6.3100: Dynamic System Modeling and Control Design

State-Space Control

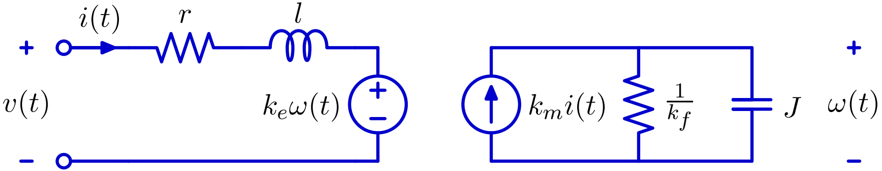
May 3, 2023

Motor Speed Control

The last few lectures have focused on state-space controllers.

Today: Apply those ideas to control the speed of a motor.

Model of the plant:



The voltage $v(t)$ represents the electrical input to the motor.

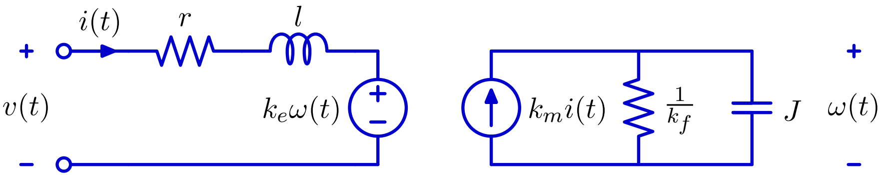
It excites a current $i(t)$, which generates a torque $k_m i(t)$ that tends to rotate the motor shaft.

The torque is resisted by the moment of inertia J and by friction (k_f).

As the motor spins, it generates a back emf ($k_e\omega(t)$) that tends to reduce the electrical current $i(t)$ drawn by the motor.

Motor Speed Control: State-Space Model

Mathematical description of the model:



Electrical port:

$$v(t) = ri(t) + l\frac{di(t)}{dt} + k_e\omega(t)$$

Mechanical port:

$$k_m i(t) = k_f\omega(t) + J\frac{d\omega(t)}{dt}$$

Determine a state-space description of this system.

Motor Speed Control: State-Space Model

Determine a state-space description of the motor.

Electrical port:

$$v(t) = ri(t) + l\frac{di(t)}{dt} + k_e\omega(t)$$

Mechanical port:

$$k_m i(t) = k_f\omega(t) + J\frac{d\omega(t)}{dt}$$

State variables: $i(t)$, $\omega(t)$.

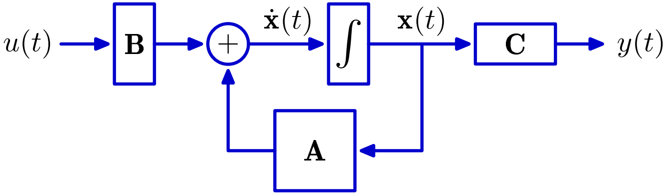
$$\frac{d}{dt}\begin{bmatrix} i(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} -\frac{r}{l} & -\frac{k_e}{l} \\ \frac{k_m}{J} & -\frac{k_f}{J} \end{bmatrix}\begin{bmatrix} i(t) \\ \omega(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{l} \\ 0 \end{bmatrix} v(t)$$

$$y(t) = \omega(t) = [0 \quad 1]\begin{bmatrix} i(t) \\ \omega(t) \end{bmatrix}$$

$$\mathbf{x}(t) = \begin{bmatrix} i(t) \\ \omega(t) \end{bmatrix}; \quad \mathbf{A} = \begin{bmatrix} -\frac{r}{l} & -\frac{k_e}{l} \\ \frac{k_m}{J} & -\frac{k_f}{J} \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} \frac{1}{l} \\ 0 \end{bmatrix}; \quad \mathbf{C} = [0 \quad 1]$$

Motor Speed Control: State-Space Model

State-space model of the plant.



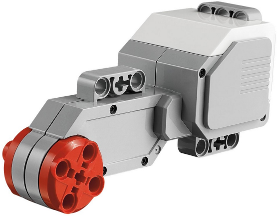
$$\frac{d}{dt}\begin{bmatrix} i(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} -\frac{r}{l} & -\frac{k_e}{l} \\ \frac{k_m}{J} & -\frac{k_f}{J} \end{bmatrix} \begin{bmatrix} i(t) \\ \omega(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{l} \\ 0 \end{bmatrix} v(t)$$

$$y(t) = \omega(t) = [0 \quad 1] \begin{bmatrix} i(t) \\ \omega(t) \end{bmatrix}$$

$$\mathbf{x}(t) = \begin{bmatrix} i(t) \\ \omega(t) \end{bmatrix}; \quad \mathbf{A} = \begin{bmatrix} -\frac{r}{l} & -\frac{k_e}{l} \\ \frac{k_m}{J} & -\frac{k_f}{J} \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} \frac{1}{l} \\ 0 \end{bmatrix}; \quad \mathbf{C} = [0 \quad 1]$$

Parameters of the Model

Lego EV3 motor parameters.

