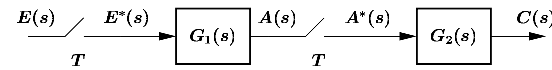
Today's EE 233 Lecture

- Analysis techniques for closed-loop DT systems.
- Review of open-loop system configurations.
- Introduce closed-loop system analysis techniques.
  - issues in deriving the output function.
  - original SFG and sampled SFG.
- State variable forms.
  - isolating the analog part of the system.
  - continuous-time to discrete-time transformations.

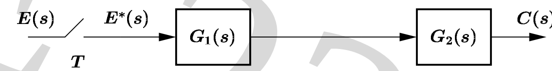
Preliminaries

- Open-loop systems.

  We derived (or at least tried to derive) the pulse transfer functions for different configurations.



$$C(z) = G_1(z)G_2(z)E(z)$$

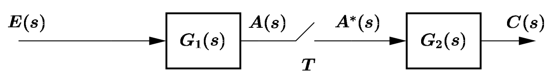


$$C(z) = [G_1G_2](z)E(z)$$

- For these two cases, it is not difficult to come up with the pulse transfer functions.

Preliminaries

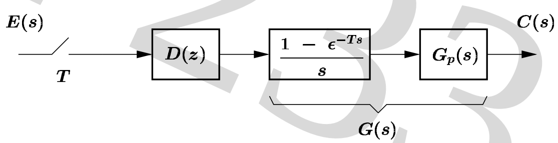
- Then, we considered



$$C(z) = G_2(z)[G_1E](z)$$

No pulse transfer function can be derived since the input term $E(z)$ cannot be factored out of $[G_1E](z)$.

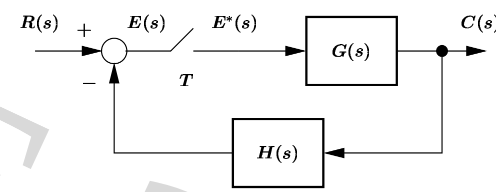
- With a digital filter.



$$C(z) = D(z)G(z)E(z)$$

Preliminaries

- Let us now consider the following closed-loop system.



- The output may be expressed as

$$C(s) = G(s)E^*(s)$$

while the error may be written as

$$E(s) = R(s) - H(s)C(s)$$

Preliminaries

- Eliminating $C(s)$ by combining the two equations gives

$$E(s) = R(s) - H(s)G(s)E^*(s)$$

- Taking the starred transform.

$$E^*(s) = R^*(s) - [GH]^*(s)E^*(s)$$

- Solving for $E^*(s)$.

$$E^*(s) = \frac{R^*(s)}{1 + [GH]^*(s)}$$

Preliminaries

- The continuous-time expression for the output is then

$$C(s) = G(s)\frac{R^*(s)}{1 + [GH]^*(s)}$$

- The discrete-time version is

$$C^*(s) = G^*(s)E^*(s) = \frac{G^*(s)R^*(s)}{1 + [GH]^*(s)}$$

$$C(z) = \frac{G(z)R(z)}{1 + [GH](z)}$$

Preliminaries

- Derivations for discrete-time closed-loop systems may not be as straightforward as the continuous-time case.

  Take for example, the output and error equations.

$$C(s) = G(s)E^*(s) \text{ and } E(s) = R(s) - H(s)C(s)$$

  We might directly solve for $C^*(s)$ by

  – getting $E^*(s)$ from second equation, and

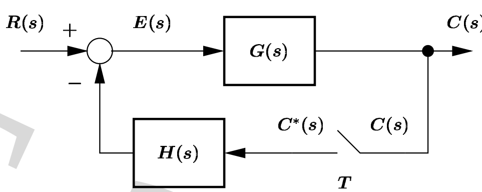
  – substituting the result into first equation.

$$C^*(s) = G^*(s)R^*(s) - G^*(s)[HC]^*(s)$$

  However, we cannot solve for transfer function since $C^*(s)$ cannot be factored out of $[HC]^*(s)$.

