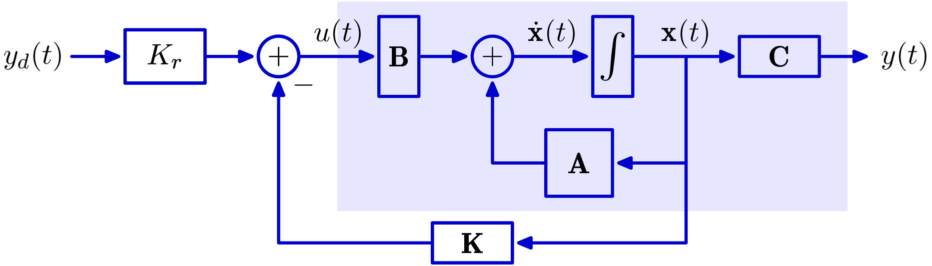
6.3100: Dynamic System Modeling and Control Design

Tracking Errors and Disturbances

November 18, 2024

Review: State-Space Design

State-Space Model:



Matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ constitute a **model of the plant** (shaded).

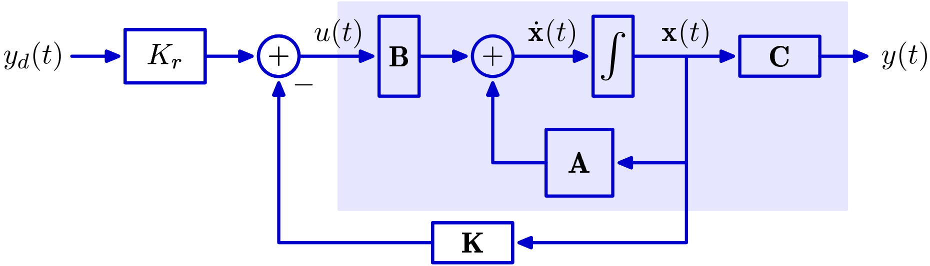
We want to design the **controller:** $\mathbf{K}$ and K_r.

Last week we discussed two methods to design $\mathbf{K}$:

- **pole placement:** choose $\mathbf{K}$ to achieve our choice of pole locations
- **linear quadratic regulator:** choose $\mathbf{K}$ to minimize a **cost function**

Pole Placement

The pole placement algorithm determines the gain $\mathbf{K}$ to locate 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:

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

which can be written as a product of first-order factors

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

Given $\mathbf{A}$ and $\mathbf{B}$, solve for the $\mathbf{K}$ that produces the desired pole locations.

Unfortunately, it's not easy to figure out an "optimal" set of pole locations.

Linear Quadratic Regulator (LQR)

The LQR method minimizes a **cost function** J that describes the relative cost (or badness) of inputs $\mathbf{u}(t)$ and responses $\mathbf{x}(t)$.

The cost function J is the time integral of a weighted sum of the squares of state variables $\mathbf{x}(t)$ and input $\mathbf{u}(t)$

$$J = \int_0^\infty \Big(\mathbf{x}^\mathbf{T}(t)\,\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^\mathbf{T}(t)\,\mathbf{R}\mathbf{u}(t)\Big)dt$$

where $\mathbf{u}(t)$ and $\mathbf{x}(t)$ are related

- by the state transition equation: $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$ and
- by the feedback constraint: $\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t)$.

and $\mathbf{Q}$ and $\mathbf{R}$ represent weights.

The "optimal" $\mathbf{K}$ is given by

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^\mathbf{T}\mathbf{S}$$

where $\mathbf{S}$ is the symmetric $n{\times}n$ solution to the **algebraic Riccati equation**:

$$\mathbf{A}^\mathbf{T}\mathbf{S} + \mathbf{S}\mathbf{A} - \mathbf{S}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\mathbf{T}\mathbf{S} + \mathbf{Q} = \mathbf{0}$$

Numerical Solutions

Fortunately there are efficient algorithms for solving both problems.

