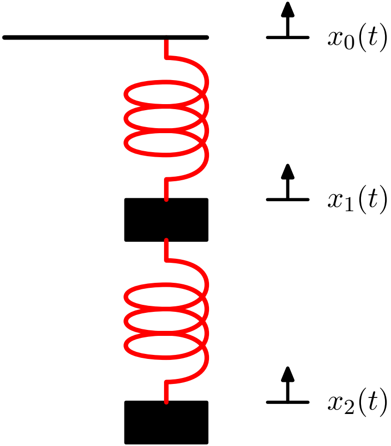
6.3100: Dynamic System Modeling and Control Design

Controlling a System with an Observer

April 29, 2024

Two-Spring System

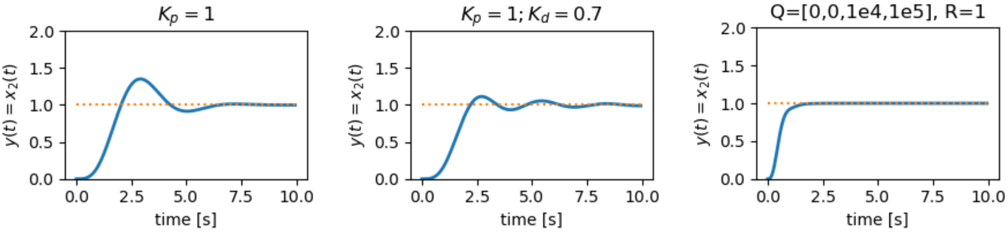
Last time, we developed classical and state-space controllers for a two-spring system.



The goal was to move the input $u(t) = x_0(t)$ so as to position the bottom mass $y(t) = x_2(t)$ at some desired location $y_d(t)$.

Comparison of Control Schemes

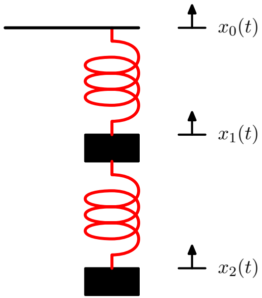
The state-space approach provided **much better performance** than either the proportional or proportional-plus-derivative approach.



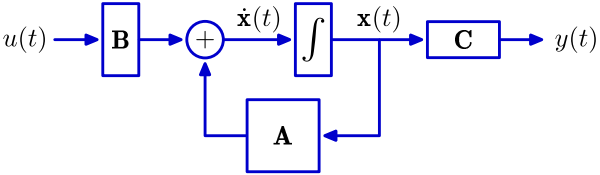
Why is state-space approach so much better?

State-Space Model

The observer-based approach builds on the state-space approach in which the plant is represented as $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$ matrices.



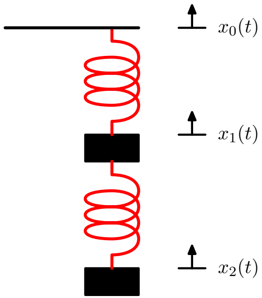
$$f_{m1} = m\ddot{x}_1(t) = k\Big(x_0(t) - x_1(t)\Big) - k\Big(x_1(t) - x_2(t)\Big) - b\dot{x}_1(t)$$
$$f_{m2} = m\ddot{x}_2(t) = k\Big(x_1(t) - x_2(t)\Big) - b\dot{x}_2(t)$$



Check Yourself

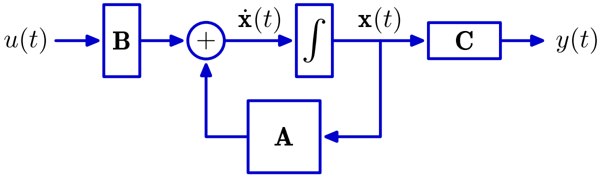
Find **A**, **B**, and **C** so that $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$ and $y = \mathbf{C}\mathbf{x}$.

How many non-zero entries are in $\mathbf{A}$?