Preliminaries

- Now, analyze the following closed-loop system.



- Writing the output and error equations.

$$C(s) = G(s)E(s) \text{ and } E(s) = R(s) - H(s)C^*(s)$$

- Eliminating $E(s)$.

$$C(s) = G(s)R(s) - G(s)H(s)C^*(s)$$

Preliminaries

- Taking the starred transform.

$$C^*(s) = [GR]^*(s) - [GH]^*(s)C^*(s)$$

- Solving for $C^*(s)$ (and also $C(z)$), we get

$$C^*(s) = \frac{[GR]^*(s)}{1 + [GH]^*(s)}, \quad C(z) = \frac{[GR](z)}{1 + [GH](z)}$$

- Substituting the $C^*(s)$ back into the $C(s)$ equation, the continuous-time version is

$$C(s) = G(s)R(s) - \frac{G(s)H(s)[GR]^*(s)}{1 + [GH]^*(s)}$$

Preliminaries

- Indeed, no (pulse) transfer function may be derived for this system. The input function does not appear as a separate factor.

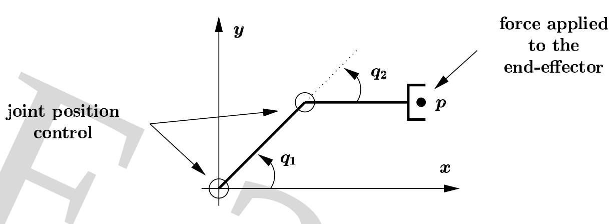
- Sampling the output signal instead of the input signal necessitates the combination of the input function with the plant transfer function in the analyses.

- Upon taking the starred transform, the input function is lost and may not be factored out from the starred expressions to reveal a transfer function for the system.

Closed-loop Discrete-time Systems
EE 233
©2002 M.C. Ramos
UP EEE Department

Preliminaries

- Example 1. Manipulator end-effector position control.



- The desired joint angle position comes from a sampled trajectory. Thus, a transfer function may be developed from joint angle input to the end-effector position.

  In contrast, a force applied to the end-effector can also be considered an input. However, a transfer function cannot be developed since the force is not sampled.

Closed-loop Discrete-time Systems
EE 233
©2002 M.C. Ramos
UP EEE Department

Derivation Using SFGs

- Difficult to determine the transfer functions for discrete-time systems.

  There is no transfer function for the ideal sampler.

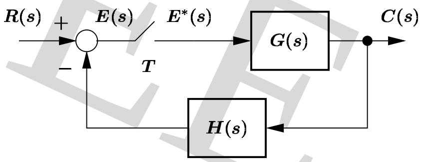
- Let us derive everything else about the system transfer function except for the part concerning the sampler.

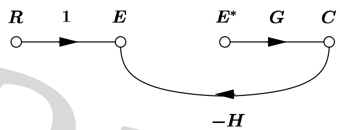
- We will perform the derivation by using a signal flow graph and omitting the sampler.

  This SFG (without the sampler) will be termed as the original SFG of the system.

Closed-loop Discrete-time Systems
EE 233
©2002 M.C. Ramos
UP EEE Department

- Consider the block diagram of a system with a sampler and the corresponding original SFG.





- After constructing the original SFG,
  - we assign a variable (e.g. $E(s)$) for the sampler input node and,
  - denote the sampler output node as the starred version of the assigned variable (e.g. $E^*(s)$).

- Write the relevant equations for the nodes, especially the system output node and the sampler input node.
- Take the starred transform of the equations. Solve for the output expression.
- Alternatively, the equations may be used to come up with a signal flow graph where Mason's gain formula can then used to derive the transfer function.

  This SFG is termed as the sampled signal flow graph.

- From our original SFG, we have

$$E(s) = R(s) - G(s)H(s)E^*(s)$$
$$C(s) = G(s)E^*(s)$$

Taking the starred transform.

$$E^*(s) = R^*(s) - [GH]^*(s)E^*(s)$$
$$C^*(s) = G^*(s)E^*(s)$$

Solving for the output.

