6.3100: Dynamic System Modeling and Control Design

Controlling a System with an Observer

April 26, 2023

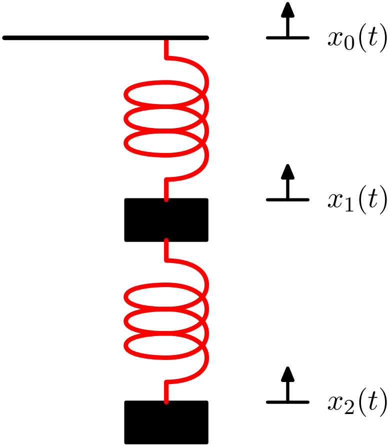
Controlling a System with an Observer

Today we will introduce a new method of control based on **observers**.

To see how this new method builds on previous ideas, let's consider all of these methods in the context of a particular problem.

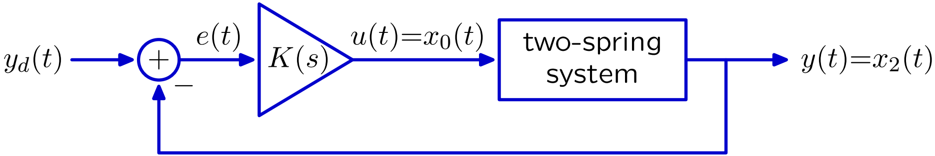
Two-Spring System

The **plant** consists of two springs and two masses. The goal is 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 $y_d(t)$.



Classical Control

A classical controller for this problem has the following form.

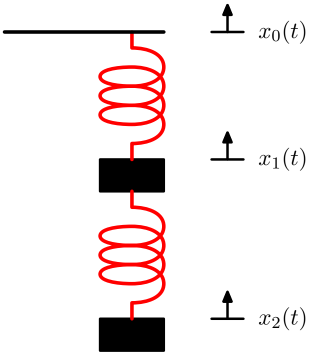


To solve this classical control problem, me must

- find the equations of motion for the plant (the two-spring system) and
- express those equations in terms of transfer function.

Two-Spring System

Equations of motion.



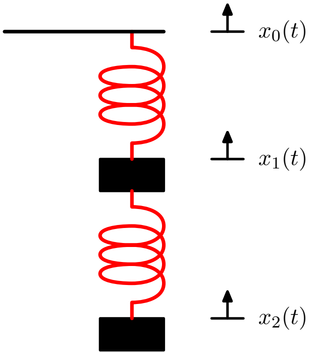
$$f_{m1} = m\ddot{x}_1(t) = k\Big(x_0(t) - x_1(t)\Big) - k\Big(x_1(t) - x_2(t)\Big) - b\dot{x}_1(t) - mg$$

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

Outputs $x_1(t)$ and $x_2(t)$ result from two separable inputs: gravity mg, which generates constant offsets, and $x_0(t)$, which determines the dynamics.

Two-Spring System

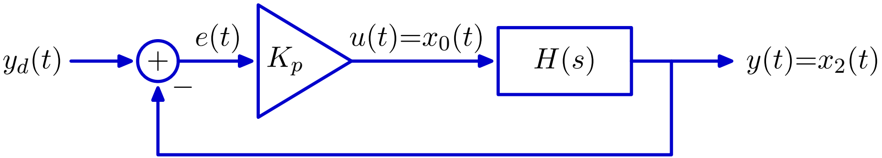
Transfer function.



$$H(s) = \frac{X_2(s)}{X_0(s)} = \frac{k^2}{(s^2m + sb + 2k)(s^2m + sb + k) - k^2}$$

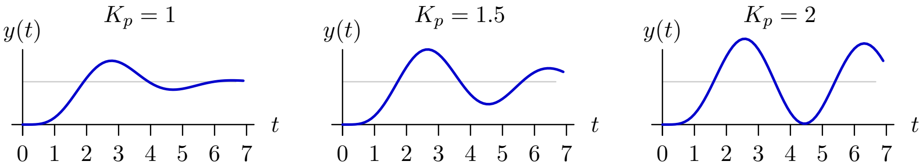
Classical Control

A proportional controller has the following form.



The feedback system is stable for only a small range of K_p: $K_p < 2.5$

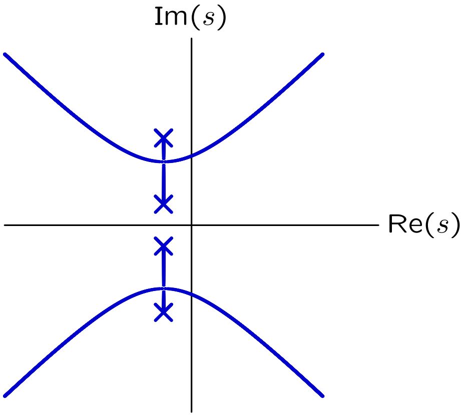
Step responses:



Slow convergence and large oscillatory overshoots.

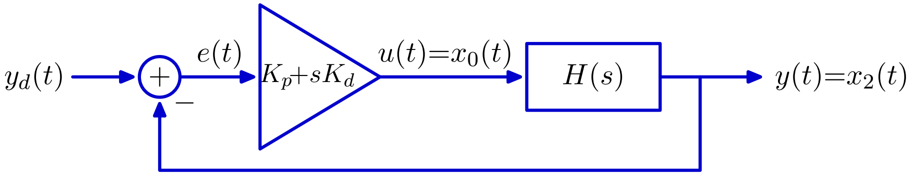
Why such poor behavior?

Root locus.