the following Python code

```
> from control import place_poles
> K = place_poles(A,B,[pole_1, pole_2, ... pole_n]).gain_matrix
or
> from control import lqr
> K,S,E = lqr(A,B,Q,R)
```

or MATLAB code

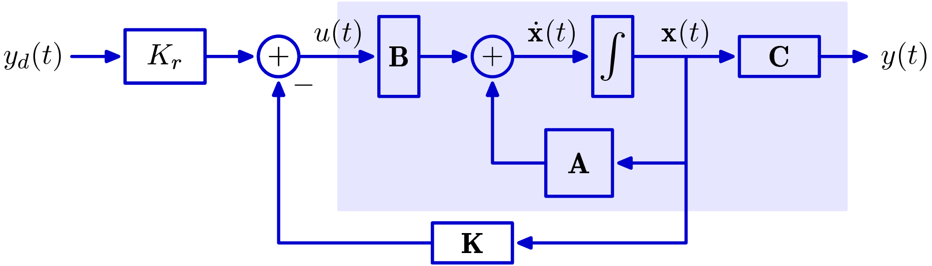
```
> K = place(A,B,[pole_1, pole_2, ... pole_n]);
or
> K,S,E = lqr(A,B,Q,R);
```

finds the optimal solutions to the place and LQR algorithms and returns

- `K`: state feedback gains,
- `S`: solution to the algebraic Riccati equation, and
- `E`: eigenvalues of the resulting closed-loop system.

Check Yourself

Use pole placement or LQR to find $\mathbf{K}$.

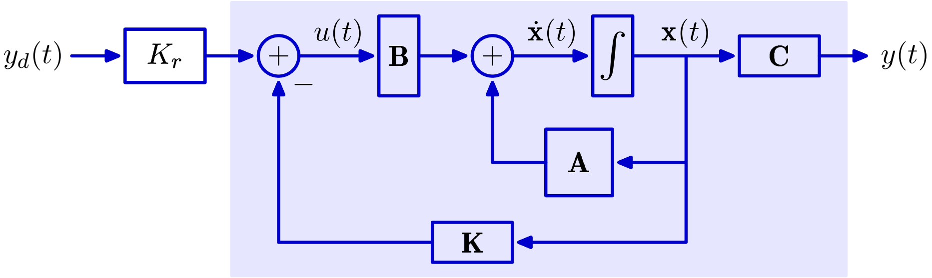


How should we choose K_r?

1. Choose K_r to maximize stability.
2. Choose K_r to minimize steady-state error.
3. Choose K_r to minimize the time constant of the step response.
4. Choose K_r to minimize overshoot in $y(t)$.
5. none of the above

Using Feedback to Reduce Tracking Errors

Using K_r to eliminate tracking errors is a **feed-forward** approach!

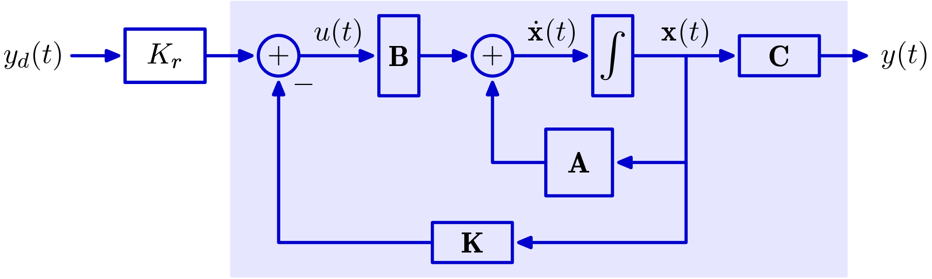


This method uses K_r to anticipate and pre-correct unwanted offsets in the rest of the system.

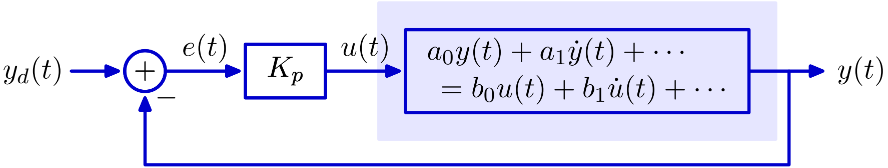
Are there similar unwanted offsets in classical controllers?

