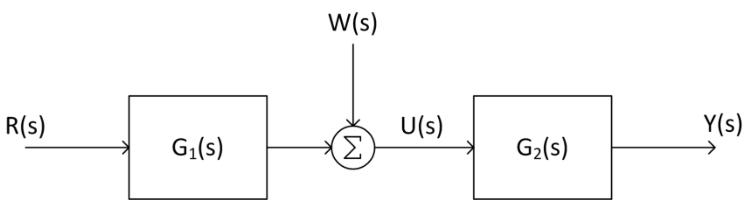
SECTION 2: BLOCK DIAGRAMS & SIGNAL FLOW GRAPHS

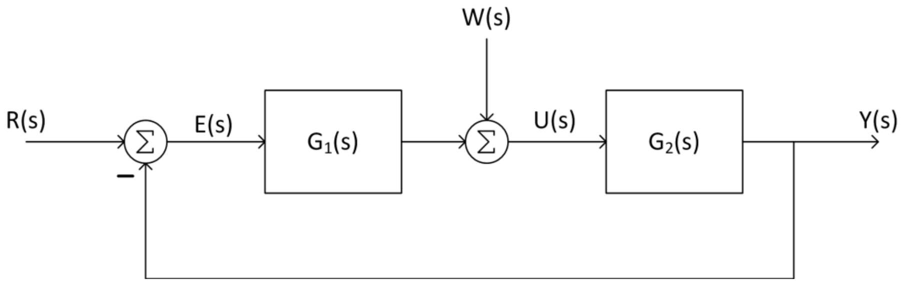
ESE 430 – Feedback Control Systems

Block Diagrams

Block Diagrams

- In the introductory section we saw examples of ***block diagrams*** to represent systems, e.g.:

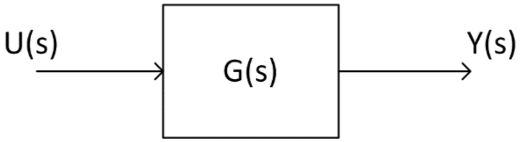




- Block diagrams consist of
  - ***Blocks*** – these represent subsystems – typically modeled by, and labeled with, a transfer function
  - ***Signals*** – inputs and outputs of blocks – signal direction indicated by arrows – could be voltage, velocity, force, etc.
  - ***Summing junctions*** – points were signals are algebraically summed – subtraction indicated by a negative sign near where the signal joins the summing junction

Standard Block Diagram Forms

- The basic input/output relationship for a single block is:



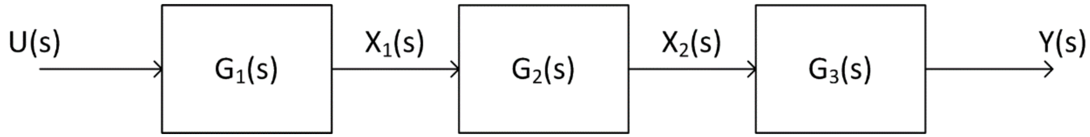
$$Y(s) = U(s) \cdot G(s)$$

- Block diagram blocks can be connected in three basic forms:
  - ***Cascade***
  - ***Parallel***
  - ***Feedback***

- We'll next look at each of these forms and derive a single-block equivalent for each

Cascade Form

- Blocks connected in ***cascade***:



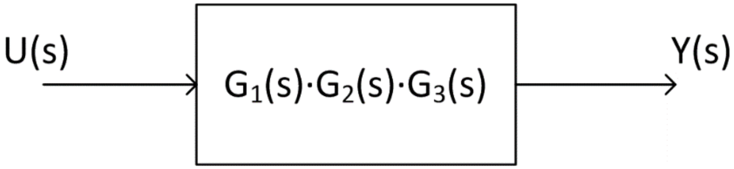
$$X_1(s) = U(s) \cdot G_1(s), \quad X_2(s) = X_1(s) \cdot G_2(s)$$

$$Y(s) = X_2(s) \cdot G_3(s) = X_1(s) \cdot G_2(s) \cdot G_3(s)$$

$$Y(s) = U(s) \cdot G_1(s) \cdot G_2(s) \cdot G_3(s) = U(s) \cdot G_{eq}(s)$$

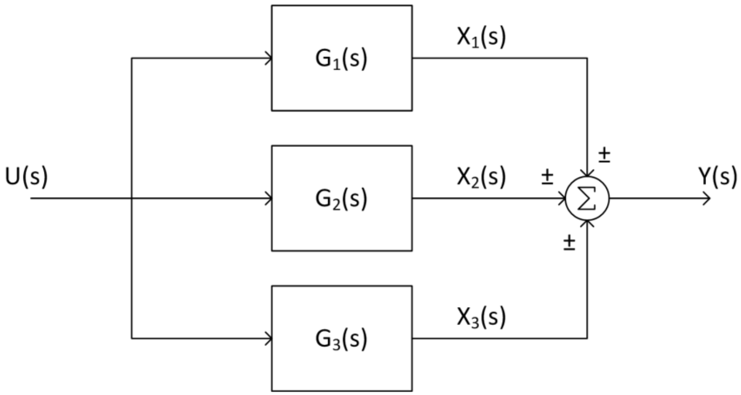
$$G_{eq}(s) = G_1(s) \cdot G_2(s) \cdot G_3(s)$$

- The equivalent transfer function of cascaded blocks is the ***product*** of the individual transfer functions



Parallel Form

- Blocks connected in parallel:



$$X_1(s) = U(s) \cdot G_1(s)$$

$$X_2(s) = U(s) \cdot G_2(s)$$

$$X_3(s) = U(s) \cdot G_3(s)$$

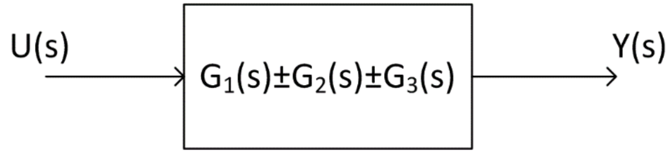
$$Y(s) = X_1(s) \pm X_2(s) \pm X_3(s)$$

$$Y(s) = U(s) \cdot G_1(s) \pm U(s) \cdot G_2(s) \pm U(s) \cdot G_3(s)$$

$$Y(s) = U(s)[G_1(s) \pm G_2(s) \pm G_3(s)] = U(s) \cdot G_{eq}(s)$$

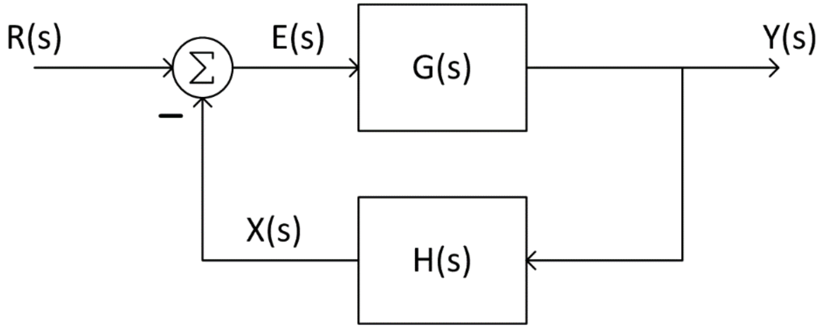
$$G_{eq}(s) = G_1(s) \pm G_2(s) \pm G_3(s)$$

- The equivalent transfer function is the ***sum*** of the individual transfer functions:



Feedback Form

- Of obvious interest to us, is the ***feedback form***:



$$Y(s) = E(s)G(s)$$

$$Y(s) = [R(s) - X(s)]G(s)$$

$$Y(s) = [R(s) - Y(s)H(s)]G(s)$$

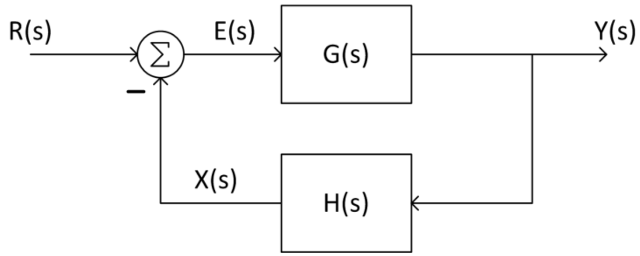
$$Y(s)[1 + G(s)H(s)] = R(s)G(s)$$

$$Y(s) = R(s) \cdot \frac{G(s)}{1 + G(s)H(s)}$$

- The ***closed-loop transfer function***, $T(s)$, is

$$T(s) = \frac{Y(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)}$$

Feedback Form



$$T(s) = \frac{G(s)}{1 + G(s)H(s)}$$

- Note that this is ***negative feedback***, for ***positive feedback***:

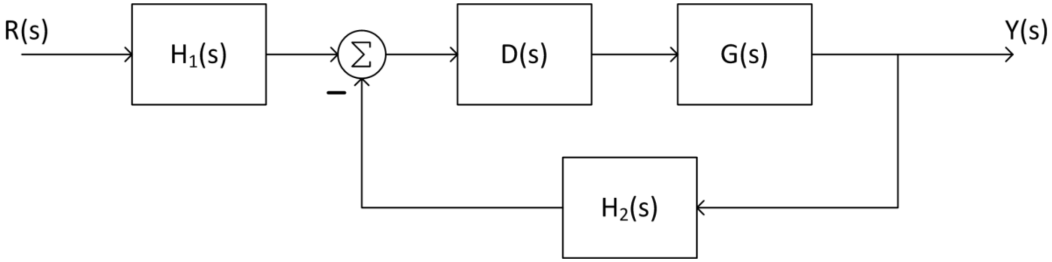
$$T(s) = \frac{G(s)}{1 - G(s)H(s)}$$

- The $G(s)H(s)$ factor in the denominator is the ***loop gain*** or ***open-loop transfer function***
- The gain from input to output with the feedback path broken is the ***forward path gain*** – here, $G(s)$
- In general:

$$T(s) = \frac{\text{forward path gain}}{1 - \text{loop gain}}$$

Closed-Loop Transfer Function - Example

- Calculate the closed-loop transfer function



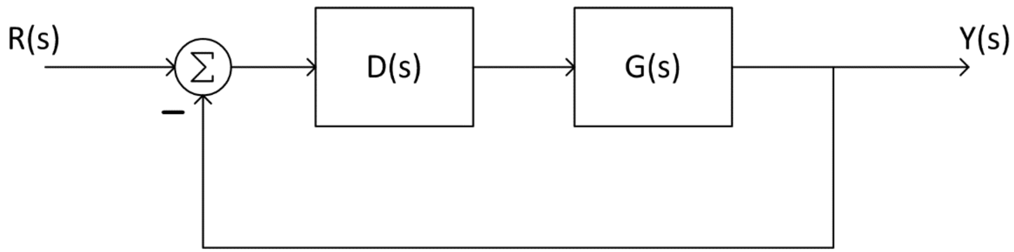
- $D(s)$ and $G(s)$ are in cascade
- $H_1(s)$ is in cascade with the feedback system consisting of $D(s)$, $G(s)$, and $H_2(s)$

$$T(s) = H_1(s) \cdot \frac{D(s)G(s)}{1 + D(s)G(s)H_2(s)}$$

$$T(s) = \frac{H_1(s)D(s)G(s)}{1 + D(s)G(s)H_2(s)}$$

Unity-Feedback Systems

- We're often interested in ***unity-feedback systems***



- Feedback path gain is unity
  - Can always reconfigure a system to unity-feedback form
- Closed-loop transfer function is:

$$T(s) = \frac{D(s)G(s)}{1 + D(s)G(s)}$$

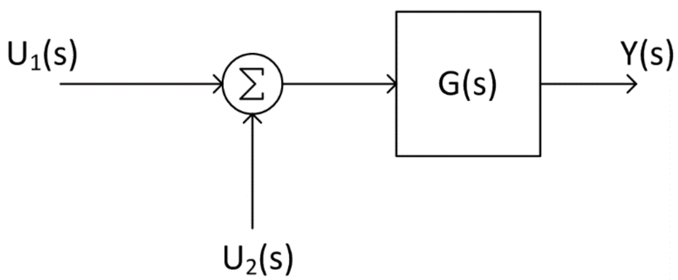
Block Diagram Manipulation

Block Diagram Algebra

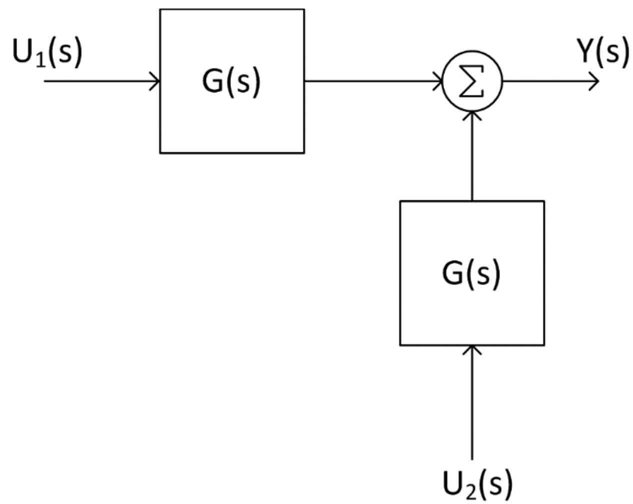
- Often want to simplify block diagrams into simpler, recognizable forms
  - To determine the equivalent transfer function
- Simplify to instances of the three standard forms, then simplify those forms
- ***Move blocks around relative to summing junctions and pickoff points*** – simplify to a standard form
  - Move blocks forward/backward past summing junctions
  - Move blocks forward/backward past pickoff points

