectors of $\mathbf{P}$: $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$ and $\begin{bmatrix} 1 \\ -2 \end{bmatrix}$.

Step 3: Find $e^{\mathbf{P}t} = \mathbf{V}e^{\mathbf{\Lambda}t}\mathbf{V^{-1}}$.

$$\begin{aligned} e^{\mathbf{P}t} &= \left[\begin{array}{c|c} 1 & 1 \\ -1 & -2 \end{array}\right] \begin{bmatrix} e^{-t} & 0 \\ 0 & e^{-2t} \end{bmatrix} \left[\begin{array}{c|c} 1 & 1 \\ -1 & -2 \end{array}\right]^{-1} \\ &= \begin{bmatrix} 1 & 1 \\ -1 & -2 \end{bmatrix} \begin{bmatrix} e^{-t} & 0 \\ 0 & e^{-2t} \end{bmatrix} \begin{bmatrix} 2 & 1 \\ -1 & -1 \end{bmatrix} \\ &= \begin{bmatrix} e^{-t} & e^{-2t} \\ -e^{-t} & -2e^{-2t} \end{bmatrix} \begin{bmatrix} 2 & 1 \\ -1 & -1 \end{bmatrix} \\ &= \begin{bmatrix} 2e^{-t} - e^{-2t} & e^{-t} - e^{-2t} \\ -2e^{-t} + 2e^{-2t} & -e^{-t} + 2e^{-2t} \end{bmatrix} \end{aligned}$$

Check Yourself

Let

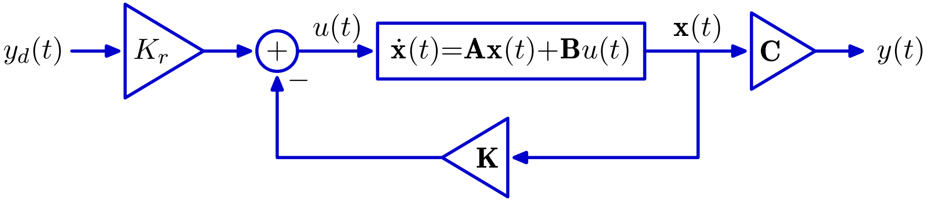
$$\mathbf{P} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Which of the following matrices equals $e^{\mathbf{P}t}$?

1. $\begin{bmatrix} e^{jt} & e^{-jt} \\ e^{-jt} & e^{jt} \end{bmatrix}$
2. $\begin{bmatrix} \cos(t) & \sin(t) \\ -\sin(t) & \cos(t) \end{bmatrix}$
3. $\begin{bmatrix} e^{t} & e^{-t} \\ e^{-t} & e^{t} \end{bmatrix}$
4. $\begin{bmatrix} te^{t} & te^{-t} \\ te^{-t} & te^{t} \end{bmatrix}$
5. none of the above

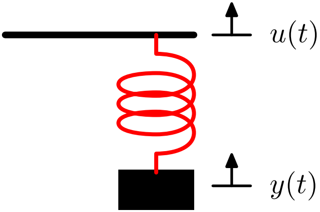
Controller Design

Optimizing the gains $\mathbf{K}$ and K_r of a state-space controller.



Controlling a Mass-Spring-Dashpot

Start by reviewing how we chose gains for a classical controller.



Model the mass-spring-dashpot system (Newton's Law):

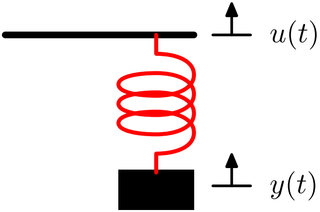
$$\underbrace{k\Big(x(t)-y(t)\Big) - b\dot{y}(t)}_{F} = \underbrace{m\ddot{y}(t)}_{ma}$$

to get the (open loop) transfer function of the plant:

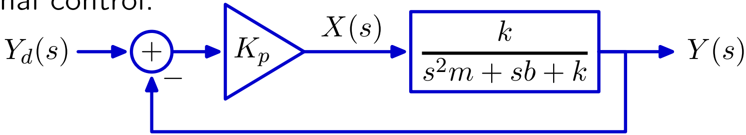
$$G(s) = \frac{Y(s)}{X(s)} = \frac{k}{s^2 m + sb + k}$$

Proportional Control of a Mass-Spring-Dashpot

Start by reviewing how we chose gains for a classical controller.



