6.3100: Dynamic System Modeling and Control Design

Discrete-Time Observer

May 8, 2023

Discrete-Time Observer

For the past few lectures, we have analyzed behaviors of **continuous-time observers**. We analyzed **convergence** of the observer and plant states:

$$\hat{\mathbf{x}}(\mathbf{t}) \rightarrow \mathbf{x}(t)$$

as well as convergence of the plant output to the desired value:

$$y(t) \rightarrow y_d(t)$$

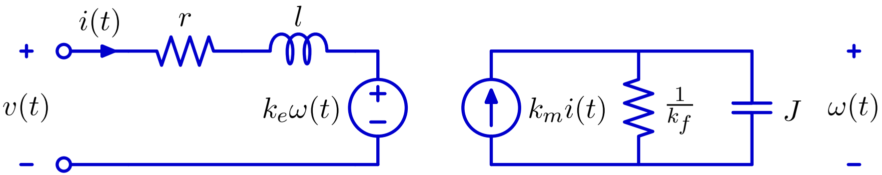
We also looked at the sensitivity of the controllers to **noise**.

But microcontrollers (such as the Teensy) are increasingly used because of their low cost and high performance. And modern microcontrollers operate in **discrete-time**.

Today: analyze systems that **combine continuous time** representations of the plant **with discrete time** implementations of control.

Motor Speed Control

We will use the motor speed control system as an example.



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

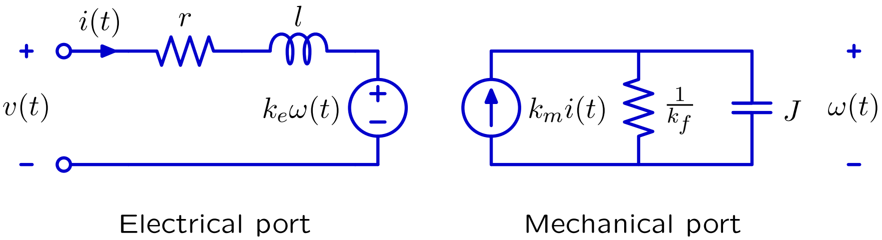
It excites a current $i(t)$, which generates a torque $k_m i(t)$ that tends to rotate the motor shaft.

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.

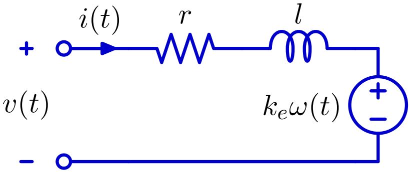
Motor Speed Control: Two-Port Model

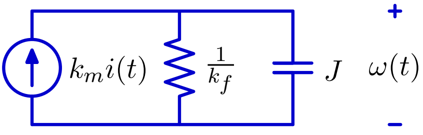
Motors have two ports: one is electrical and one is mechanical.



Motor Speed Control: Mathematical Representation

Simple circuit analysis provides a mathematical representation.



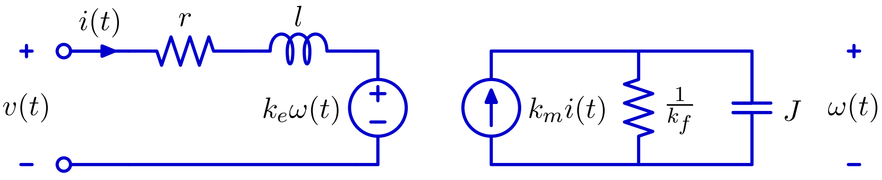


$$v(t) = ri(t) + l\frac{di(t)}{dt} + k_e\omega(t)$$

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

Motor Speed Control: Matrix Representation

The equations are conveniently represented by a pair of matrix equations.



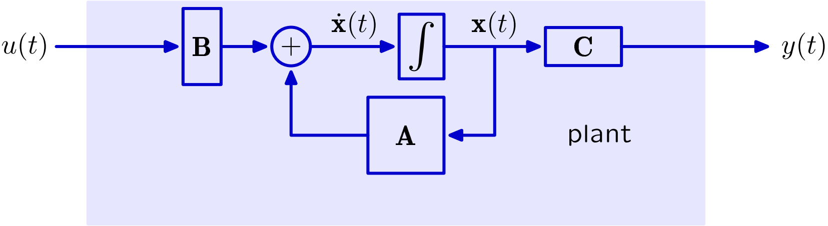
$$v(t) = ri(t) + l\frac{di(t)}{dt} + k_e\omega(t) \qquad k_m i(t) = k_f\omega(t) + J\frac{d\omega(t)}{dt}$$

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

$$\underbrace{\omega(t)}_{y(t)} = \underbrace{[0 \quad 1]}_{\mathbf{C}}\underbrace{\begin{bmatrix} i(t) \\ \omega(t) \end{bmatrix}}_{\mathbf{x}(t)}$$

State-Space Model

The matrix equations provide a complete representation of the **plant**.

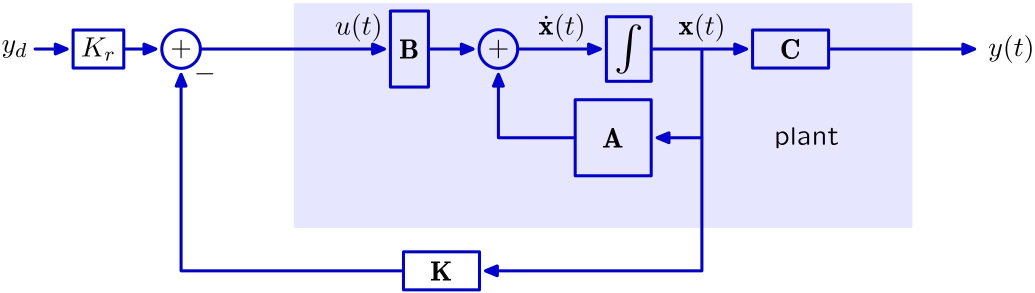


$$\frac{d}{dt}\underbrace{\begin{bmatrix} i(t) \\ \omega(t) \end{bmatrix}}_{\mathbf{x}(t)} = \underbrace{\begin{bmatrix} -\frac{r}{l} & -\frac{k_e}{l} \\ \frac{k_m}{J} & -\frac{k_f}{J} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} i(t) \\ \omega(t) \end{bmatrix}}_{\mathbf{x}(t)} + \underbrace{\begin{bmatrix}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

$$\underbrace{\omega(t)}_{y(t)} = \underbrace{\begin{bmatrix} 0 & 1 \end{bmatrix}}_{\mathbf{C}} \underbrace{\begin{bmatrix} i(t) \\ \omega(t) \end{bmatrix}}_{\mathbf{x}(t)}$$