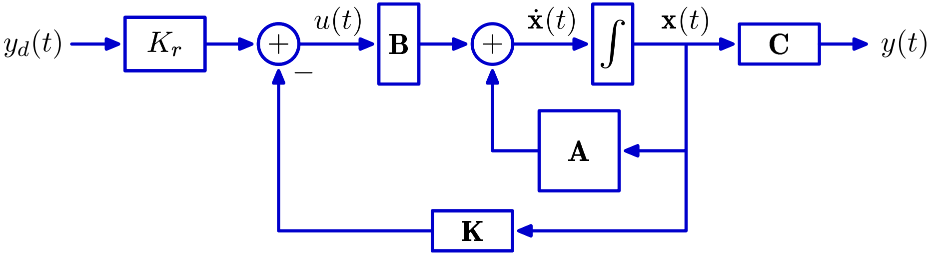


$$
\begin{aligned}
r &= 7\,\Omega \\
l &= 0.005\,\text{H} \\
k_e &= 0.46\,\text{volts/(radian/sec)} \\
k_m &= 0.3\,\text{Nm/(radian/sec)} \\
k_f &= 0.00073\,\text{Nm/(radian/sec)} \\
J &= 0.0015\,\text{Nm/(radian/sec}^2)
\end{aligned}
$$

$$
A = \begin{bmatrix} -1400 & -92 \\ 200 & -0.5 \end{bmatrix} \quad B = \begin{bmatrix} 200 \\ 0 \end{bmatrix} \quad C = \begin{bmatrix} 0 & 1 \end{bmatrix}
$$

State-Space Controller

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



We can find $\mathbf{K}$ using pole placement:

```
K = place(A,B,[poles])
```

or LQR:

```
Q = diag([1,1,1,1])
R = 1
K = lqr(A,B,Q,R)
```

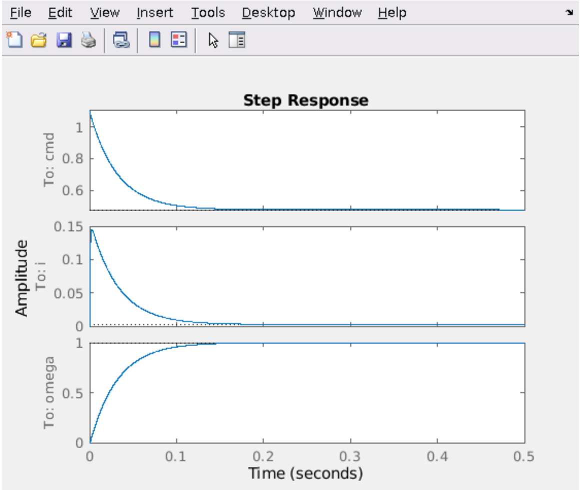
and

```
Kr = -1/(C*((A-BK)\B)
```

Choosing the Feedback Matrix K

Try LQR with `Q = diag([1,1])` and `R = 1`.

Result: `K = [0.1597 0.6305]`

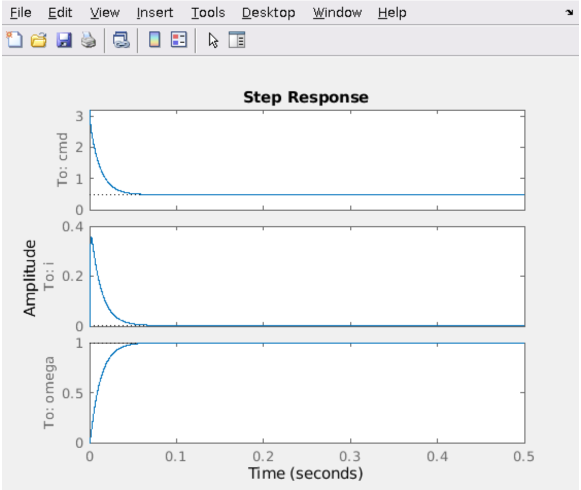


Is this good? Can we do better?

Choosing the Feedback Matrix K

Try reducing the weight on $u(t)$: `Q = diag([1,1])` and `R = 0.1`.

Result: `K = [1.0273 2.7185]`

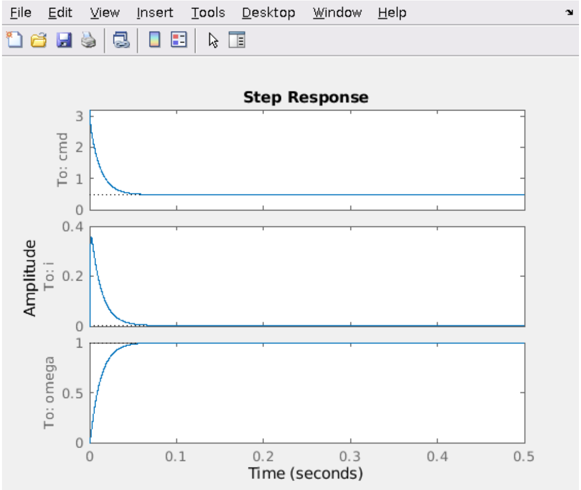


Definitely faster! Any drawbacks?

Choosing the Feedback Matrix K

Try reducing the weight on $u(t)$: `Q = diag([1,1])` and `R = 0.1`.