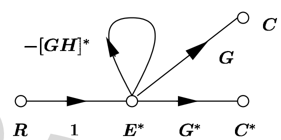
$$E^*(s) = \frac{R^*(s)}{1 + [GH]^*(s)}$$

$$C^*(s) = \frac{G^*(s)}{1 + [GH]^*(s)}R^*(s)$$

- Recalling the previous three equations.

$$C(s) = G(s)E^*(s)$$
$$E^*(s) = R^*(s) - [GH]^*(s)E^*(s)$$
$$C^*(s) = G^*(s)E^*(s)$$

We come up with the sampled SFG.

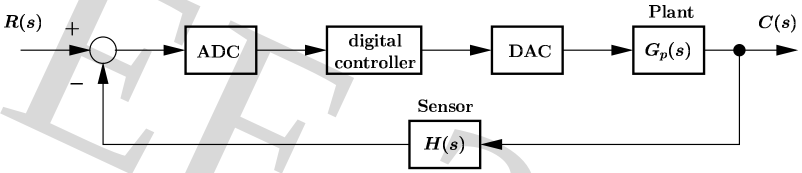


- From the SFG, $C^*(s)$ and $C(s)$ may be written as

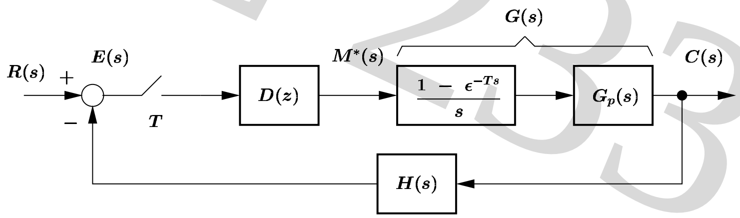
$$C^*(s) = \frac{G^*(s)}{1 + [GH]^*(s)}R^*(s)$$

$$C(s) = \frac{G(s)}{1 + [GH]^*(s)}R^*(s)$$

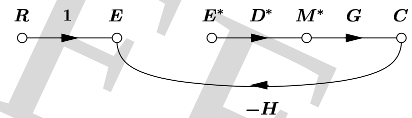
- Example 2. Consider the closed-loop control system.



The system may be modeled as



- Original SFG, and equations for nodes E and C.



$$E = R - GHD^*E^*$$
$$C = GD^*E^*$$

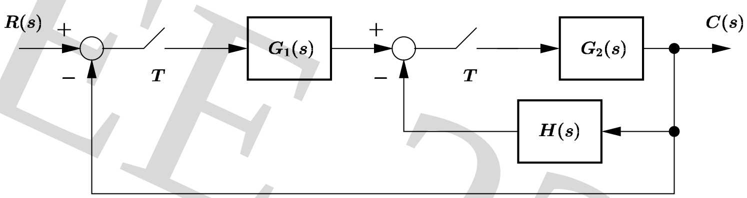
- Taking the starred transform, and solving for $E^*(s)$.

$$E^* = R^* - [GH]^*D^*E^* \Rightarrow E^* = \frac{R^*}{1 + [GH]^*D^*}$$

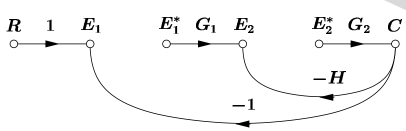
Solving for $C(z)$ from $C^*(s)$ and $E^*(s)$.

$$C^* = G^*D^*E^* \Rightarrow C(z) = \frac{G(z)D(z)}{1 + [GH](z)D(z)}R(z)$$

- Example 3. Consider the following system model.