State-Space Model + State-Space Controller

This motor model was then put into a feedback loop that was designed to make the output speed $y(t) = \omega(t)$ track the desired speed $y_d(t)$.



where $\mathbf{K}$ is found using pole placement:

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

or LQR:

`Q = diag([1,1,1,1])` and `R = 1`

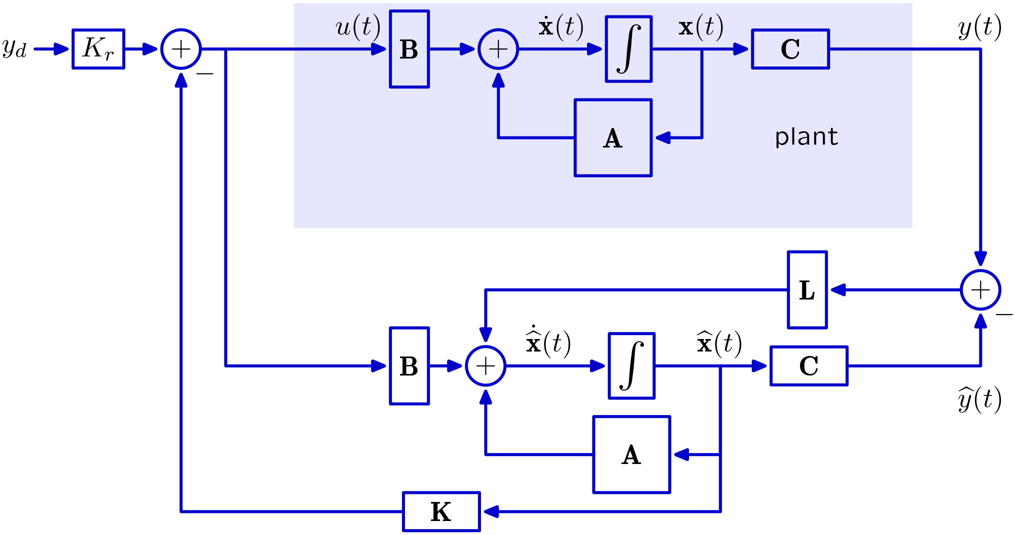
```
K = lqr(A,B,Q,R)
```

and

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

State-Space Model + Observer

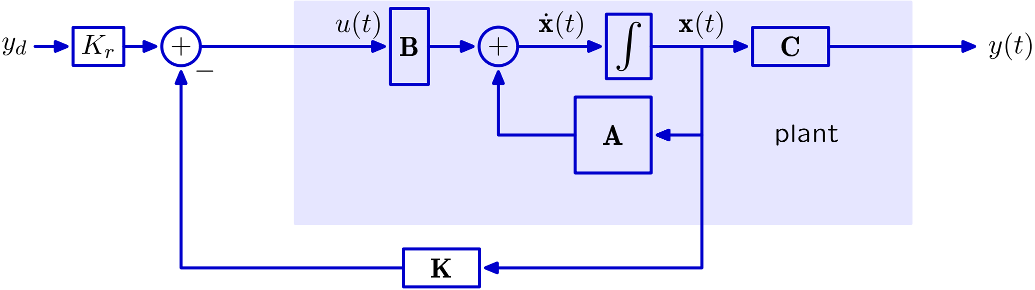
We also analyzed the performance of an observer-based controller.



Must specify both $\mathbf{K}$ and $\mathbf{L}$.

Effects of Sensor Noise

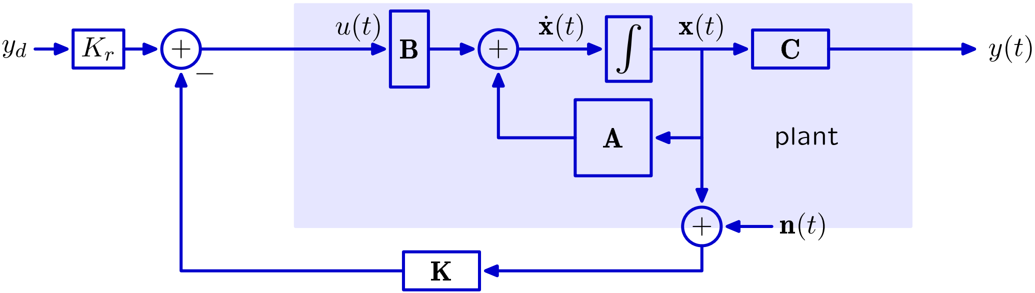
We looked at noise performance for both simple state-space controller ...



We focused on sensing (measurement) noise at the interface between the plant and the controller.

Effects of Sensor Noise

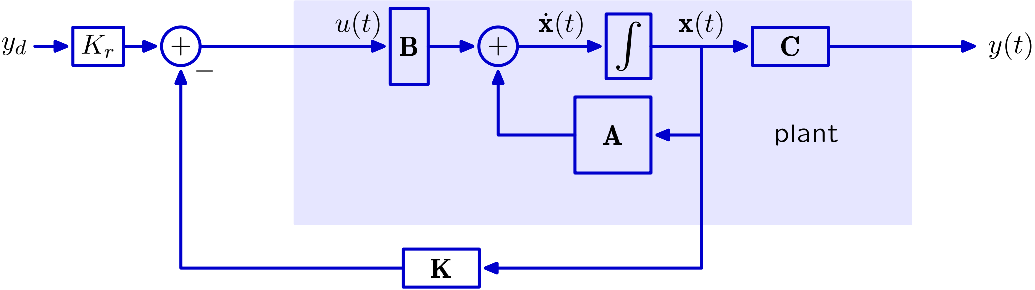
We looked at noise performance for both simple state-space controller ...