Compare Classical and State-Space Controllers

State-Space Controller:

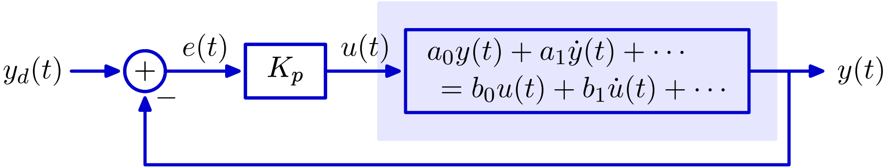


Classical Proportional Controller



Check Yourself

Determine the steady-state error of the following system.



Which of the following correctly describes the steady-state error?

1. $e(\infty)$ equals zero
2. $e(\infty)$ approaches zero with increasing K_p
3. $e(\infty)$ approaches zero with increasing K_p if $b_0 \neq 0$
4. feedback tends to reduce the steady-state error
5. none of the above

Tracking Errors

Tracking error refers to the difference between the output $y(t)$ of a feedback system and its desired value $y_d(t)$.

Tracking errors are especially important in some applications:

- automotive cruise control
- industrial robot (e.g., automotive assembly)
- landing a spacecraft on the moon

  ...

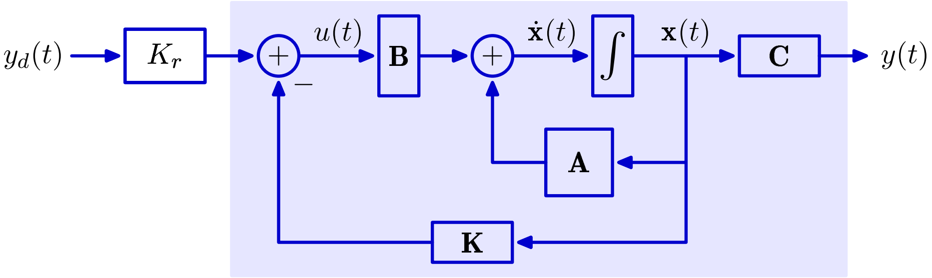
Tracking errors can be eliminated by setting K_r as follows:

$$K_r = \frac{-1}{\mathbf{C}(\mathbf{A}-\mathbf{B}\mathbf{K})^{-1}\mathbf{B}}$$

Unfortunately, tracking errors will still occur if the model parameters ($\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$) do not accurately represent the physical plant (which is inevitable).

Using Feedback to Reduce Tracking Errors

Using K_r to eliminate tracking errors is a **feed-forward** approach!



This method uses K_r to anticipate and pre-correct unwanted offsets in the rest of the system.

Fortunately, there is an alternative.

We can use **feedback** to dynamically reduce tracking errors.

Using Feedback to Reduce Tracking Errors

Approach: assign a state $w(t)$ to accumulate the tracking error $y(t) - y_d(t)$.

$$w(t) = \int_0^t \Big(y(\tau) - y_d(\tau)\Big) d\tau$$

Then use pole placement or LQR to design gains $\mathbf{K}$ to "optimally" reduce this tracking error along with the other state variables to zero.

Incorporate $w(t)$ into the state-space representation of the system.

Compute the derivative of $w(t)$:

$$\frac{dw(t)}{dt} = y(t) - y_d(t) = \mathbf{C}\mathbf{x}(t) - y_d(t)$$

Note that if $w(t)$ converges, then $\dot{w}(t) \to 0$ and the steady-state output error goes to zero.

Combine this equation with the original state equations:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) \; ; \quad y(t) = \mathbf{C}\mathbf{x}(t)$$

by defining a new augmented state vector:

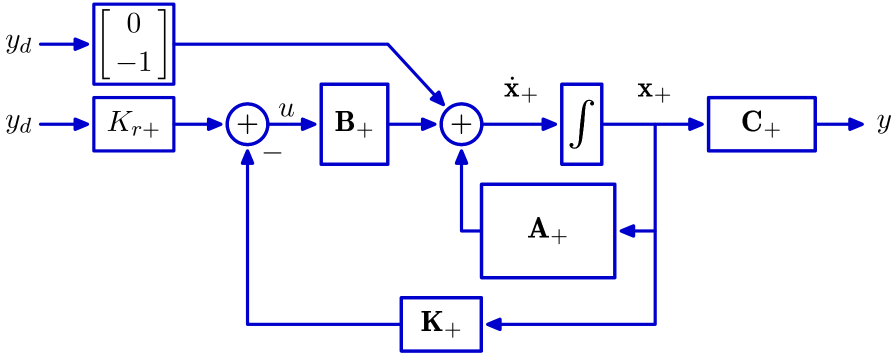
$$\mathbf{x}_+(t) = \begin{bmatrix} \mathbf{x}(t) \\ w(t) \end{bmatrix}$$

