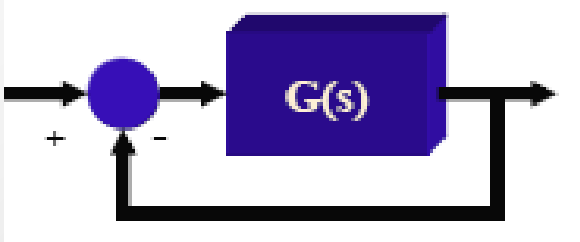
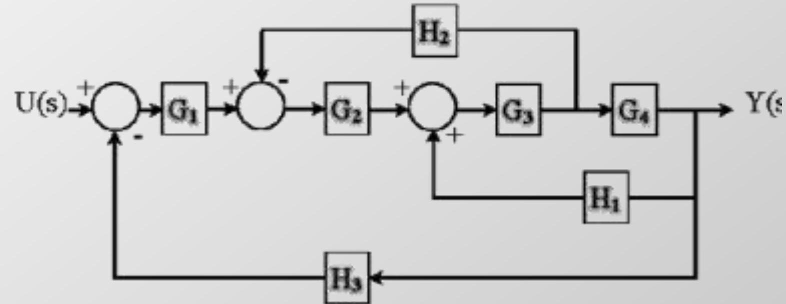
Mathematical Models of Systems



Outlines

- Block Diagram Models
- Basic Elements of Signal Flow Graph
- Basic Properties
- Definitions of Signal Flow Graph Terms
- Mason Theory
- Examples

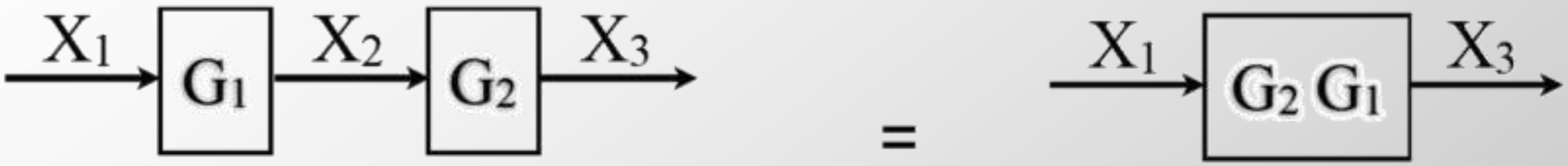
Block Diagram Models

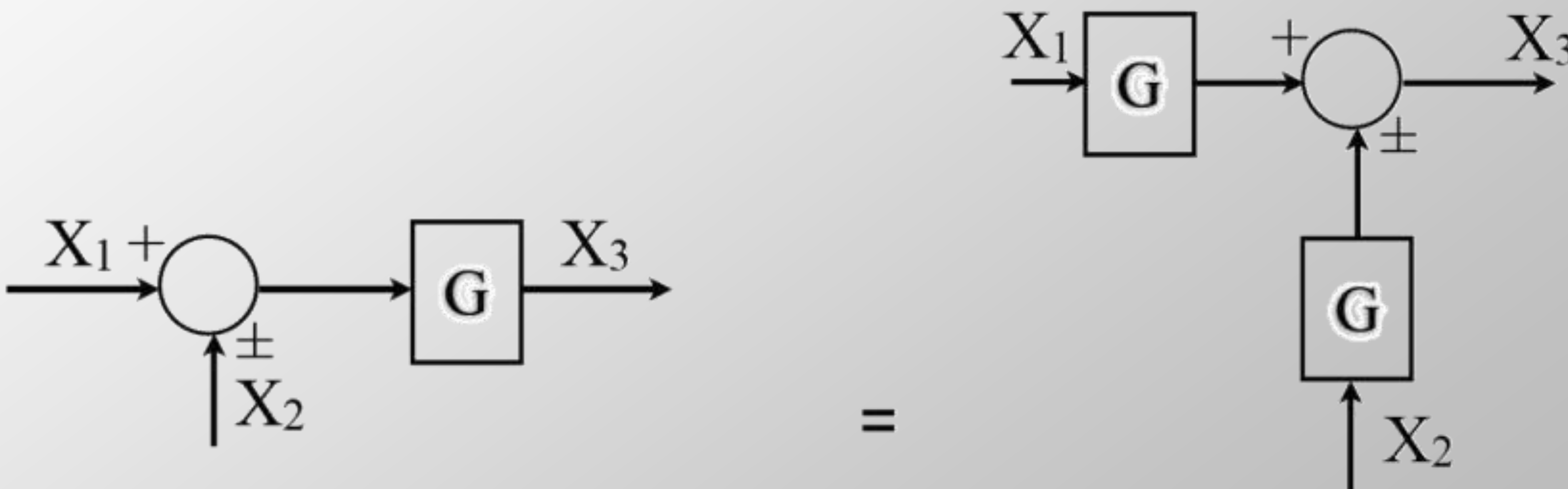


- Visualize input output relations
- Useful in design and realization of (linear) components
- Helps understand flow of information between internal variables.
- Are equivalent to a set of linear algebraic equations