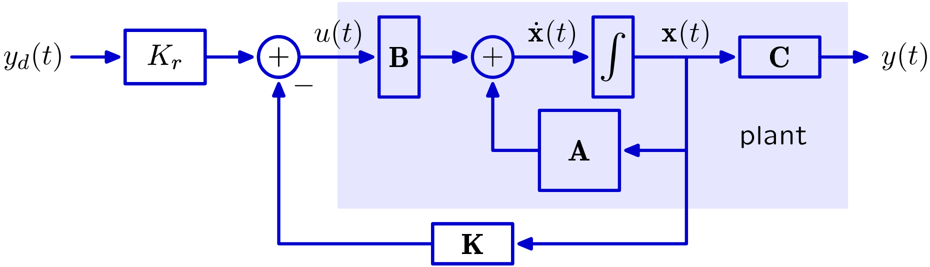
Result: `K = [1.0273 2.7185]`



Definitely faster! Any drawbacks? **Larger gains can increase noise.**

Effects of Sensor Noise

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



The feedback matrix $\mathbf{K}$ is larger for $R = 0.1$ than it is for $R = 1$.

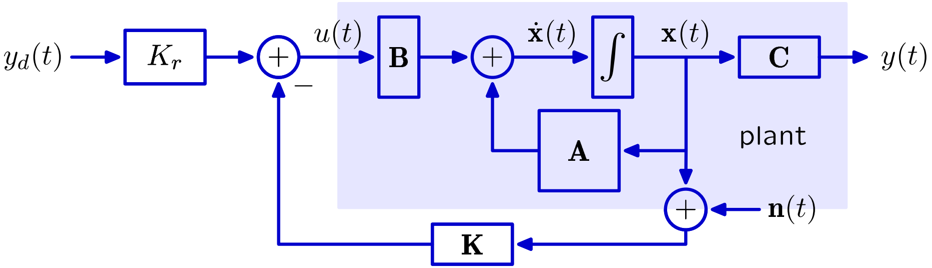
$\mathbf{K} = [0.1597\ 0.6305]$ when $R = 1.0$.
$\mathbf{K} = [1.0273\ 2.7185]$ when $R = 0.1$.

$u(t)$ is bigger because $\mathbf{K}$ is bigger.

How does a bigger $\mathbf{K}$ result in more noise?

Effects of Sensor Noise

In the last lecture, we looked at how measurement noise can enter a system.



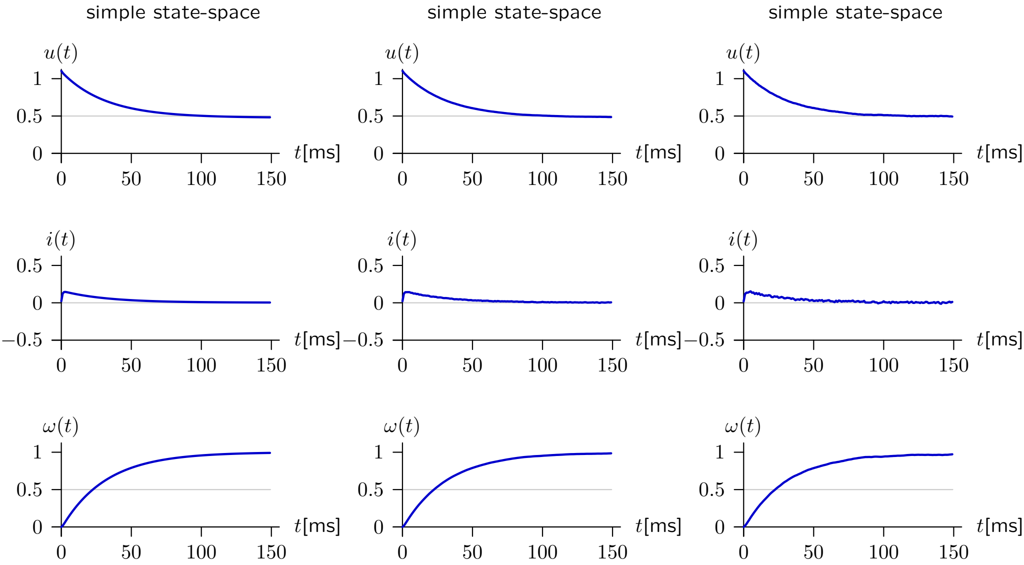
Notice that the amount of noise $\mathbf{n}(t)$ that is added to $u(t)$ is scaled by $\mathbf{K}$.

The feedback matrix $\mathbf{K}$ is larger for $R = 0.1$ than it is for $R = 1$.

$\mathbf{K} = [0.1597\ 0.6305]$ when $R = 1.0$.
$\mathbf{K} = [1.0273\ 2.7185]$ when $R = 0.1$.

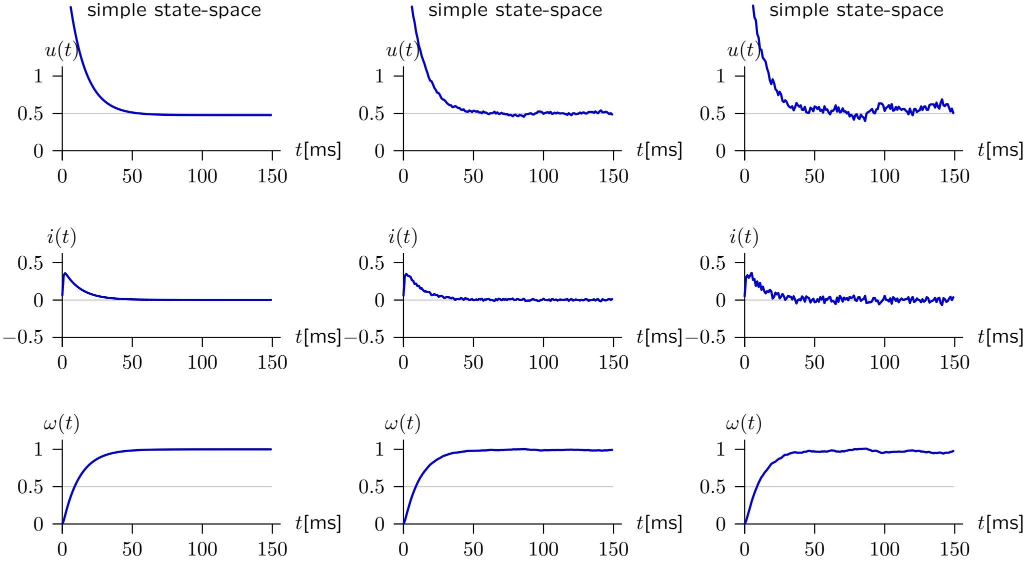
Effect of K on Noise Performance when R=1