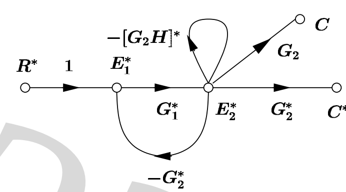
The original SFG and node equations are



$$E_1 = R - G_2E_2^*$$
$$E_2 = G_1E_1^* - G_2HE_2^*$$
$$C = G_2E_2^*$$

- Using the starred transform to get the sampled SFG.

$$E_1^* = R^* - G_2^*E_2^*$$
$$E_2^* = G_1^*E_1^* - [G_2H]^*E_2^*$$
$$C^* = G_2^*E_2^*$$



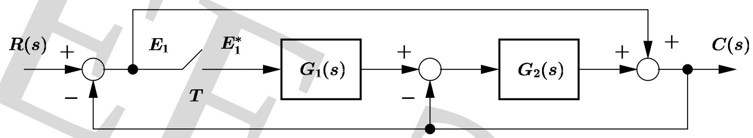
Using Mason's gain rule,

$$C^*(s) = \frac{G_1^*(s)G_2^*(s)}{1 + G_1^*(s)G_2^*(s) + [G_2H]^*(s)}R^*(s)$$

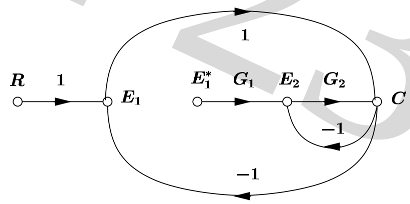
$$C(z) = \frac{G_1(z)G_2(z)}{1 + G_1(z)G_2(z) + [G_2H](z)}R(z)$$

Derivation Using SFGs

- **Example 4.** Consider the following system model.



The original SFG is



Derivation Using SFGs

- From the SFG, the equations for nodes E_1 and C are

$$E_1 = R - C$$

$$C = E_1 + G_2[G_1E_1^* - C]$$

Substituting, we get

$$C = R - C + G_1G_2E_1^* - G_2C$$

$$[2 + G_2]C = R + G_1G_2E_1^*$$

Thus,

$$C = \frac{1}{2 + G_2}R + \frac{G_1G_2}{2 + G_2}E_1^*$$

$$E_1 = \frac{1 + G_2}{2 + G_2}R - \frac{G_1G_2}{2 + G_2}E_1^*$$

Closed-loop Discrete-time Systems
EE 233

©2002 M.C. Ramos
UP EEE Department

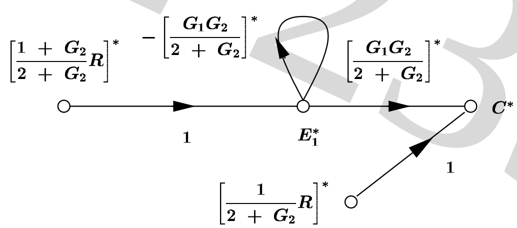
Derivation Using SFGs

- Taking the starred transform,

$$C^* = \left[\frac{1}{2 + G_2}R\right]^* + \left[\frac{G_1G_2}{2 + G_2}\right]^* E_1^*$$

$$E_1^* = \left[\frac{1 + G_2}{2 + G_2}R\right]^* - \left[\frac{G_1G_2}{2 + G_2}\right]^* E_1^*$$

Drawing the sampled SFG using the above equations.



Closed-loop Discrete-time Systems
EE 233

©2002 M.C. Ramos
UP EEE Department

Derivation Using SFGs

- Employing Mason's gain rule, we get

$$C^*(s) = \left[\frac{1}{2 + G_2}R\right]^*(s) + \frac{\left[\frac{G_1G_2}{2 + G_2}\right]^*(s)}{1 + \left[\frac{G_1G_2}{2 + G_2}\right]^*(s)} \left[\frac{1 + G_2}{2 + G_2}R\right]^*(s)$$

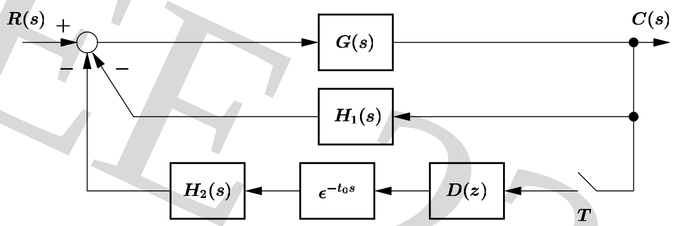
- No transfer function may be written for the system since the input to $G_2(s)$ is not sampled.