Using Feedback to Reduce Tracking Errors

Express both the original system equations and the tracking equation as a single first-order matrix equation in the augmented state.

$$\underbrace{\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{w}(t) \end{bmatrix}}_{\dot{\mathbf{x}}_+} = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & 0 \end{bmatrix}}_{\mathbf{A}_+} \underbrace{\begin{bmatrix} \mathbf{x} \\ w \end{bmatrix}}_{\mathbf{x}_+} + \underbrace{\begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix}}_{\mathbf{B}_+} u(t) + \begin{bmatrix} 0 \\ -1 \end{bmatrix} y_d(t) \ ; \qquad y(t) = \underbrace{\begin{bmatrix} \mathbf{C} & 0 \end{bmatrix}}_{\mathbf{C}_+} \underbrace{\begin{bmatrix} \mathbf{x} \\ w \end{bmatrix}}_{\mathbf{x}_+}$$

The block diagram shows two entry points for $y_d(t)$. Do we need both?

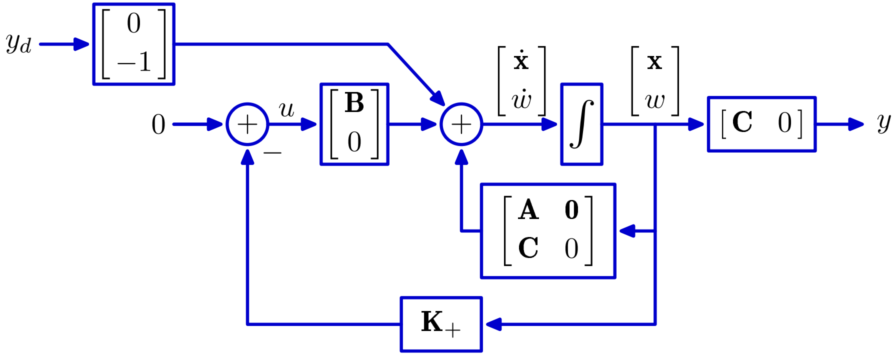


Using Feedback to Reduce Tracking Errors

Express both the original system equations and the tracking equation as a first-order matrix equation in the augmented state.

$$\underbrace{\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{w}(t) \end{bmatrix}}_{\dot{\mathbf{x}}_+} = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & 0 \end{bmatrix}}_{\mathbf{A}_+} \underbrace{\begin{bmatrix} \mathbf{x} \\ w \end{bmatrix}}_{\mathbf{x}_+} + \underbrace{\begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix}}_{\mathbf{B}_+} u(t) + \begin{bmatrix} 0 \\ -1 \end{bmatrix} y_d(t) \; ; \qquad y(t) = \underbrace{\begin{bmatrix} \mathbf{C} & 0 \end{bmatrix}}_{\mathbf{C}_+} \underbrace{\begin{bmatrix} \mathbf{x} \\ w \end{bmatrix}}_{\mathbf{x}_+}$$

Retain only upper y_d path; write augmented matrices as composites.

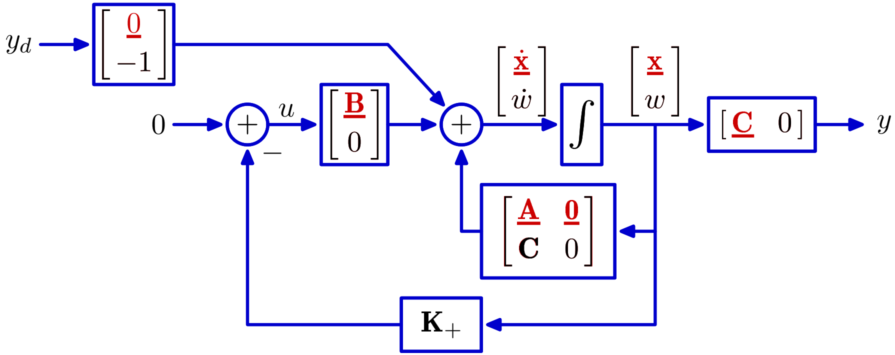


Using Feedback to Reduce Tracking Errors

Express both the original system equations and the tracking equation as a first-order matrix equation in the augmented state.