Low (left), medium (center), and high (right) values of $\mathbf{n}(t)$.



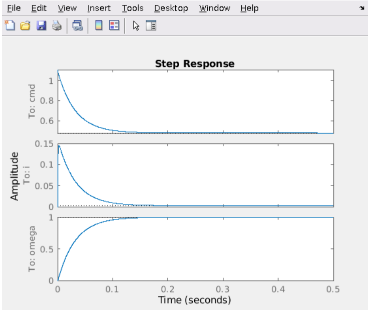
Effect of K on Noise Performance when R=0.1

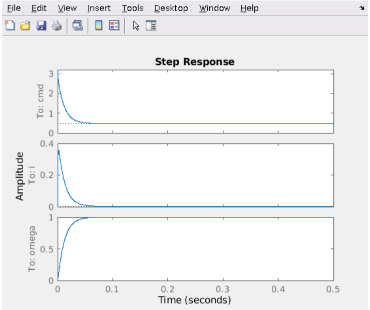
Low (left), medium (center), and high (right) values of $\mathbf{n}(t)$.

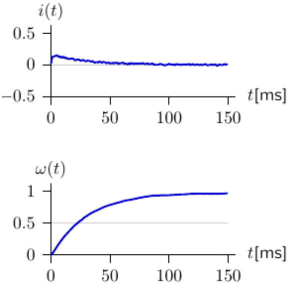


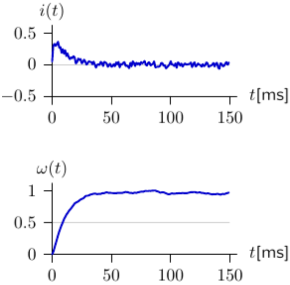
Speed/Noise Tradeoff

Higher gains can increase speed, but they also tend to increase noise.



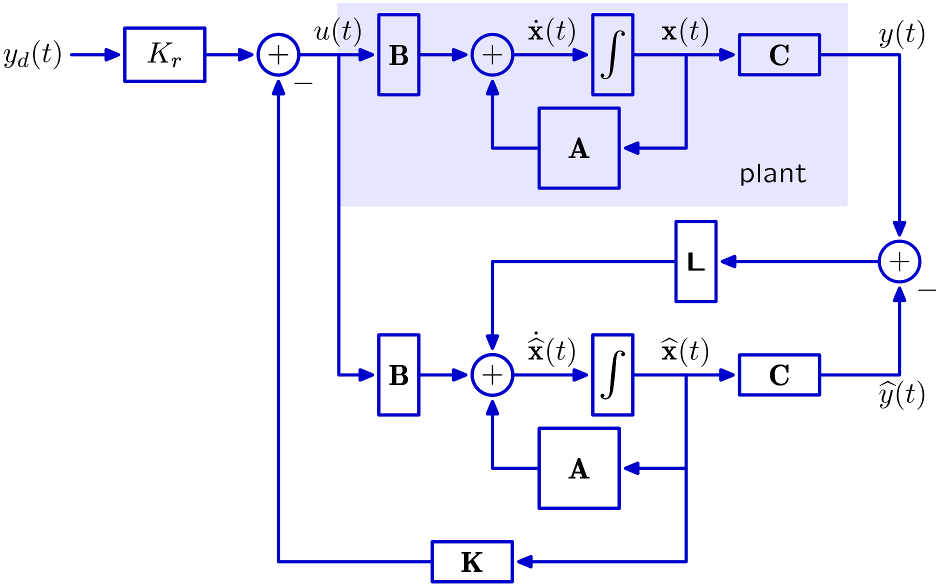






Design Tradeoffs with an Observer

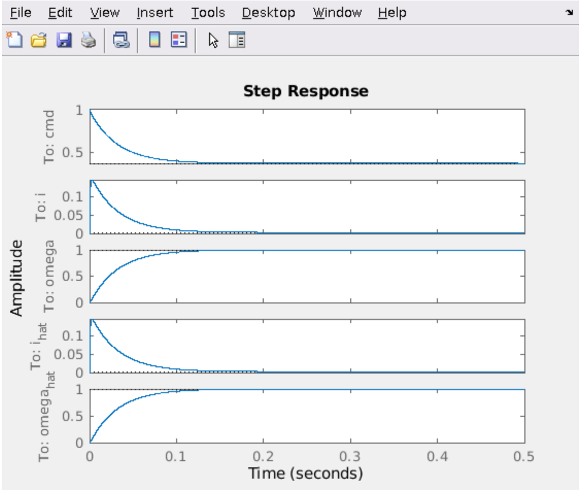
General form: Now we must choose both $\mathbf{K}$ and $\mathbf{L}$.