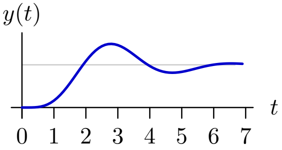
Classical Control

Proportional plus derivative performance is similar to that for proportional.

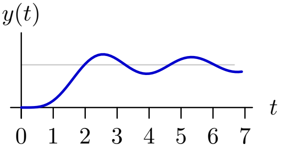


Step responses:

Proportional controller

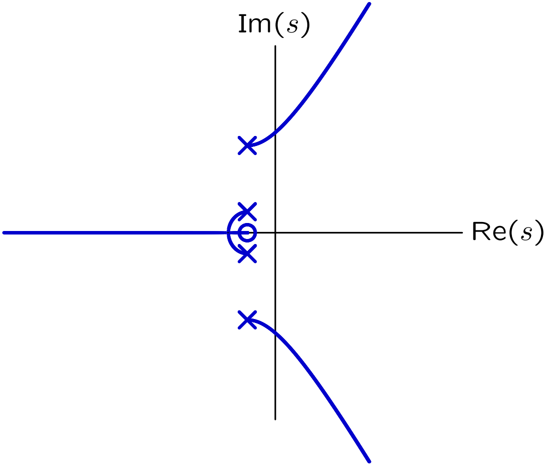


Proportional plus derivative



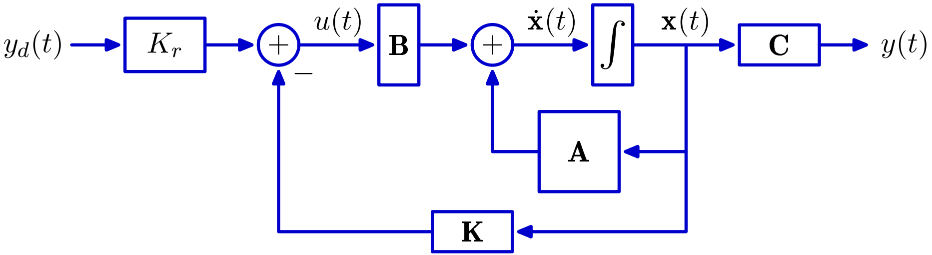
Classical Control

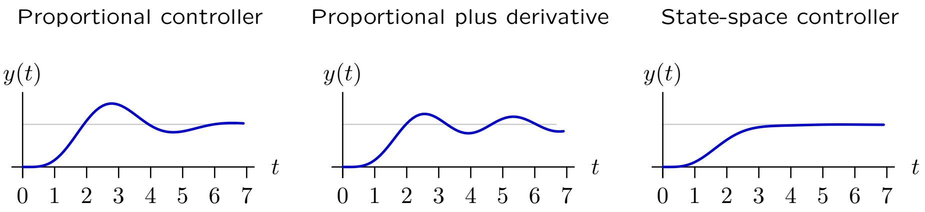
Root locus.



State-Space Control

State-space control is **much** better.



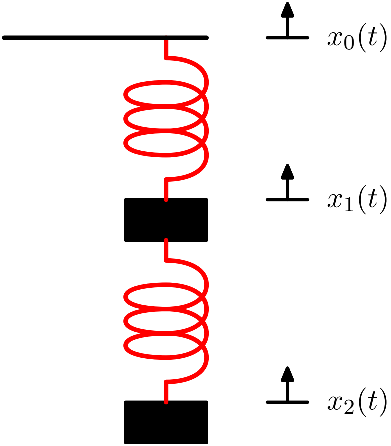


What is it about state-space control that allows better performance?

Two-Spring System

The state-space approach uses information from $x_2(t)$ **and** $x_1(t)$.

The combination of $x_1(t)$ and $x_2(t)$ is much more powerful than $x_2(t)$ alone.



Beyond State-Space Control

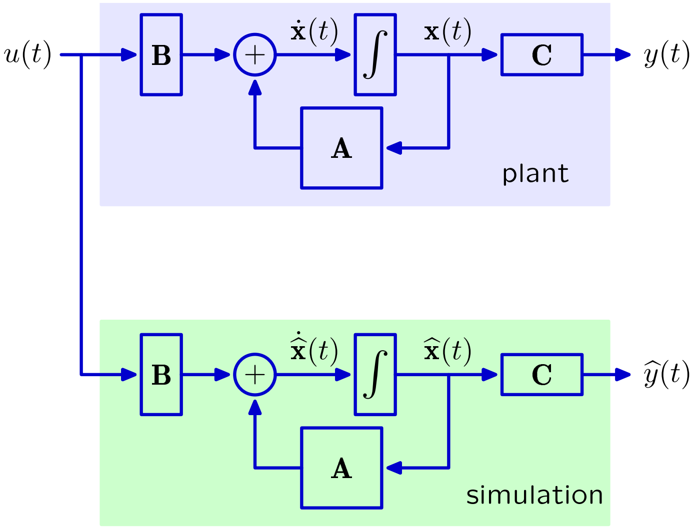
However, to feed back information about $x_1(t)$, we must **measure** $x_1(t)$.

What if it's not possible to measure $x_1(t)$.

Idea: Could we simulate the unmeasured states?