Moving Blocks Back Past a Summing Junction

- The following two block diagrams are equivalent:



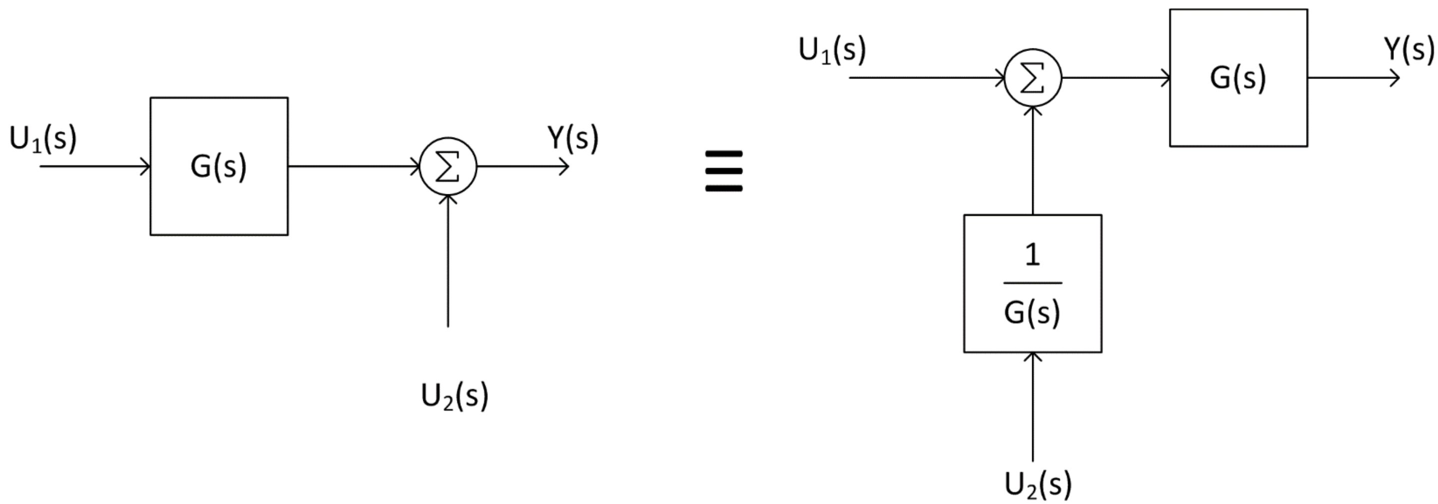
≡



$$Y(s) = [U_1(s) + U_2(s)]G(s) = U_1(s)G(s) + U_2(s)G(s)$$

Moving Blocks Forward Past a Summing Junction

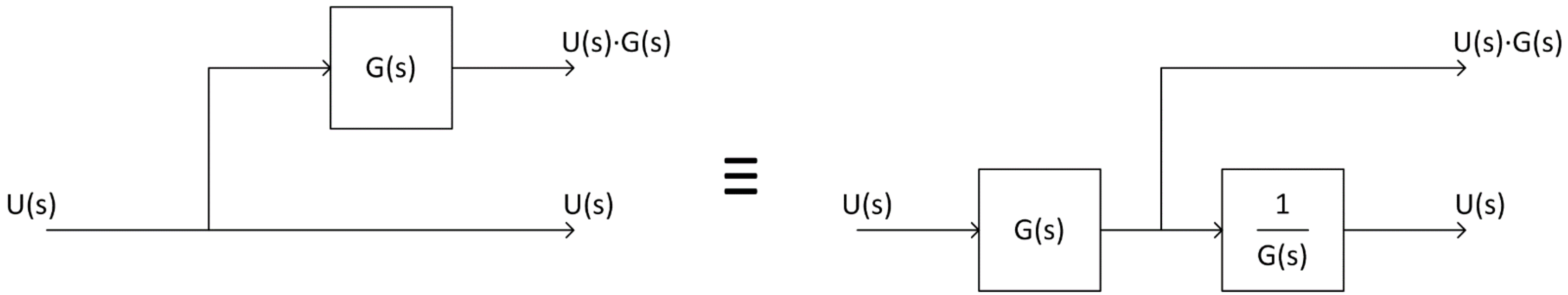
- The following two block diagrams are equivalent:



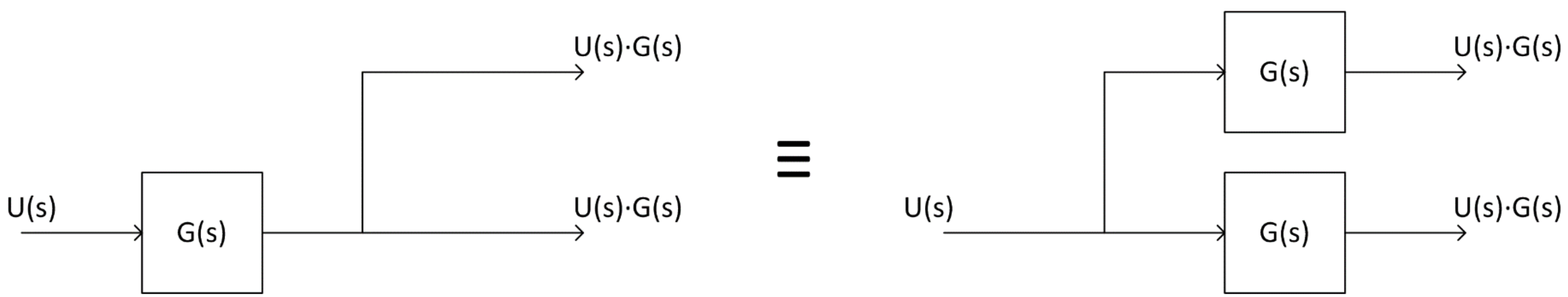
$$Y(s) = U_1(s)G(s) + U_2(s) = \left[U_1(s) + U_2(s)\frac{1}{G(s)}\right]G(s)$$

Moving Blocks Relative to Pickoff Points

- We can move blocks backward past pickoff points:

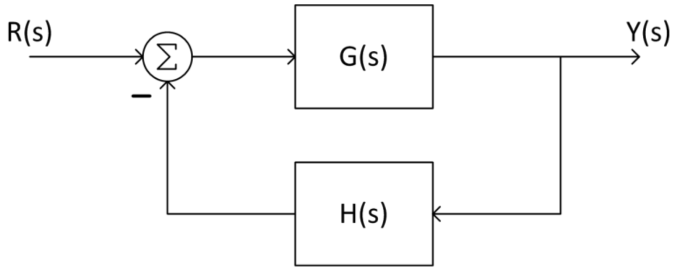


- And, we can move them forward past pickoff points:

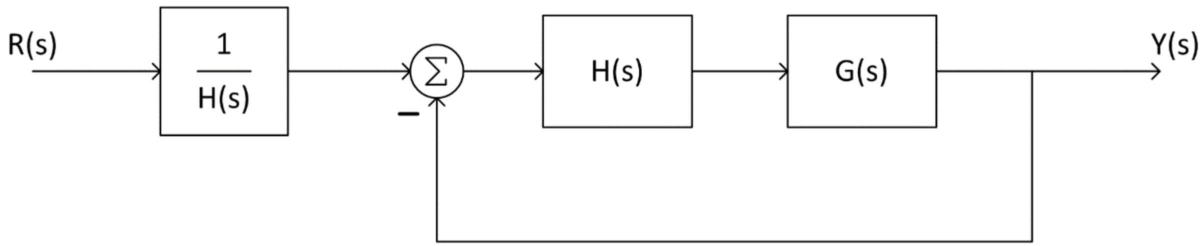


Block Diagram Simplification – Example 1

- Rearrange the following into a unity-feedback system



- Move the feedback block, $H(s)$, forward, past the summing junction
- Add an inverse block on $R(s)$ to compensate for the move

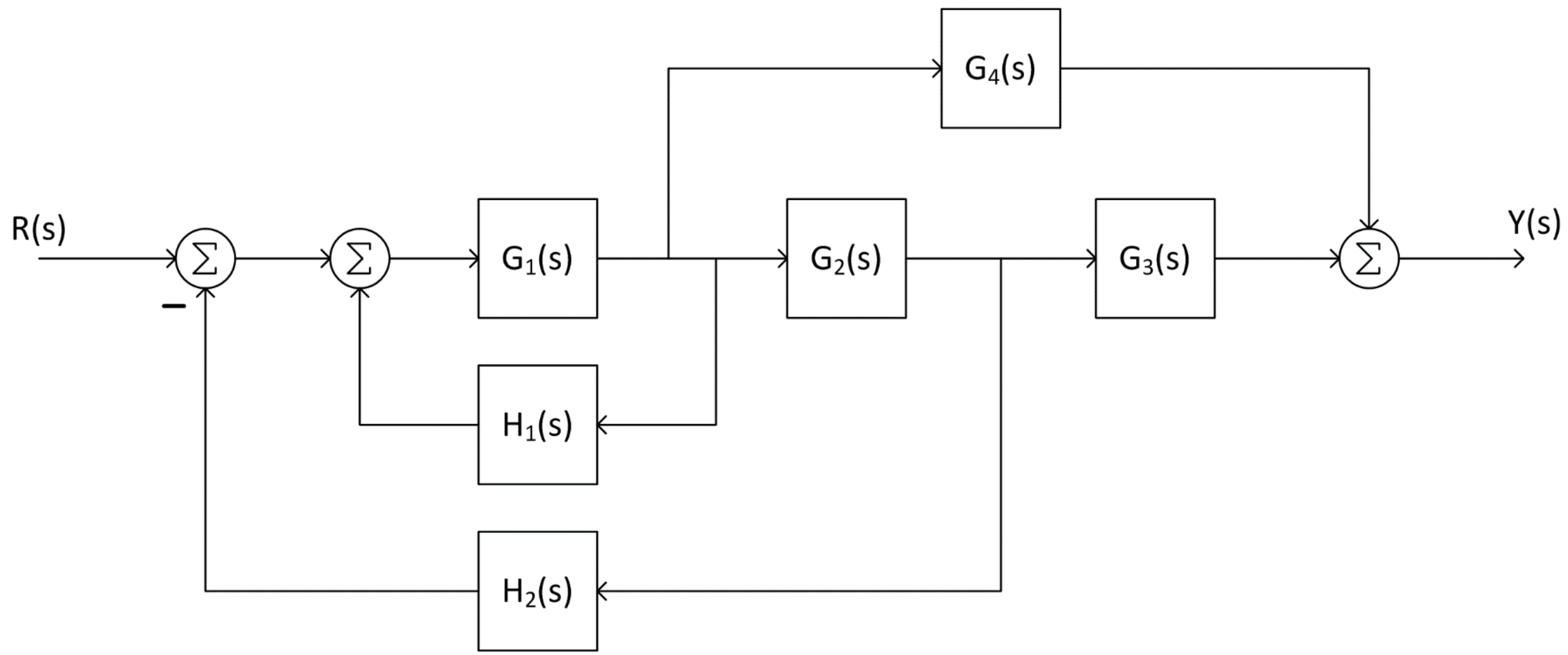


- Closed-loop transfer function:

$$T(s) = \frac{\frac{1}{H(s)}H(s)G(s)}{1 + G(s)H(s)} = \frac{G(s)}{1 + G(s)H(s)}$$