Choosing the Matrix L

Try LQR with `Q = diag([1,1])` and `R = 1` for both $\mathbf{K}$ and $\mathbf{L}$.

Result: `K = [0.1597 0.6305]` and `L = [-0.0024;0.0377]`

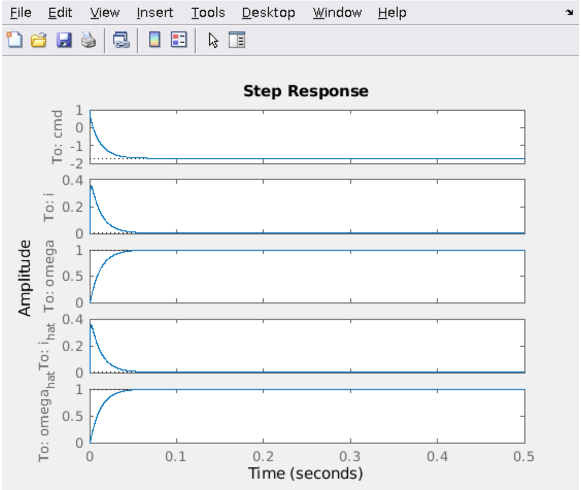


Is this good? Can we do better?

Choosing the Matrix L

Try more aggressive: `Q = diag([1,1])` and `R = 0.1` for both $\mathbf{K}$ and $\mathbf{L}$.

Result: `K = [1.0273 2.7185]` and `L = [-0.0237;0.3727]`

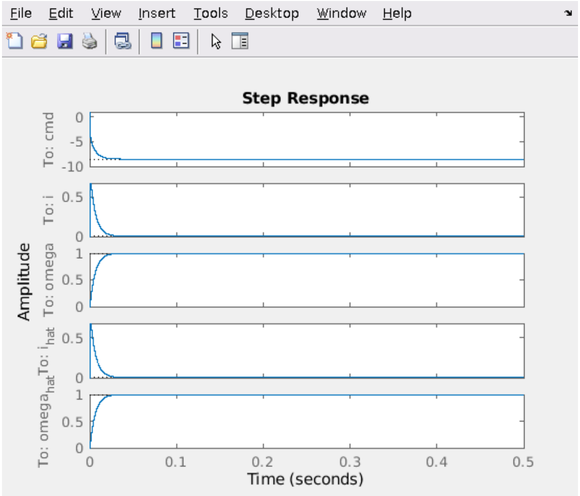


Similar to simple state-space controller: higher gain → faster response.

Choosing the Matrix L

Try more aggressive: `Q = diag([1,1])` and `R = 0.01` for both $\mathbf{K}$ and $\mathbf{L}$.

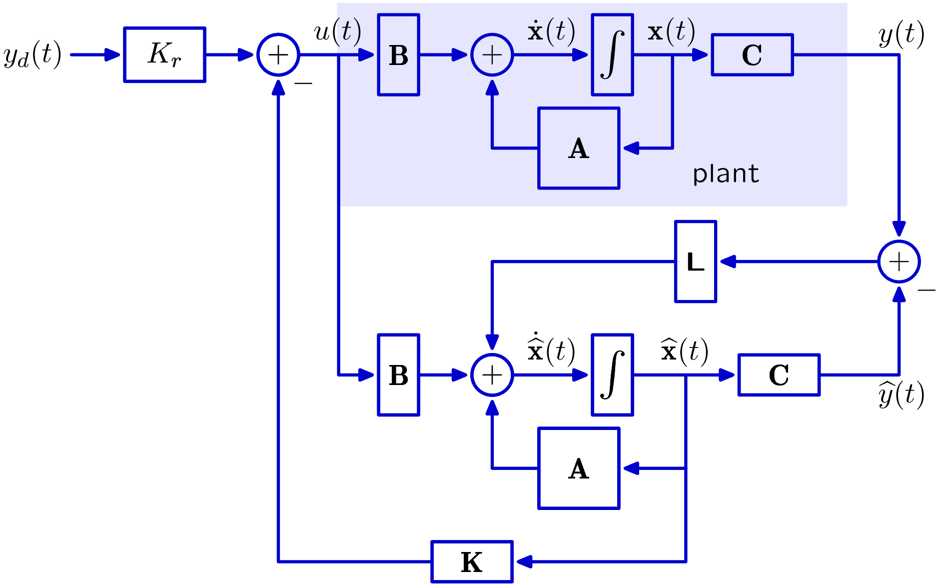
Result: `K = [5.9630 9.5199]` and `L = [-0.2135;3.3658]`



Even higher gain → even faster responses.