We focused on sensing (measurement) noise at the interface between the plant and the controller.

Discrete-Time Control

Today we will look at a different issue that affects performance.



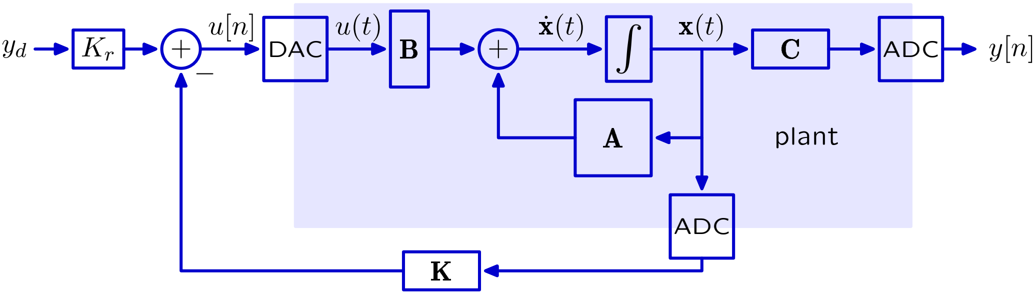
Hybrid representation: continuous-time plant with discrete-time control.

The state $\mathbf{x}(t)$ of the plant must be converted to discrete time to process in a digital controller (such as the Teensy).

The resulting discrete-time command $u[n]$ must be converted to continuous time for the plant.

Discrete-Time Control

Today we will look at a different issue that affects performance.



Hybrid representation: continuous-time plant with discrete-time control.

The state $\mathbf{x}(t)$ of the plant must be converted to discrete time to process in a digital controller (such as the Teensy).

The resulting discrete-time command $u[n]$ must be converted to continuous time for the plant.

Analog-To-Digital Conversion

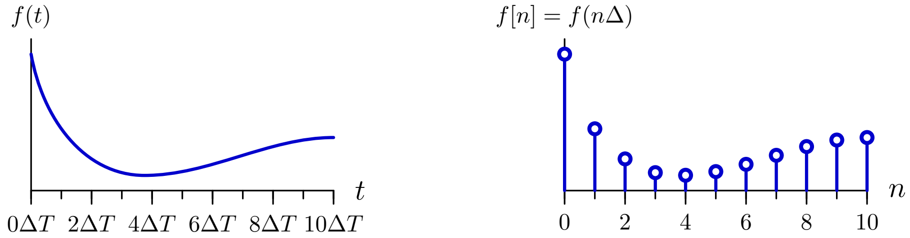
Analog-to-digital conversion entails two types of transformations.

Sampling: process by which a function of real domain is transformed into a function of integer domain.

Quantization: process by which a continuous range of amplitudes is represented by a finite range of integers.

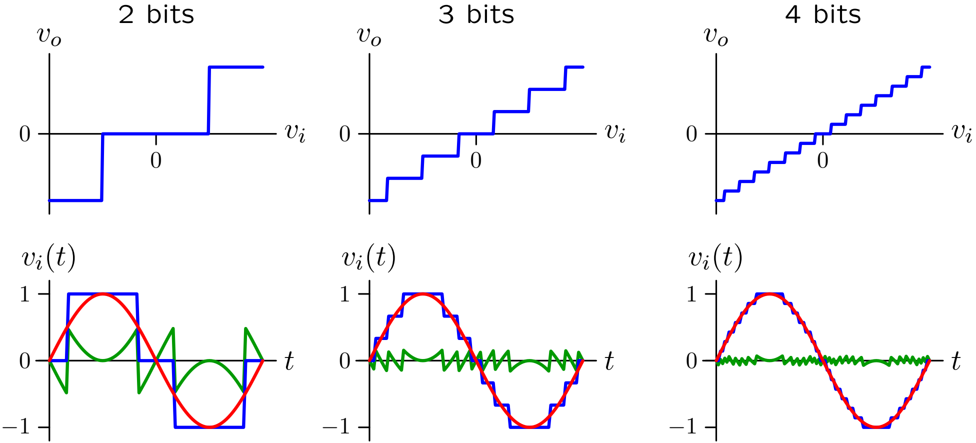
Sampling

A function of real domain is transformed into a function of integer domain.



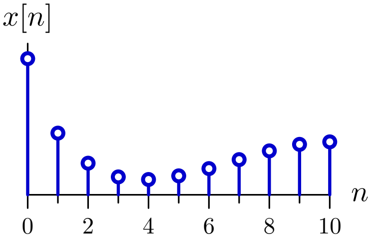
Quantization

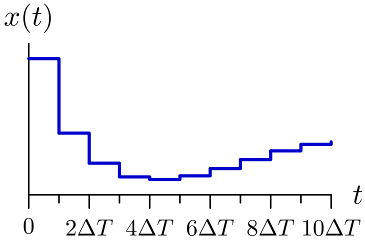
Quantization: process by which a continuous range of amplitudes is represented by a finite range of integers.