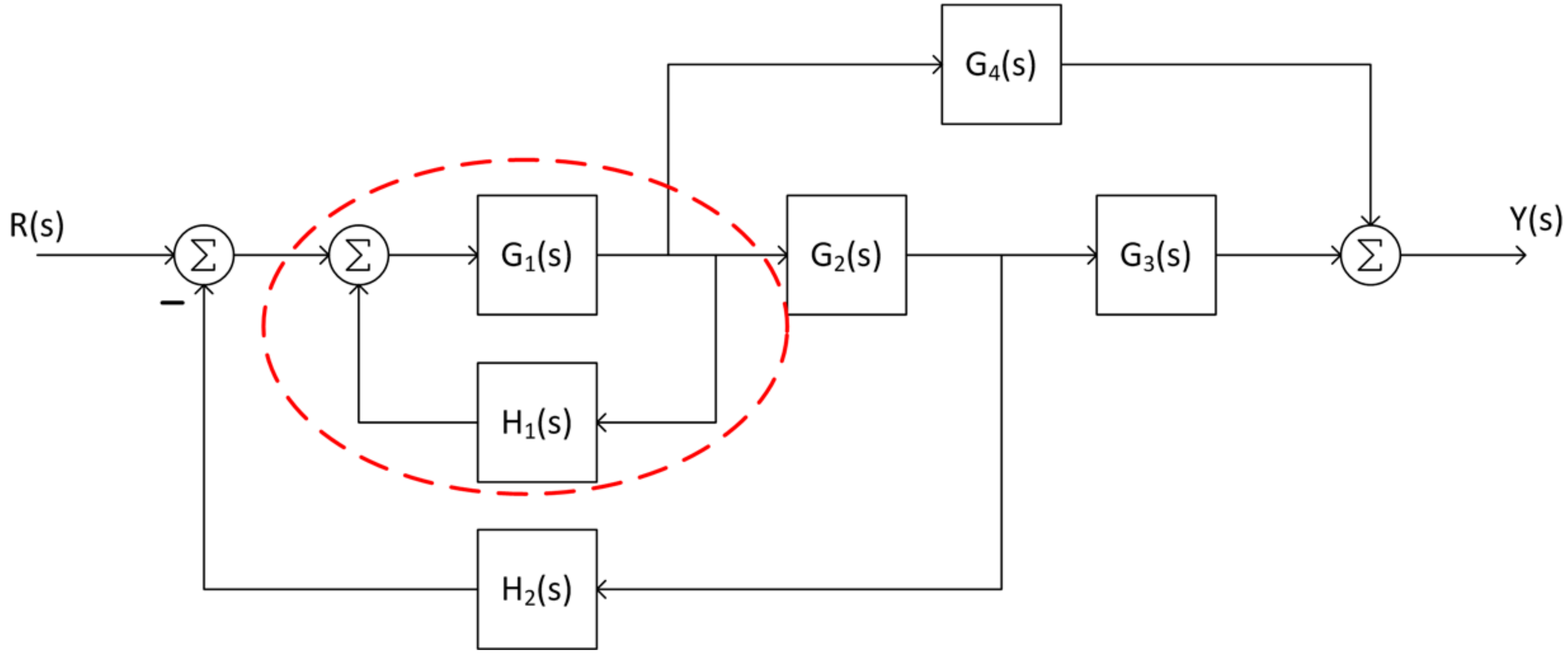
Block Diagram Simplification – Example 2

- Find the closed-loop transfer function of the following system through block-diagram simplification



Block Diagram Simplification – Example 2

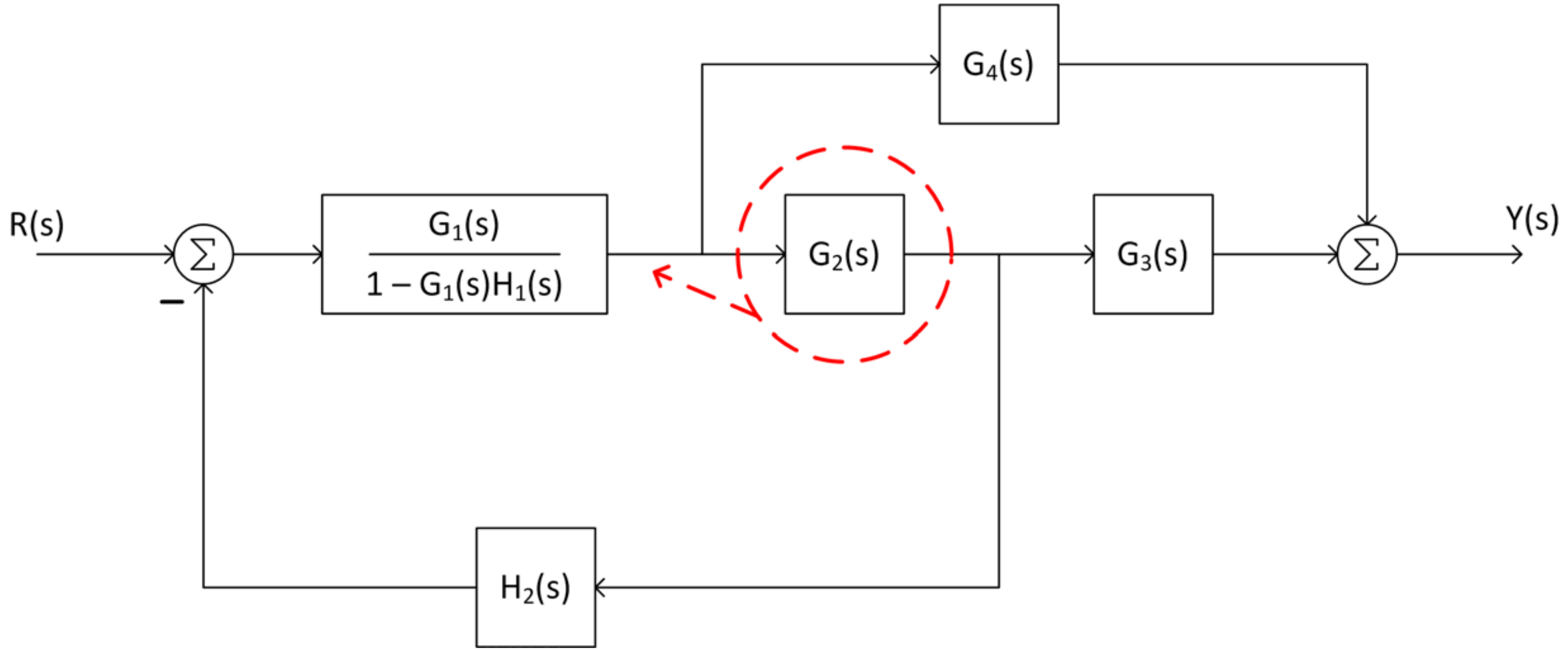
- $G_1(s)$ and $H_1(s)$ are in feedback form



$$G_{eq}(s) = \frac{G_1(s)}{1 - G_1(s)H_1(s)}$$

Block Diagram Simplification – Example 2

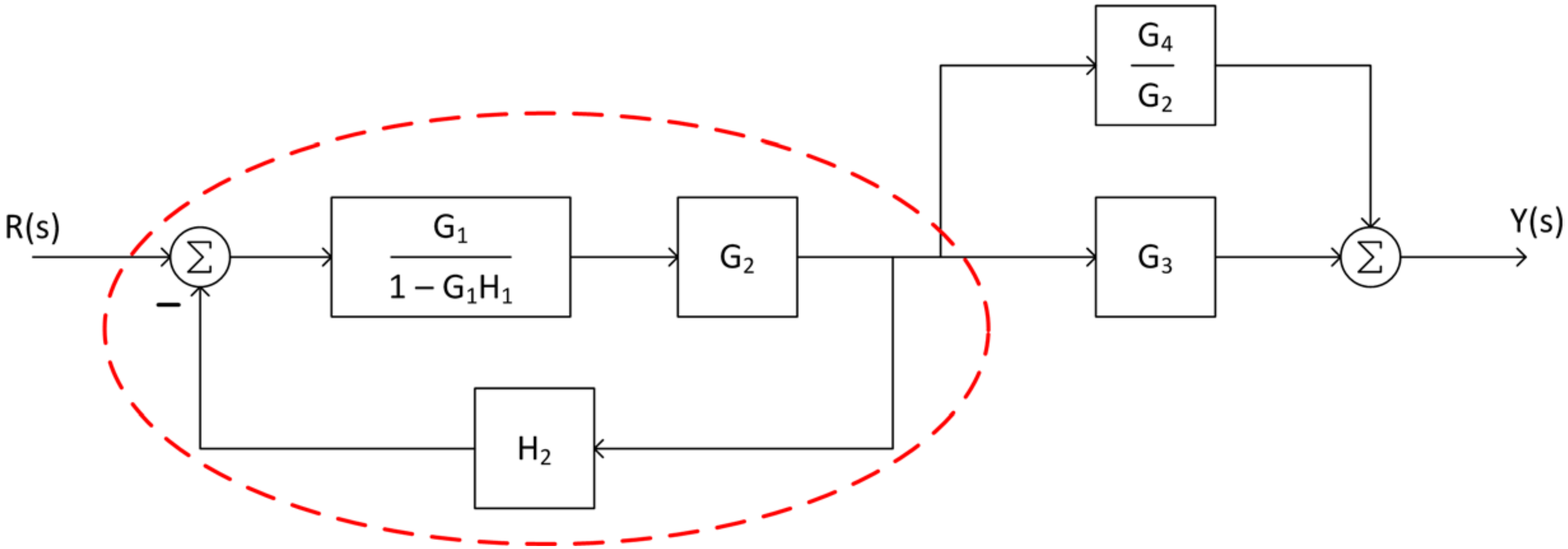
- Move $G_2(s)$ backward past the pickoff point



- Block from previous step, $G_2(s)$, and $H_2(s)$ become a feedback system that can be simplified

Block Diagram Simplification – Example 2

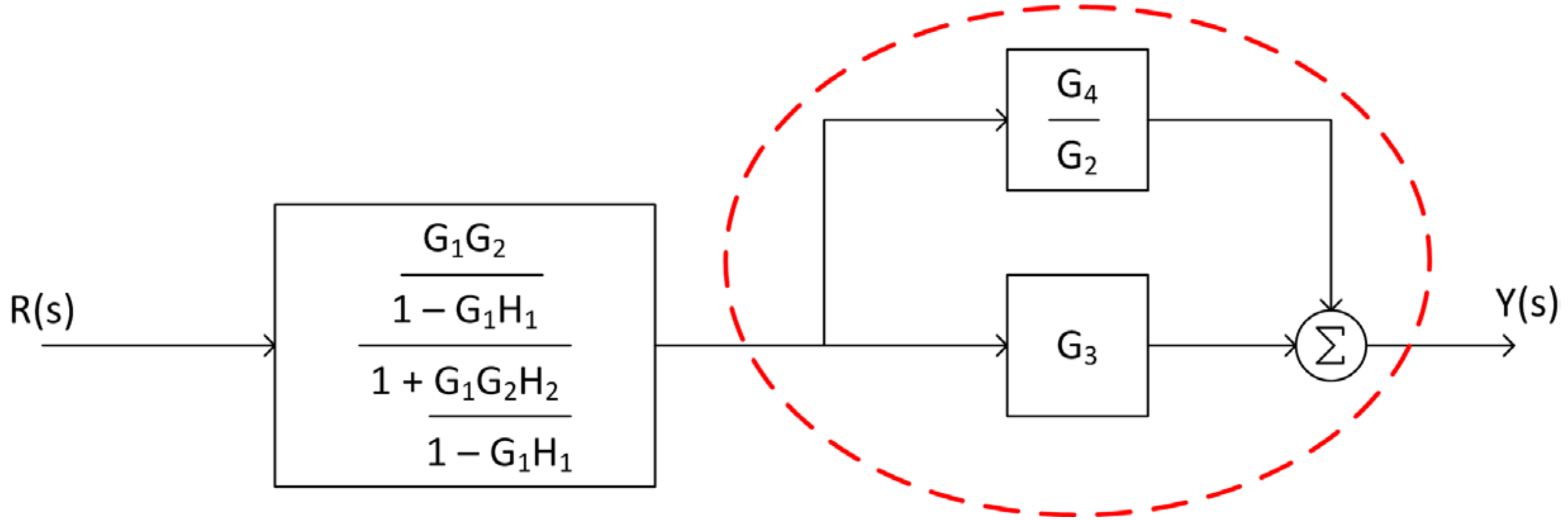
- Simplify the feedback subsystem
- Note that we've dropped the function of s notation, (s), for clarity



$$G_{eq}(s) = \frac{\frac{G_1G_2}{1 - G_1H_1}}{1 + \frac{G_1G_2H_2}{1 - G_1H_1}} = \frac{G_1G_2}{1 - G_1H_1 + G_1G_2H_2}$$