Closed-loop Discrete-time Systems
EE 233

©2002 M.C. Ramos
UP EEE Department

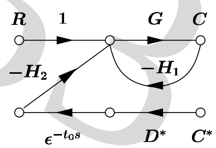
Derivation Using SFGs

- **Example 5.** System with a digital controller and delay.



From the original SFG,

$$C = \frac{GR}{1 + GH_1} - \frac{GH_2\epsilon^{-t_0 s}}{1 + GH_1}D^*C^*$$



Derivation Using SFGs

- Thus,

$$C(z) = \left[\frac{GR}{1 + GH_1}\right](z) - \left[\frac{GH_2}{1 + GH_1}\right](z,m)D(z)C(z)$$

where $mT = T - t_0$ for $t_0 < T$.
Solving for $C(z)$, we get

$$C(z) = \frac{\left[\frac{GR}{1 + GH_1}\right](z)}{1 + \left[\frac{GH_2}{1 + GH_1}\right](z,m)D(z)}$$

- What happens for $t_0 > T$?

State Variable Forms

- We will now look at a technique for expressing the system equations in state variable form.

- The system will be redrawn to separate the continuous-time part and the sampling blocks.

  A discrete-time model will be derived for the continuous-time system.

  The discrete-time model for the whole system would then be written.

- This technique is applicable for closed-loop systems as well as for open-loop systems.

State Variable Forms

- How do you derive a discrete-time model from a continuous-time model?

  We need to first derive a state variable model from the continuous-time system transfer function.

  Then, we will convert this state variable model to the equivalent DT state variable model.

- Let us start from a general form of a transfer function.

$$G(s) = \frac{b_{n-1}s^{n-1} + b_{n-2}s^{n-2} + \ldots + b_0}{s^n + a_{n-1}s^{n-1} + \ldots + a_0}$$

- Introducing a dummy variable $E(s)$ and using $G(s) = Y(s)/R(s)$,

$$\frac{Y(s)}{R(s)} = \frac{b_{n-1}s^{n-1} + b_{n-2}s^{n-2} + \ldots + b_0}{s^n + a_{n-1}s^{n-1} + \ldots + a_0} \cdot \frac{E(s)}{E(s)}$$

- Splitting the above equation gives

$$Y(s) = (b_{n-1}s^{n-1} + b_{n-2}s^{n-2} + \ldots + b_0)E(s)$$
$$R(s) = (s^n + a_{n-1}s^{n-1} + \ldots + a_0)E(s)$$

- Recalling the Laplace transform for the derivative operator, we can assign state variables as

$$\begin{aligned}
E(s) &\rightarrow e(t) \triangleq x_1(t) \\
sE(s) &\rightarrow \dot{e}(t) = \dot{x}_1(t) \triangleq x_2(t) \\
s^2E(s) &\rightarrow \ddot{e}(t) = \dot{x}_2(t) \triangleq x_3(t) \\
&\vdots \\
s^{n-1}E(s) &\rightarrow \frac{d^{n-1}}{dt^{n-1}}e(t) = \dot{x}_{n-1}(t) \triangleq x_n(t) \\
s^nE(s) &\rightarrow \frac{d^n}{dt^n}e(t) = \dot{x}_n(t)
\end{aligned}$$

We now have all the state equations except for $\dot{x}_n(t)$.