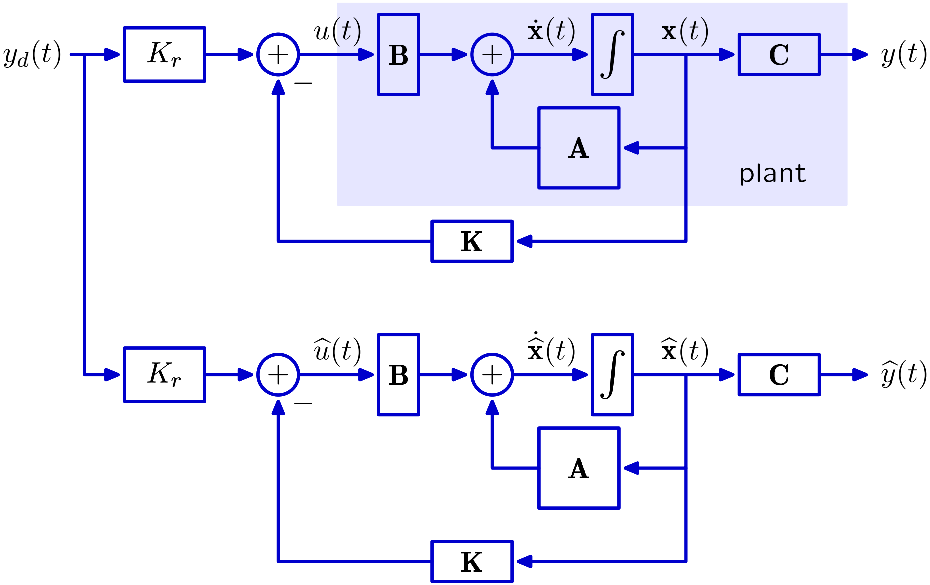
Observers

An **observer** is a **simulation** of the plant that is used to provide information about unmeasured states. This **simulation** will be part of the controller!



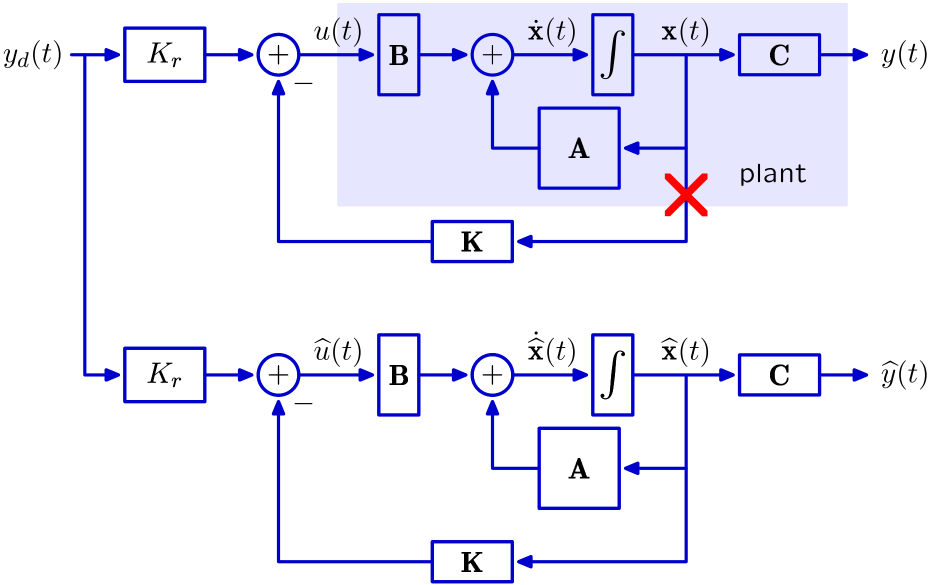
Observers

We can build state-space controllers for both the plant and the simulation. If our model of the plant (**A, B, C**) is perfect, then $\widehat{\mathbf{x}}(t)=\mathbf{x}(t)$ and $\widehat{y}(t)=y(t)$.



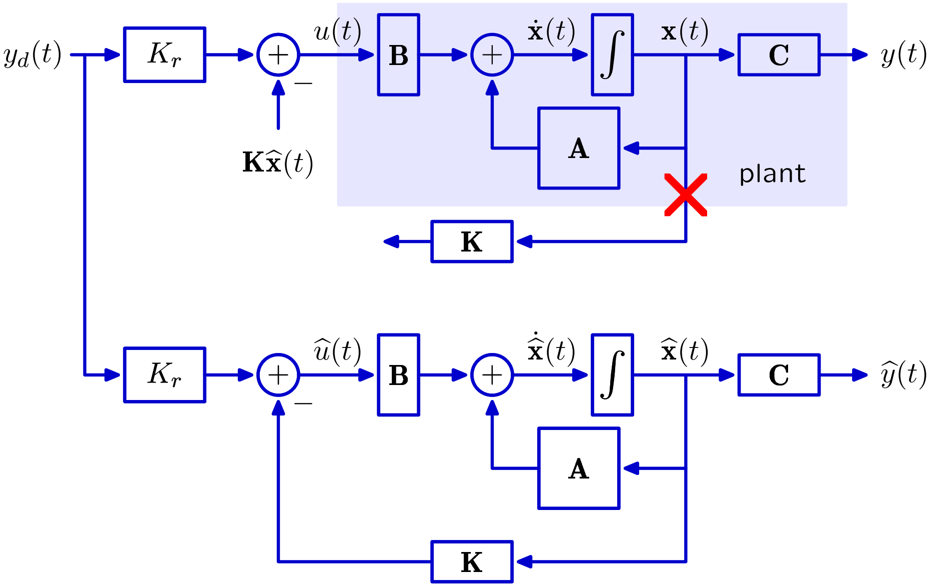
Observers

Recall the problem with designing a state-space controller for the two-springs system: the plant did not provide outputs for all of the states $\mathbf{x}(t)$.



