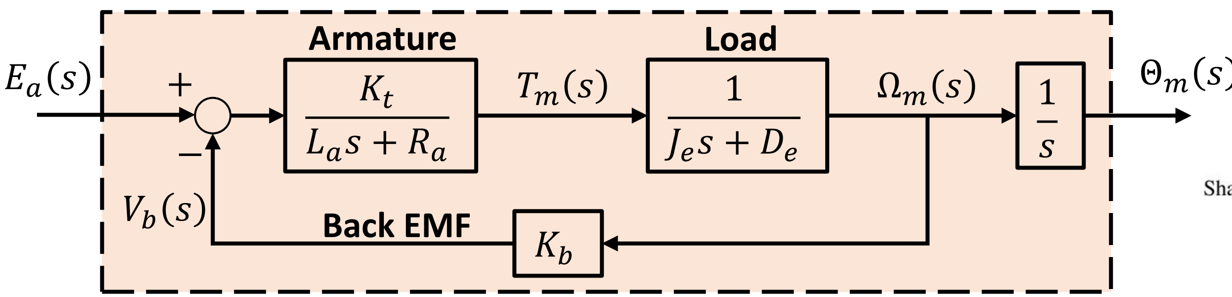
Ch5: Block Diagram Reduction

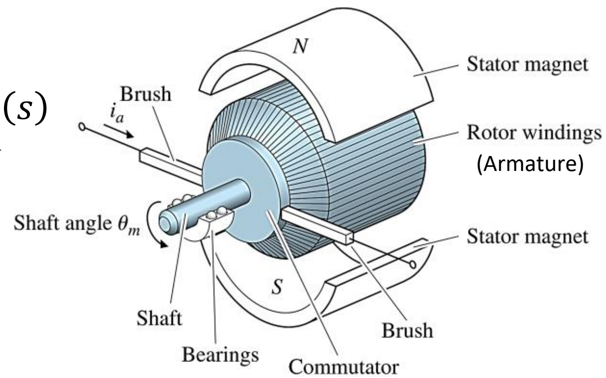
Contents:

Introduction

Introduction

- **Complicated systems** usually consist of the **interconnection** of **many subsystems**.
- Differing from a purely abstract mathematical representation of these systems, **Block Diagrams** and **Signal-Flow Graphs** can depict the **interrelationships** that exist among the various components and subsystems **more realistically** and it is possible to evaluate the contribution of each component to the overall performance of the system.



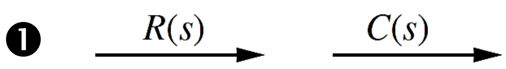


To analysis these systems, we represent them as **a single transfer function**. In this chapter, we will develop techniques to reduce each representation to a single transfer function.

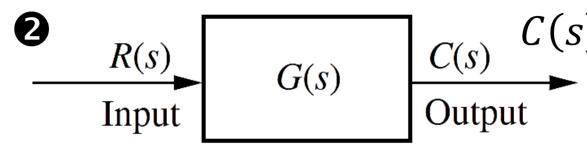
Block Diagrams

Block Diagram Components

Block is a symbol for the mathematical operation (transfer function) on the input signal that produces the output. Blocks are connected by arrows to indicate the direction of the flow of signals.

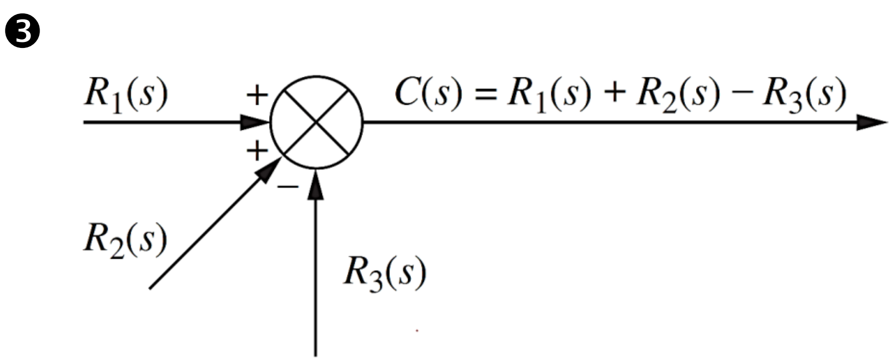


Signals



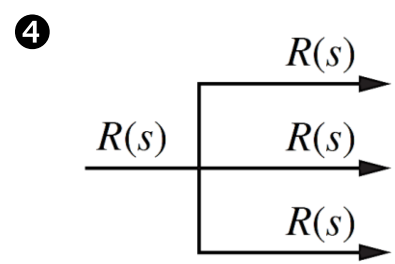
$$C(s) = R(s)G(s)$$

System



Summing Junction

(**Note**: The quantities being added or subtracted must have the **same** dimensions and units.)

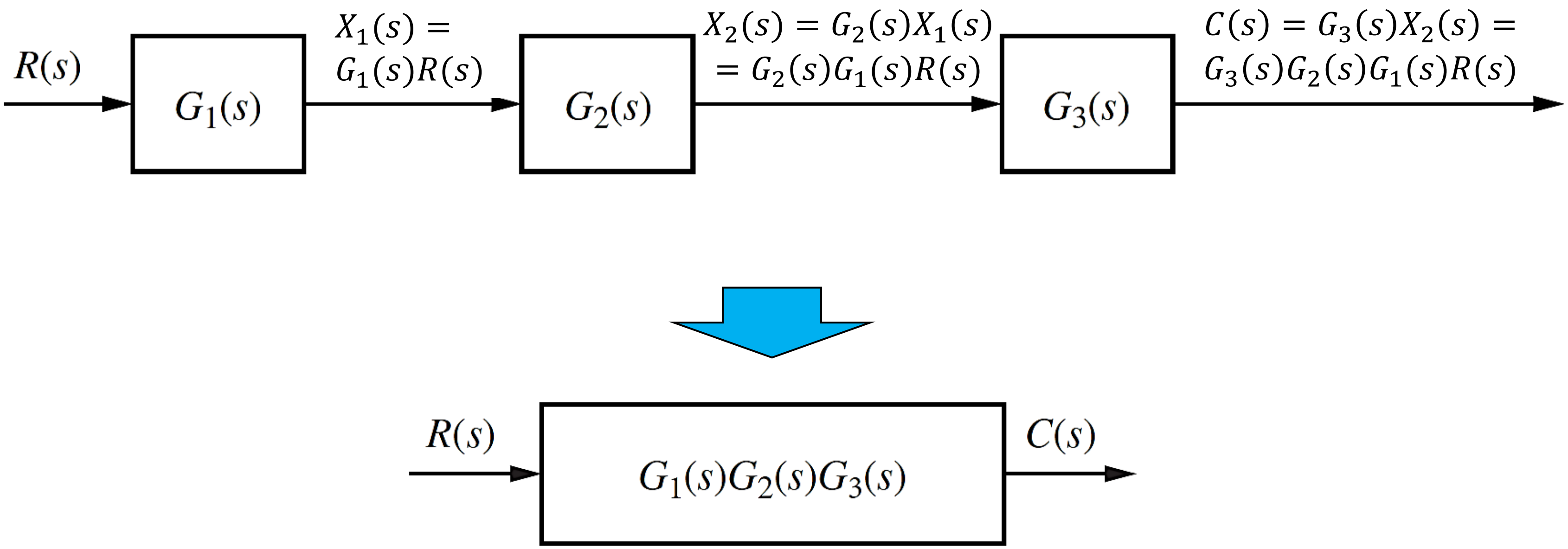


Pickoff Point or Branch Point

Common Configurations for Multiple Subsystems

(1) Cascade Form: The equivalent transfer function is the product of the subsystems' transfer functions.