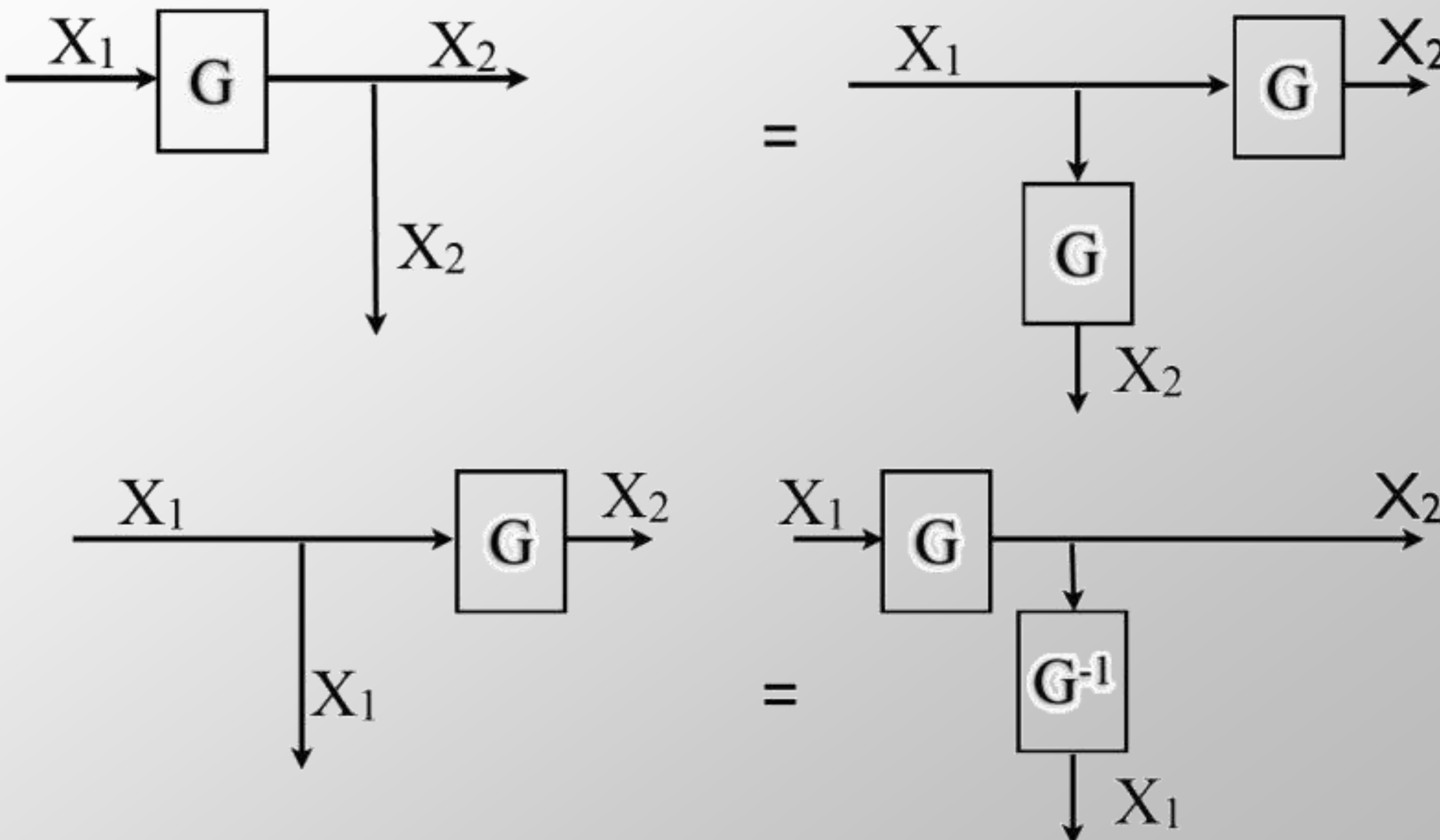
  (of rational functions).
- Mainly relevant where there is a cascade of information flow