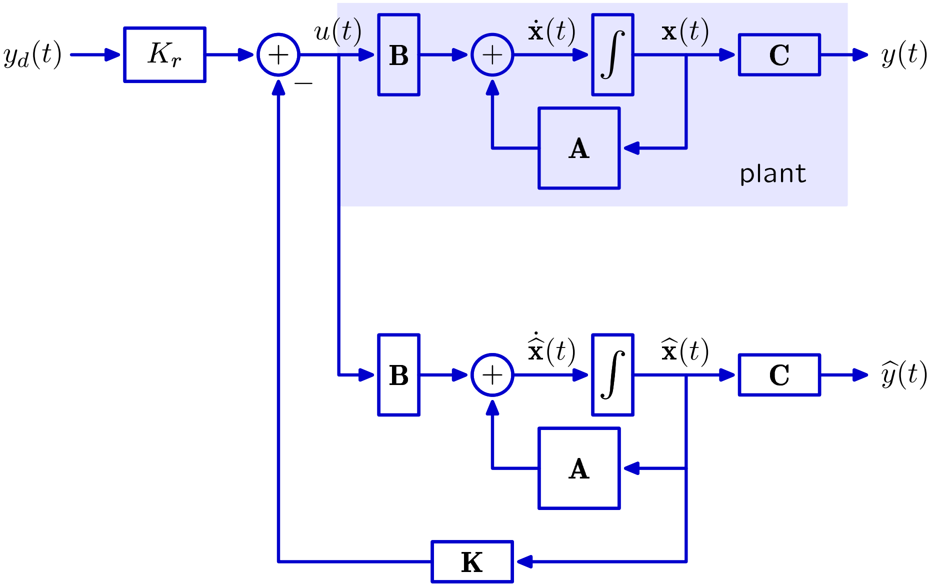
Observers

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



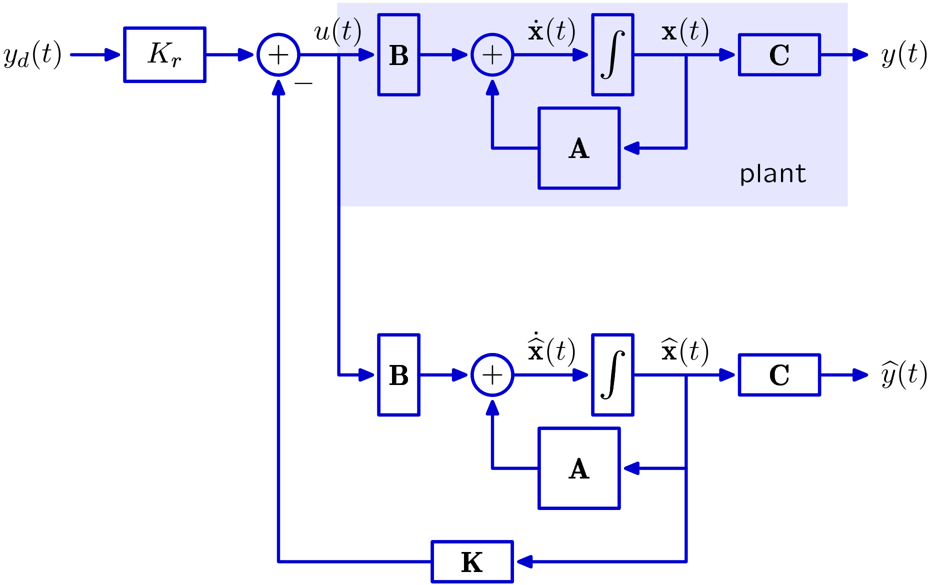
Observers

The resulting structure provides feedback from all **simulated** states $\hat{\mathbf{x}}(t)$. But there is a problem. What's wrong with this scheme?



Observers

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 $\widehat{\mathbf{x}}(t)$.