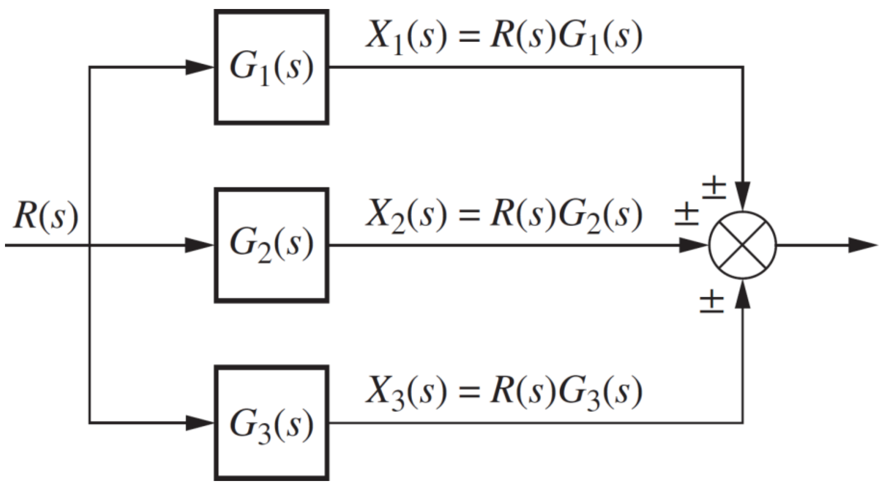
$$\boxed{G_e(s) = G_1(s)G_2(s)G_3(s)}$$



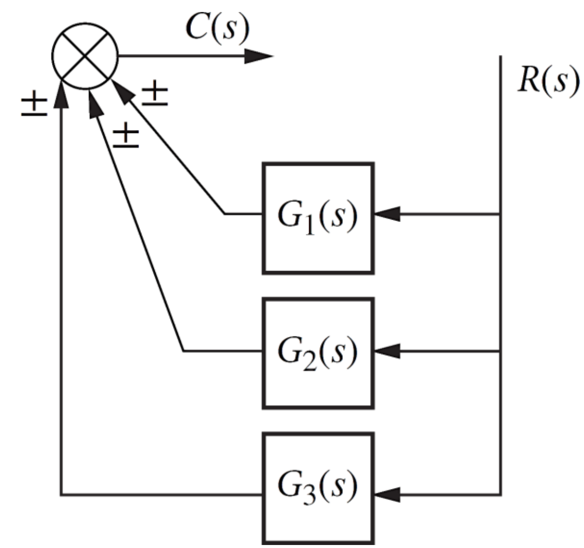
Common Configurations for Multiple Subsystems

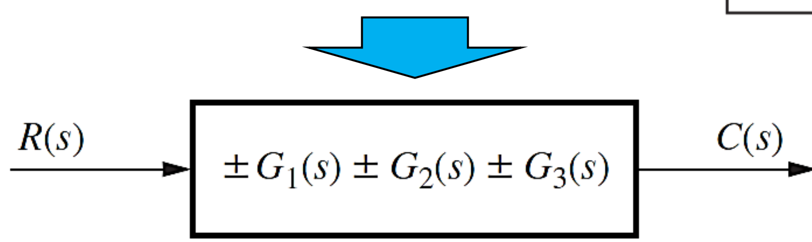
(2) Parallel Form: The equivalent transfer function is the algebraic sum of the subsystems' transfer functions.

$$\boxed{G_e(s) = \pm G_1(s) \pm G_2(s) \pm G_3(s)}$$



or





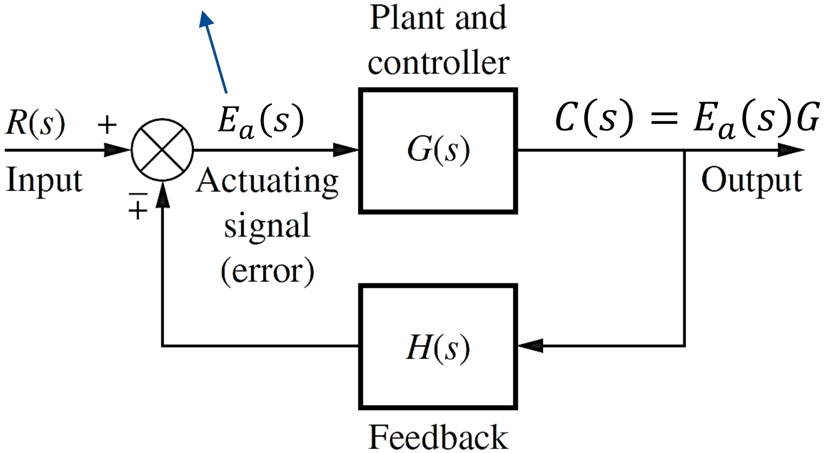
Common Configurations for Multiple Subsystems

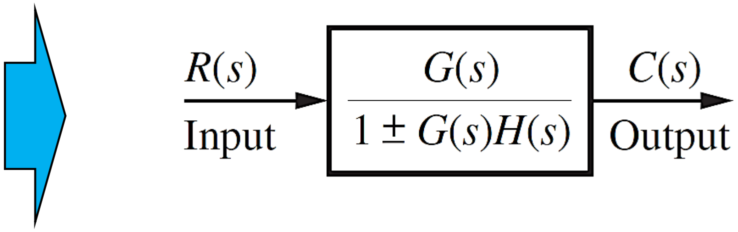
(3) Feedback Form: It is the basis for study of control systems engineering. By considering the simplified model of feedback control system, the equivalent, or **closed-loop**, transfer function is derived as:

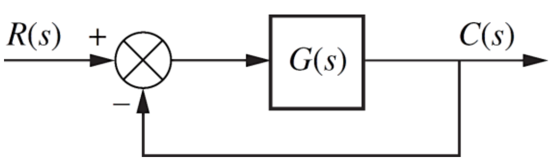
$$G_e(s) = \frac{G(s)}{1 \pm G(s)H(s)}$$

$$E_a(s) = R(s) \mp C(s)H(s)$$

$$C(s) = E_a(s)G(s)$$







Note: When $H(s) = 1$, the system is called **Unity Feedback**.