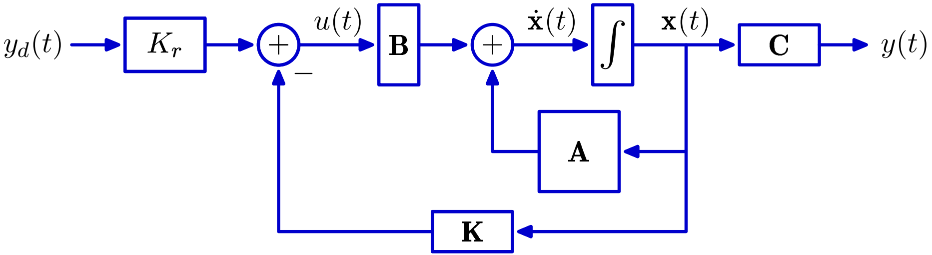
$$f_{m1} = m\ddot{x}_1(t) = k\Big(x_0(t) - x_1(t)\Big) - k\Big(x_1(t) - x_2(t)\Big) - b\dot{x}_1(t)$$

$$f_{m2} = m\ddot{x}_2(t) = k\Big(x_1(t) - x_2(t)\Big) - b\dot{x}_2(t)$$



State-Space Controller

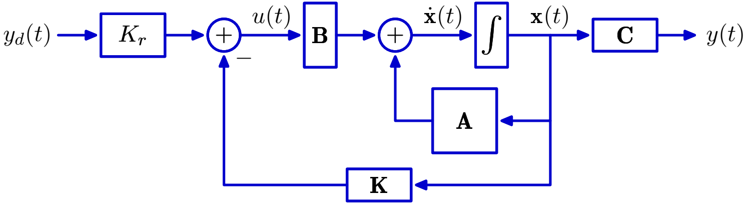
A state-space **controller** can then be expressed as follows.



How do we find $\mathbf{K}$ and K_r?

Check Yourself

Assume that we will **implement the controller** with a microprocessor.



Which pseudo-code snippet best describes a function `step` whose input is the current state `x` and output is the next command `u`?

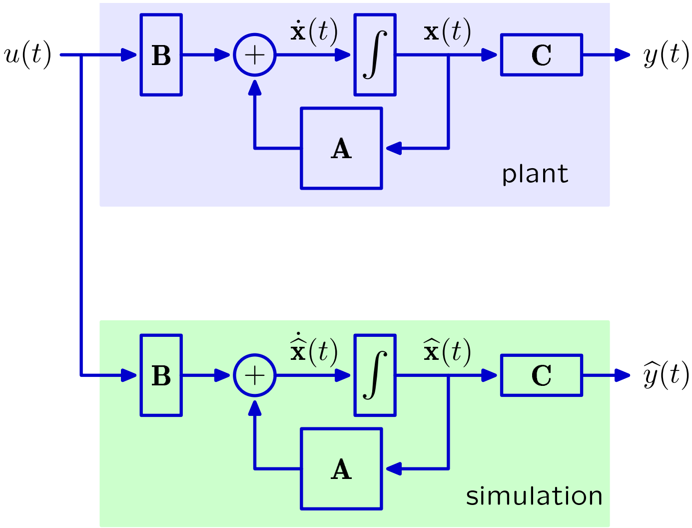
```
def step_v1(x):
    x += e**((A-B*K)*DeltaT)*x + B*Kr*yd
    return Kr*yd-K*x
def step_v2(x):
    x += (I+(A-B*K)*DeltaT)*x + B*Kr*yd
    return Kr*yd-K*x
def step_v3(x):
    return Kr*yd-K*x
```

Observers

By contrast, observer-based controllers **explicitly** depend on $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$.

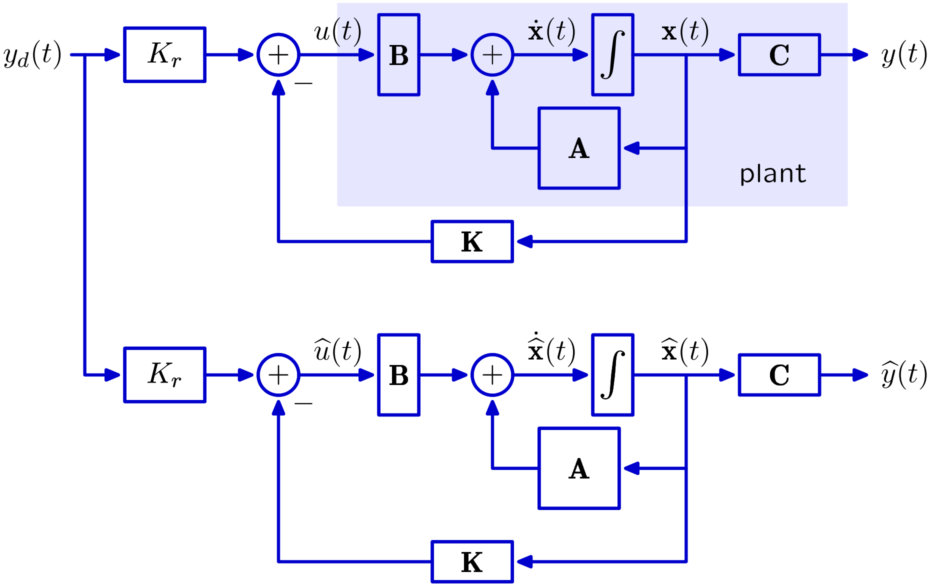
Observers (Recap)

An **observer** is a **simulation** of the plant that is used by the controller – i.e., the simulation is part of the controller!