Block Diagram Manipulation Rules

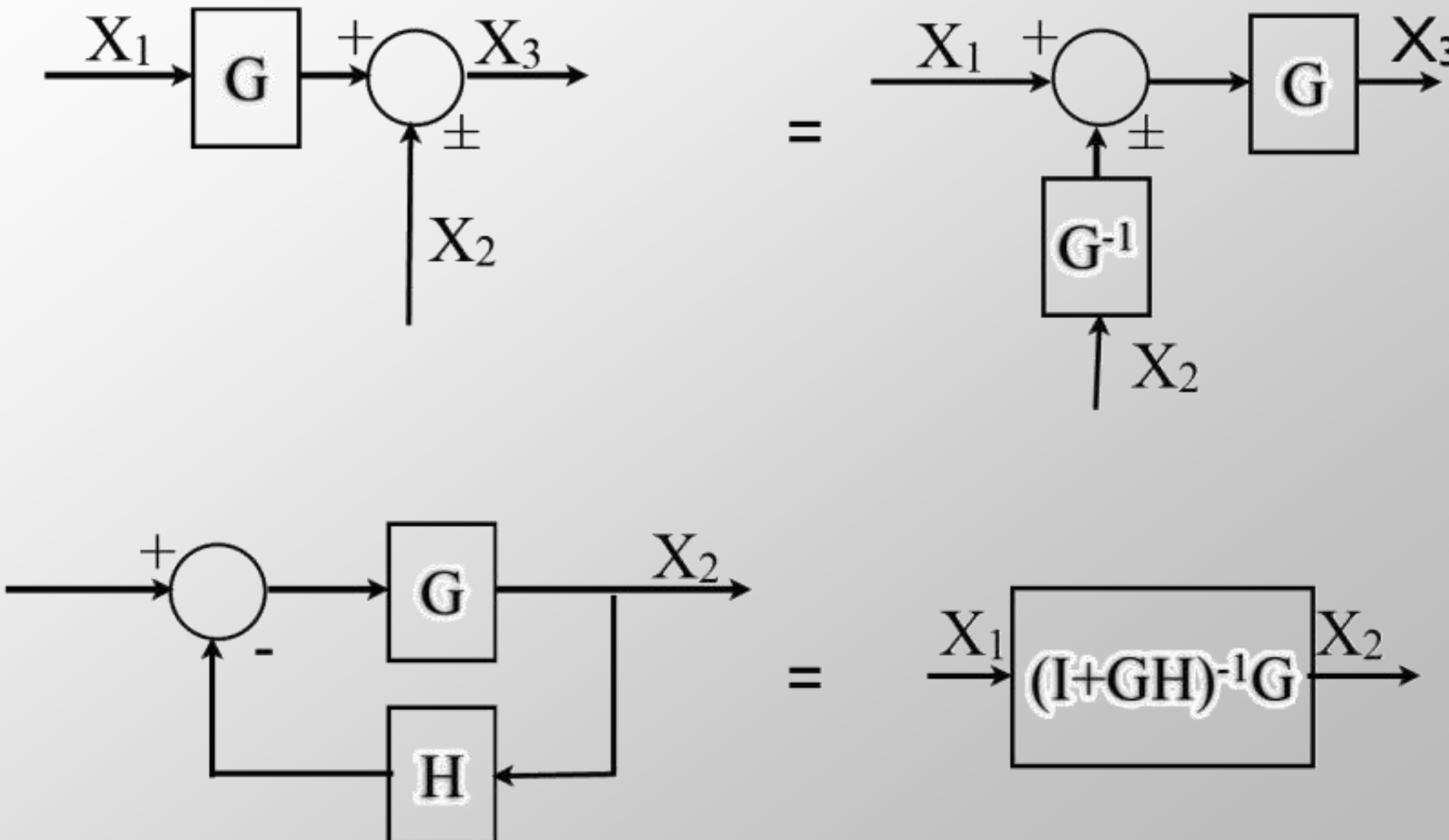




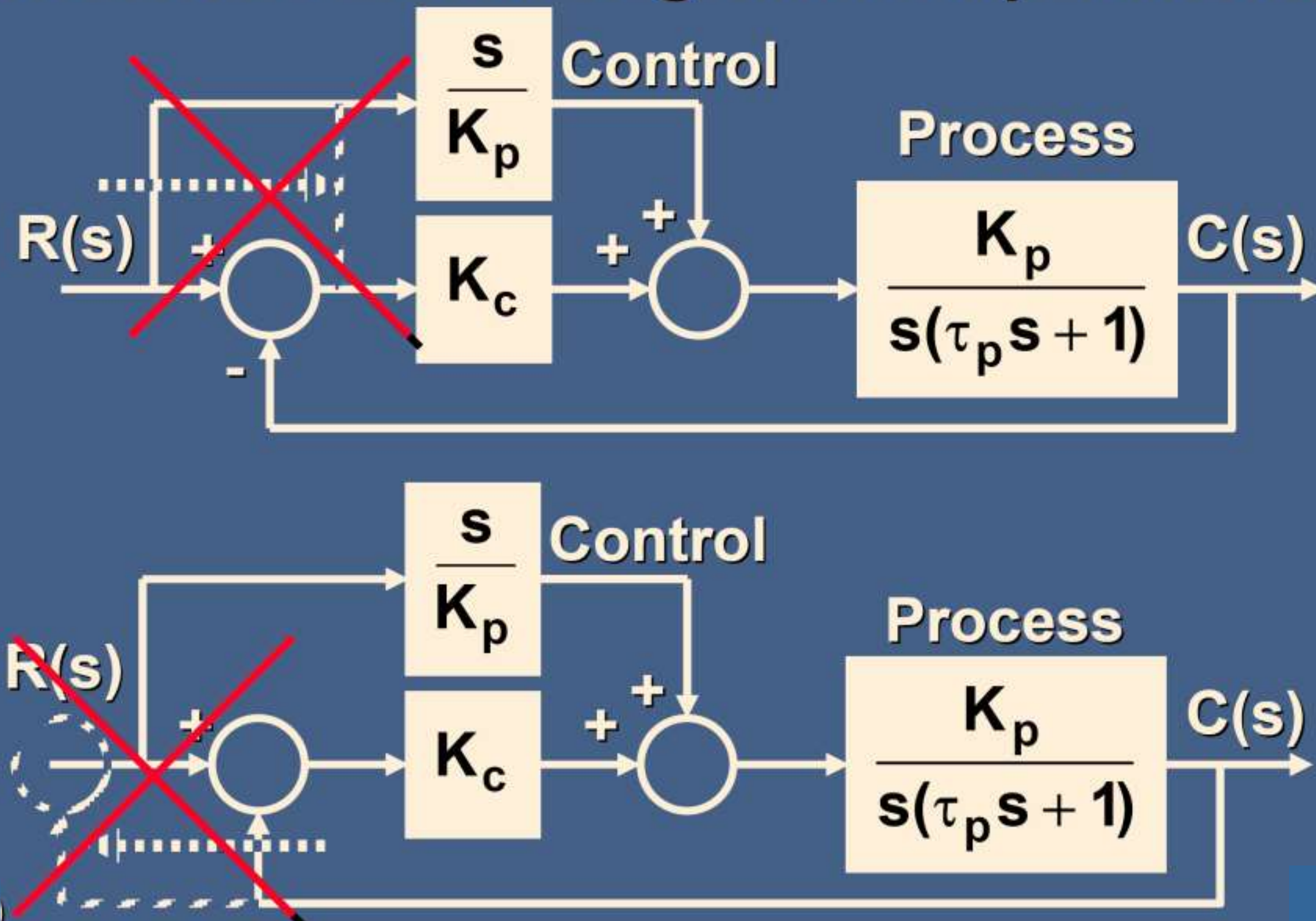