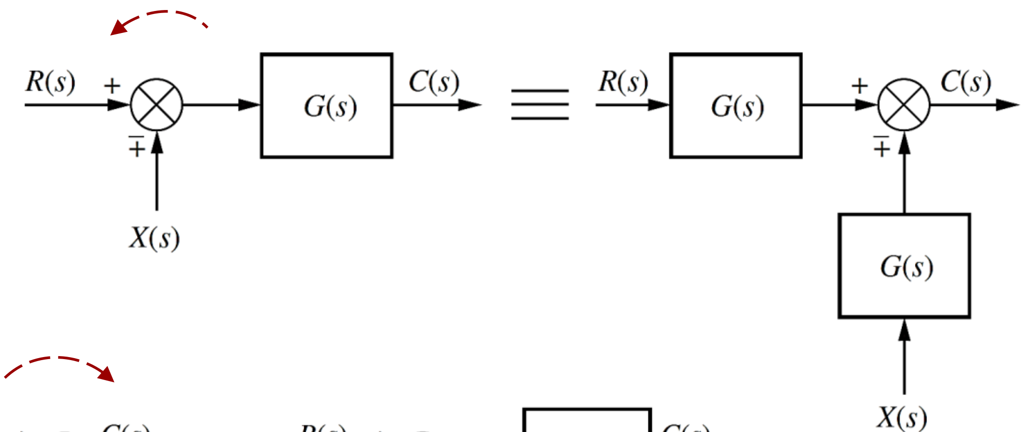
Note: The product, $G(s)H(s)$, is called the **open-loop transfer function**, or **loop gain**.

Moving Blocks to Create Familiar Forms

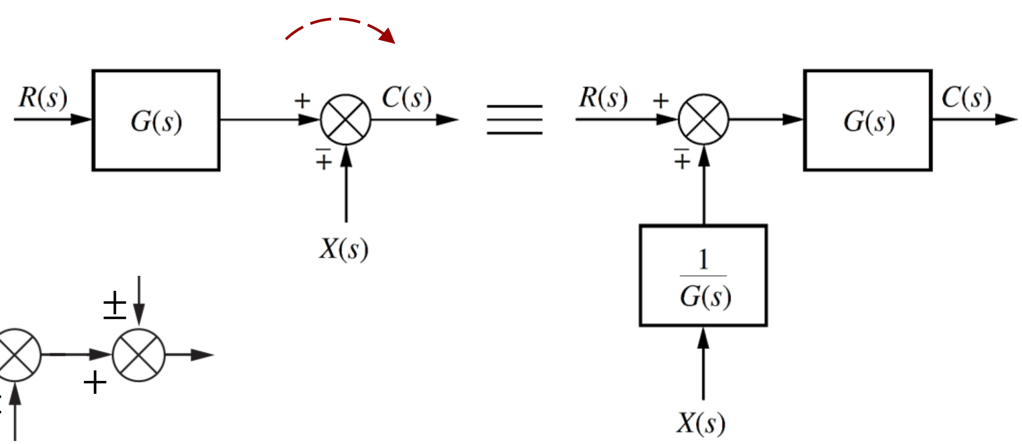
Familiar forms (cascade, parallel, and feedback) are not always apparent in a block diagram. Hence, blocks should be moved to the left and right of summing junctions and pickoff (branch) points to establish familiar forms and **reduce** a block diagram to a **single** transfer function.

(1) Block diagram algebra for summing junctions:

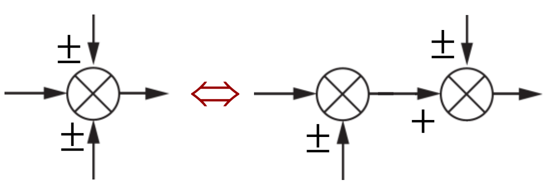
❶



❷



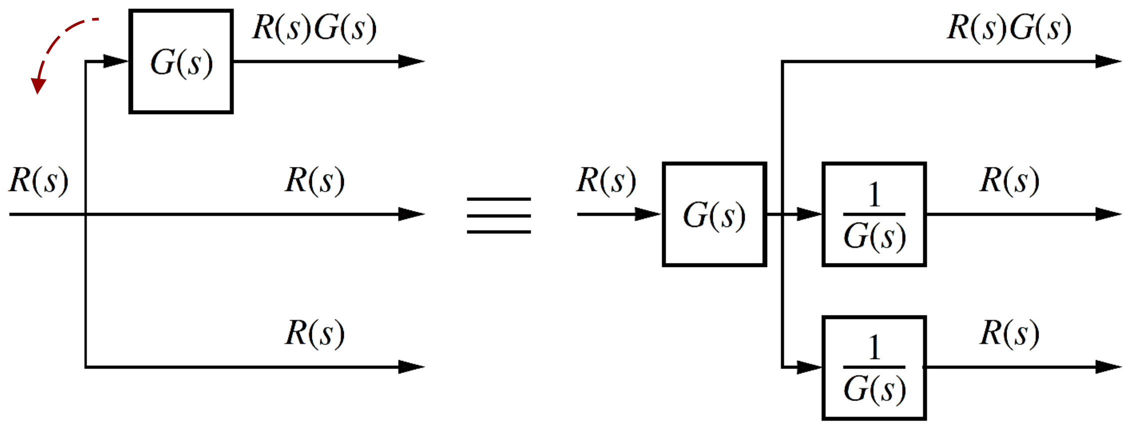
Note:



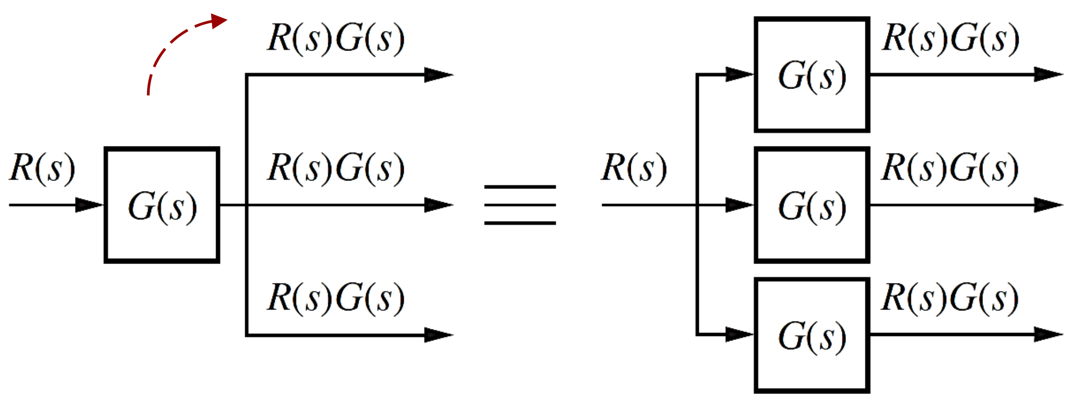
Moving Blocks to Create Familiar Forms

(2) Block diagram algebra for pickoff (branch) points:

❶

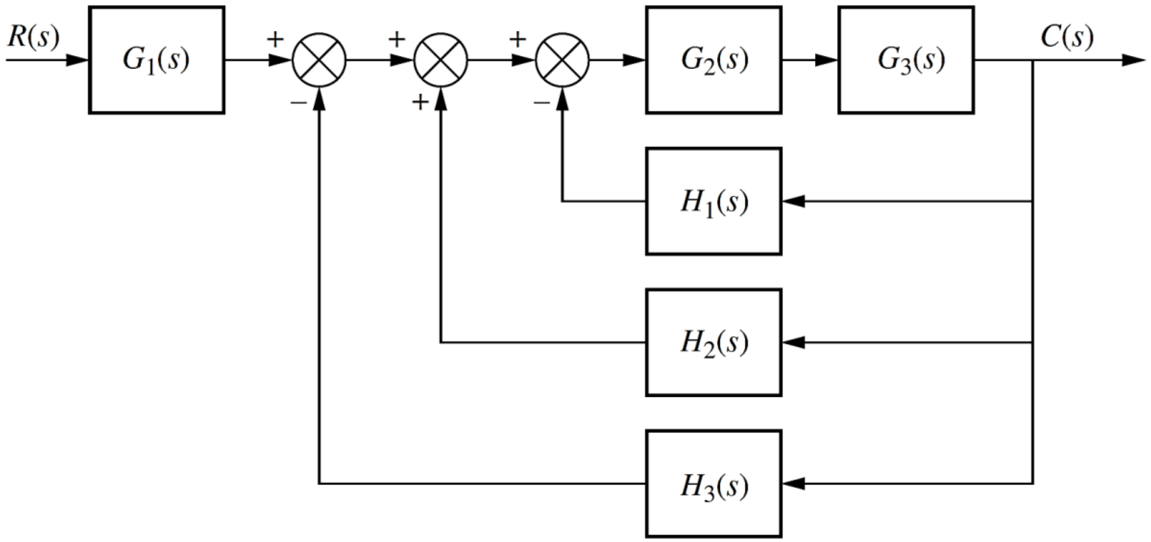


❷



Example

Reduce the block diagram to a single transfer function.

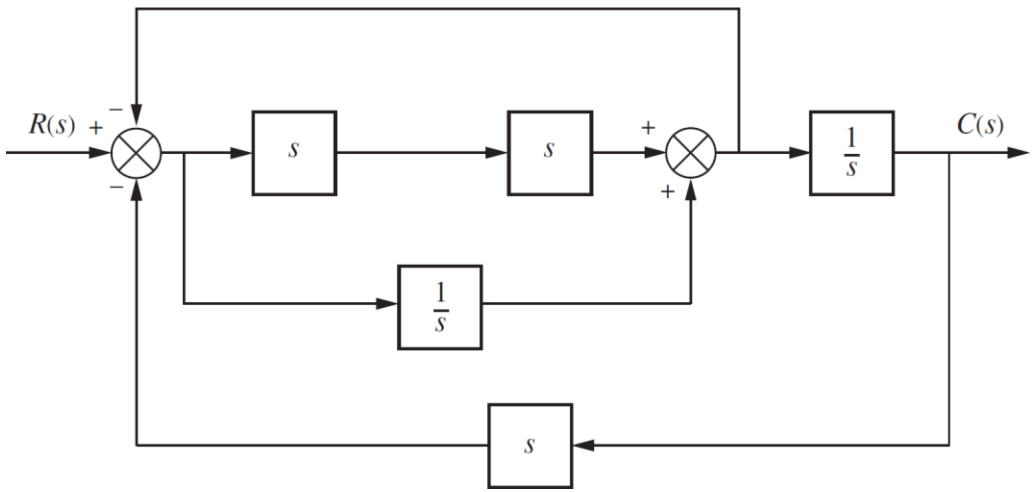


Answer:

$$G(s) = \frac{G_3(s)G_2(s)G_1(s)}{1 + G_3(s)G_2(s)[H_1(s) - H_2(s) + H_3(s)]}$$

Example

Find the equivalent transfer function, $T(s) = C(s)/R(s)$ for the system shown.

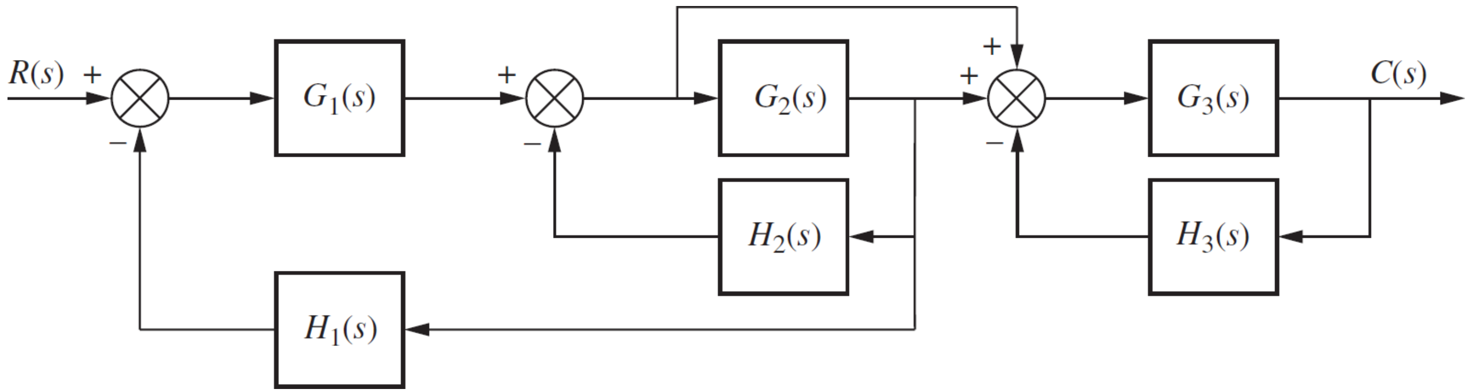


Answer:

$$T(s) = \frac{s^3 + 1}{2s^4 + s^2 + 2s}$$

Example

Reduce the block diagram to a single transfer function.

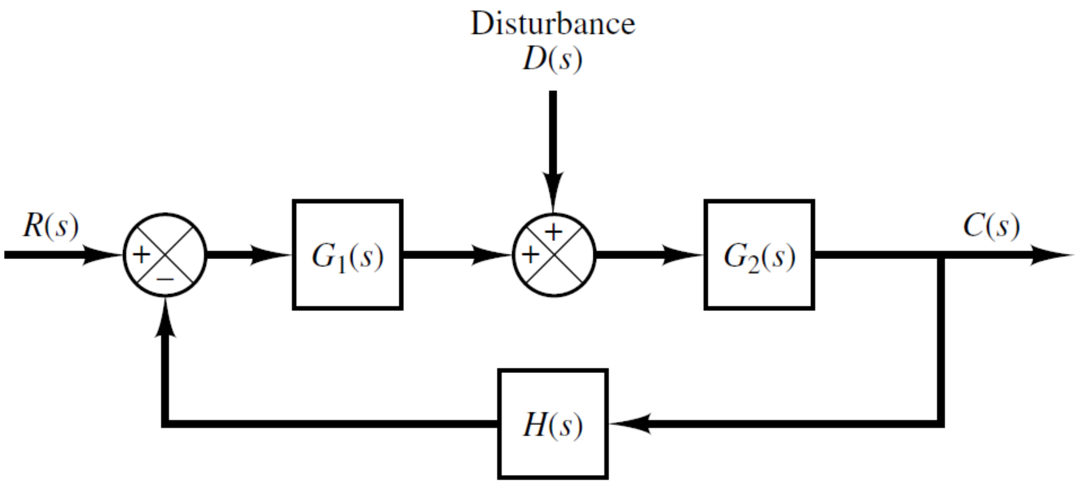


Answer:

$$G(s) = \frac{G_1(s)G_3(s)[1 + G_2(s)]}{[1 + G_2(s)H_2(s) + G_1(s)G_2(s)H_1(s)][1 + G_3(s)H_3(s)]}$$

Example

Reduce the block diagram to a single transfer function $C(s)/D(s)$ when $R(s) = 0$.