Digital-To-Analog Conversion

Digital-to-analog conversion **reconstructs** an analog signal from its digital representation. **zero-order hold**

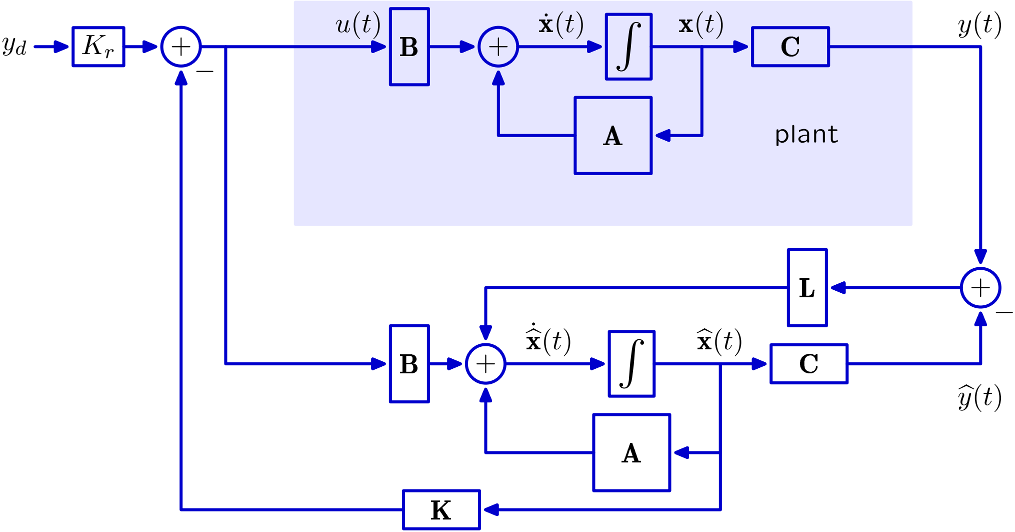




While these methods of conversion are common, there are numerous other schemes (specialized for audio, images, etc.).

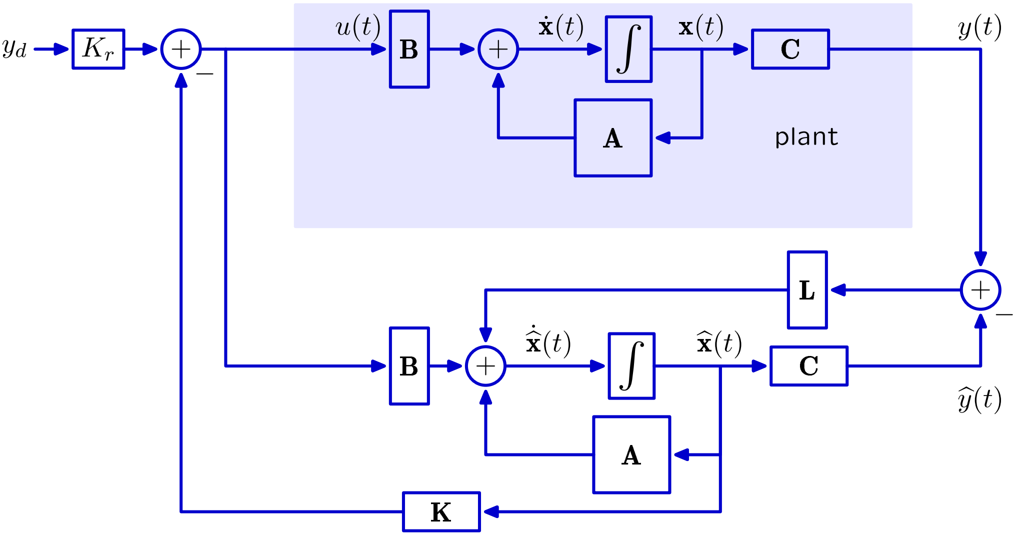
Check Yourself

List the changes that are needed to convert the controller to discrete time.



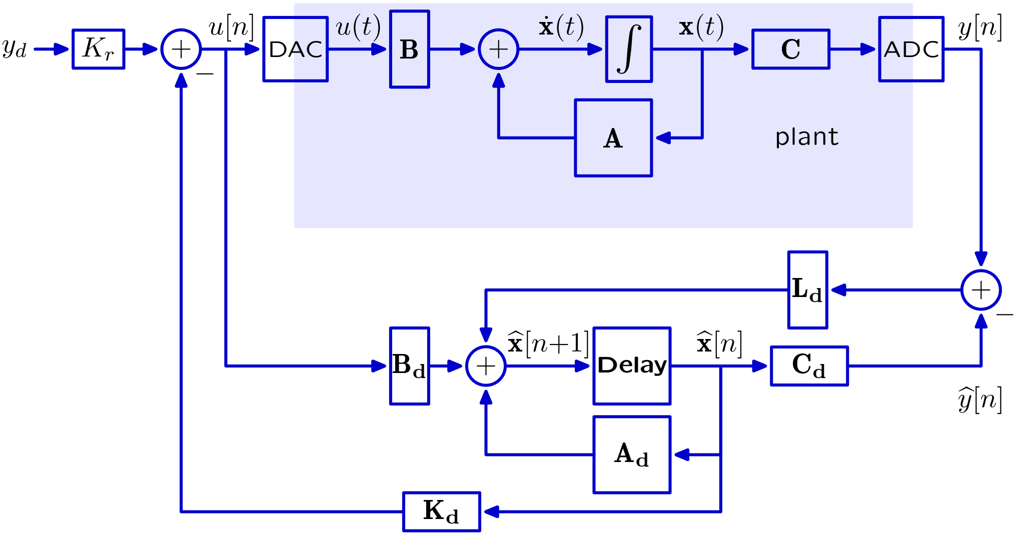
Check Yourself

List the changes that are needed to convert the controller to discrete time.



Check Yourself

List the changes that are needed to convert the controller to discrete time.