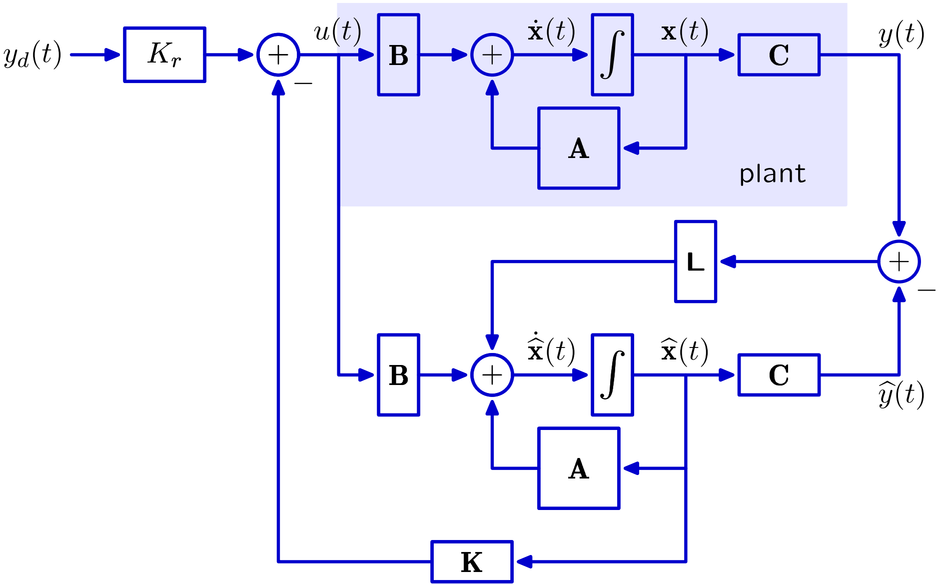


Observers

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

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

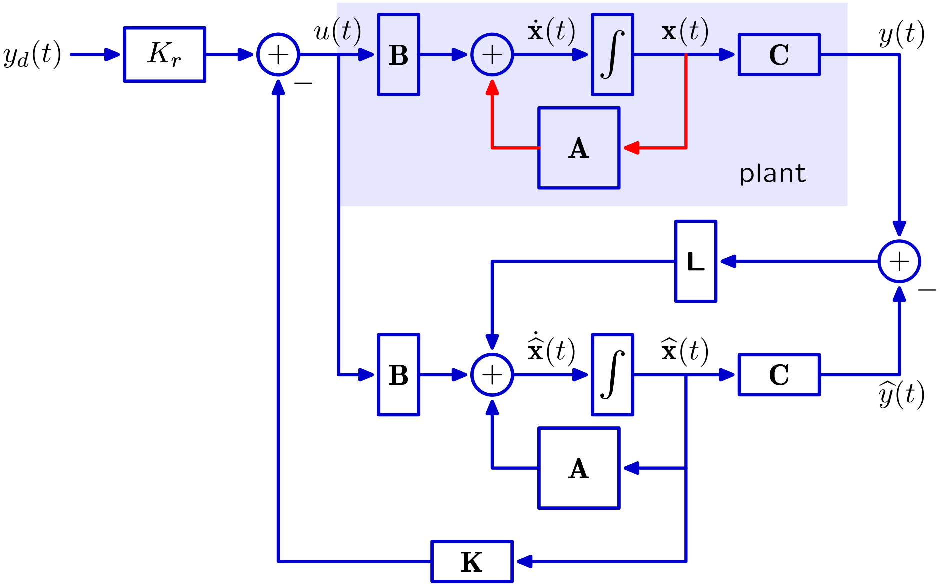
Then use that signal (times $\mathbf{L}$) to correct $\dot{\widehat{\mathbf{x}}}(t)$.



Observers

Plant dynamics:

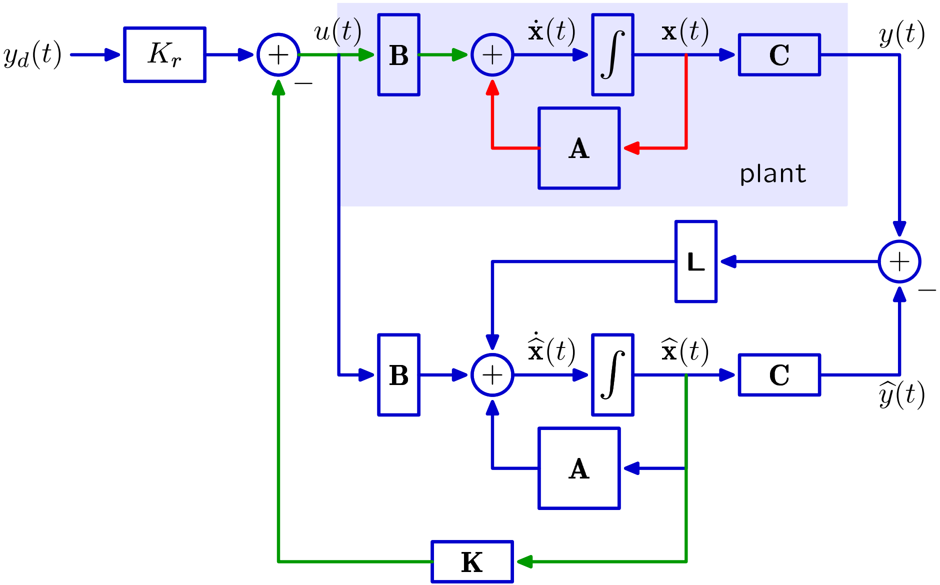
$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) - \mathbf{B}\mathbf{K}\widehat{\mathbf{x}}(t) + \mathbf{B}K_r y_d(t)$$



Observers

Plant dynamics:

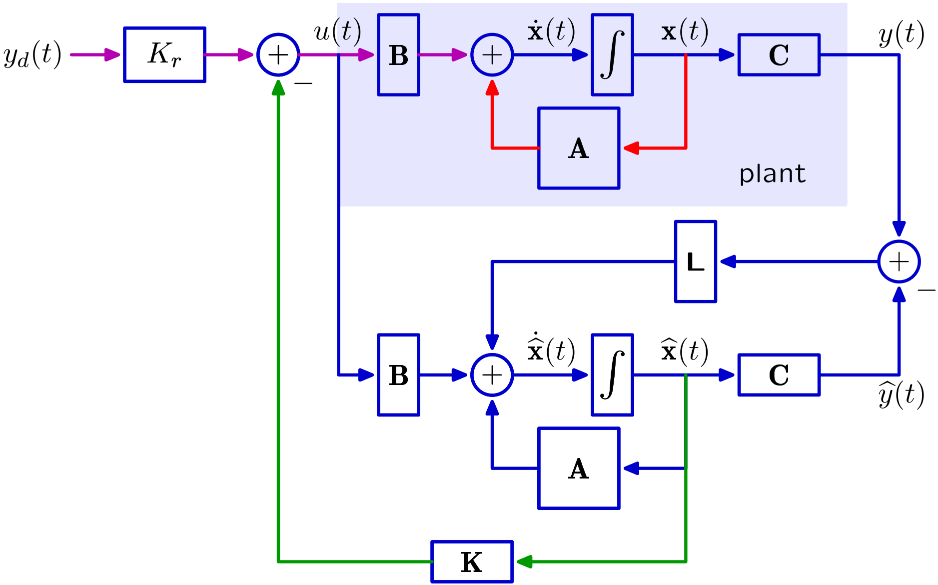
$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) - \mathbf{B}\mathbf{K}\widehat{\mathbf{x}}(t) + \mathbf{B}K_r y_d(t)$$



Observers

Plant dynamics:

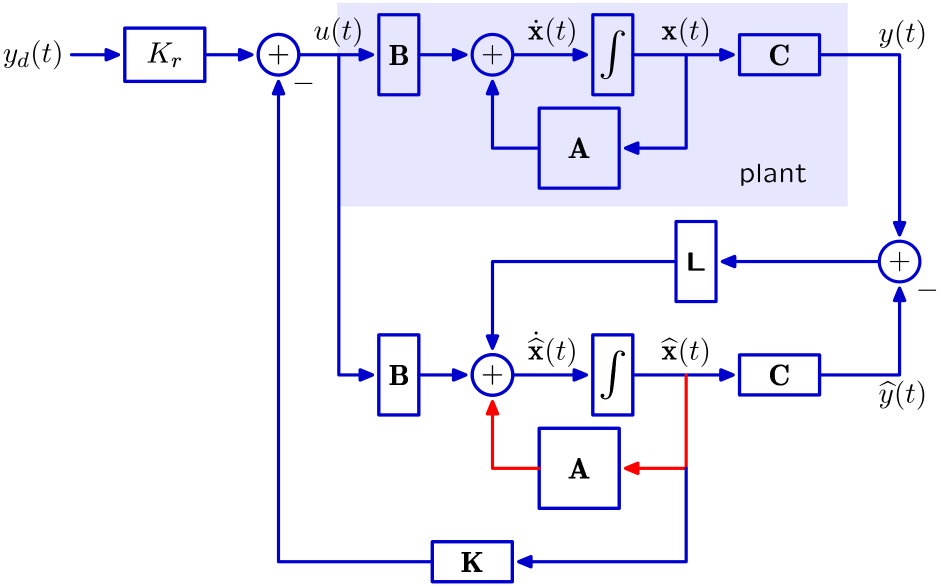
$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) - \mathbf{B}\mathbf{K}\widehat{\mathbf{x}}(t) + \mathbf{B}K_r y_d(t)$$



Observers

Simulation dynamics:

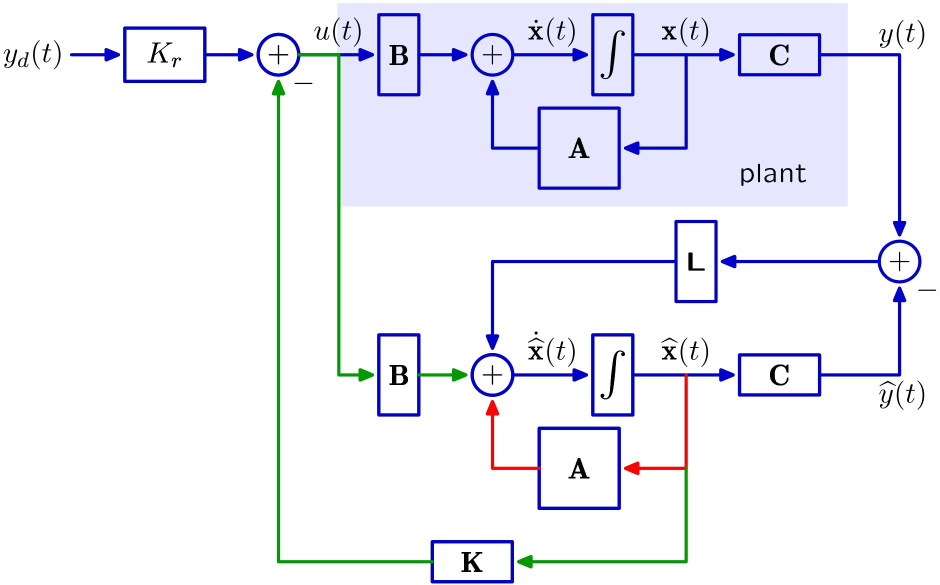
$$\dot{\widehat{\mathbf{x}}}(t) = \mathbf{A}\widehat{\mathbf{x}}(t) - \mathbf{B}\mathbf{K}\widehat{\mathbf{x}}(t) + \mathbf{B}K_r y_d(t) + \mathbf{L}(y(t) - \widehat{y}(t))$$



Observers

Simulation dynamics:

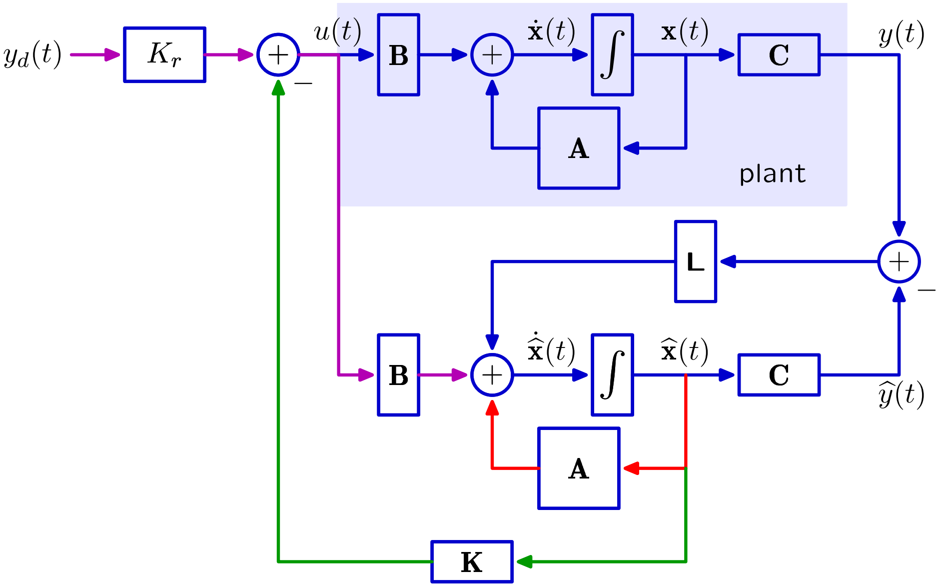
$$\dot{\widehat{\mathbf{x}}}(t) = \mathbf{A}\widehat{\mathbf{x}}(t) - \mathbf{B}\mathbf{K}\widehat{\mathbf{x}}(t) + \mathbf{B}K_r y_d(t) + \mathbf{L}(y(t) - \widehat{y}(t))$$