$$\underbrace{\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{w}(t) \end{bmatrix}}_{\dot{\mathbf{x}}_+} = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & 0 \end{bmatrix}}_{\mathbf{A}_+} \underbrace{\begin{bmatrix} \mathbf{x} \\ w \end{bmatrix}}_{\mathbf{x}_+} + \underbrace{\begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix}}_{\mathbf{B}_+} u(t) + \begin{bmatrix} 0 \\ -1 \end{bmatrix} y_d(t) \; ; \qquad y(t) = \underbrace{\begin{bmatrix} \mathbf{C} & 0 \end{bmatrix}}_{\mathbf{C}_+} \underbrace{\begin{bmatrix} \mathbf{x} \\ w \end{bmatrix}}_{\mathbf{x}_+}$$

Check that the original homogeneous equations are correctly represented.

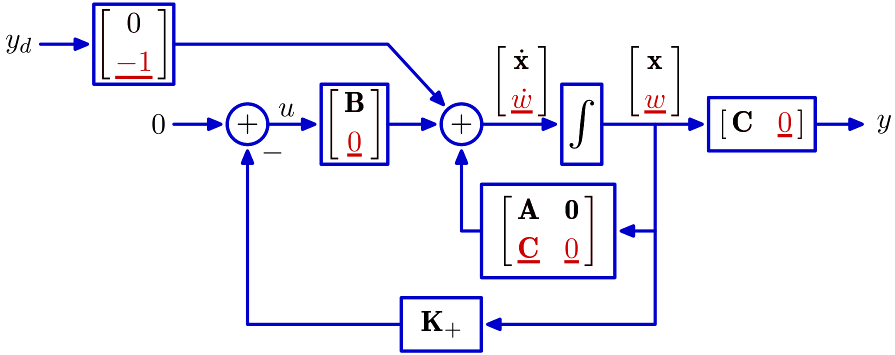


Using Feedback to Reduce Tracking Errors

Express both the original system equations and the tracking equation as a first-order matrix equation in the augmented state.

$$\underbrace{\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{w}(t) \end{bmatrix}}_{\dot{\mathbf{x}}_+} = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & 0 \end{bmatrix}}_{\mathbf{A}_+} \underbrace{\begin{bmatrix} \mathbf{x} \\ w \end{bmatrix}}_{\mathbf{x}_+} + \underbrace{\begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix}}_{\mathbf{B}_+} u(t) + \begin{bmatrix} 0 \\ -1 \end{bmatrix} y_d(t) \; ; \qquad y(t) = \underbrace{\begin{bmatrix} \mathbf{C} & 0 \end{bmatrix}}_{\mathbf{C}_+} \underbrace{\begin{bmatrix} \mathbf{x} \\ w \end{bmatrix}}_{\mathbf{x}_+}$$

Check that the integral equation is correctly represented.

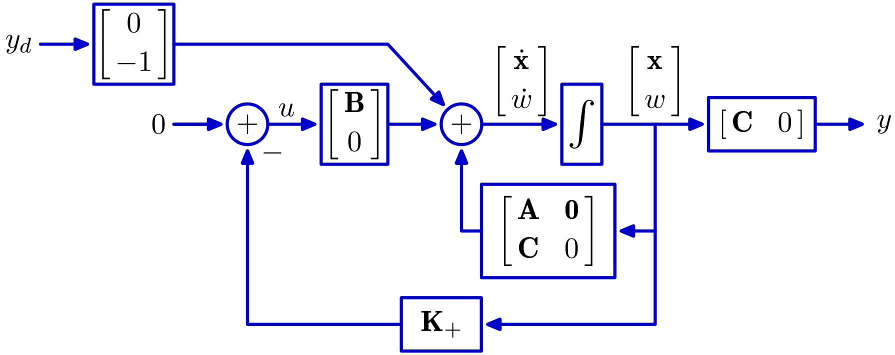


Using Feedback to Reduce Tracking Errors

Express both the original system equations and the tracking equation as a first-order matrix equation in the augmented state.