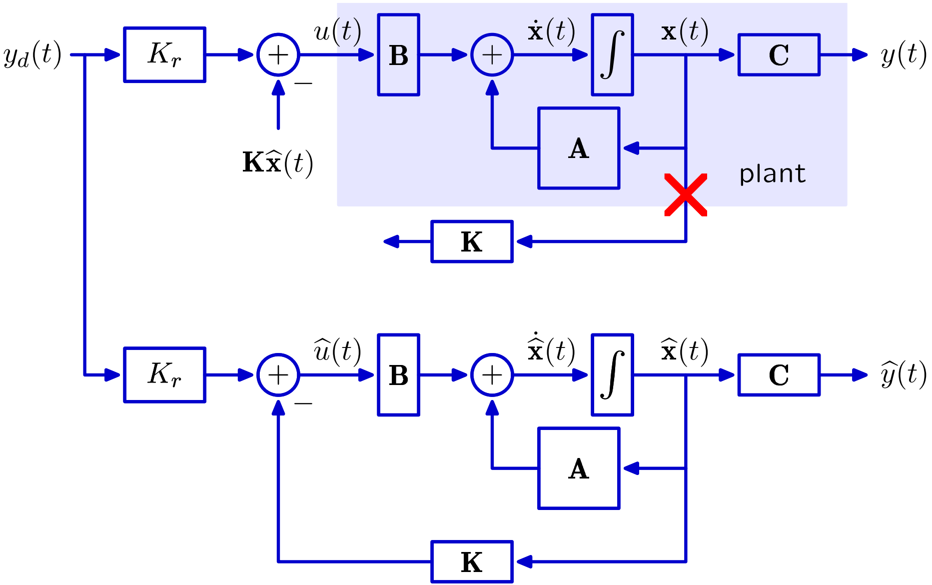
Observers (Recap)

We can build state-space controllers for both the plant and the simulation.



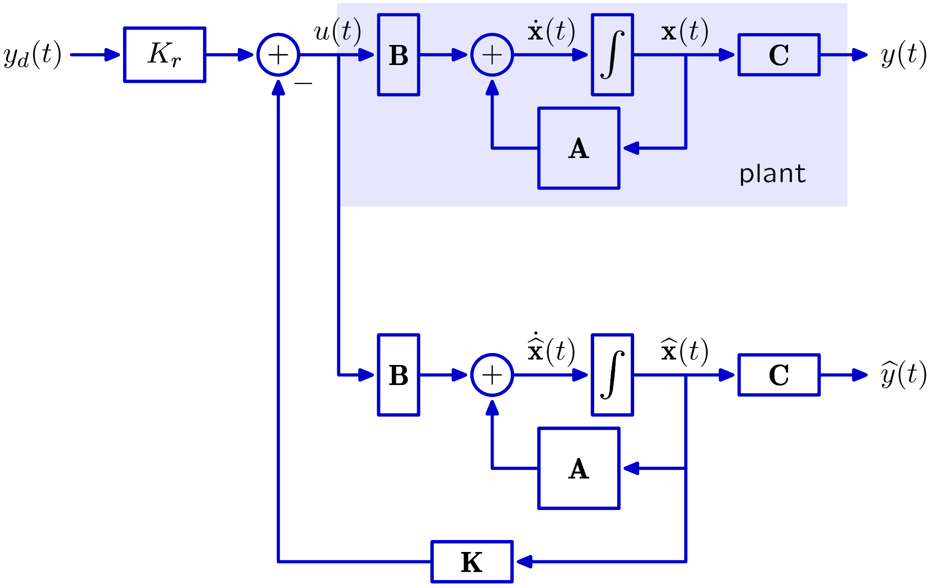
Observers (Recap)

If our model of the plant (**A**, **B**, **C**) is perfect, then $\widehat{\mathbf{x}}(t) = \mathbf{x}(t)$ and we can replace $\mathbf{K}\mathbf{x}(t)$ with $\mathbf{K}\widehat{\mathbf{x}}(t)$. This substitution also makes $u(t) = \widehat{u}(t)$.