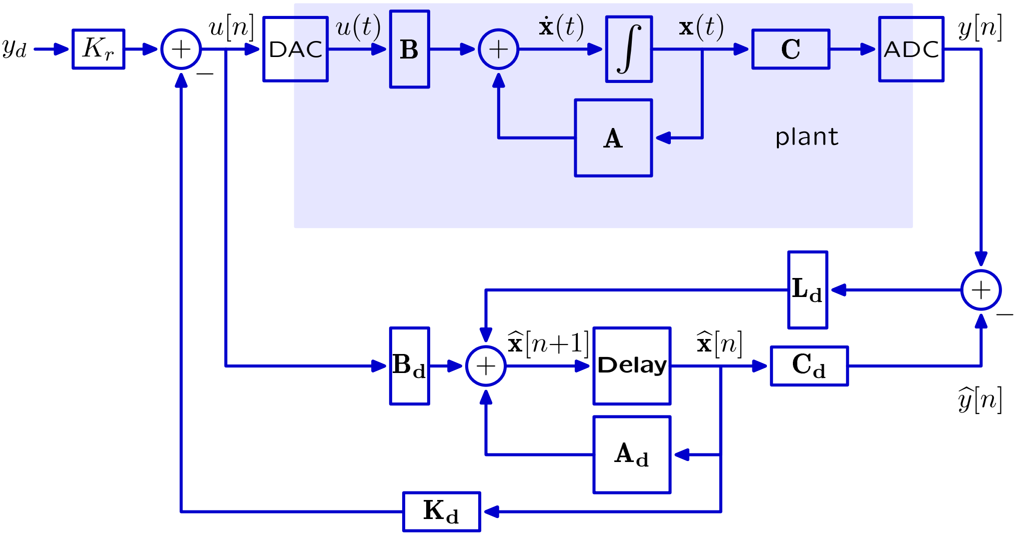
Signals outside plant must be discrete time: $y(t) \to y[n]$; $u[n] \to u(t)$.

Integrator in observer $\to$ delay: $\widehat{\mathbf{x}}(t) \to \widehat{\mathbf{x}}[n]$; $\dot{\widehat{\mathbf{x}}}(t) \to \widehat{\mathbf{x}}[n+1]$

Control matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{L}$, and $\mathbf{K}$ must be converted to discrete versions.

Check Yourself

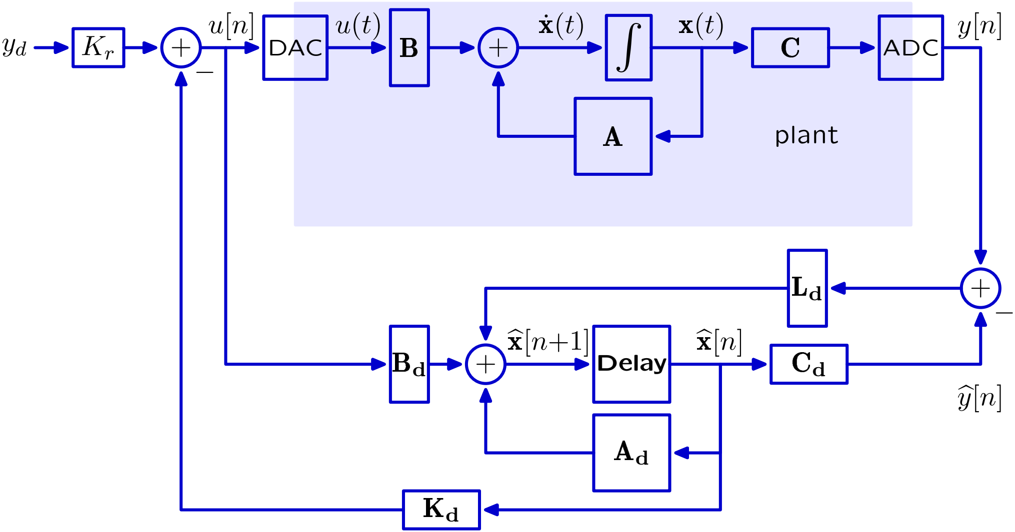
List the changes that are needed to convert the controller to discrete time.



What criteria should we use to determine $\mathbf{A_d}, \mathbf{B_d}$, and $\mathbf{C_d}$?

Check Yourself

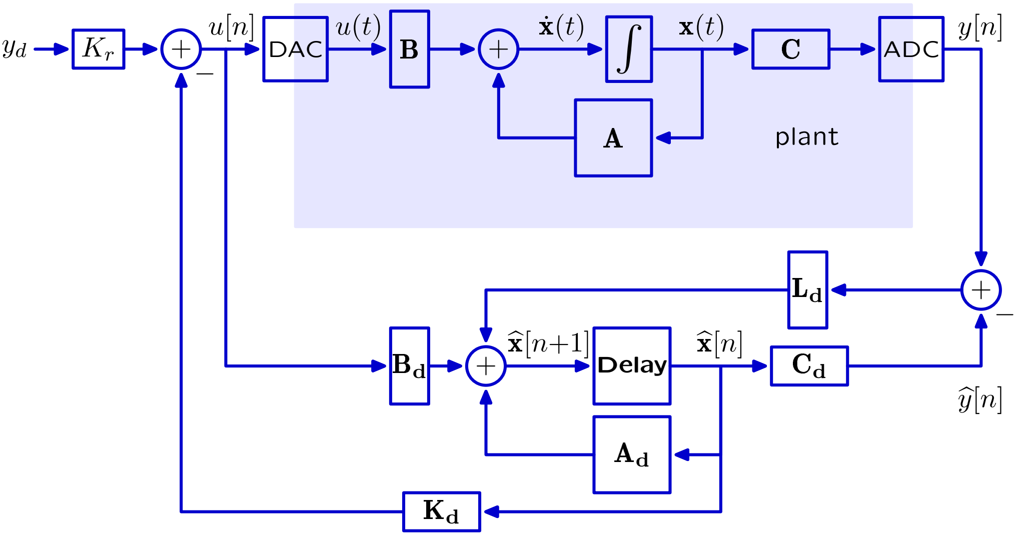
List the changes that are needed to convert the controller to discrete time.



1. $\widehat{\mathbf{x}}[n+1] = \dot{\mathbf{x}}(n\Delta T)$; for all n
2. $\widehat{\mathbf{x}}[n] = \mathbf{x}(n\Delta T)$; for all n
3. $\widehat{y}[n] = y(n\Delta T)$; for all n
4. $\mathbf{x}(t) = \widehat{\mathbf{x}}[\lfloor \frac{t}{\Delta T} \rfloor]$; for $n\Delta T < t < (n+1)\Delta T$
5. none of the above

Check Yourself

List the changes that are needed to convert the controller to discrete time.