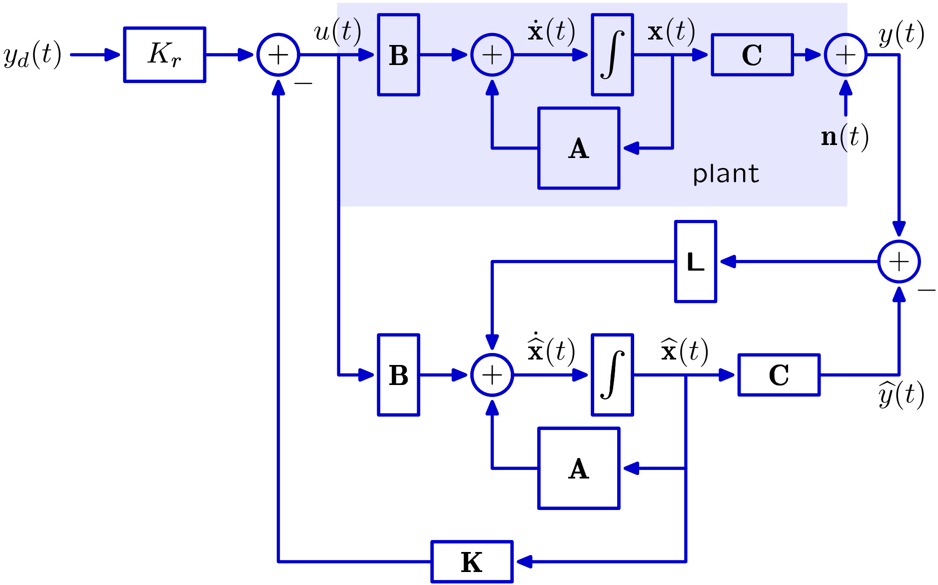
Effects of Sensor Noise

Model effects of noise on the observer system.



Effects of Sensor Noise

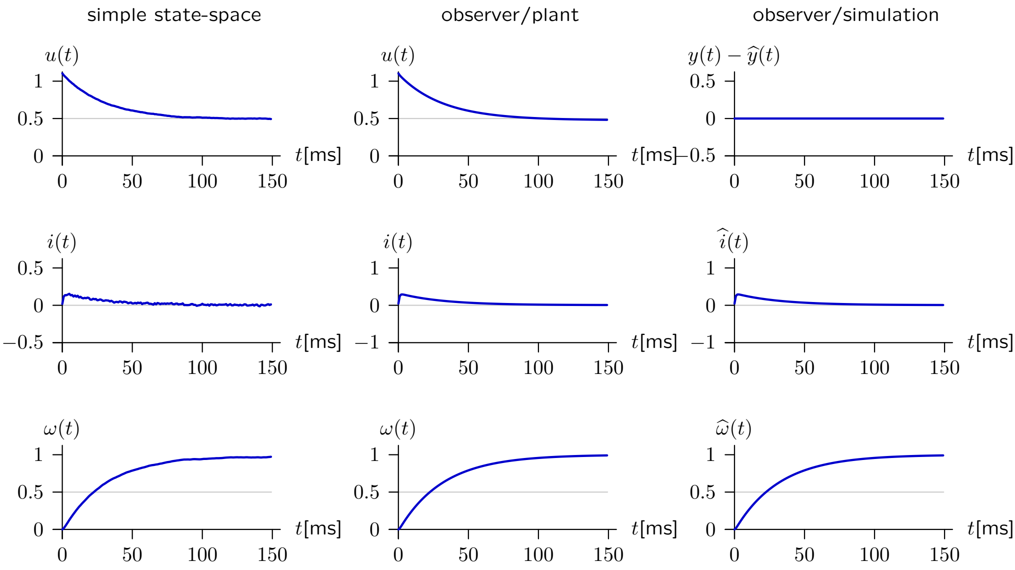
The important source of measurement noise is in the measurement of the output $y(t)$.



Effects of K and L

Choose `K` based on `Q=diag([1,1])` and `R=1`.

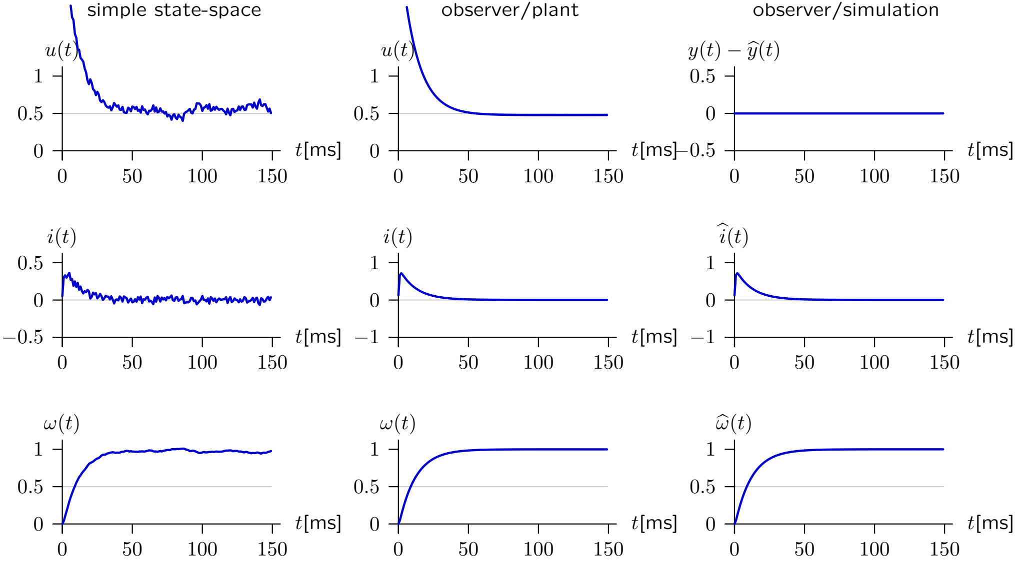
Choose `L` based on `Q=diag([1,1])` and `R=1`.