Observers

Simulation dynamics:

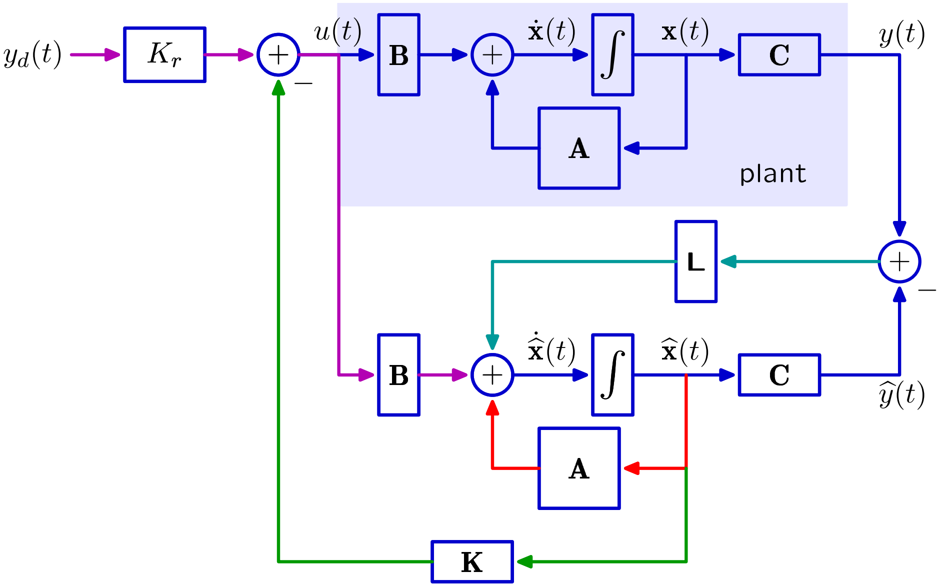
$$\dot{\widehat{\mathbf{x}}}(t) = \mathbf{A}\widehat{\mathbf{x}}(t) - \mathbf{B}\mathbf{K}\widehat{\mathbf{x}}(t) + \mathbf{B}K_r y_d(t) + \mathbf{L}(y(t) - \widehat{y}(t))$$



Observers

Simulation dynamics:

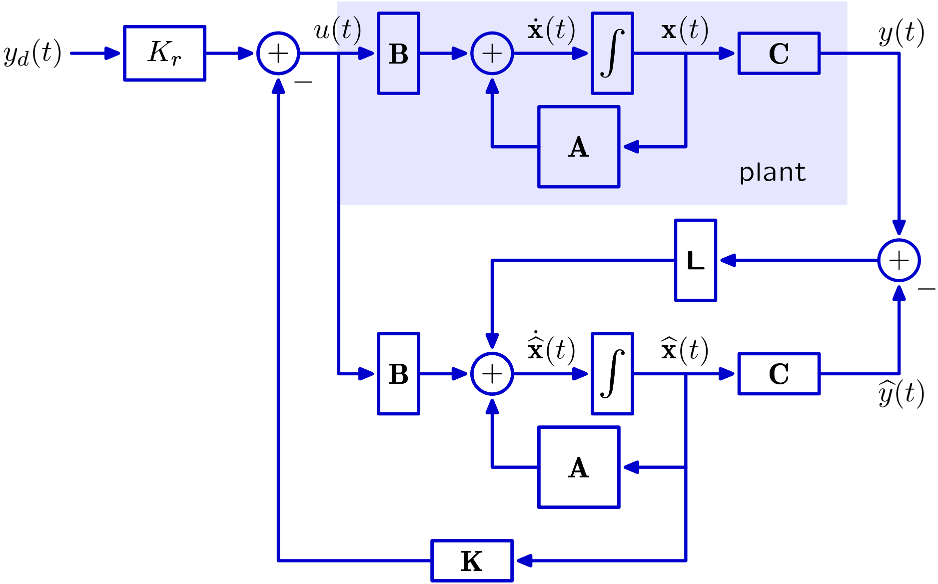
$$\dot{\widehat{\mathbf{x}}}(t) = \mathbf{A}\widehat{\mathbf{x}}(t) - \mathbf{B}\mathbf{K}\widehat{\mathbf{x}}(t) + \mathbf{B}K_r y_d(t) + \mathbf{L}(y(t) - \widehat{y}(t))$$



Observers

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

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



Observers

Dynamics:

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

Define $\mathbf{e}(t)$ to be the difference between the plant and simulation states:

$$\mathbf{e}(t) = \mathbf{x}(t) - \widehat{\mathbf{x}}(t)$$

Subtract $\dot{\widehat{\mathbf{x}}}(t)$ from $\dot{\mathbf{x}}(t)$ to find the derivative of $\mathbf{e}(t)$:

$$\dot{\mathbf{e}}(t) = \mathbf{A}\mathbf{e}(t) - \mathbf{L}\Big(y(t) - \widehat{y}(t)\Big) = \mathbf{A}\mathbf{e}(t) - \mathbf{L}\mathbf{C}\mathbf{e}(t)$$

Append the $\dot{\mathbf{x}}(t)$ and $\dot{\mathbf{e}}(t)$ to make a new **combined** state vector:

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{e}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A}-\mathbf{B}\mathbf{K} & \mathbf{B}\mathbf{K} \\ \mathbf{0} & \mathbf{A}-\mathbf{L}\mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{e}(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} K_r y_d(t)$$

Notice that the resulting matrix equation has the same form as the original state evolution equation:

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

where $\mathbf{A}, \mathbf{B}$, and $\mathbf{x}(t)$ have been extended to include error terms.