Block Diagram Manipulation Rules



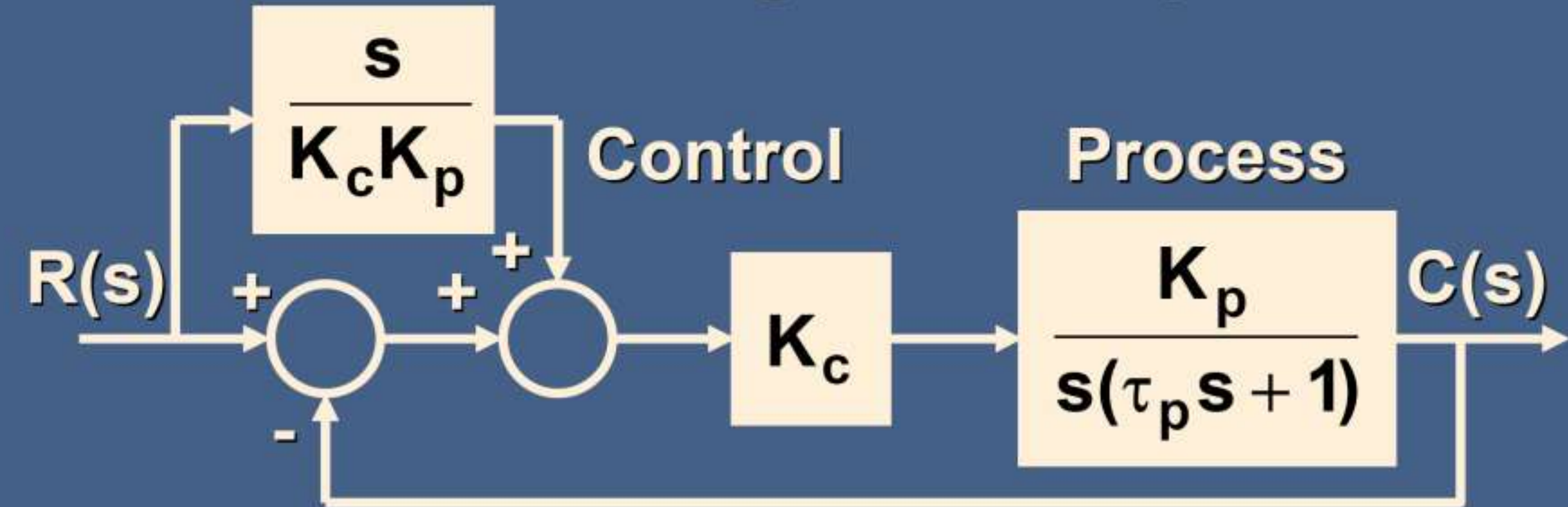