Observers (Recap)

The resulting structure provides feedback from all **simulated** states $\widehat{\mathbf{x}}(t)$. Unfortunately even small differences between the plant and simulation can lead to large differences between $\mathbf{x}(t)$ and $\widehat{\mathbf{x}}(t)$.

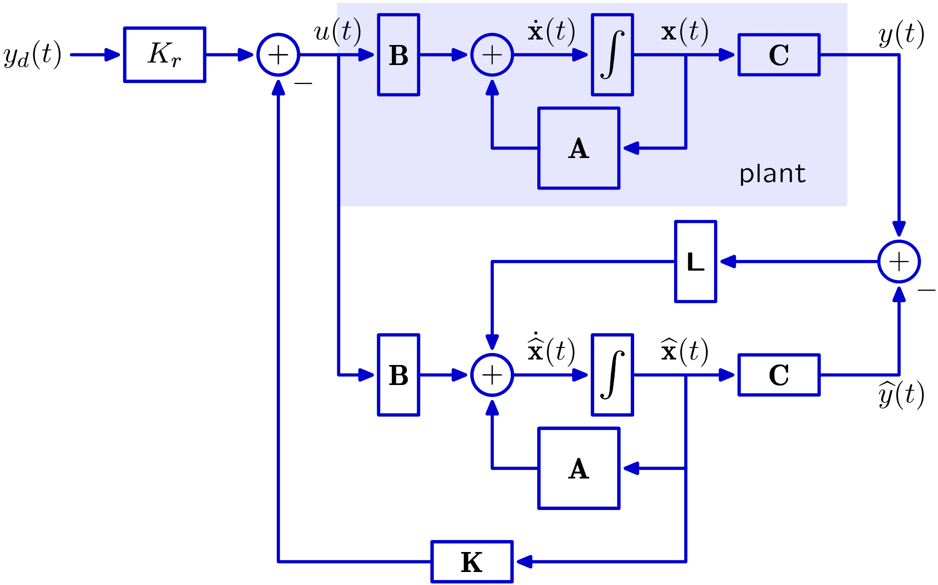


Observers (Recap)

Fortunately, we can use **feedback** to correct simulation errors!

Calculate the difference between $y(t)$ and $\widehat{y}(t)$.

Then use that signal (times $\mathbf{L}$) to correct $\dot{\widehat{\mathbf{x}}}(t)$.



Observers

Dynamics:

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

$$\dot{\widehat{\mathbf{x}}}(t) = \mathbf{A}\widehat{\mathbf{x}}(t) - \mathbf{B}\mathbf{K}\widehat{\mathbf{x}}(t) + \mathbf{B}K_r y_d(t) + \mathbf{L}\Big(y(t) - \widehat{y}(t)\Big)$$

Matrix form:

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\widehat{\mathbf{x}}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & -\mathbf{B}\mathbf{K} \\ \mathbf{L}\mathbf{C} & \mathbf{A}-\mathbf{L}\mathbf{C}-\mathbf{B}\mathbf{K} \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \widehat{\mathbf{x}}(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{B} \end{bmatrix} K_r y_d(t)$$

$$\begin{bmatrix} y(t) \\ \widehat{y}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \widehat{\mathbf{x}}(t) \end{bmatrix}$$

Choose $\mathbf{L}$ to optimize the eigenvalues of $\mathbf{A}^T - \mathbf{C}^T\mathbf{L}^T$.

Choose $\mathbf{K}$ to optimize the eigenvalues of $\mathbf{A} - \mathbf{B}\mathbf{K}$.

```
L = place(A.',C.',[poles]).'
K = place(A,B,[poles])
```

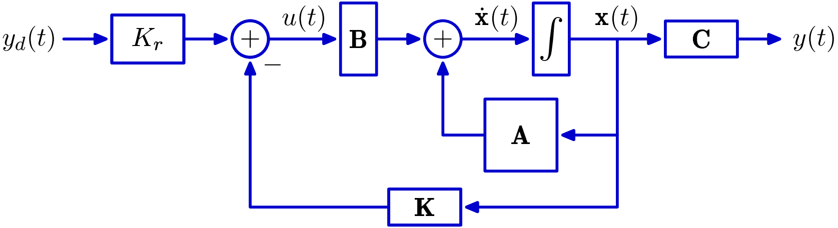
or

```
L = lqr(A.',C.',Q,R).'
K = lqr(A,B,Q,R)
```

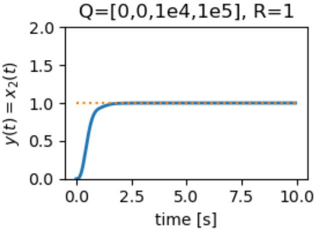
Simulating an Observer-Based Control System

If the simulation ($\mathbf{A}, \mathbf{B}, \mathbf{C}$, and $\mathbf{D}$) matches the plant **exactly** then the observer-based system is equivalent to full-state, state-space feedback.