Answer:

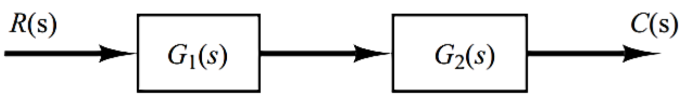
$$\frac{C(s)}{D(s)} = \frac{G_2(s)}{1 + G_1(s)G_2(s)H(s)}$$

Using MATLAB and Control System Toolbox

Block Diagram Reduction
Method 1: Using series, parallel, feedback

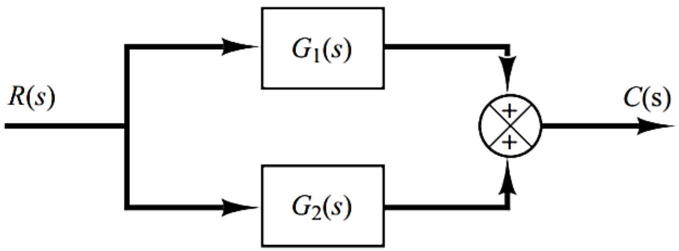
The closed-loop transfer function is obtained using the following commands successively.

(a)



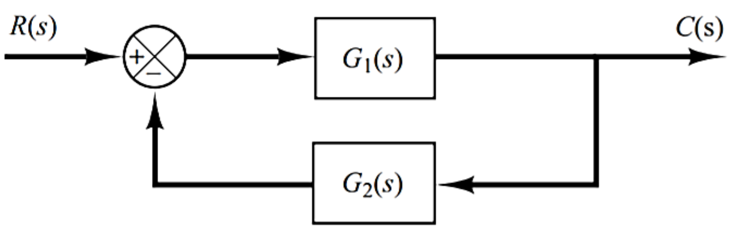

```
series(G1,G2)
```

(b)




```
parallel(G1,G2)
```

(c)

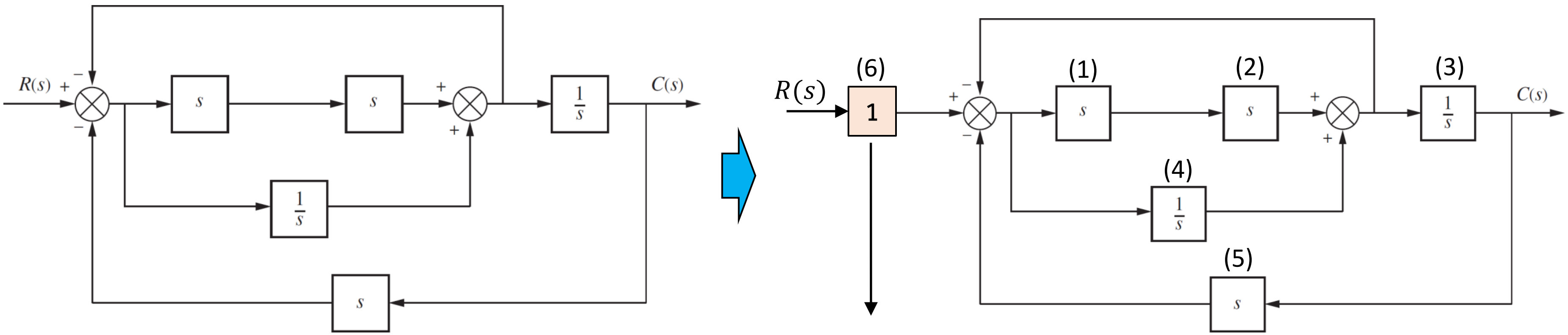



```
feedback(G1,G2,sign)
```

sign is –1 for negative-feedback systems or +1 for positive-feedback systems.

Block Diagram Reduction
Method 2: Using append, connect

This method, which defines the topology of the system, is used effectively for complicated systems. Let's see an example:




```
s = tf('s');
G1 = s;
G2 = s;
G3 = 1/s;
G4 = 1/s;
G5 = s;
G6 = 1;
system = append(G1,G2,G3,G4,G5,G6);
```

- If the input is not connected directly to a block, add an auxiliary block "1".
- Number all the blocks.
- Define the transfer functions.
- Append the blocks using the command G=append(G1,G2,G3,G4, ...Gn), where the Gi are the transfer functions of the blocks.
- The position of a block (transfer function) in the append argument is based on its number.

Block Diagram Reduction Using append, connect

```
input = 6;
output = 3;
Q= [1 -2 -4 -5 6
    2  1  0  0 0
    3  2  4  0 0
    4 -2 -4 -5 6
    5  3  0  0 0
    6  0  0  0 0];
T = connect(system,Q,input,output);
G = tf(T);
G = minreal(G)
```

These define the input to block 6 as the **external input** and the output of block 3 as the **external output**.

- To determine how all of the blocks are interconnected, a matrix that has a row for each block is formed.
- The first column contains the block's number.
- Subsequent columns contain the numbers of the blocks from which the inputs come (the order is immaterial and use negative sign for negative inputs).

minreal (G) is used to cancel possible common terms in the numerator and denominator.

$$G(s) = \frac{s^3 + 1}{2s^4 + s^2 + 2s}$$

Signal-Flow Graphs and Mason’s Rule

Signal-Flow Graph Components

Signal-Flow Graphs are an alternative to block diagrams, which are more compact. They consist of **branches** (**lines**), which represent **systems**, and **nodes**, which represent **signals**.

A system is represented by a line with an arrow showing the direction of signal flow through the system. Transfer function $G(s)$ is written adjacent to the line.

A signal is a node with the signal's name written adjacent to the node.