$$\underbrace{\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{w}(t) \end{bmatrix}}_{\dot{\mathbf{x}}_+} = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & 0 \end{bmatrix}}_{\mathbf{A}_+} \underbrace{\begin{bmatrix} \mathbf{x} \\ w \end{bmatrix}}_{\mathbf{x}_+} + \underbrace{\begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix}}_{\mathbf{B}_+} u(t) + \begin{bmatrix} 0 \\ -1 \end{bmatrix} y_d(t) \; ; \qquad y(t) = \underbrace{[\, \mathbf{C} \quad 0 \,]}_{\mathbf{C}_+} \underbrace{\begin{bmatrix} \mathbf{x} \\ w \end{bmatrix}}_{\mathbf{x}_+}$$

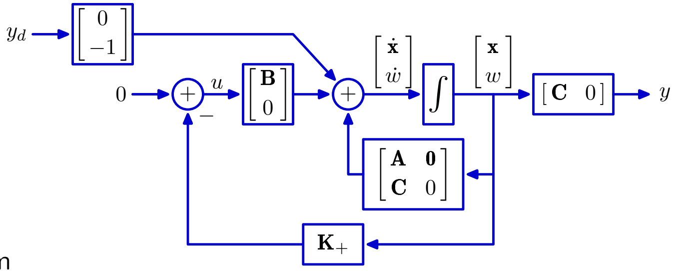
Can we replace $\mathbf{K}_+$ with $[\, \mathbf{K} \quad K_{int} \,]$?



Check Yourself

Express the following system

$$\underbrace{\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{w}(t) \end{bmatrix}}_{\dot{\mathbf{x}}_+} = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & 0 \end{bmatrix}}_{\mathbf{A}_+} \underbrace{\begin{bmatrix} \mathbf{x} \\ w \end{bmatrix}}_{\mathbf{x}_+} + \underbrace{\begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix}}_{\mathbf{B}_+} u(t) + \begin{bmatrix} 0 \\ -1 \end{bmatrix} y_d(t) \; ; \quad y(t) = \underbrace{\begin{bmatrix} \mathbf{C} & 0 \end{bmatrix}}_{\mathbf{C}_+} \underbrace{\begin{bmatrix} \mathbf{x} \\ w \end{bmatrix}}_{\mathbf{x}_+}$$



in the form

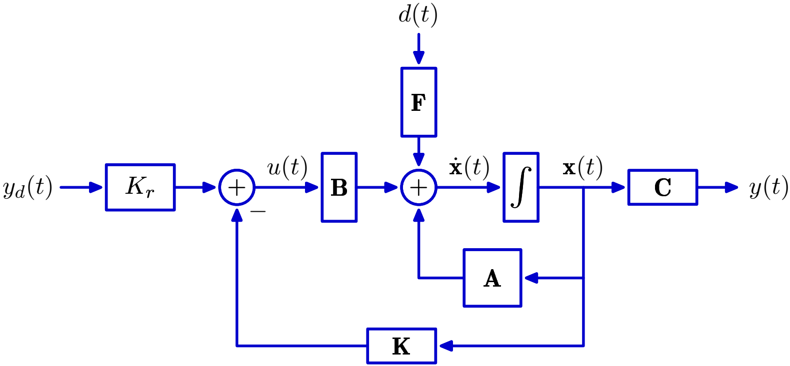
$$\dot{\mathbf{x}}_+(t) = \mathbf{A_{clp}}\, \mathbf{x}_+(t) + \mathbf{B_{clp}}\, y_d(t)$$

Which (if any) of the following definitions are true?