Full-state, state-space feedback:

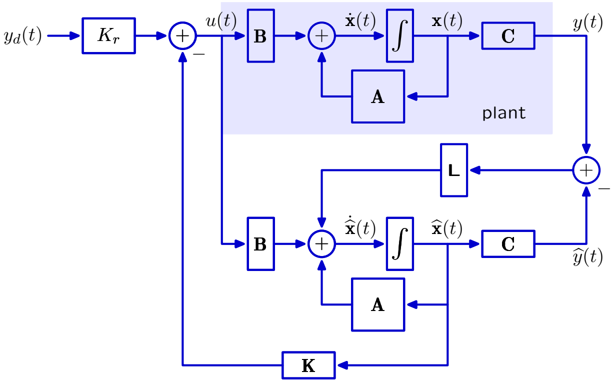


Step response:

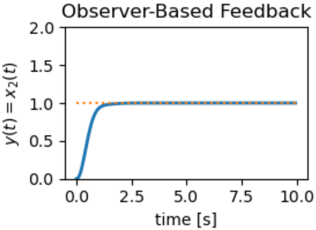


Simulating an Observer-Based Control System

Observer-based feedback:

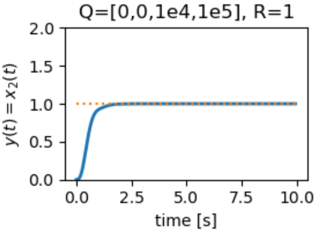


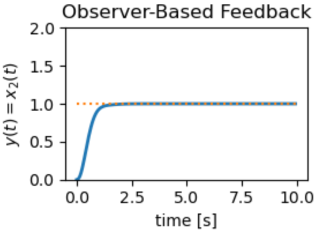
Step response:



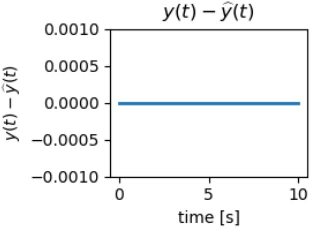
Simulating an Observer-Based Control System

If the simulation $(\mathbf{A}, \mathbf{B}, \mathbf{C}$, and $\mathbf{D})$ matches the plant **exactly** then the observer-based system is equivalent to full-state, state-space feedback.



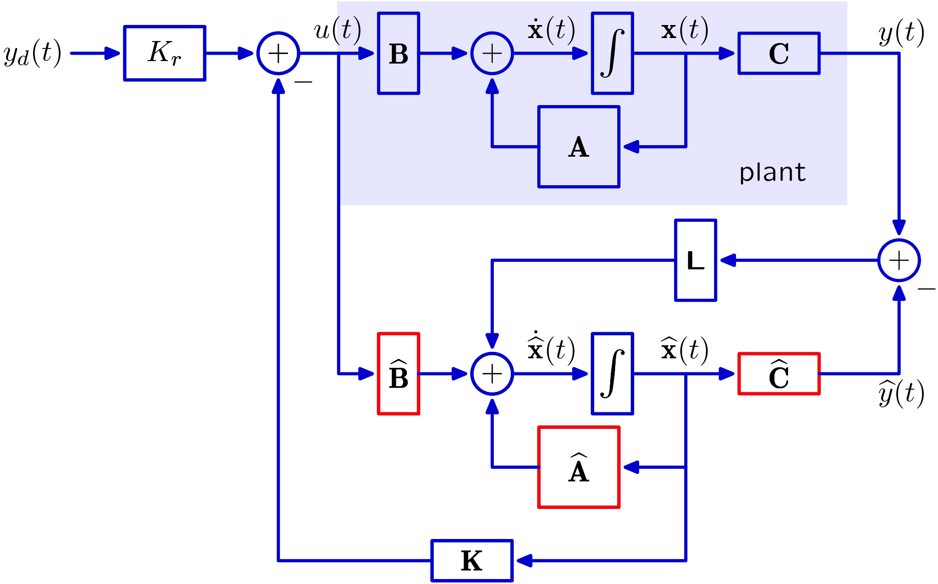


Even more compellingly, the difference between the measured and simulated outputs is very close to zero.