- Expanding and taking the inverse s-transform of

$$R(s) = (s^n + a_{n-1}s^{n-1} + \ldots + a_0)E(s)$$

gives us the state equation for $\dot{x}_n(t)$.

$$\begin{aligned}
\dot{x}_n(t) = &-a_0x_1(t) - a_1x_2(t) - \ldots \\
&- a_{n-1}x_n(t) + u(t)
\end{aligned}$$

The rest of the state equations are simply,

$$\begin{aligned}
\dot{x}_1(t) &= x_2(t) \\
\dot{x}_2(t) &= x_3(t) \\
&\vdots \\
\dot{x}_{n-1}(t) &= x_n(t)
\end{aligned}$$

- In matrix form,

$$\dot{x}(t) = Ax(t) + Bu(t)$$

where

$$x(k) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} \Rightarrow \dot{x}(t) = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_n(t) \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ 0 & 0 & 0 & \ldots & 0 \\ & & \vdots & & \\ -a_0 & -a_1 & -a_2 & \ldots & -a_{n-1} \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

State Variable Forms

- The output equation is obtained by expanding and taking the inverse s-transform of

$$Y(s) = (b_{n-1}s^{n-1} + b_{n-2}s^{n-2} + \ldots + b_0)E(s)$$

which gives in matrix form,

$$y(t) = [b_0\ b_1\ \ldots\ b_{n-1}] \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}$$

or $y(t) = Cx(t)$ where $C = [b_0\ b_1\ \ldots\ b_{n-1}]$.

- Now, how do we go from

$$\dot{x}(t) = A_c x(t) + B_c u(t)$$
$$y(t) = C_c x(t)$$

to discrete-time equivalent

$$x(k+1) = A_d x(k) + B_d u(k)$$
$$y(k) = C_d x(k)$$

- Basically, we find a solution for the continuous-time equations for one sampling period T.

We set the input to the CT system to be

$$u(t) = u(kT) \qquad kT \leq t < (k+1)T$$

- Solve the state equation first.

$$\dot{x}(t) = A_c x(t) + B_c u(t)$$

- Use linearity and supersposition.

If $u(t) = 0$ for all t, then the solution is

$$x_h(t) = \epsilon^{A_c t} c_1$$

where c_1 is some constant vector.

With no initial conditions, and taking into account that input is some constant $\bar{u}$ for the sampling interval T,

$$x_p(t) = -A_c^{-1} B_c \bar{u} \qquad \text{assuming } A_c^{-1} \text{ exists.}$$

- Thus, if $x(0)$ denotes the initial condition of the system

$$x(t) = \epsilon^{A_c t}[x(0) + A_c^{-1} B_c \bar{u}] - A_c^{-1} B_c \bar{u}$$
$$= \epsilon^{A_c t} x(0) + [\epsilon^{A_c t} - I] A_c^{-1} B_c \bar{u}$$

- We calculate the state of the system after one sampling period T by computing relative to the start of the sampling interval.

Note that at the start of the sampling interval, the system state is $x(kT)$. Thus, we use $x(kT)$ as our initial condition for that time interval.

Furthermore, the input is constant for the sampling period, i.e. $\bar{u} = u(kT)$.

State Variable Forms

- The state of the system after one sampling period T denoted by $x[(k+1)T]$ is

$$x[(k+1)T] = \epsilon^{A_cT}x(kT) + [\epsilon^{A_cT} - I]A_c^{-1}B_cu(kT)$$

- For a discrete-time system, we are only interested in the states at the sampling points.

  From the above equation we can now develop our DT state equation as

$$x[(k+1)T] = A_dx(kT) + B_du(kT)$$

where $A_d = \epsilon^{A_cT}$ and $B_d = [\epsilon^{A_cT} - I]A_c^{-1}B_c$.

State Variable Forms

- As for the DT output equation, it is the CT output equation considered for discrete times $t = kT$. Thus,

$$y(t) = C_cx(t) \Rightarrow y(kT) = C_dx(kT)$$

where $C_d = C_c$.