Block Diagram Simplification – Example 2

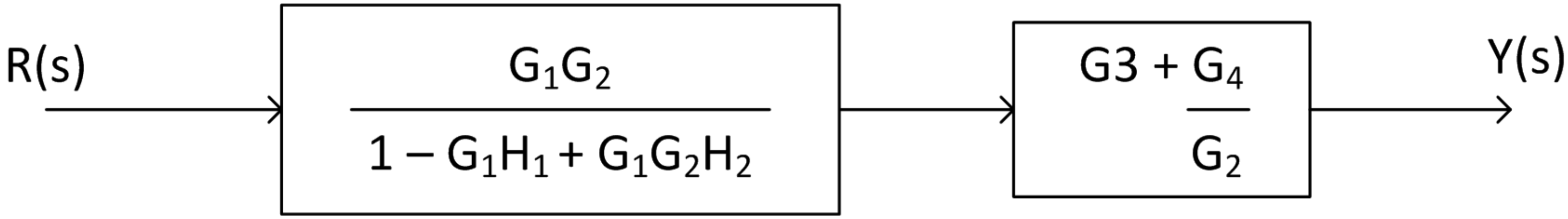
- Simplify the two parallel subsystems



$$G_{eq}(s) = G_3 + \frac{G_4}{G_2}$$

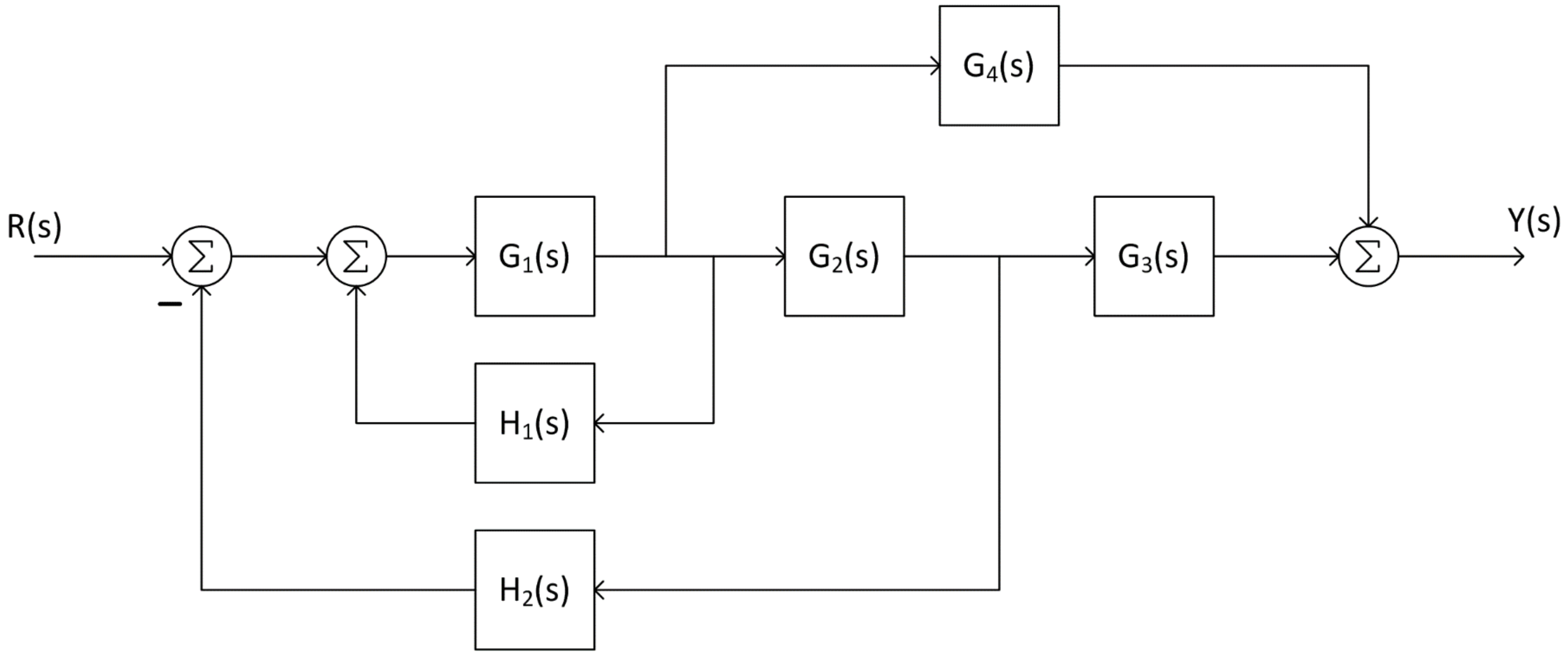
Block Diagram Simplification – Example 2

- Now left with two cascaded subsystems
  - Transfer functions multiply



$$G_{eq}(s) = \frac{G_1 G_2 G_3 + G_1 G_4}{1 - G_1 H_1 + G_1 G_2 H_2}$$

Block Diagram Simplification – Example 2



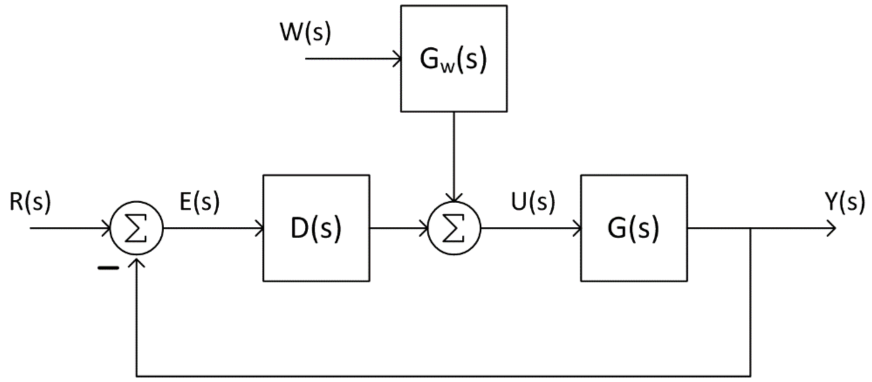
- The equivalent, close-loop transfer function is

$$T(s) = \frac{G_1G_2G_3 + G_1G_4}{1 - G_1H_1 + G_1G_2H_2}$$

Multiple-Input Systems

Multiple Input Systems

- Systems often have more than one input
  - E.g., reference, $R(s)$, and disturbance, $W(s)$



- Two transfer functions:
  - From reference to output

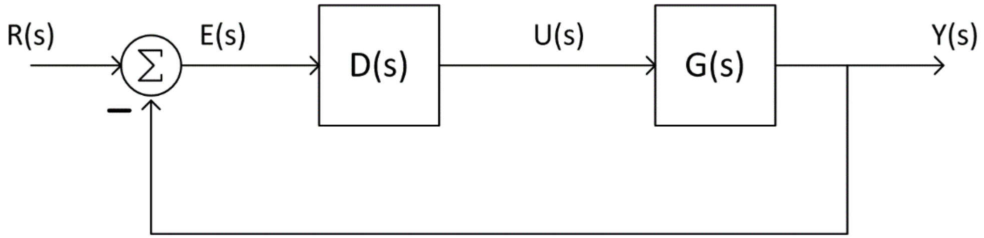
$$T(s) = Y(s)/R(s)$$

  - From disturbance to output

$$T_w(s) = Y(s)/W(s)$$

Transfer Function – Reference

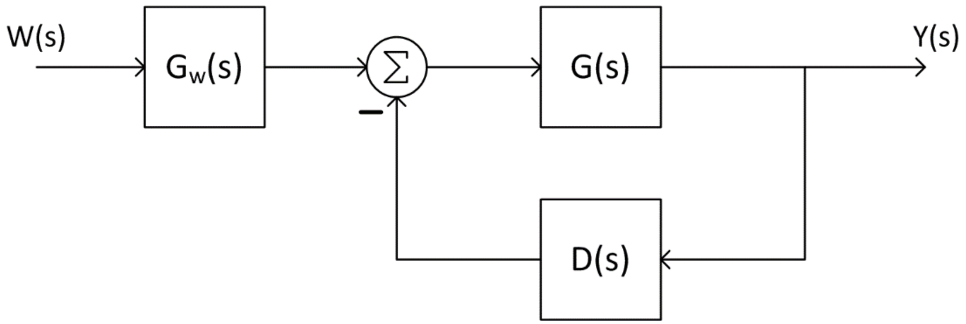
- Find transfer function from $R(s)$ to $Y(s)$
  - A linear system – superposition applies
  - Set $W(s) = 0$



$$T(s) = \frac{Y(s)}{R(s)} = \frac{D(s)G(s)}{1 + D(s)G(s)}$$

Transfer Function – Reference

- Next, find transfer function from $W(s)$ to $Y(s)$
  - Set $R(s) = 0$
  - System now becomes:



$$T_w(s) = \frac{Y(s)}{W(s)} = \frac{G_w(s)G(s)}{1 + D(s)G(s)}$$

Multiple Input Systems

- Two inputs, two transfer functions

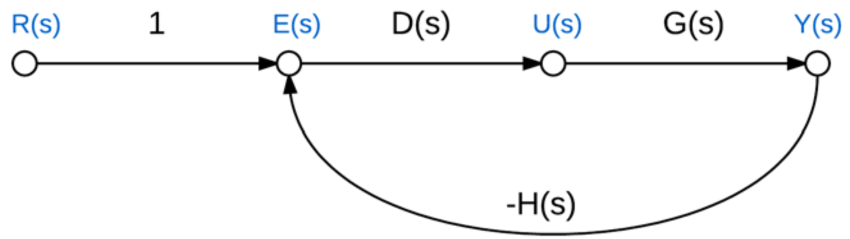
$$T(s) = \frac{D(s)G(s)}{1+D(s)G(s)} \quad \text{and} \quad T_w(s) = \frac{G_w(s)G(s)}{1+D(s)G(s)}$$

- $D(s)$ is the controller transfer function
  - Ultimately, we'll determine this
  - We have control over both $T(s)$ and $T_w(s)$
- What do we want these to be?
  - Design $T(s)$ for desired performance
  - Design $T_w(s)$ for disturbance rejection

Signal Flow Graphs

Signal Flow Graphs

- An alternative to block diagrams for graphically describing systems



- Signal flow graphs consist of:
  - ***Nodes*** –represent signals
  - ***Branches*** –represent system blocks
- Branches labeled with system transfer functions
- Nodes (sometimes) labeled with signal names
- Arrows indicate signal flow direction
- Implicit summation at nodes
  - Always a positive sum
  - Negative signs associated with branch transfer functions

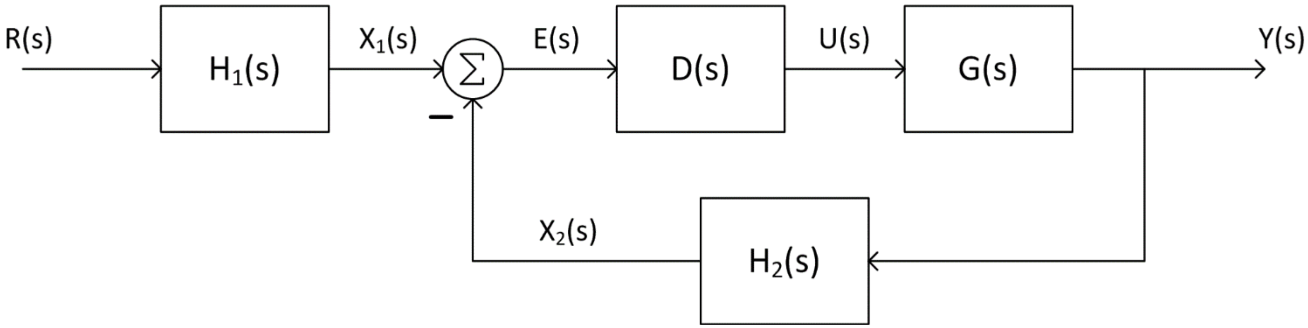
Block Diagram → Signal Flow Graph

- To convert from a block diagram to a signal flow graph:

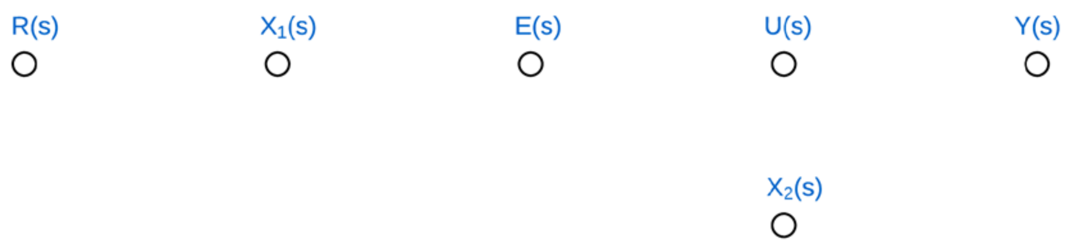
  1. Identify and label all signals on the block diagram
  2. Place a node for each signal
  3. Connect nodes with branches in place of the blocks
     - Maintain correct direction
     - Label branches with corresponding transfer functions
     - Negate transfer functions as necessary to provide negative feedback
  4. If desired, simplify where possible

Signal Flow Graph – Example 1

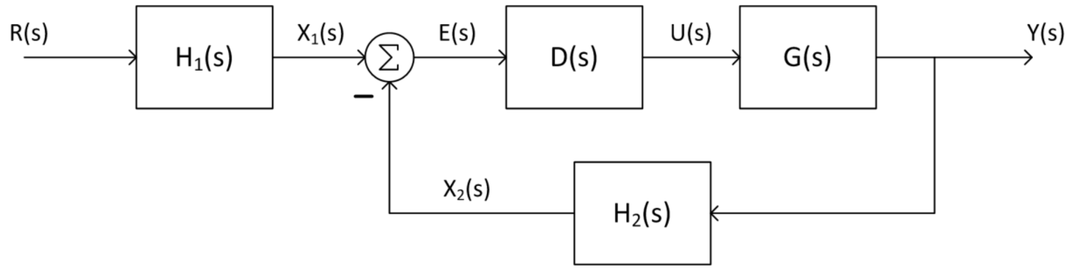
- Convert to a signal flow graph



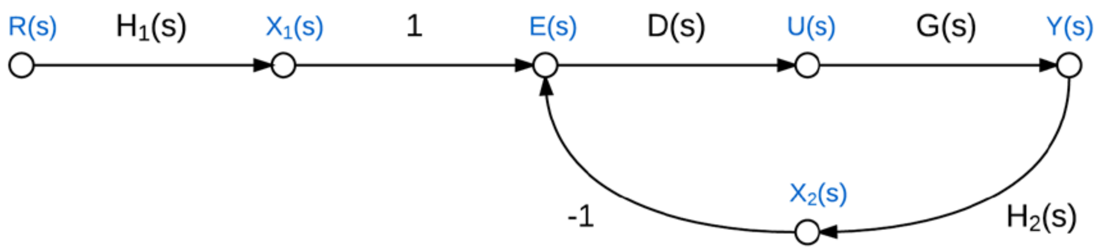
- Label any unlabeled signals
- Place a node for each signal



Signal Flow Graph – Example 1

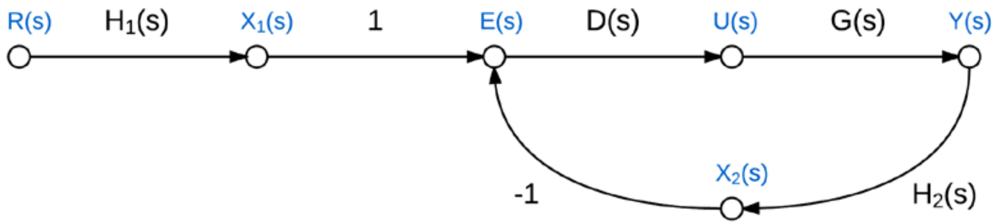


- Connect nodes with branches, each representing a system block

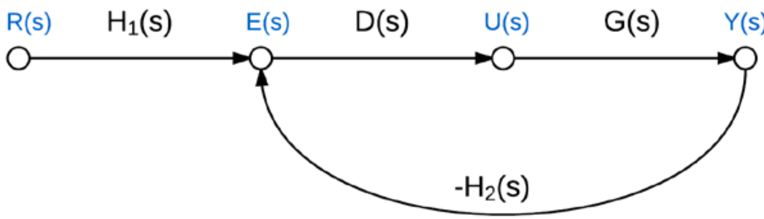


- Note the -1 to provide negative feedback of $X_2(s)$

Signal Flow Graph – Example 1

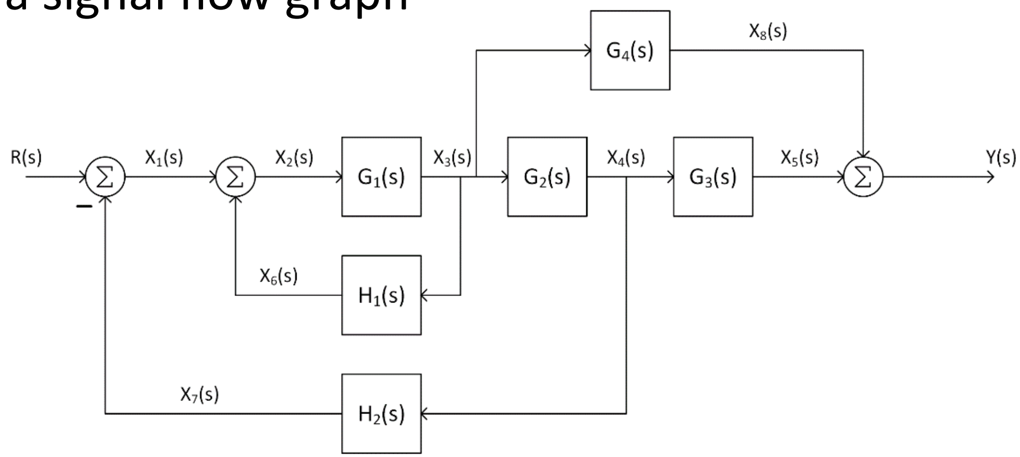


- Nodes with a single input and single output can be eliminated, if desired
  - This makes sense for $X_1(s)$ and $X_2(s)$
  - Leave $U(s)$ to indicate separation between controller and plant

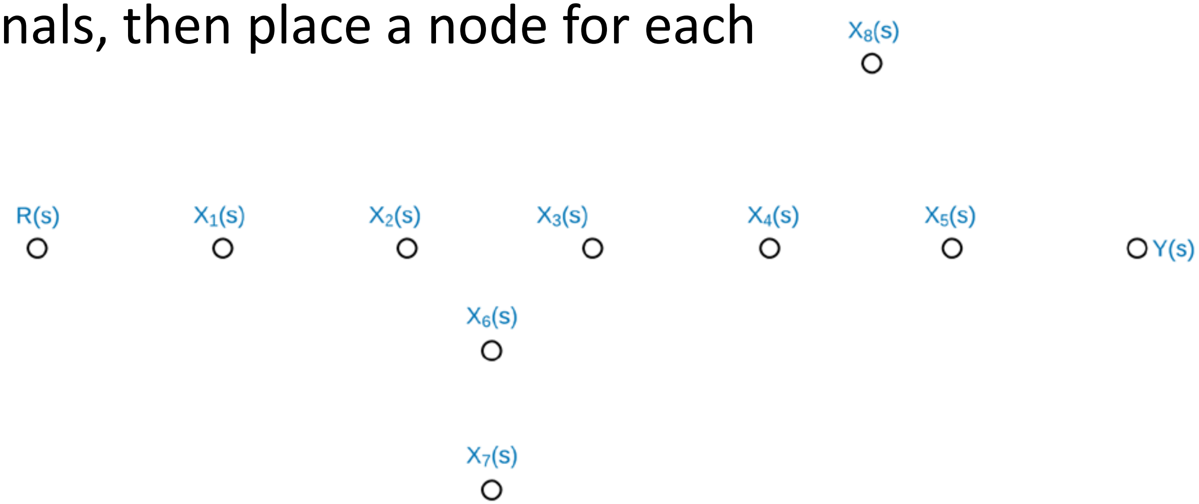


Signal Flow Graph – Example 2

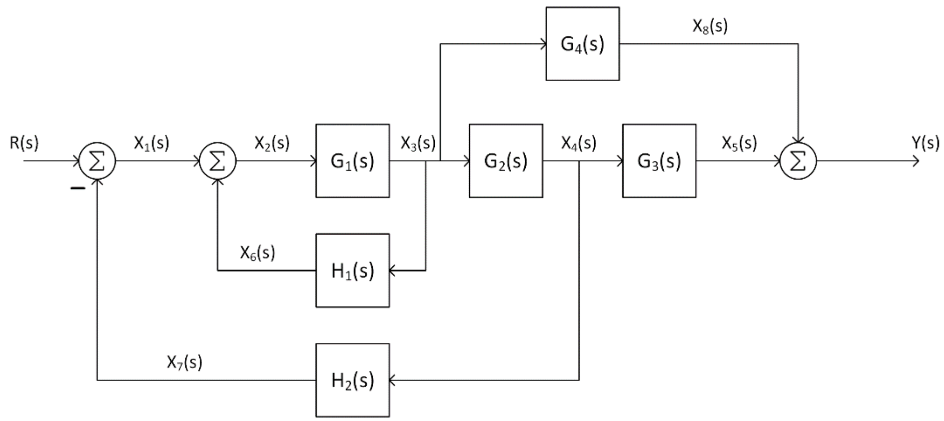
- Revisit the block diagram from earlier
  - Convert to a signal flow graph



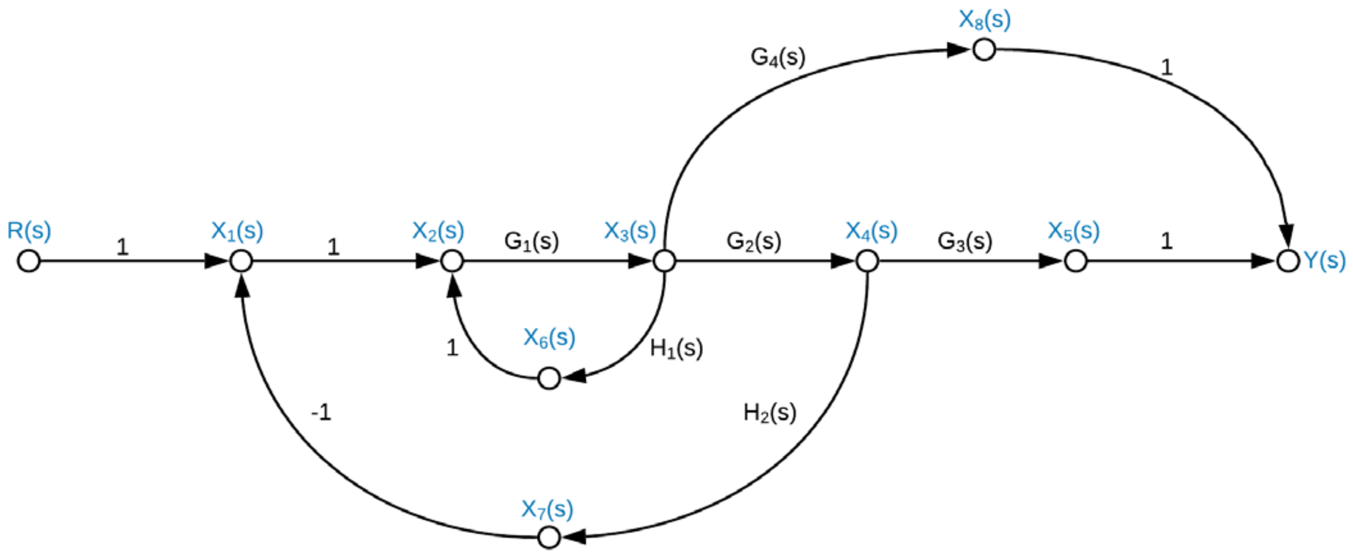
- Label all signals, then place a node for each