What's the relation between $\mathbf{A}$ and $\mathbf{A_d}$? Between $\mathbf{B}$ and $\mathbf{B_d}$? $\mathbf{C}$ and $\mathbf{C_d}$?

Discrete-Time Observer Matrices

We want the state of the observer at time index n ($\widehat{\mathbf{x}}[n]$) to track the state of the plant ($\mathbf{x}(t)$) at the corresponding time $t = n\Delta T$.

Assume that the states of the plant and observer are equal at $t = n\Delta T$:

$$\widehat{\mathbf{x}}[n] = \mathbf{x}(n\Delta T)$$

Since $u(t)$ is the output of a digital-to-analog converter, its value is constant at $u[n]$ for $n\Delta T \le t \le (n{+}1)\Delta T$. With constant input over this period,

$$\mathbf{x}(t) = e^{\mathbf{A}(t-n\Delta T)}\,\mathbf{x}(n\Delta T) + \left(e^{\mathbf{A}(t-n\Delta T)} - \mathbf{I}\right)\mathbf{A}^{-\mathbf{1}}\mathbf{B}\,u[n]$$

which is easy to proved by showing that this expression is the solution to

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

Since $u(t) = u[n]$ is constant over the interval $n\Delta T < t < (n{+}1)\Delta T$:

$$\dot{\mathbf{x}}(t) = \mathbf{A}e^{\mathbf{A}(t-n\Delta T)}\mathbf{x}(n\Delta T) + \mathbf{A}e^{\mathbf{A}(t-n\Delta T)}\mathbf{A}^{-\mathbf{1}}\mathbf{B}u[n]$$

$$\mathbf{A}\mathbf{x}(t) + \mathbf{B}u[n] = \mathbf{A}e^{\mathbf{A}(t-n\Delta T)}\,\mathbf{x}(n\Delta T) + \mathbf{A}\left(e^{\mathbf{A}(t-n\Delta T)} - \mathbf{I}\right)\mathbf{A}^{-\mathbf{1}}\mathbf{B}\,u[n] + \mathbf{B}u[n]$$

Discrete-Time Observer Matrices

We want the state of the observer at time index n ($\widehat{\mathbf{x}}[n]$) to track the state of the plant ($\mathbf{x}(t)$) at the corresponding time $t = n\Delta T$.

Assume that the states of the plant and observer are equal at $t = n\Delta T$:

$$\widehat{\mathbf{x}}[n] = \mathbf{x}(n\Delta T)$$

Since $u(t)$ is the output of a digital-to-analog converter, its value is constant at $u[n]$ for $n\Delta T \leq t \leq (n+1)\Delta T$. With constant input over this period,

$$\mathbf{x}(t) = e^{\mathbf{A}(t-n\Delta T)}\,\mathbf{x}(n\Delta T) + \left(e^{\mathbf{A}(t-n\Delta T)} - \mathbf{I}\right)\mathbf{A}^{-\mathbf{1}}\mathbf{B}\,u[n]$$

Discrete-Time Observer Matrices

We want the state of the observer at time index n ($\widehat{\mathbf{x}}[n]$) to track the state of the plant ($\mathbf{x}(t)$) at the corresponding time $t = n\Delta T$.

Assume that the states of the plant and observer are equal at $t = n\Delta T$:

$$\widehat{\mathbf{x}}[n] = \mathbf{x}(n\Delta T)$$

Since $u(t)$ is the output of a digital-to-analog converter, its value is constant at $u[n]$ for $n\Delta T \le t \le (n{+}1)\Delta T$. With constant input over this period,

$$\mathbf{x}(t) = e^{\mathbf{A}(t-n\Delta T)}\,\mathbf{x}(n\Delta T) + \left(e^{\mathbf{A}(t-n\Delta T)}{-}\mathbf{I}\right)\mathbf{A}^{-\mathbf{1}}\mathbf{B}\,u[n]$$

It follows that

$$\mathbf{x}((n{+}1)\Delta T) = e^{\mathbf{A}\Delta T}\,\mathbf{x}(n\Delta T) + \left(e^{\mathbf{A}\Delta T}{-}\mathbf{I}\right)\mathbf{A}^{-\mathbf{1}}\mathbf{B}\,u[n]$$

and the state of the observer will track samples of the state of the plant if

$$\widehat{\mathbf{x}}[n{+}1] = \mathbf{A_d}\widehat{\mathbf{x}}[n] + \mathbf{B_d}u[n]$$

if

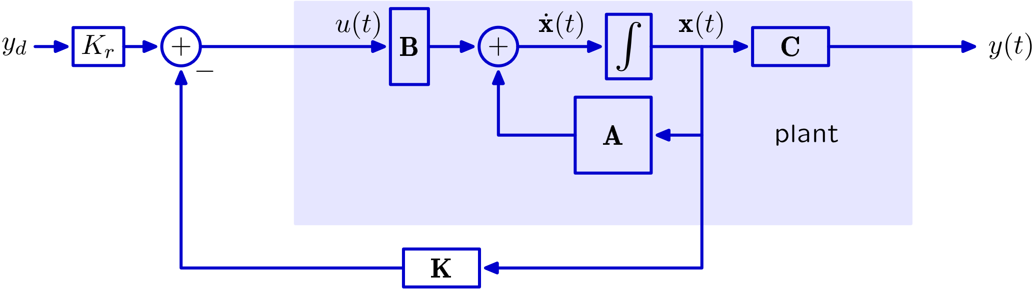
$$\mathbf{A_d} = e^{\mathbf{A}\Delta T}$$

$$\mathbf{B_d} = \left(e^{\mathbf{A}\Delta T}{-}\mathbf{I}\right)\mathbf{A}^{-\mathbf{1}}\mathbf{B}$$