Proportional control:

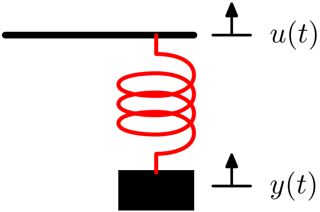


Closed-loop transfer function:

$$H(s) = \frac{Y(s)}{Y_d(s)} = \frac{kK_p}{s^2m + sb + k(1+K_p)}$$

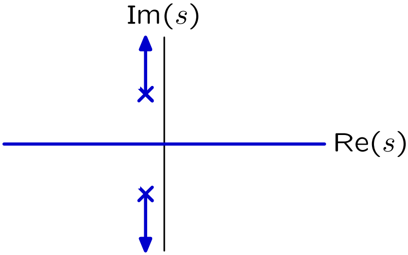
Closed-loop poles are roots of denominator: $s_1, s_2 = \dfrac{-b \pm \sqrt{b^2 - 4mk(1+K_p)}}{2m}$

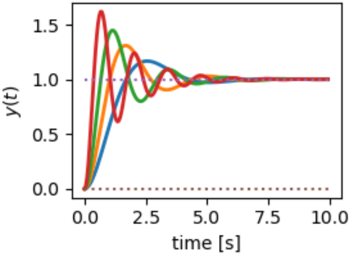
Root-Locus Analysis of Proportional Control



Consider the closed-loop poles $s_1, s_2 = \dfrac{-b \pm \sqrt{b^2 - 4mk(1+K_p)}}{2m}$

when $m = 1$, $b = 1.4$, $k = 2$, and K_p increases from 0 to ∞.

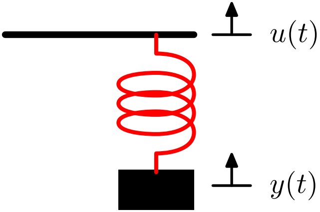




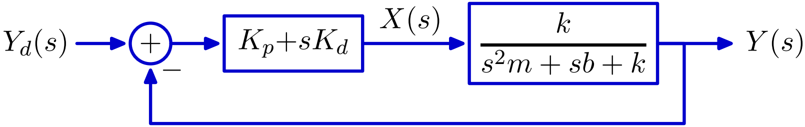
As $K_p \uparrow$, frequency of ringing $\uparrow$, but peaks decay with same time constant.

Proportional Plus Derivative Control

Adding a derivative term helps.



Same open-loop transfer function, different controller:



Closed-loop transfer function:

$$H(s) = \frac{Y(s)}{Y_d(s)} = \frac{k(K_p+sK_d)}{s^2m + s(b+kK_d) + k(1+K_p)}$$

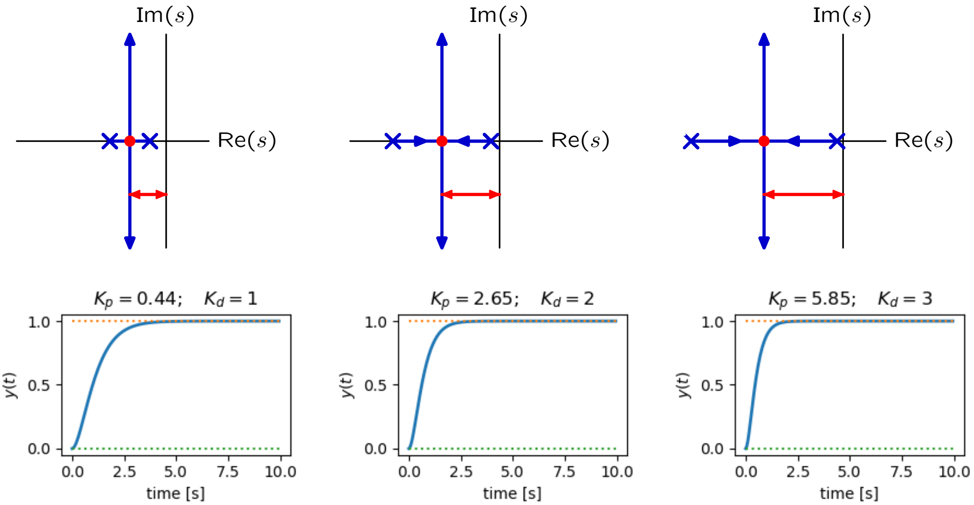
Closed-loop poles: $s = \dfrac{-b-kK_d \pm \sqrt{(b+kK_d)^2 - 4mk(1+K_p)}}{2m}$

Root-Locus Analysis of PD Control

Increasing K_d enables faster closed-loop poles (red dots) without overshoot.

$$s = \frac{-b - kK_d \pm \sqrt{(b + kK_d)^2 - 4mk(1 + K_p)}}{2m}$$

$m = 1;\ b = 1.4;\ k = 2;\ K_d = 1, 2, 3;\ K_p\colon 0 \to \infty.$



More Advanced Methods in Classical Control

Derivative feedback is just one way to optimize a classical controller.