Parameter Errors

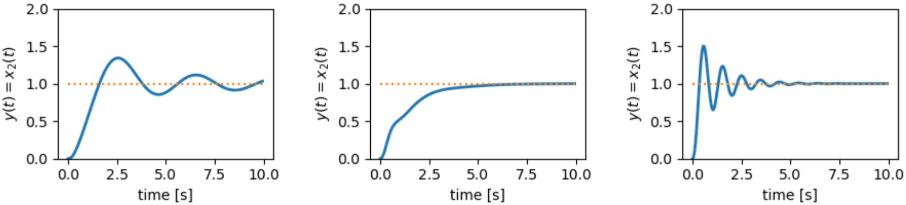
What happens if the simulation does not accurately represent the plant?



Check Yourself

What if the simulation does not accurately represent the plant?

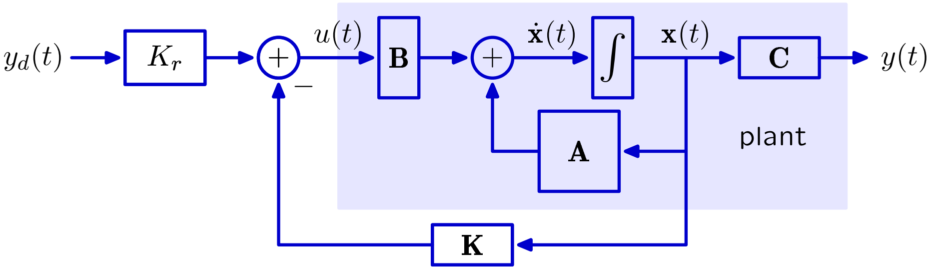
Which plot (if any) shows results when the model of the springs is not as stiff as the physical springs in the plant?



1. left panel
2. center panel
3. right panel
4. none of the above

Noise Performance

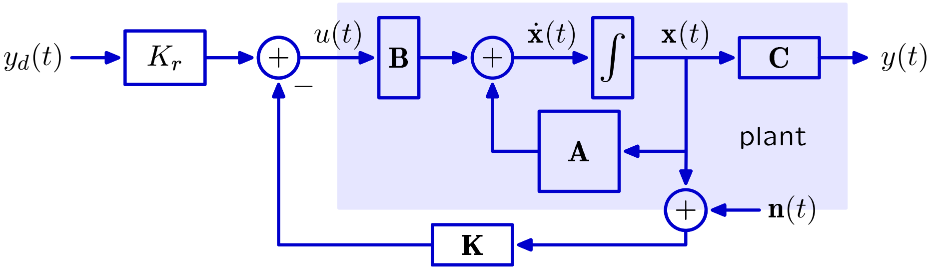
Feedback control can be significantly degraded by noise that is introduced by the sensors that provide information about the plant to the controller.



Effects of Sensor Noise

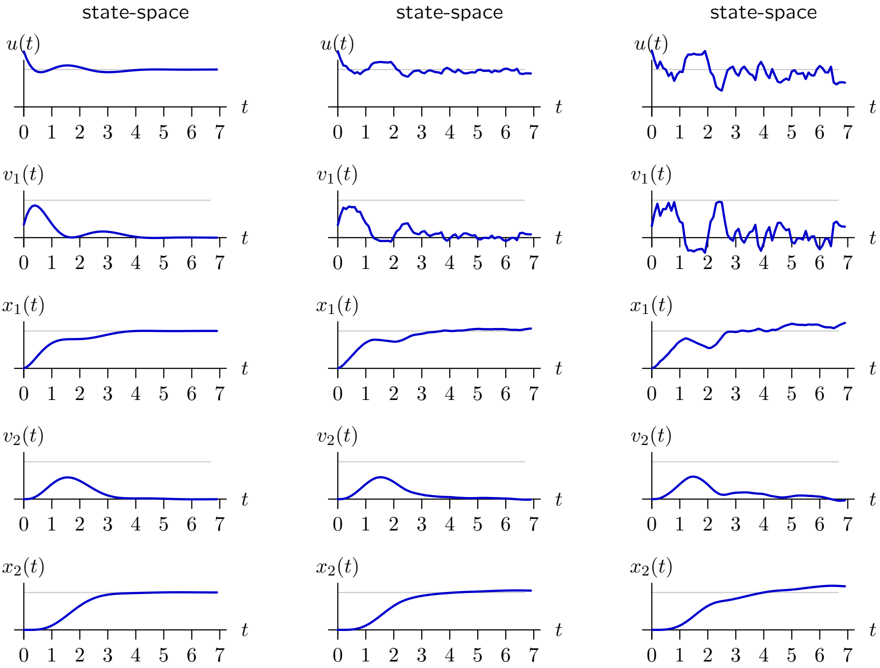
Consider the effect of sensor noise on full-state state-space control.