Effects of K and L

Choose `K` based on `Q=diag([1,1])` and `R=0.1`.

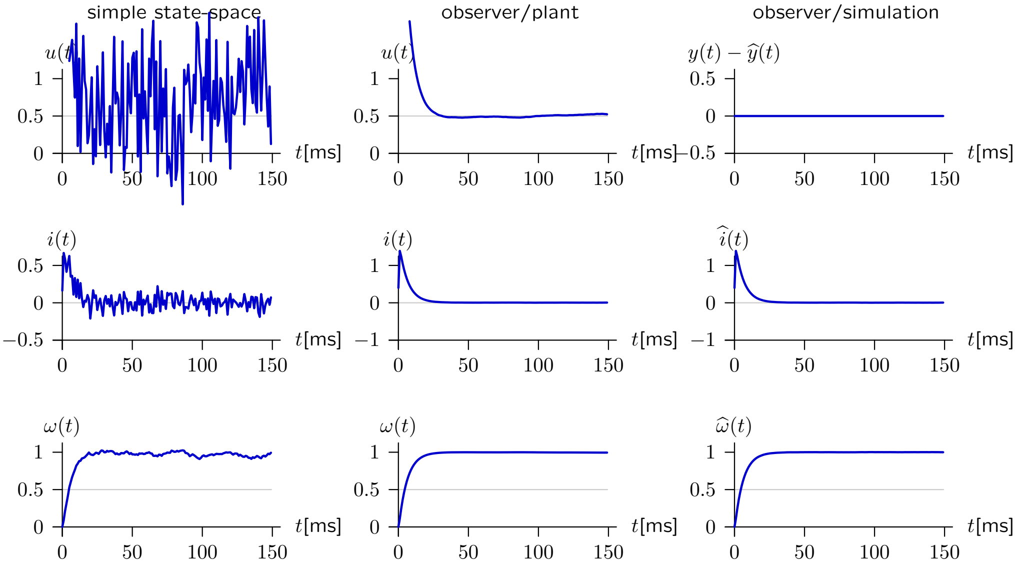
Choose `L` based on `Q=diag([1,1])` and `R=0.1`.



Effects of K and L

Choose `K` based on `Q=diag([1,1])` and `R=0.01`.

Choose `L` based on `Q=diag([1,1])` and `R=0.01`.



Effects of K and L

Parameters.

$\mathbf{Q}$	$\mathbf{R}$	$\mathbf{K}$	eig$(\mathbf{A}-\mathbf{BK})$	$\mathbf{L}$	eig$(\mathbf{A^T}-\mathbf{C^T}\mathbf{L^T})$
[1,1]	1	[0.1597,0.6305]	-1401,-31.6	[-0.0024;0.0377]	-1387,-13.8
[1,1]	0.1	[1.0273,2.7185]	-1522 -84.0	[-0.0237;0.3727]	-1387 -14.1
[1,1]	0.01	[5.9630,9.5199]	-2428,-164.9	[-0.2135;03.3658]	-1387 -17.1

Effects of K and L

Parameters.

$\mathbf{Q}$	$\mathbf{R}$	$\mathbf{K}$	eig($\mathbf{A}-\mathbf{BK}$)	$\mathbf{L}$	eig($\mathbf{A^T}-\mathbf{C^T L^T}$)
[1,1]	1	[0.1597,0.6305]	-1401,-31.6	[-0.0024;0.0377]	-1387,-13.8
[1,1]	0.1	[1.0273,2.7185]	-1522 -84.0	[-0.0237;0.3727]	-1387 -14.1
[1,1]	0.01	[5.9630,9.5199]	-2428,-164.9	[-0.2135;03.3658]	-1387 -17.1

For each entry, the dominant eigenvalue of $\mathbf{A^T}-\mathbf{C^T L^T}$ is greater than the dominant eigenvalue of $\mathbf{A}-\mathbf{BK}$.

The feedback from the observer to the plant happens **before** the observer has had a chance to synchronize with the plant!

We need faster eigenvalues of $\mathbf{A^T}-\mathbf{C^T L^T}$.

Effects of K and L

Parameters.

$\mathbf{Q}$	$\mathbf{R}$	$\mathbf{K}$	eig($\mathbf{A}-\mathbf{BK}$)	$\mathbf{L}$	eig($\mathbf{A^T}-\mathbf{C^T}\mathbf{L^T}$)
[1,1]	1	[0.1597,0.6305]	-1401,-31.6	[-0.0024;0.0377]	-1387,-13.8
[1,1]	0.1	[1.0273,2.7185]	-1522 -84.0	[-0.0237;0.3727]	-1387 -14.1
[1,1]	0.01	[5.9630,9.5199]	-2428,-164.9	[-0.2135;03.3658]	-1387 -17.1
[1,100]	0.1	[4.0127,31.1396]	-1102 -1102	[-1.3463;21.0007]	-1387 -35
[1,100]	0.01	[[11.6545,99.4957]	-1865 -1865	[-5.3925;88.0755]	-1387 -102