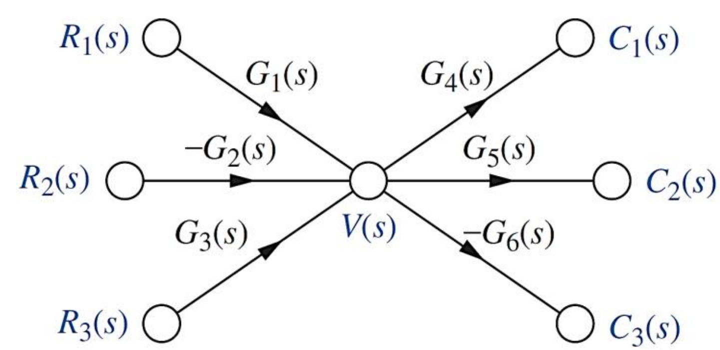
- Each signal is the sum of signals flowing into it:

$$V(s) = R_1(s)G_1(s) - R_2(s)G_2(s) + R_3(s)G_3(s)$$

$$C_1(s) = V(s)G_4(s)$$

$$C_2(s) = V(s)G_5(s)$$

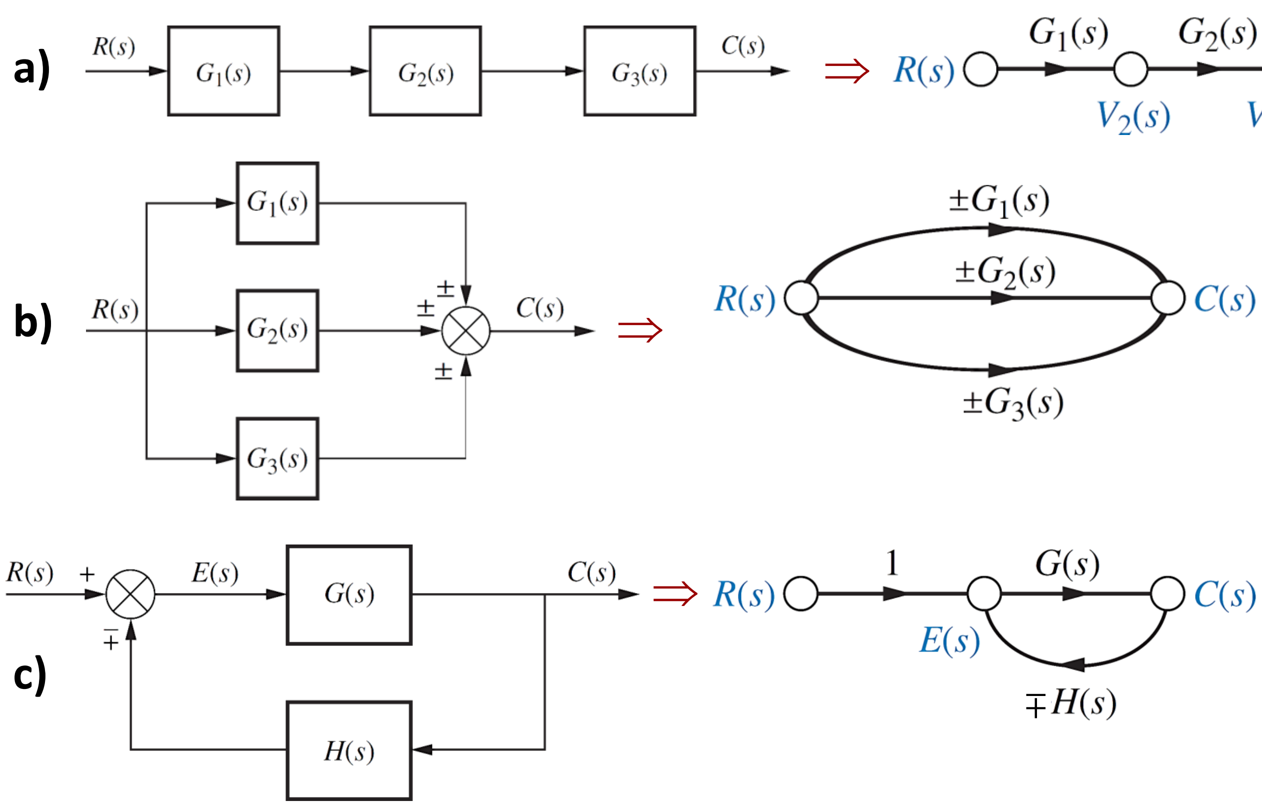
$$C_3(s) = -V(s)G_6(s)$$



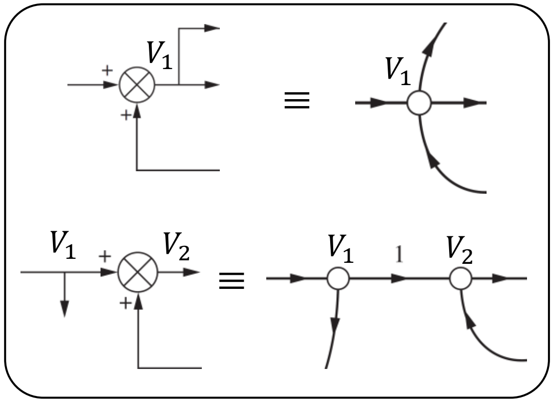
Drawing Signal-Flow Graphs

To convert the block diagrams into signal-flow graphs, start by drawing the signal nodes of the system. Next interconnect the signal nodes with system branches.

Signal-Flow Graphs the **cascaded**, **parallel**, and **feedback** forms:

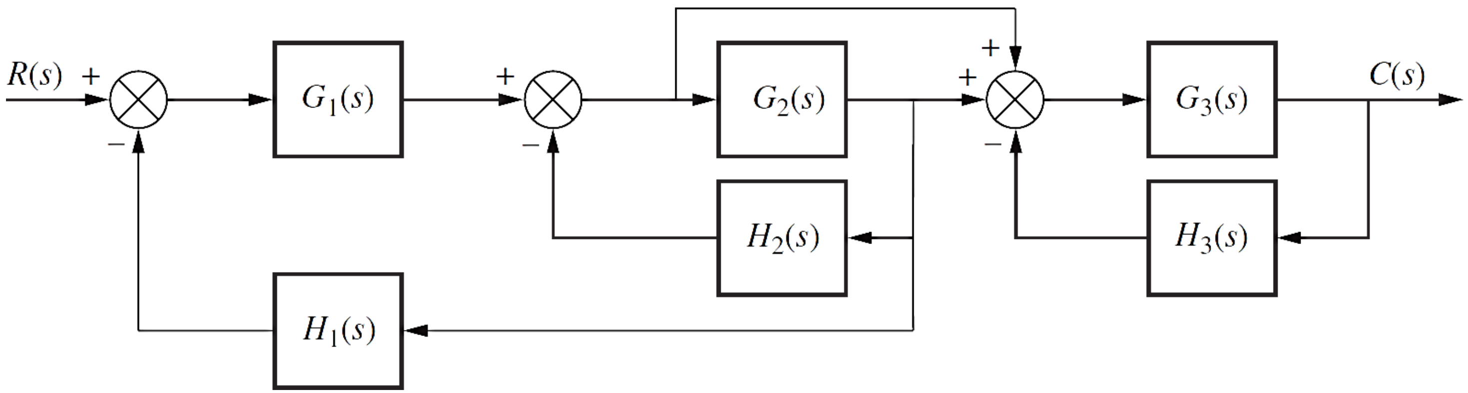


Note:



Example

Convert the block diagram to a signal-flow graph.