$$\mathbf{C_d} = \mathbf{C}$$

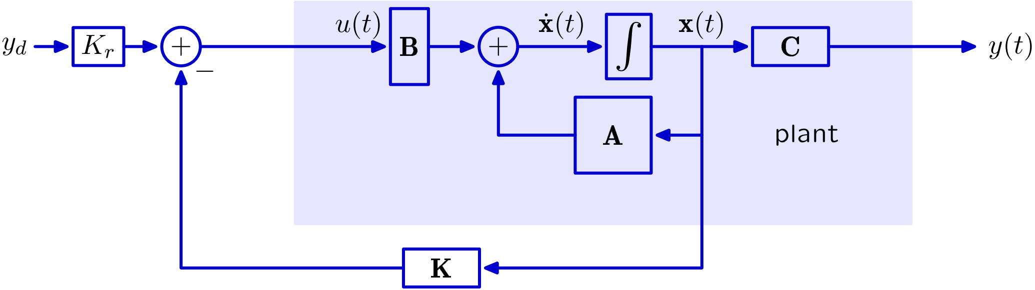
Check Yourself

List the changes that are needed to convert the controller to discrete time.



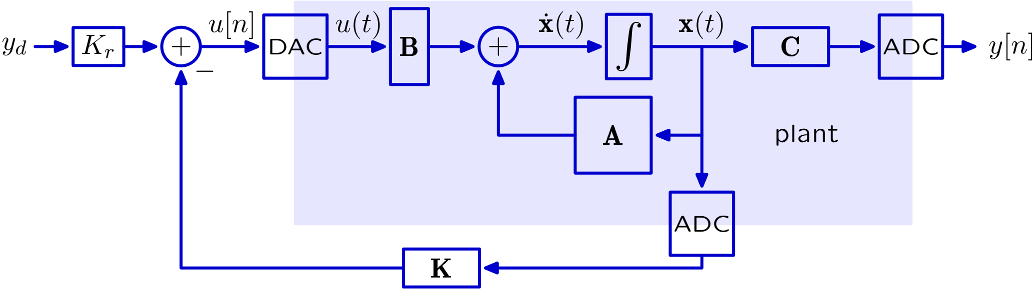
Check Yourself

List the changes that are needed to convert the controller to discrete time.



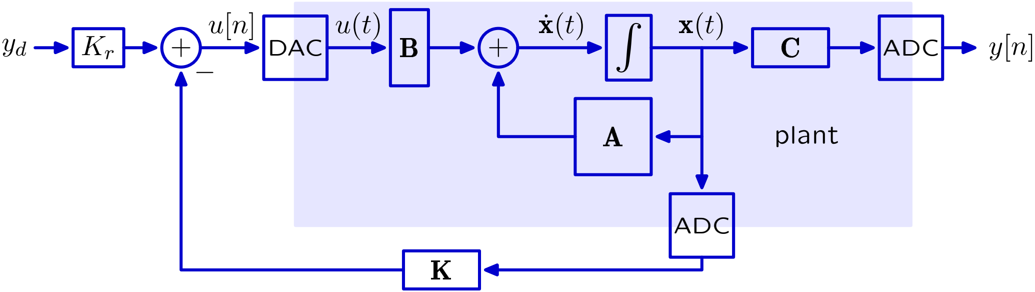
Check Yourself

List the changes that are needed to convert the controller to discrete time.



Check Yourself

List the changes that are needed to convert the controller to discrete time.



Add analog-to-digital converters to outputs of plant and digital-to-analog converters to input to plant.

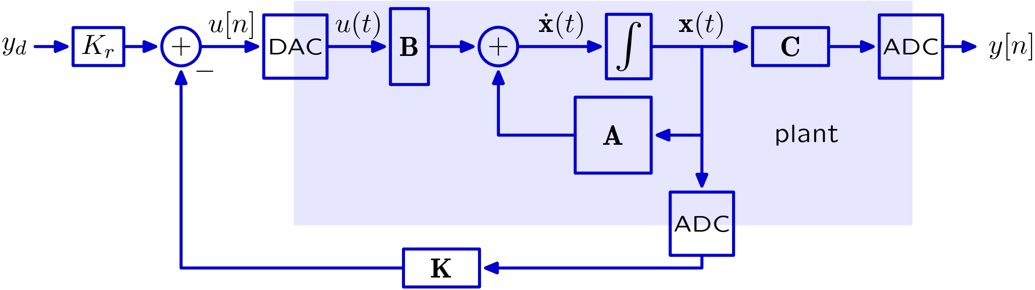
How about $\mathbf{K}$?

Is this a $\mathbf{K}$ or $\mathbf{K_d}$?

How should it be computed?

Check Yourself

List the changes that are needed to convert the controller to discrete time.

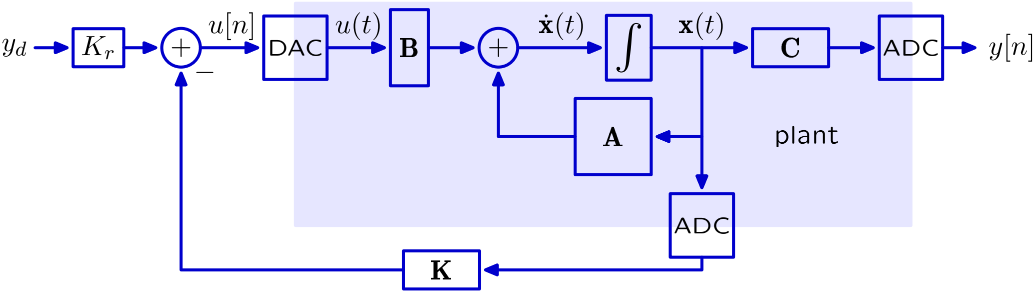


Consider the motor model with $\Delta T = 0.0001$ s. Which of these is better?

```
K1 =  lqr(A,B,Q,R)
K2 = dlqr(A,B,Q,R)
K3 =  lqr(I+A*DeltaT,B*DeltaT,Q,R)
K4 = dlqr(I+A*DeltaT,B*DeltaT,Q,R)
K5 =  lqr(expm(A*DeltaT),(expm(A*DeltaT)-I)*A\B,Q,R)
K6 = dlqr(expm(A*DeltaT),(expm(A*DeltaT)-I)*A\B,Q,R)
```

Check Yourself

List the changes that are needed to convert the controller to discrete time.