How will this noise affect performance?



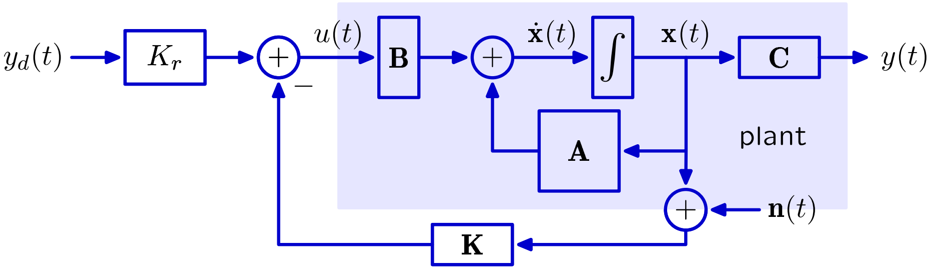
Check Yourself

Effects of noise are greater for mass 1 than for mass 2.



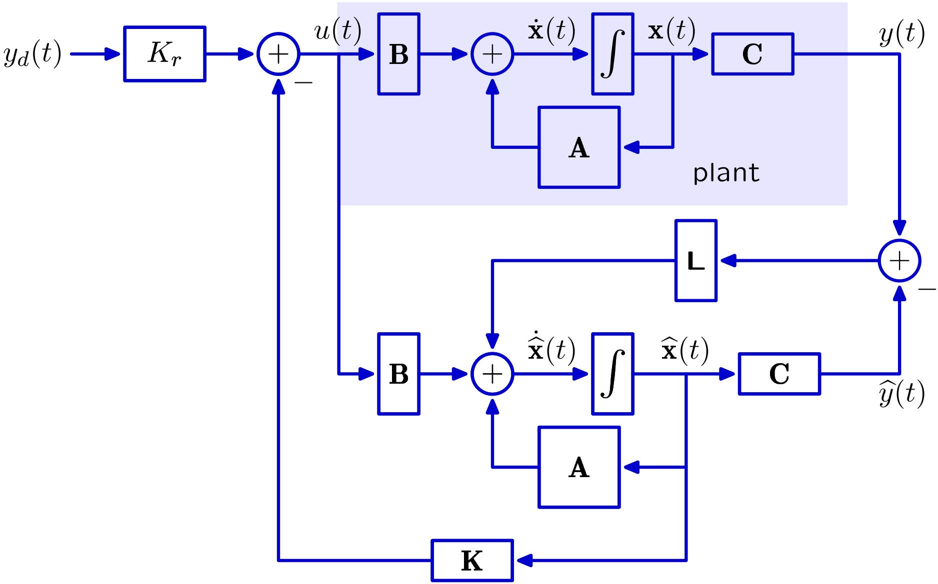
Check Yourself

Why are effects of noise greater for mass 1 than for mass 2?



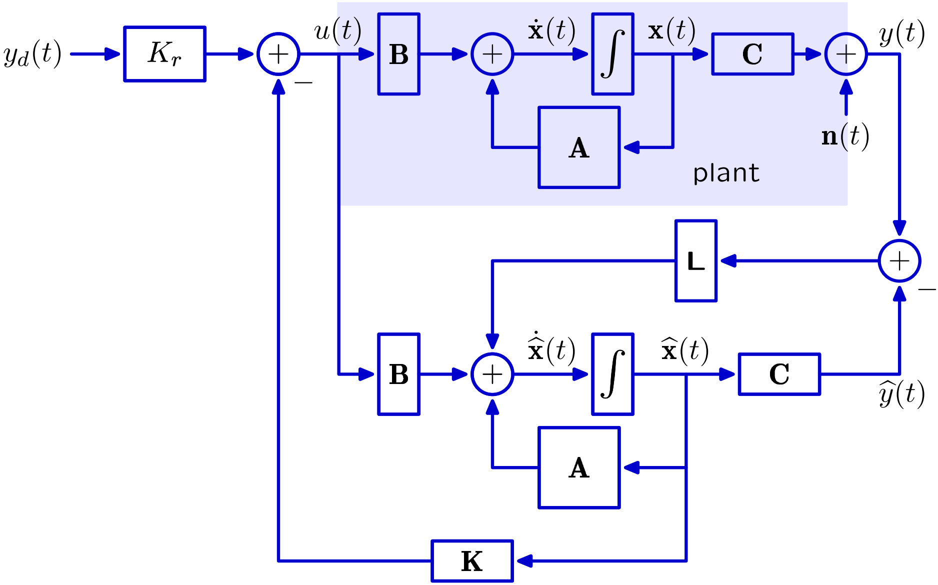
Effects of Sensor Noise

How should we model sensor noise with an observer?