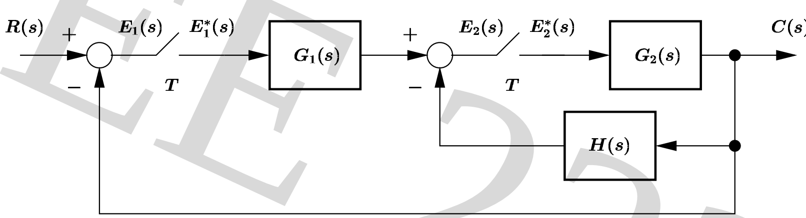
- Since we are primarily studying discrete-time systems, we usually drop the subscripts and the sampling period notations from our discrete-time equations.

$$x(k+1) = Ax(k) + Bu(k)$$
$$y(k) = Cx(k)$$

where $A = \epsilon^{A_cT}$, $B = [\epsilon^{A_cT} - I]A_c^{-1}B_c$ and $C = C_c$.

State Variable Forms

- Example 6. Let $T = 0.1$ s. Derive the discrete-time state variable model for



$$G_1(s) = \frac{1 - \epsilon^{-Ts}}{s}G_{p1}(s), \quad G_2(s) = \frac{1 - \epsilon^{-Ts}}{s}G_{p2}(s)$$

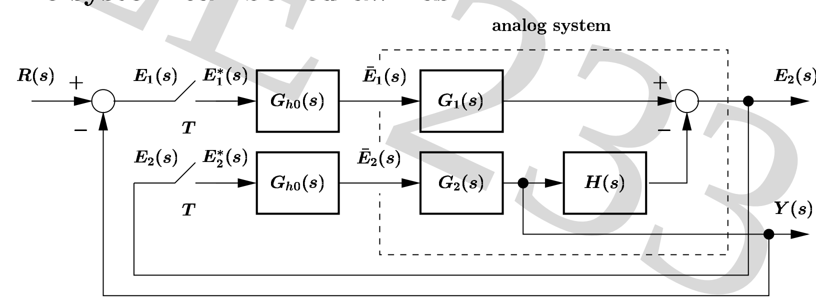
$$H(s) = \frac{10}{s+10}$$

State Variable Forms

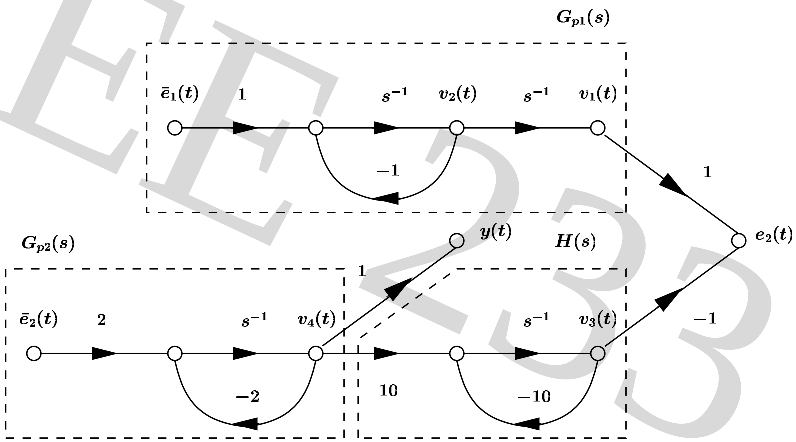
- The plant transfer functions are

$$G_{p1}(s) = \frac{1}{s(s+1)} \text{ and } G_{p2}(s) = \frac{2}{s+2}$$

- The system can be redrawn as



- The SFG of the analog system is



- The continous-time state equations can be written from the SFG.

$$\dot{v}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -10 & 10 \\ 0 & 0 & 0 & -2 \end{bmatrix} v(t) + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} \bar{e}_1(t) \\ \bar{e}_2(t) \end{bmatrix}$$

$$\begin{bmatrix} y(t) \\ e_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & -1 & 0 \end{bmatrix} v(t)$$

or

$$\dot{v}(t) = A_c v(t) + B_c \begin{bmatrix} \bar{e}_1(t) \\ \bar{e}_2(t) \end{bmatrix} \text{ and } \begin{bmatrix} y(t) \\ e_2(t) \end{bmatrix} = C_c v(t)$$

- The DT state equations will now be derived from the continuous-time equations.

$$v(k+1) = Av(k) + B \begin{bmatrix} e_1(k) \\ e_2(k) \end{bmatrix}$$

$$\begin{bmatrix} y(k) \\ e_2(k) \end{bmatrix} = Cv(k)$$

Notice that $e_2(k)$ is the discrete-time versions of both $e_2(t)$ and $\bar{e}_2(t)$.