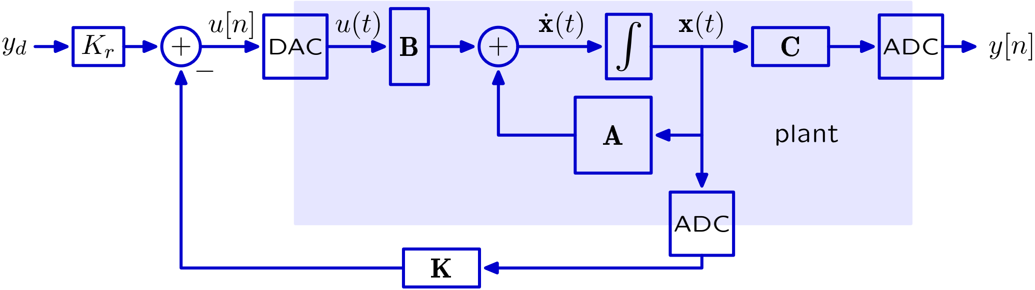


Consider the motor model with $\Delta T = 0.0001$ s. Which of these is better?

```
K1 =  lqr(A,B,Q,R)
K2 = dlqr(A,B,Q,R)
K3 =  lqr(I+A*DeltaT,B*DeltaT,Q,R)
K4 = dlqr(I+A*DeltaT,B*DeltaT,Q,R)
K5 =  lqr(expm(A*DeltaT),(expm(A*DeltaT)-I)*A\B,Q,R)
K6 = dlqr(expm(A*DeltaT),(expm(A*DeltaT)-I)*A\B,Q,R)
```

Check Yourself

List the changes that are needed to convert the controller to discrete time.

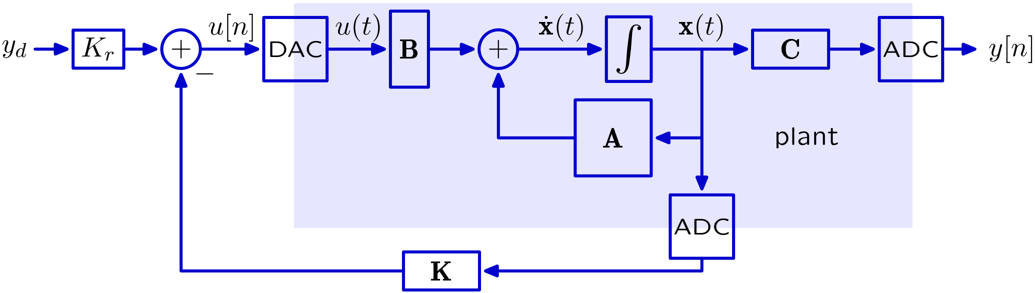


Consider the motor model with $\Delta T = 0.0001$ s. Which of these is better?

```
                                                           K
K1 =  lqr(A,B,Q,R)                               [  0.1597      0.6305]
K2 = dlqr(A,B,Q,R)                               [ -7.0024     -0.4600]
K3 =  lqr(I+A*DeltaT,B*DeltaT,Q,R)               [186.0068   9300.4668]
K4 = dlqr(I+A*DeltaT,B*DeltaT,Q,R)               [  0.1545      0.6287]
K5 =  lqr(expm(A*DeltaT),(expm(A*DeltaT)-I)*A\B,Q,R)  [ 97.8197 10016.3798]
K6 = dlqr(expm(A*DeltaT),(expm(A*DeltaT)-I)*A\B,Q,R)  [  0.1546      0.6288]
```

Check Yourself

List the changes that are needed to convert the controller to discrete time.



Consider the motor model with $\Delta T = 0.0001$ s. Which of these is better?

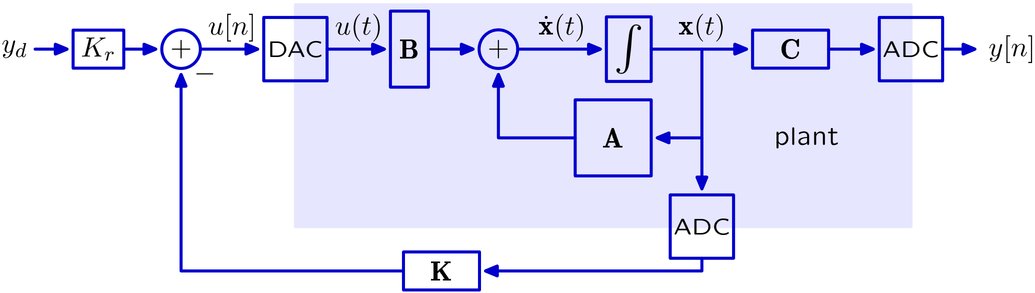
Since ΔT is small compared to the dynamics of the system, the path from $\mathbf{x}(t)$ to $u(t)$ is nearly instantaneous. Therefore we can approximate the behavior of this hybrid system as purely continuous.

$\rightarrow$ `K1` is a **reasonable approximation**.

`K2` never makes sense: one cannot run `dlqr` on a CT evolution matrix.

Check Yourself

List the changes that are needed to convert the controller to discrete time.



The evolution matrices for `K5` and `K6` are discrete approximations to the CT matrices $\mathbf{A}$ and $\mathbf{B}$.

→ `K5` doesn't make sense: cannot run `lqr` on CT matrices.

→ `K6` computes the result of a DT approximation to the CT system.

Similarly, `K3` and `K4` represent **first-order** DT approximations to the CT.

→ `K3` doesn't make sense: cannot run `lqr` on CT matrices.

→ `K4` represents a reasonable DT approximation to the CT system.