Other advanced classical methods include

- integral feedback for PID control,
- optimizing gain and phase margins,
- lead compensation, lag compensation, lead/lag compensation, and
- many other techniques.

Much of the design power of these advanced methods results from their ability to **move poles and zeros** to locations that are more favorable for

- stability,
- disturbance rejection,
- noise immunity, etc.

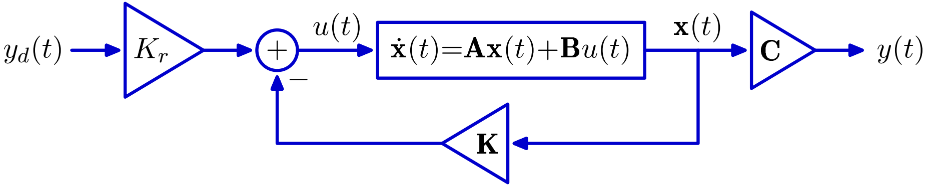
We can similarly optimize state-space controllers.

And the state-space formulation is much more powerful!

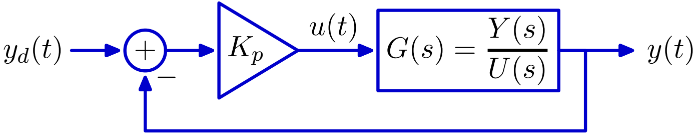
Stopped Here

Check Yourself: Proportional Control in State Space

Choose $\mathbf{K}$ and K_r so that the state-space controller:



is equivalent to a proportional controller:

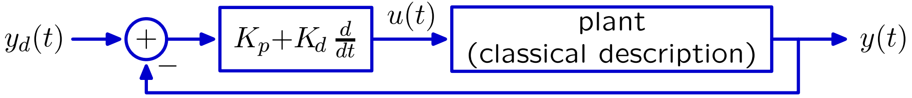


What $\mathbf{K}$ and K_r implement a proportional controller?

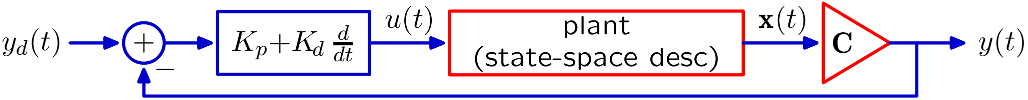
1. $\mathbf{K}=K_r\mathbf{C}$ and $K_r = K_p$
2. $\mathbf{K}=\mathbf{C}\mathbf{B}$ and $K_r = 1$
3. $\mathbf{K}=\mathbf{C}$ and $K_r = K_p$
4. $\mathbf{K}=K_p\mathbf{C}$ and $K_r = K_p$
5. none of the above

Proportional Plus Derivative Control in State-Space

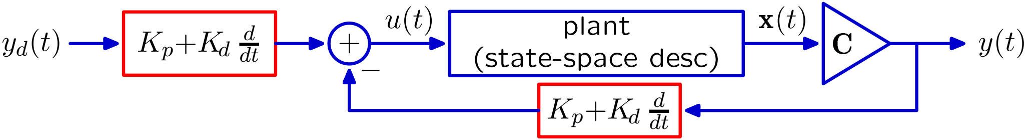
Start with a classical system with PD control:



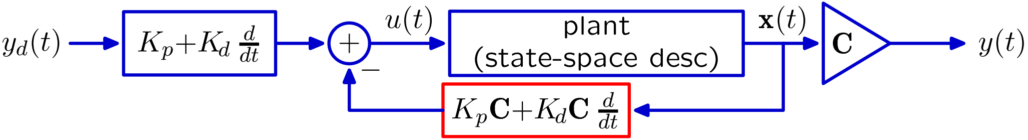
Replace classical description of plant with equiv. state-space description:



Distribute the PD controller over the inputs to the subtractor:



Feedback $\mathbf{x}(t)$ instead of $y(t)$:



Result is a state-space controller with $K_r = K_p + K_d \frac{d}{dt}$ and $\mathbf{K} = K_p\mathbf{C} + K_d\mathbf{C} \frac{d}{dt}$.