Mason's Rule: Definitions

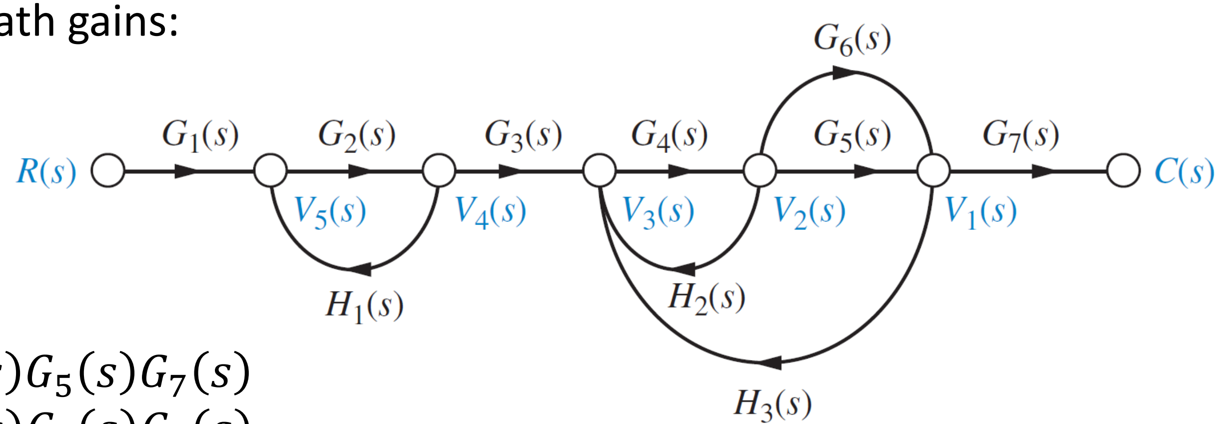
Mason's Rule is used for **reducing** a signal-flow graph to a single transfer function.

Some basic definitions for using Mason's Rule:

1. Forward-Path Gain:

The product of gains found by traversing a path from the **input** node to the **output** node of the signal-flow graph in the direction of signal flow, without repetition of any node.

Ex: There are two forward-path gains:



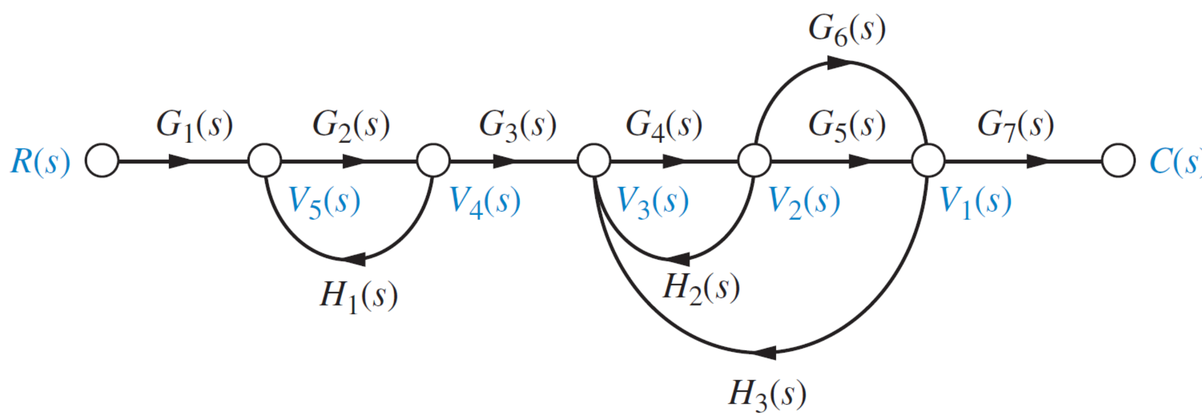
$$T_1 = G_1(s)G_2(s)G_3(s)G_4(s)G_5(s)G_7(s)$$
$$T_2 = G_1(s)G_2(s)G_3(s)G_4(s)G_6(s)G_7(s)$$

Mason's Rule: Definitions

2. Loop Gain:
The product of branch gains found by traversing a path that **starts** at a node and **ends** at the same node, following the direction of the signal flow, without passing through any other node more than once.

Ex: There are four loop gains:

$$L_1 = G_2(s)H_1(s)$$
$$L_2 = G_4(s)H_2(s)$$
$$L_3 = G_4(s)G_5(s)H_3(s)$$
$$L_4 = G_4(s)G_6(s)H_3(s)$$



3. Non-Touching Loops:
Loops that do not have any **nodes** in common.

Ex: Loop L_1 does not touch loops L_2, L_3, and L_4.

Mason's Rule: Formula

The transfer function of a system represented by a signal-flow graph:

$$G(s) = \frac{C(s)}{R(s)} = \frac{\sum_{k=1}^{N} T_k \Delta_k}{\Delta}$$

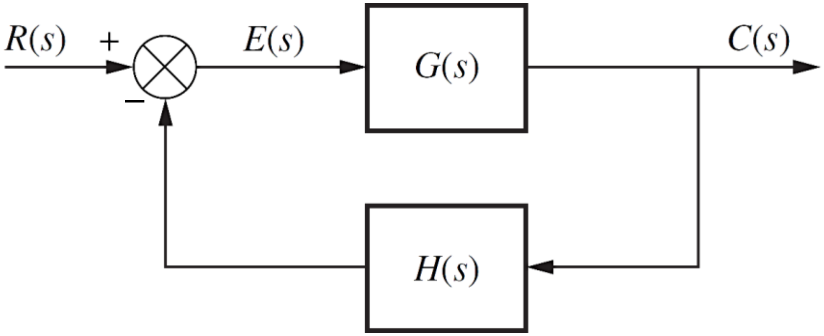
N: Total number of forward paths,
T_k: The kth forward-path gain,
Δ: The determinant of the graph, i.e.,:

$$\Delta = 1 - \sum L_i + \sum L_i L_j - \sum L_i L_j L_k + \cdots$$

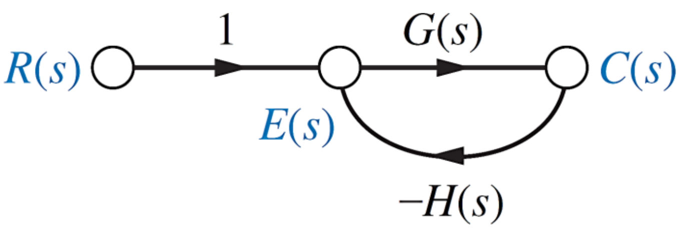
L_i: Loop gains,
$L_i L_j$: Product of the loop gains of any two non-touching loops,
$L_i L_j L_k$: Product of the loop gains of any three pairwise non-touching loops,
Δ_k: The cofactor value of Δ for the kth forward path. It is formed by **eliminating** from Δ those loop gains that touch the kth forward path.

Example

Find the transfer function using Mason's rule.