Observers

Combined dynamics: state + observer.

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{e}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A}-\mathbf{BK} & \mathbf{BK} \\ \mathbf{0} & \mathbf{A}-\mathbf{LC} \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{e}(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} K_r y_d(t)$$

Because the evolution matrix has a Block Upper Triangular form, its determinant (and therefore the corresponding poles) are the union of those of $\mathbf{A}-\mathbf{BK}$ and those of $\mathbf{A}-\mathbf{LC}$.

$$\det\left(\begin{bmatrix} \mathbf{M_{11}} & \mathbf{M_{12}} \\ \mathbf{0} & \mathbf{M_{22}} \end{bmatrix}\right) = \det(\mathbf{M_{11}}) \times \det(\mathbf{M_{22}})$$

The independence of these eigenvalues allows us to independently choose the poles of $\mathbf{A}-\mathbf{BK}$ and $\mathbf{A}-\mathbf{LC}$.

This allows us to pick an $\mathbf{L}$ to give fast decay of observer state errors (going from $\mathbf{x}(t)$ to $\widehat{\mathbf{x}}(t)$) relative to tracking errors (going from $y_d(t)$ to $y(t)$).

Observers

Since $\mathbf{e}(t) = \mathbf{x}(t) - \widehat{\mathbf{x}}(t)$, we can formulate the combined dynamics of the plant and observer in terms of a state vector $\begin{bmatrix} \mathbf{x}(t) \\ \widehat{\mathbf{x}}(t) \end{bmatrix}$ instead of $\begin{bmatrix} \mathbf{x}(t) \\ \mathbf{e}(t) \end{bmatrix}$:

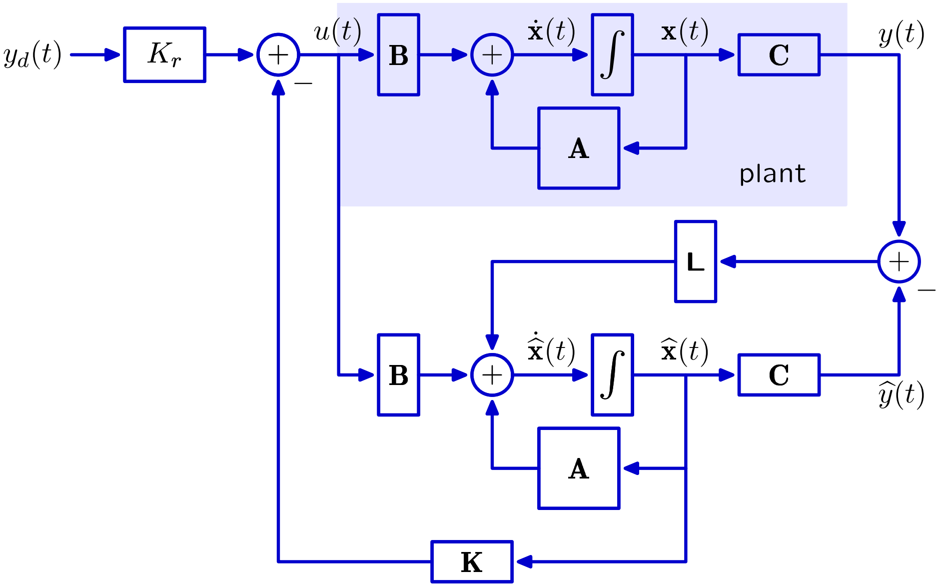
These new equations are then as follows:

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

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

Choosing L

How can we choose **L** to make the simulated states $\widehat{\mathbf{x}}(t)$ converge to $\mathbf{x}(t)$?



Choosing L

How can we choose **L** to make the simulated states $\widehat{\mathbf{x}}(t)$ converge to $\mathbf{x}(t)$?

In the normal state-space model, we choose the control vector **K** based on the eigenvalues of plant dynamics:

$$s\mathbf{X}(s) = \mathbf{AX}(s) - \mathbf{BKX}(s) + \mathbf{B}KrY_d$$

Choose **K** to optimize properties of the eigenvalues of **A**-**BK**.

For the observer, we similarly choose the feedback vector **L** based on the eigenvalues of the error dynamics:

$$s\mathbf{E}(s) = \mathbf{AE}(s) - \mathbf{LCE}(s)$$

Choose **L** to optimize properties of the eigenvalues of **A**-**LC**.

The **K** and **L** problems have a similar form – but they are not identical. The form can be made identical by transposition, i.e., optimize the eigenvalues of the transpose $\mathbf{A}^T - \mathbf{C}^T\mathbf{L}^T$ (which are identical to those of **A**-**LC**).

Choosing L

Since optimizing **K** and **L** can be cast into problems with the same form, the optimizations can be solved using the same methods.

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

or

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

Summary

Today we formulated a new approach to control based on **observers**.

- An observer is a simulation of the plant that is part of the controller.
- The biggest challenge in designing an observer is keeping its state up-to-date with that of the plant.
- We can feedback the difference between the measured and simulated outputs to correct the simulated states.