- The matrix C is derived from

$$C = C_c = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & -1 & 0 \end{bmatrix}$$

- If A_c^{-1} exists, matrices A and B may be computed from

$$A = \epsilon^{A_c T} \text{ and } B = [\epsilon^{A_c T} - I]A_c^{-1}B_c$$

However, for this case, A_c is singular.

- We may expand out the factor $[\epsilon^{A_c T} - I]A_c^{-1}$ by Taylor series expansion to eliminate the need to compute A_c^{-1}.

$$\epsilon^{A_c T} = I + A_c T + A_c^2 \frac{T^2}{2!} + A_c^3 \frac{T^3}{3!} + \ldots$$

$$[\epsilon^{A_c T} - I]A_c^{-1} = IT + A_c \frac{T^2}{2!} + A_c^2 \frac{T^3}{3!} + \ldots$$

State Variable Forms

- Alternatively, we may use Octave's `c2d` function.

```
>> [A,B] = sys2ss( c2d( ss2sys(Ac,Bc,Cc), T))

 A =
     1.00000  0.09516  0.00000  0.00000
     0.00000  0.90484  0.00000  0.00000
     0.00000  0.00000  0.36788  0.56356
     0.00000  0.00000  0.00000  0.81873

 B =
     0.00484  0.00000
     0.09516  0.00000
     0.00000  0.06856
     0.00000  0.18127
```


State Variable Forms

- Our discrete-time version for the analog part of the system is now known.

$$\begin{aligned} v(k+1) &= Av(k) + B\begin{bmatrix} e_1(k) \\ e_2(k) \end{bmatrix} \\ \begin{bmatrix} y(k) \\ e_2(k) \end{bmatrix} &= Cv(k) \end{aligned}$$

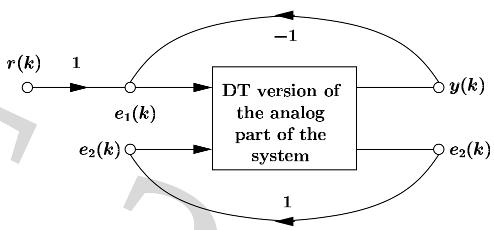
- Now we need to get everything together.
  - the input $r(k)$ should appear in the state equation.
  - the output equation should be for $y(k)$ only.

  We need to get expressions for $e_1(k)$ and $e_2(k)$.

State Variable Forms

- Look at the SFG of the whole system.

  $e_1(k) = r(k) - y(k)$



- If we decompose the C matrix into $\begin{bmatrix} C_1 \\ C_2 \end{bmatrix}$, we get

$$\begin{aligned} y(k) = C_1 v(k) &\Rightarrow e_1(k) = r(k) - C_1 v(k) \\ e_2(k) &= C_2 v(k) \end{aligned}$$

State Variable Forms

- Decomposing the B matrix into $[B_1\ B_2]$, we can write

$$v(k+1) = Av(k) + [B_1\ B_2]\begin{bmatrix} r(k) - C_1 v(k) \\ C_2 v(k) \end{bmatrix}$$

- Thus, we have a discrete-time state variable model for our system.

$$\begin{aligned} v(k+1) &= [A - B_1C_1 + B_2C_2]v(k) + B_1 r(k) \\ y(k) &= C_1 v(k) \end{aligned}$$

Summary

- **Review of open-loop systems.**
- **Closed-loop system analysis techniques.**
  - **starred transform.**
  - **original SFG.**
  - **sampled SFG.**
- **State variable forms.**
  - **isolating the analog part of the system.**
  - **performing continuous-time to discrete-time transformations.**