Solution:



$N = 1$ (There is only 1 forward path)

$T_1 = G(s)$

$L_1 = -G(s)H(s)$ (There is only 1 loop)

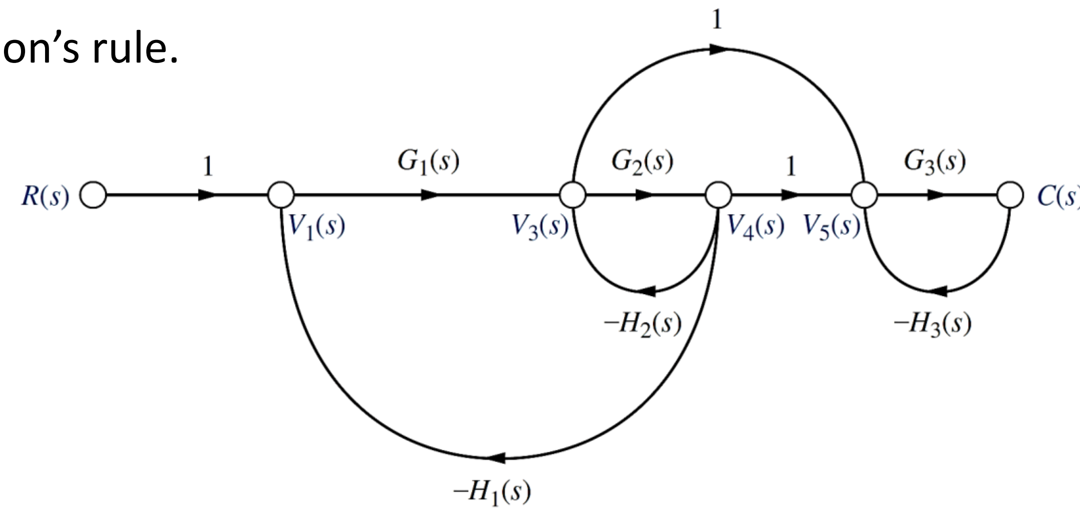
$$\Delta = 1 - \big(-G(s)H(s)\big) = 1 + G(s)H(s)$$

$$\Delta_1 = 1$$

$$\Rightarrow G(s) = \frac{C(s)}{R(s)} = \frac{T_1 \Delta_1}{\Delta} = \frac{G(s)}{1 + G(s)H(s)}$$

Example

Find the transfer function using Mason's rule.



Solution:

$N = 2$ (There are 2 forward paths)

$T_1 = G_1(s)G_2(s)G_3(s)$

$T_2 = G_1(s)G_3(s)$

$L_1 = -G_1(s)G_2(s)H_1(s)$

$L_2 = -G_2(s)H_2(s)$ (There are 3 loops)

$L_3 = -G_3(s)H_3(s)$

$\Delta = 1 - (L_1 + L_2 + L_3) + (L_1 L_3 + L_2 L_3)$

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

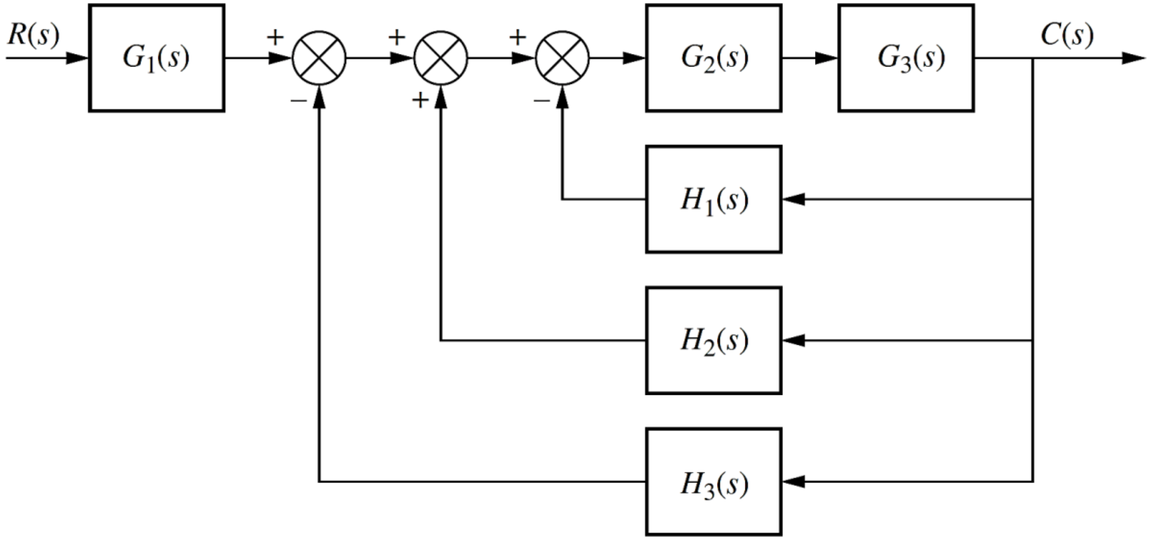
$\Delta_2 = 1 - 0 + 0 = 1$

$$\Rightarrow \quad G(s) = \frac{C(s)}{R(s)} = \frac{T_1\Delta_1 + T_2\Delta_2}{\Delta}$$

$$= \frac{G_1(s)G_3(s)[1 + G_2(s)]}{[1 + G_2(s)H_2(s) + G_1(s)G_2(s)H_1(s)][1 + G_3(s)H_3(s)]}$$

Example

Find the transfer function using Mason's rule.

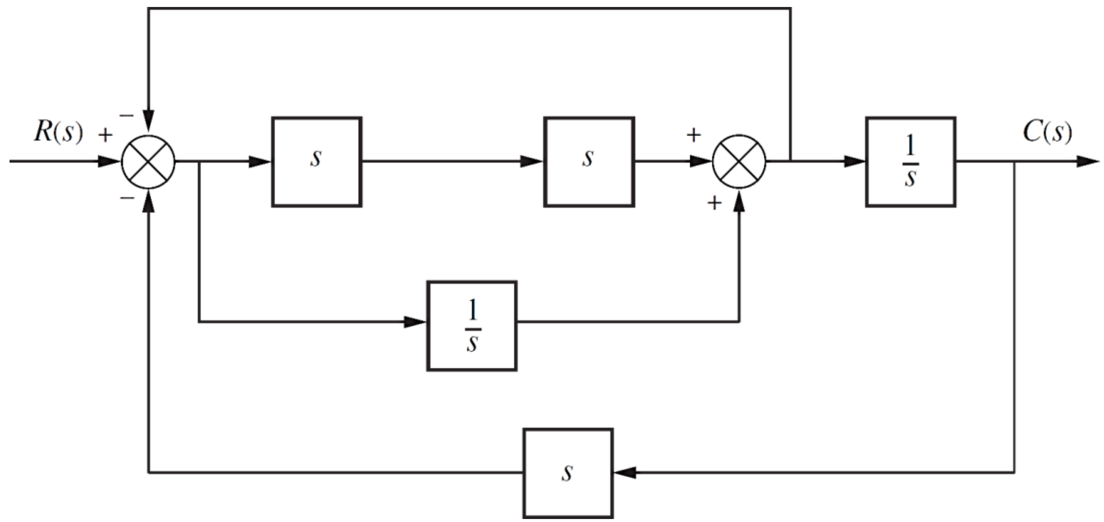


Answer:

$$G(s) = \frac{G_3(s)G_2(s)G_1(s)}{1 + G_3(s)G_2(s)[H_1(s) - H_2(s) + H_3(s)]}$$

Example

Find the transfer function using Mason's rule.



Answer:

$$T(s) = \frac{s^3 + 1}{2s^4 + s^2 + 2s}$$