Effects of K and L

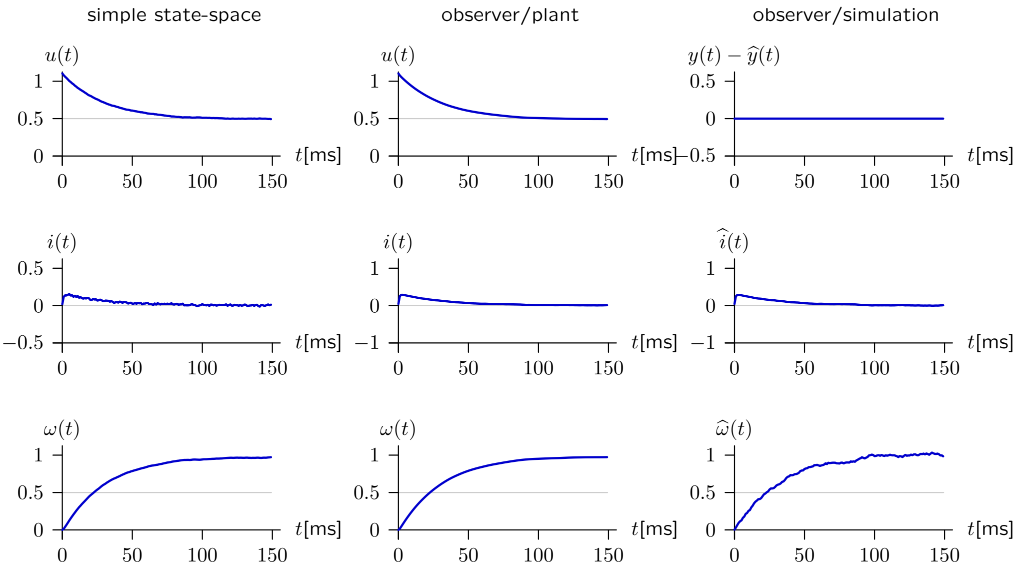
Use the red parameters.

$\mathbf{Q}$	$\mathbf{R}$	$\mathbf{K}$	eig$(\mathbf{A}-\mathbf{BK})$	$\mathbf{L}$	eig$(\mathbf{A^T}-\mathbf{C^T}\mathbf{L^T})$
[1,1]	1	[0.1597,0.6305]	-1401,-31.6	[-0.0024;0.0377]	-1387,-13.8
[1,1]	0.1	[1.0273,2.7185]	-1522 -84.0	[-0.0237;0.3727]	-1387 -14.1
[1,1]	0.01	[5.9630,9.5199]	-2428,-164.9	[-0.2135;03.3658]	-1387 -17.1
[1,100]	0.1	[4.0127,31.1396]	-1102 -1102	[-1.3463;21.0007]	-1387 -35
[1,100]	0.01	[[11.6545,99.4957]	-1865 -1865	[-5.3925;88.0755]	-1387 -102

Effects of K and L

Choose `K` based on `Q=diag([1,1])` and `R=1`.

Choose `L` based on `Q=diag([1,100])` and `R=0.01`.



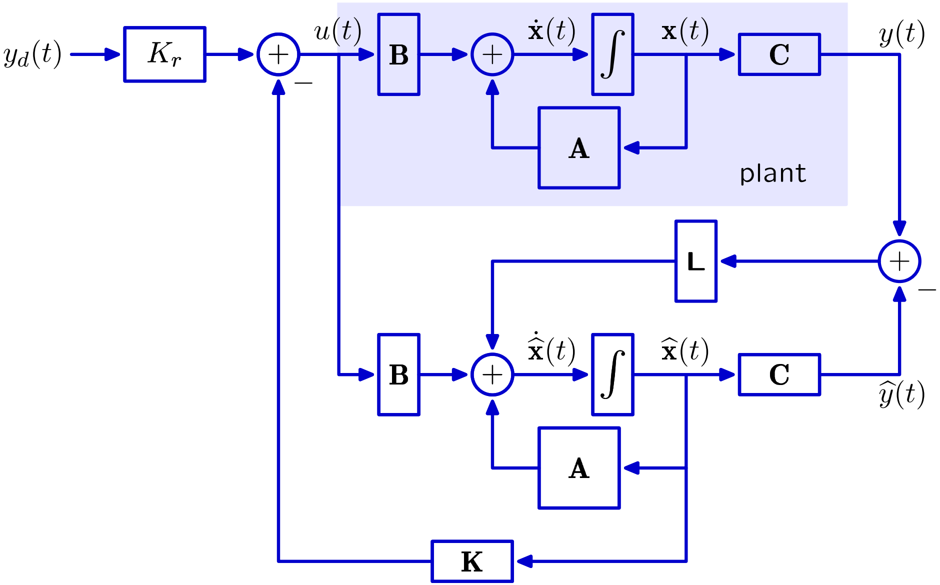
Effects of K and L

As with the simple state-space controller, higher gains can increase speed.

Extra care is needed when designing an observer. Make sure that the observer stabilizes to the plant **before** the observer is used to provide feedback to the plant.

State-Space Controller

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



Express the controller algorithm in pseudo-code.

State-Space Controller

Express the controller algorithm in pseudo-code.

Assume the `step` function (below) is executed once every `DeltaT` seconds.

```
global xhat
void step(){
  y = get_output_y()
  xhat = xhat + DeltaT * ((A-B*K)*xhat + B*Kr*yd + L*(y-C*xhat))
  put_command_u(Kr*yd-K*xhat)
}
```