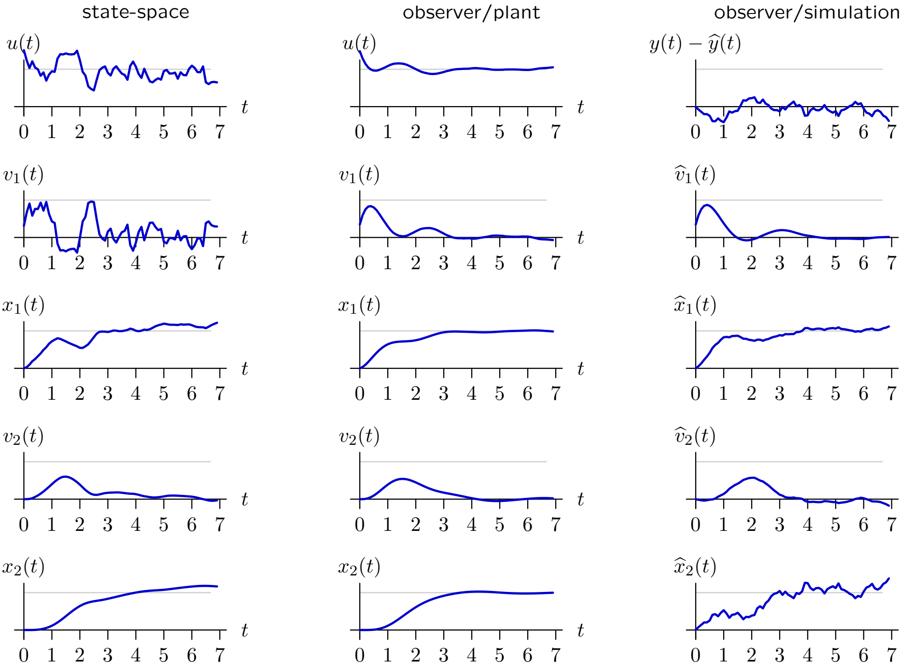
Effects of Sensor Noise

How will this noise affect performance of the control system?



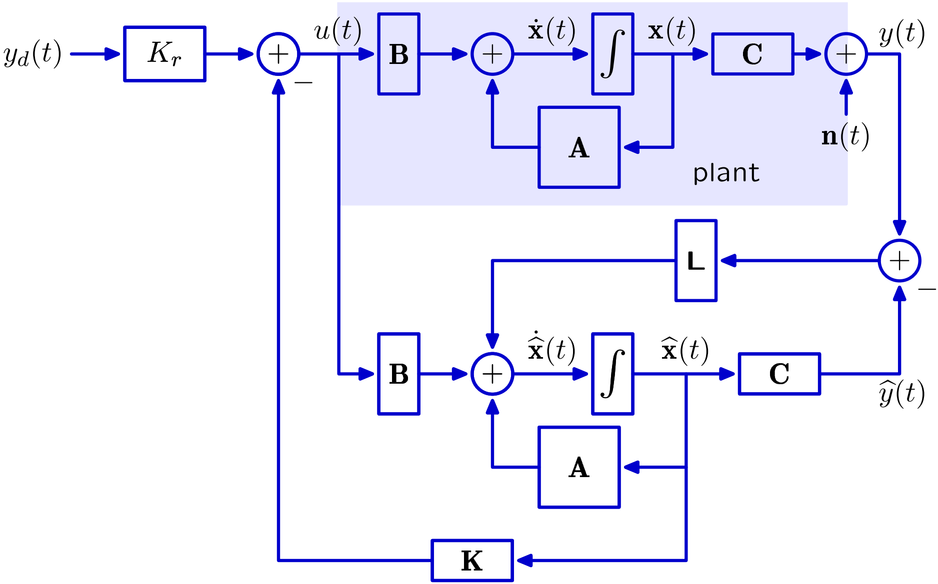
Results

Why is there a lot of noise in $\widehat{x}_2$ but not x_2?



Effects of Sensor Noise

Why is there a lot of noise in $\widehat{x}_2$ but not x_2?



Summary

Full-state, state-space controllers have access to more information and can thereby be more effect than proportional, PD, or PID controllers.

Full-state, state-space controllers require sensors for every state, which may not be feasible in many real-life systems.

Observers use a simulation of the plant to provide information about states without physically measuring them.