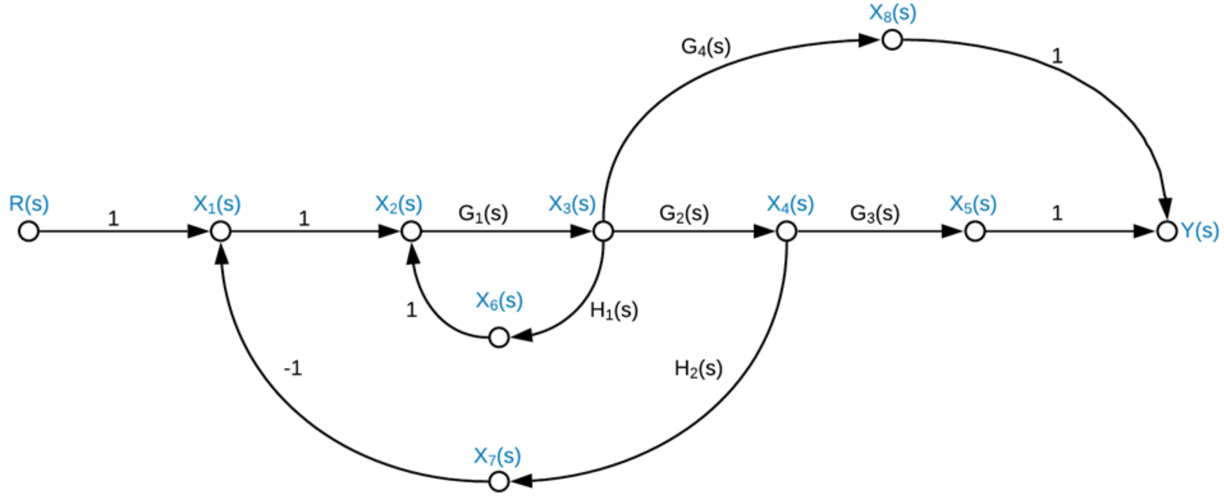
Signal Flow Graph – Example 2



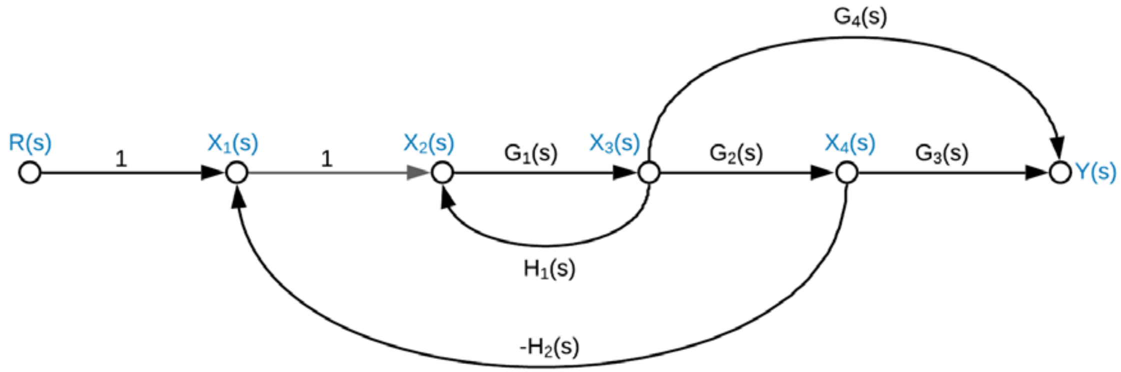
- Connect nodes with branches



Signal Flow Graph – Example 2



- Simplify – eliminate $X_5(s)$, $X_6(s)$, $X_7(s)$, and $X_8(s)$



Signal Flow Graphs vs. Block Diagrams

- Signal flow graphs and block diagrams are ***alternative***, though ***equivalent***, tools for graphical representation of interconnected systems

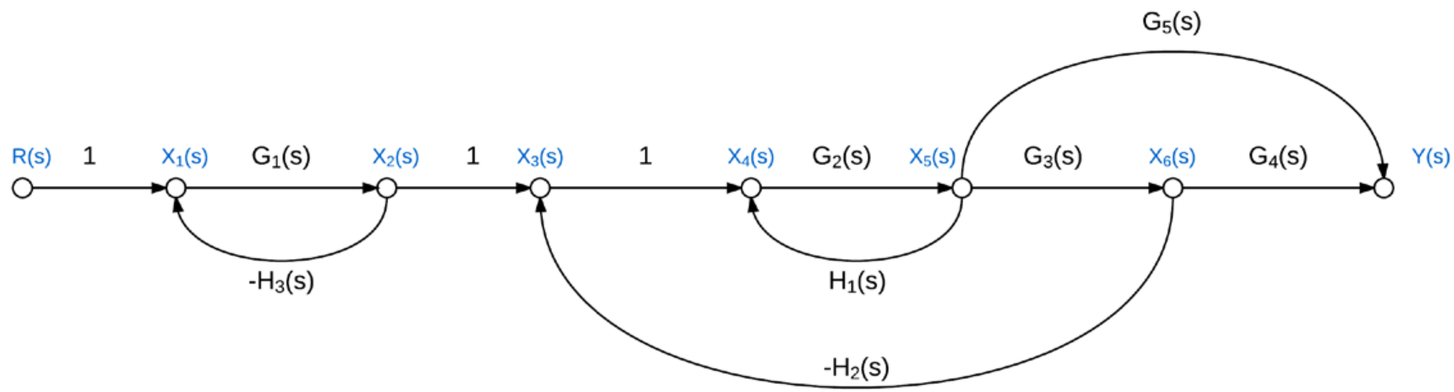
- A generalization (not a rule)
  - ***Signal flow graphs*** – more often used when dealing with ***state-space*** system models
  - ***Block diagrams*** – more often used when dealing with ***transfer function*** system models

Mason's Rule

Mason's Rule

- We've seen how to reduce a complicated block diagram to a single input-to-output transfer function
  - Many successive simplifications
- ***Mason's rule*** provides a formula to calculate the same overall transfer function
  - Single application of the formula
  - Can get complicated
- Before presenting the Mason's rule formula, we need to define some terminology

Loop Gain

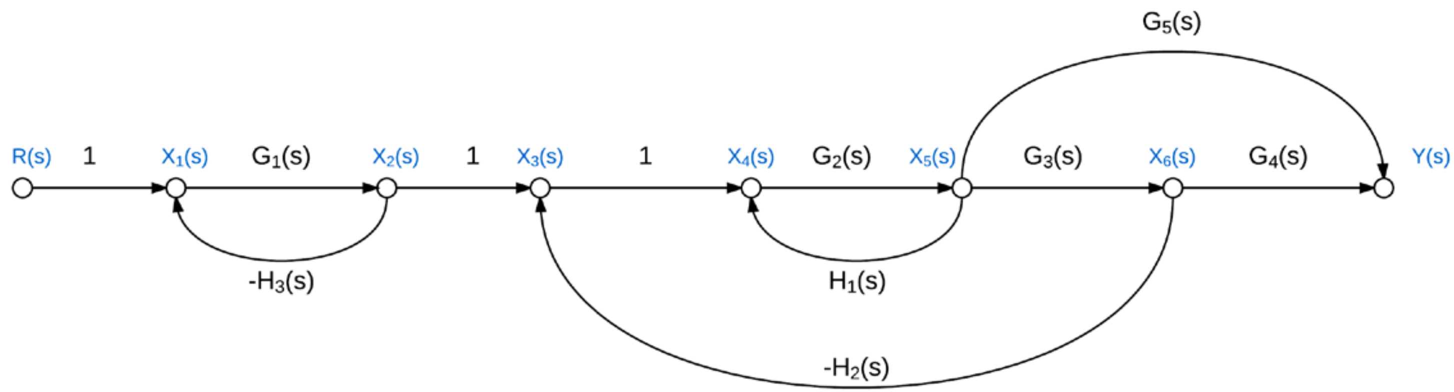


- ***Loop gain*** – total gain (product of individual gains) around any path in the signal flow graph
  - Beginning and ending at the same node
  - Not passing through any node more than once

- Here, there are three loops with the following gains:

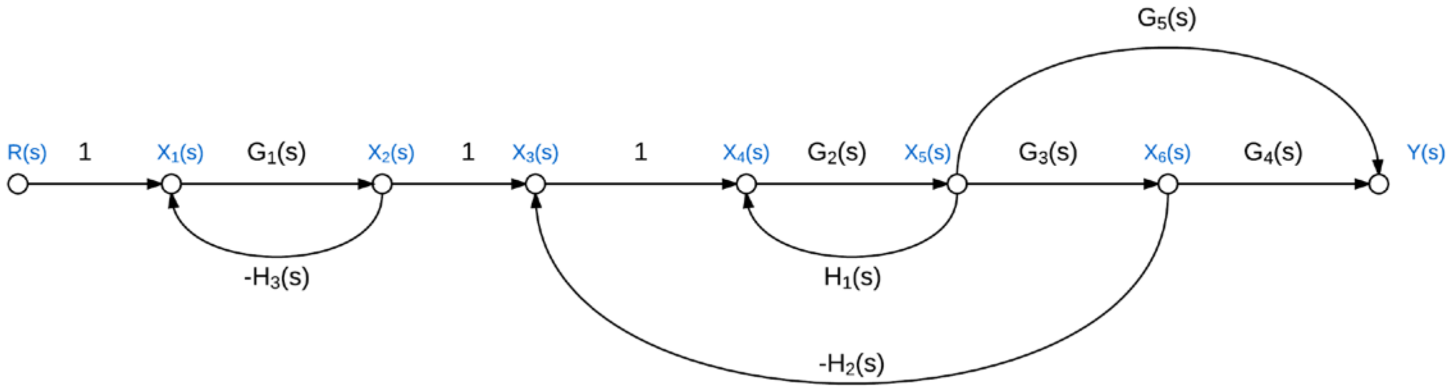
1. $-G_1H_3$
2. G_2H_1
3. $-G_2G_3H_2$

Forward Path Gain



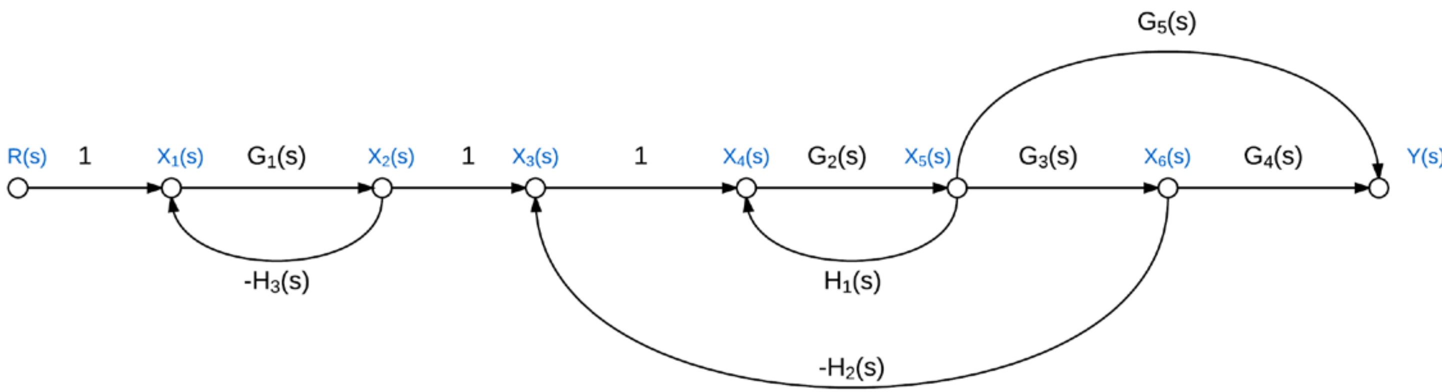
- ***Forward path gain*** – gain along any path from the input to the output
  - Not passing through any node more than once
- Here, there are two forward paths with the following gains:
  1. $G_1G_2G_3G_4$
  2. $G_1G_2G_5$

Non-Touching Loops



- ***Non-touching loops*** – loops that do not have any nodes in common
- Here,
  1. $-G_1H_3$ does not touch G_2H_1
  2. $-G_1H_3$ does not touch $-G_2G_3H_2$

Non-Touching Loop Gains



- ***Non-touching loop gains*** – the *product* of loop gains from non-touching loops, taken two, three, four, or more at a time
- Here, there are only two *pairs* of non-touching loops
  1. $[-G_1 H_3] \cdot [G_2 H_1]$
  2. $[-G_1 H_3] \cdot [-G_2 G_3 H_2]$

Mason's Rule

$$T(s) = \frac{Y(s)}{R(s)} = \frac{1}{\Delta}\sum_{k=1}^{P} T_k \Delta_k$$

where

P = # of forward paths

T_k = gain of the k^{th} forward path

$\Delta = 1 - \Sigma$(loop gains)

$+\Sigma$(non-touching loop gains taken two-at-a-time)