1. $\mathbf{A_{clp}} = \mathbf{A}_+ - \mathbf{B}_+\mathbf{K}_+$
2. $\mathbf{A_{clp}} = \mathbf{A} - \mathbf{BK}$
3. $\mathbf{B_{clp}} = [0, -1]^T$
4. $\mathbf{B_{clp}} = [\mathbf{B}, 0]^T$
5. $\mathbf{B_{clp}} = \mathbf{BK}$

Disturbances

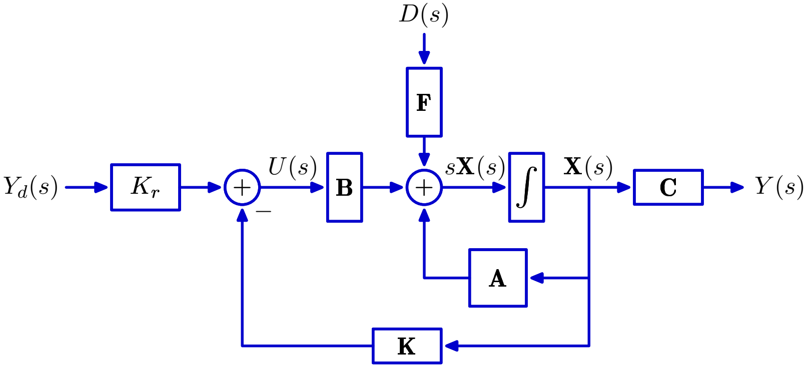
Disturbance $d(t)$ adds to the value of $\dot{x}(t)$ as shown below.



What's the size of $\mathbf{F}$? What do the entries in $\mathbf{F}$ represent?

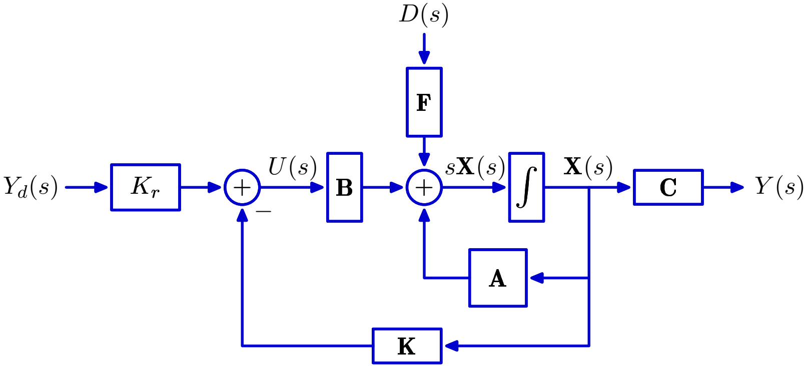
Disturbances

Let $H(s)$ represent the transfer function from $Y_d(s)$ to $Y(s)$ when $D(s) = 0$. Find a linear algebraic expression for $H(s)$ in terms of the matrices below.

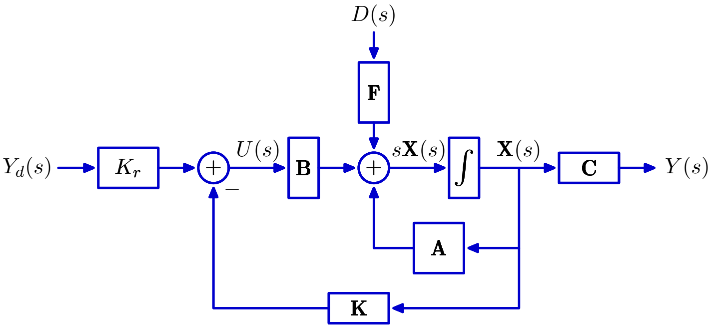


Disturbances

Let $G(s)$ represent the transfer function from $D(s)$ to $Y(s)$ when $Y_d(s) = 0$. Find a linear algebraic expression for $G(s)$ in terms of the matrices below.



Check Yourself



$$H(s) = \frac{Y(s)}{Y_d(s)} = \mathbf{C}\Big(s\mathbf{I}-(\mathbf{A}-\mathbf{B}\mathbf{K})\Big)^{-1}\mathbf{B}K_r$$

$$G(s) = \frac{Y(s)}{D(s)} = \mathbf{C}\Big(s\mathbf{I}-(\mathbf{A}-\mathbf{B}\mathbf{K})\Big)^{-1}\mathbf{F}$$

Which statements are true if only the first component of $\mathbf{F}$ is nonzero?

1. $G(s)$ represents a disturbance applied to $\dot{x}_1$ and observed at $y(t)$.
2. $G(s)$ represents a disturbance applied to d and observed at $x_1(t)$.
3. Only $x_1(t)$ is affected by a disturbance $d(t)$.
4. $G(s)$ and $H(s)$ have the same poles.

Next Time

Observer-based control.