More Advanced Methods in Classical Control

Much of the design power of the more advanced methods results from their ability to **move poles and zeros** to locations that are more favorable for

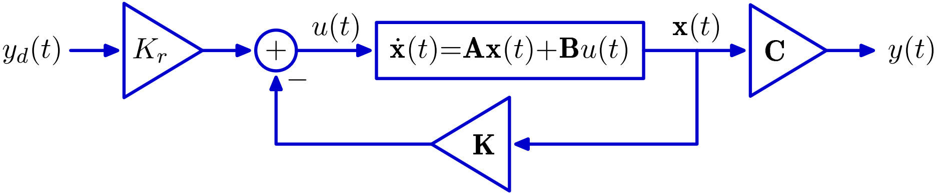
- stability,
- disturbance rejection,
- noise immunity, etc.

We can similarly optimize state-space controllers.

And the state-space formulation is much more powerful!

Pole Placement

With the correct choice of gains $\mathbf{K}$ and K_r, we can move the closed-loop poles of a state-space model **anywhere** in the complex plane.



The closed-loop poles of a state-space model are equal to the roots of its characteristic polynomial:

$$\left|s\mathbf{I}-(\mathbf{A}-\mathbf{B}\mathbf{K})\right| = \mathbf{0}$$

Fundamental theorem of algebra: an n^{th} order polynomial as n roots.

Factor theorem: each root determines a first-order factor.

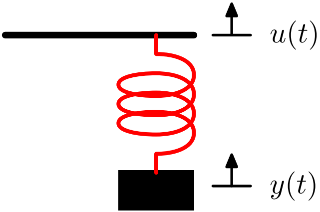
$\rightarrow$ characteristic polynomial can be written as a product of first-order terms:

$$\left|s\mathbf{I}-(\mathbf{A}-\mathbf{B}\mathbf{K})\right| = \prod_{i=1}^{n}(s-s_i) = 0$$

LHS: n^{th} order polynomial in s (pole locations)

RHS: same polynomial, but coeff's in terms of desired pole locations s_i.

Example: Pole Placement for Mass-Spring-Dashpot



Plant:

$$\underbrace{k\Big(u(t)-y(t)\Big) - b\dot{y}(t)}_{F} = \underbrace{m\ddot{y}(t)}_{ma}$$

Rewrite this second-order differential equation as two **first-order** equations:

$$\underbrace{\frac{d}{dt}\begin{bmatrix} y(t) \\ \dot{y}(t) \end{bmatrix}}_{\dot{\mathbf{x}}(t)} = \underbrace{\begin{bmatrix} 0 & 1 \\ -k/m & -b/m \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} y(t) \\ \dot{y}(t) \end{bmatrix}}_{\mathbf{x}(t)} + \underbrace{\begin{bmatrix} 0 \\ k/m \end{bmatrix}}_{\mathbf{B}} u(t)$$

which can be expressed as a single matrix equation:

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

Example: Pole Placement for Mass-Spring-Dashpot

For $m = 1$, $b = 1.4$, and $k = 2$:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -2 & -1.4 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

With $\mathbf{K} = [\, k_1 \quad k_2 \,]$:

$$|s\mathbf{I} - (\mathbf{A} - \mathbf{BK})| = \left| \begin{bmatrix} s & -1 \\ 2+2k_1 & s+1.4+2k_2 \end{bmatrix} \right| = s^2 + (1.4+2k_2)s + (2+2k_1)$$

To place poles at **s = −0.5 and s = −1**, set

$$s^2 + (1.4+2k_2)s + (2+2k_1) = (s+0.5)(s+1) = s^2 + 1.5s + 0.5$$

$\rightarrow k_1 = -0.75$ and $k_2 = 0.05$.

Alternatively, to place poles at **s = −0.5 and s = −0.6**, set

$$s^2 + (1.4+2k_2)s + (2+2k_1) = (s+0.5)(s+0.6) = s^2 + 1.1s + 0.3$$

$\rightarrow k_1 = -0.85$ and $k_2 = -0.15$.

Check Yourself

Let $\mathbf{A}$ represent the system matrix and $\mathbf{B}$ represent the input matrix for a state-space control system, where these matrices are given by

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Which $\mathbf{K}$ vector will produce closed-loop poles at 0 and -2?

1. $\mathbf{K} = [\,1 \quad 2\,]$
2. $\mathbf{K} = [\,3 \quad 4\,]$
3. $\mathbf{K} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$
4. $\mathbf{K} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$
5. none of the above

Pole Placement

With full-state feedback, the gains $\mathbf{K}$ can be adjusted to produce **ANY** set of n closed-loop poles! $\rightarrow$ **much more powerful than classical methods!**

The design problem shifts ...

- from finding gains to optimize pole locations (classical view)
- to finding pole locations to optimize performance (modern view).

Examples: Next Lecture