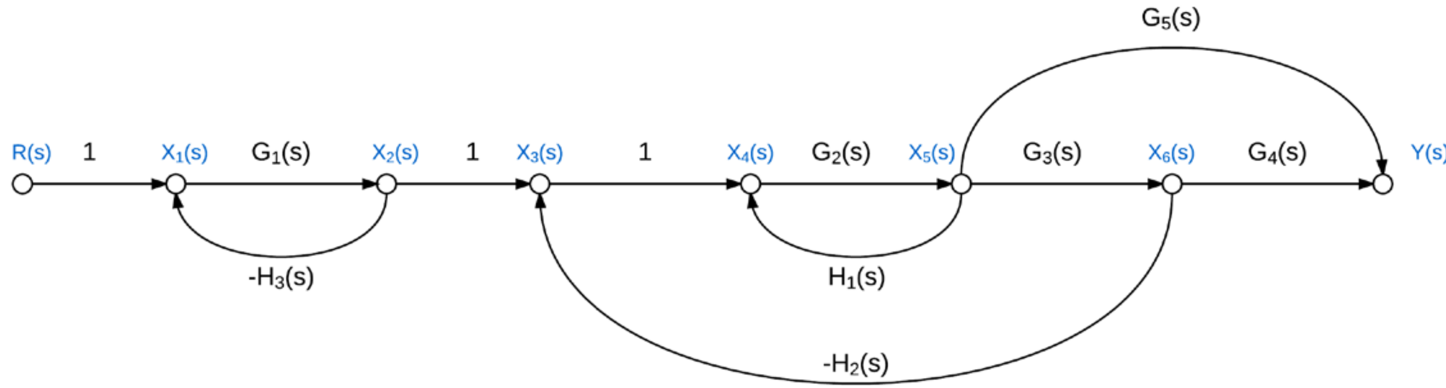
$-\Sigma$(non-touching loop gains taken three-at-a-time)

$+\Sigma$(non-touching loop gains taken four-at-a-time)

$-\Sigma \ldots$

$\Delta_k = \Delta - \Sigma$(loop gain terms in Δ that touch the k^{th} forward path)

Mason's Rule - Example



- # of forward paths:

  $P = 2$

- Forward path gains:

  $T_1 = G_1 G_2 G_3 G_4$

  $T_2 = G_1 G_2 G_5$

- Σ(loop gains):

  $-G_1 H_3 + G_2 H_1 - G_2 G_3 H_2$

- Σ(NTLGs taken two-at-a-time):

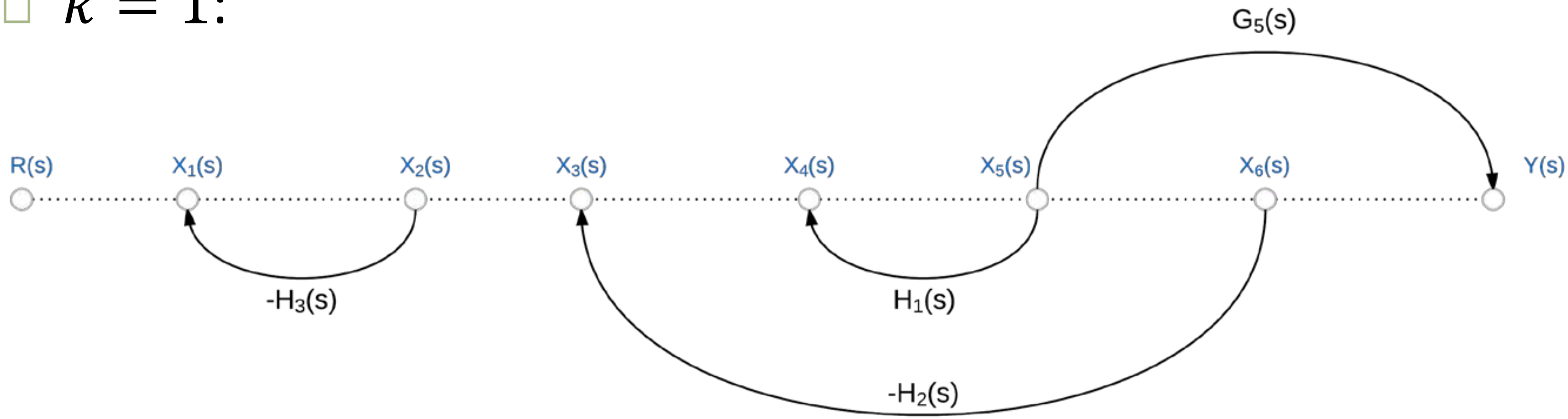
  $(-G_1 H_3 G_2 H_1) + (G_1 H_3 G_2 G_3 H_2)$

- Δ:

$$\Delta = 1 - \left(-G_1 H_3 + G_2 H_1 - G_2 G_3 H_2\right) + \left(-G_1 H_3 G_2 H_1 + G_1 H_3 G_2 G_3 H_2\right)$$

Mason's Rule – Example - Δ_k

- Simplest way to find Δ_k terms is to calculate Δ with the k^{th} path removed – must remove *nodes* as well
- $k = 1$:

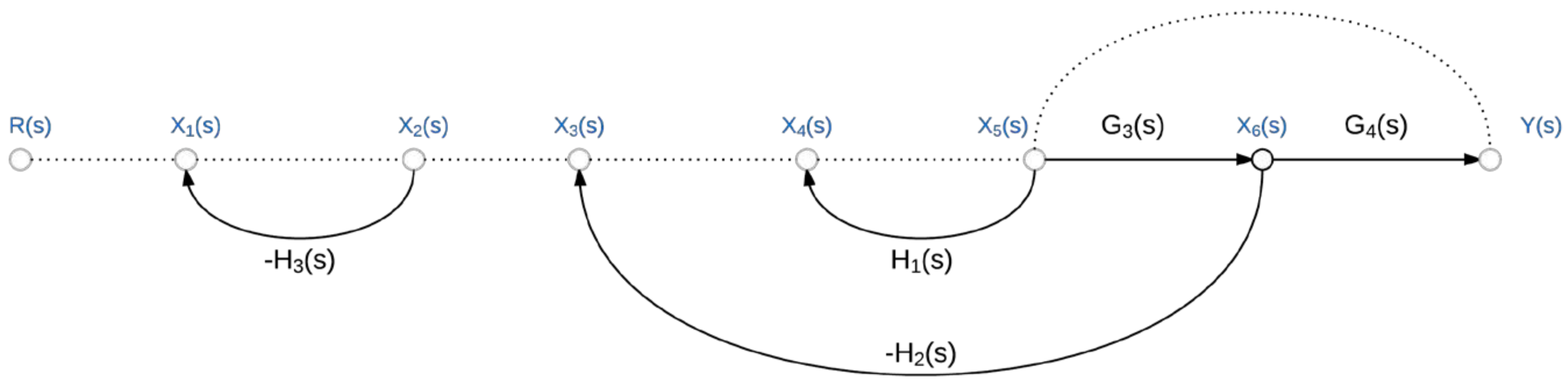


- With forward path 1 removed, there are no loops, so

$$\Delta_1 = 1 - 0$$
$$\Delta_1 = 1$$

Mason's Rule – Example - Δ_k

- $k = 2$:



- Similarly, removing forward path 2 leaves no loops, so

$$\Delta_2 = 1 - 0$$
$$\Delta_2 = 1$$

Mason's Rule - Example

- For our example:

$$P = 2$$

$$T_1 = G_1 G_2 G_3 G_4$$

$$T_2 = G_1 G_2 G_5$$

$$\Delta = 1 + G_1 H_3 - G_2 H_1 + G_2 G_3 H_2 - G_1 H_3 G_2 H_1 + G_1 H_3 G_2 G_3 H_2$$

$$\Delta_1 = 1$$

$$\Delta_2 = 1$$

$$T(s) = \frac{Y(s)}{R(s)} = \frac{1}{\Delta} \sum_{k=1}^{P} T_k \Delta_k$$

- The closed-loop transfer function:

$$T(s) = \frac{T_1 \Delta_1 + T_2 \Delta_2}{\Delta}$$

$$T(s) = \frac{G_1 G_2 G_3 G_4 + G_1 G_2 G_5}{1 + G_1 H_3 - G_2 H_1 + G_2 G_3 H_2 - G_1 H_3 G_2 H_1 + G_1 H_3 G_2 G_3 H_2}$$

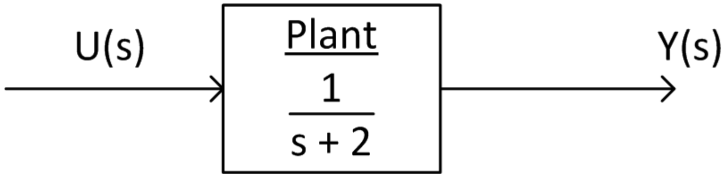
Preview of Controller Design

Controller Design – Preview

- We now have the tools necessary to determine the transfer function of closed-loop feedback systems
- Let's take a closer look at how feedback can help us achieve a desired response
  - Just a preview – this is the objective of the whole course

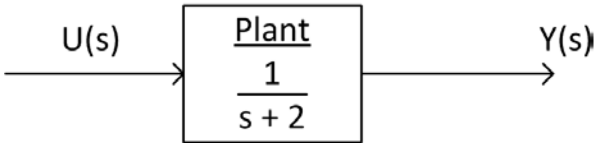
---

- Consider a simple first-order system

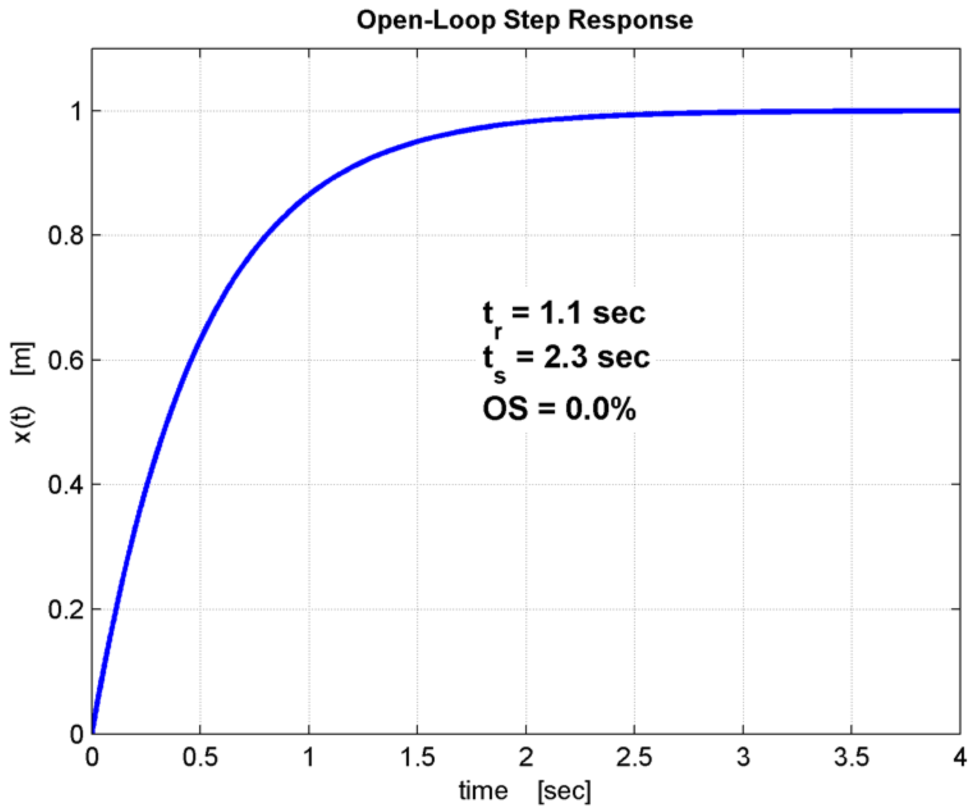


- A single real pole at $s = -2\,\frac{rad}{sec}$

Open-Loop Step Response

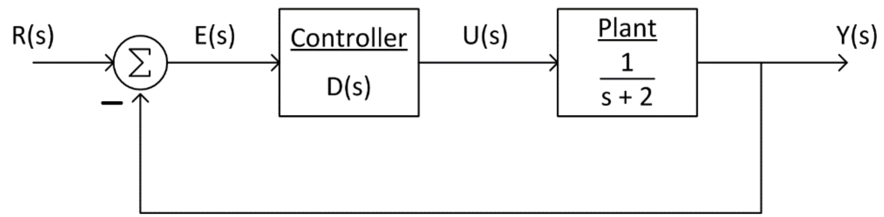


- This system exhibits the expected first-order step response
  - No overshoot or ringing



Add Feedback

- Now let's enclose the system in a feedback loop



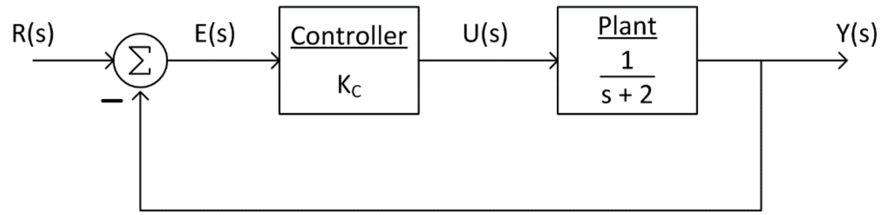
- Add controller block with transfer function $D(s)$
- Closed-loop transfer function becomes:

$$T(s) = \frac{D(s)\frac{1}{s+2}}{1 + D(s)\frac{1}{s+2}} = \frac{D(s)}{s + 2 + D(s)}$$

- Clearly the addition of feedback and the controller changes the transfer function – but how?
  - Let's consider a couple of example cases for $D(s)$

Add Feedback

- First, consider a simple gain block for the controller

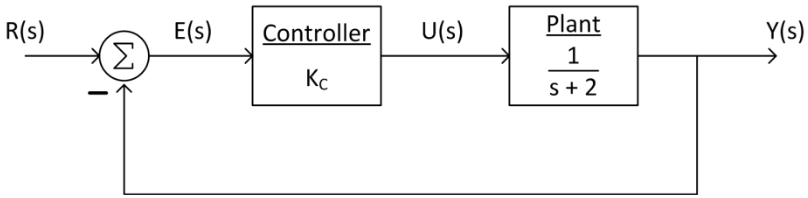


- Error signal, $E(s)$, amplified by a constant gain, K_C
  - A proportional controller, with gain K_C
- Now, the closed-loop transfer function is:

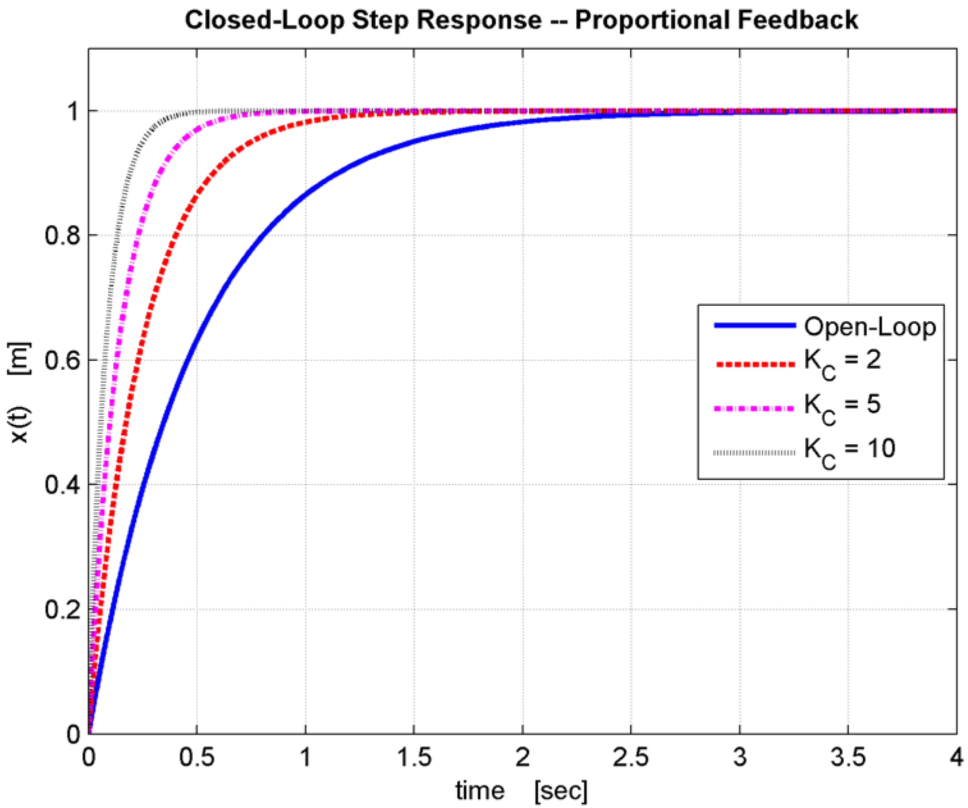
$$T(s) = \frac{\frac{K_C}{s+2}}{1 + \frac{K_C}{s+2}} = \frac{K_C}{s + 2 + K_C}$$

- A single real pole at $s = -(2 + K_C)$
  - Pole moved to a higher frequency
  - A faster response

Open-Loop Step Response

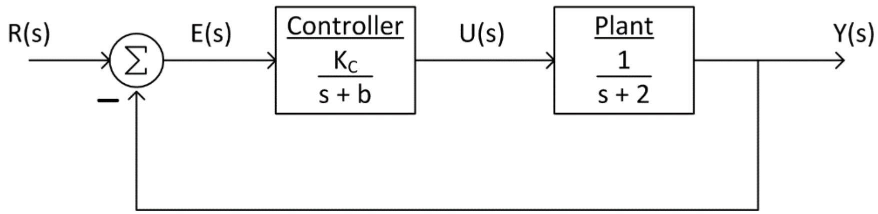


- As feedback gain increases:
  - Pole moves to a higher frequency
  - Response gets faster



First-Order Controller

- Next, allow the controller to have some dynamics of its own

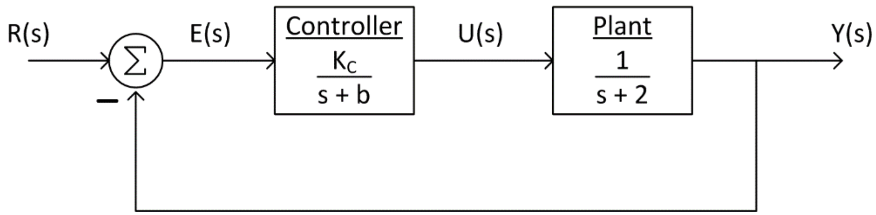


- Now the controller is a first-order block with gain K_C and a pole at $s = -b$
- This yields the following closed-loop transfer function:

$$T(s) = \frac{\frac{K_C}{(s+b)}\frac{1}{(s+2)}}{1 + \frac{K_C}{(s+b)}\frac{1}{(s+2)}} = \frac{K_C}{s^2 + (2+b)s + 2b + K_C}$$

- The closed-loop system is now ***second-order***
  - One pole from the plant
  - One pole from the controller

First-Order Controller



$$T(s) = \frac{K_C}{s^2 + (2+b)s + 2b + K_C}$$

- Two closed-loop poles:

$$s_{1,2} = -\frac{(b+2)}{2} \pm \frac{\sqrt{b^2 - 4b + 4 - 4K_C}}{2}$$

- Pole locations determined by b and K_C
  - Controller parameters – we have control over these
  - Design the controller to place the poles where we want them
- So, where do we want them?
  - Design to performance specifications
  - Risetime, overshoot, settling time, etc.

Design to Specifications

- The second-order closed-loop transfer function

$$T(s) = \frac{K_C}{s^2 + (2+b)s + 2b + K_C}$$

can be expressed as

$$T(s) = \frac{K_C}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{K_C}{s^2 + 2\sigma s + \omega_n^2}$$

- Let's say we want a closed-loop response that satisfies the following specifications:
  - $\%OS \leq 5\%$
  - $t_s \leq 600\ msec$
- Use $\%OS$ and t_s specs to determine values of ζ and σ
  - Then use ζ and σ to determine K_C and b

Determine ζ from Specifications

- Overshoot and damping ratio, ζ, are related as follows:

$$\zeta = \frac{-\ln(OS)}{\sqrt{\pi^2 + \ln^2(OS)}}$$

- The requirement is $\%OS \leq 5\%$, so

$$\zeta \geq \frac{-\ln(0.05)}{\sqrt{\pi^2 + \ln^2(0.05)}} = 0.69$$

- Allowing some margin, set $\zeta = 0.75$

Determine σ from Specifications

- Settling time ($\pm 1\%$) can be approximated from σ as

$$t_s \approx \frac{4.6}{\sigma}$$

- The requirement is $t_s \leq 600\,msec$
- Allowing for some margin, design for $t_s = 500\,msec$

$$t_s \approx \frac{4.6}{\sigma} = 500\,msec \quad \rightarrow \quad \sigma = \frac{4.6}{500\,msec}$$

which gives

$$\sigma = 9.2\frac{rad}{sec}$$

- We can then calculate the natural frequency from ζ and σ

$$\omega_n = \frac{\sigma}{\zeta} = \frac{9.2}{0.75} = 12.27\frac{rad}{sec}$$

Determine Controller Parameters from σ and ω_n

- The characteristic polynomial is

$$s^2 + (2+b)s + 2b + K_C = s^2 + 2\sigma s + \omega_n^2$$

- Equating coefficients to solve for b:

$$2 + b = 2\sigma = 18.4$$
$$b = 16.4$$

and K_c:

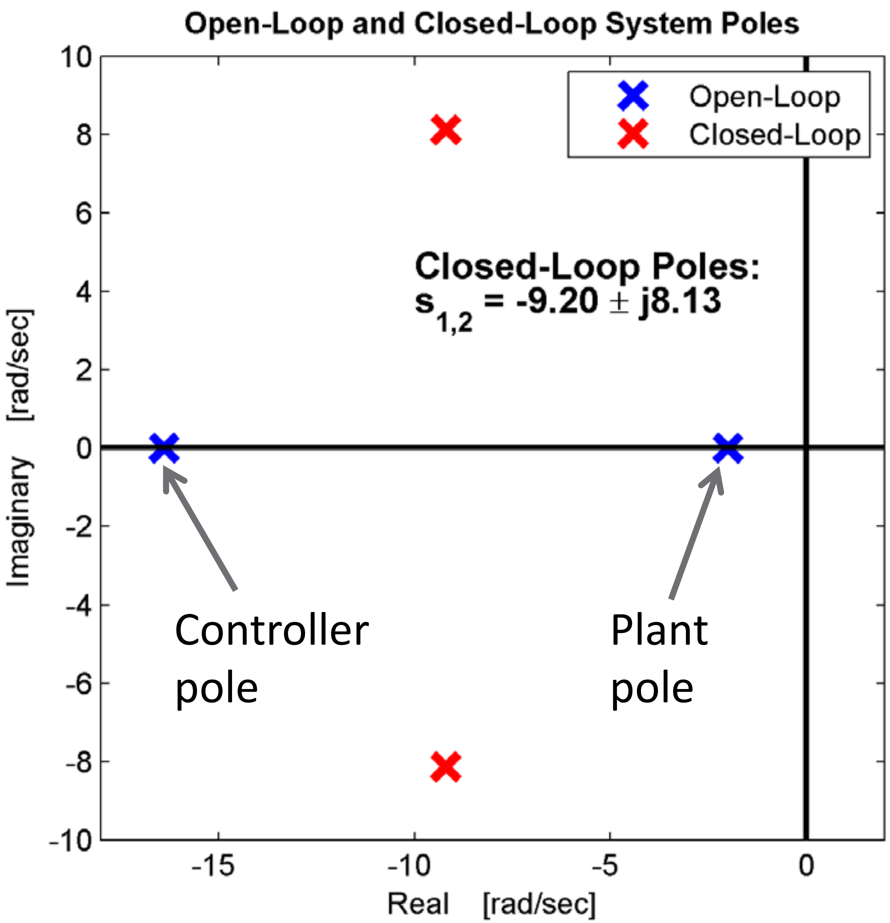
$$2b + K_C = \omega_n^2 = (12.27)^2 = 150.5$$
$$K_C = 150.5 - 2 \cdot 16.4 = 117.7 \quad \rightarrow \quad 118$$
$$K_c = 118$$

- The controller transfer function is

$$D(s) = \frac{118}{s + 16.4}$$

Closed-Loop Poles

- Closed-loop system is now second order
- Controller designed to place the two closed-loop poles at desirable locations:
  - $s_{1,2} = -9.2 \pm j8.13$
  - $\zeta = 0.75$
  - $\omega_n = 12.3$



Closed-Loop Step Response

- Closed-loop step response satisfies the specifications
- Approximations were used
  - If requirements not met - ***iterate***