Block Diagram Manipulation Rules

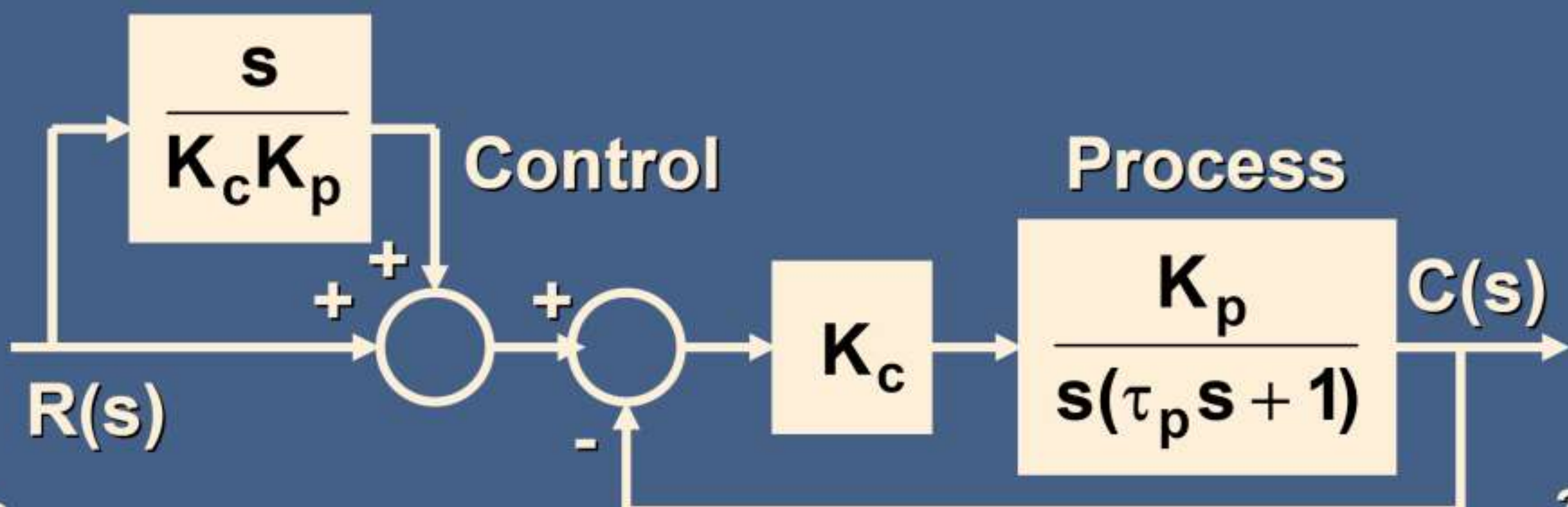


Incorrect Block Diagram Manipulations

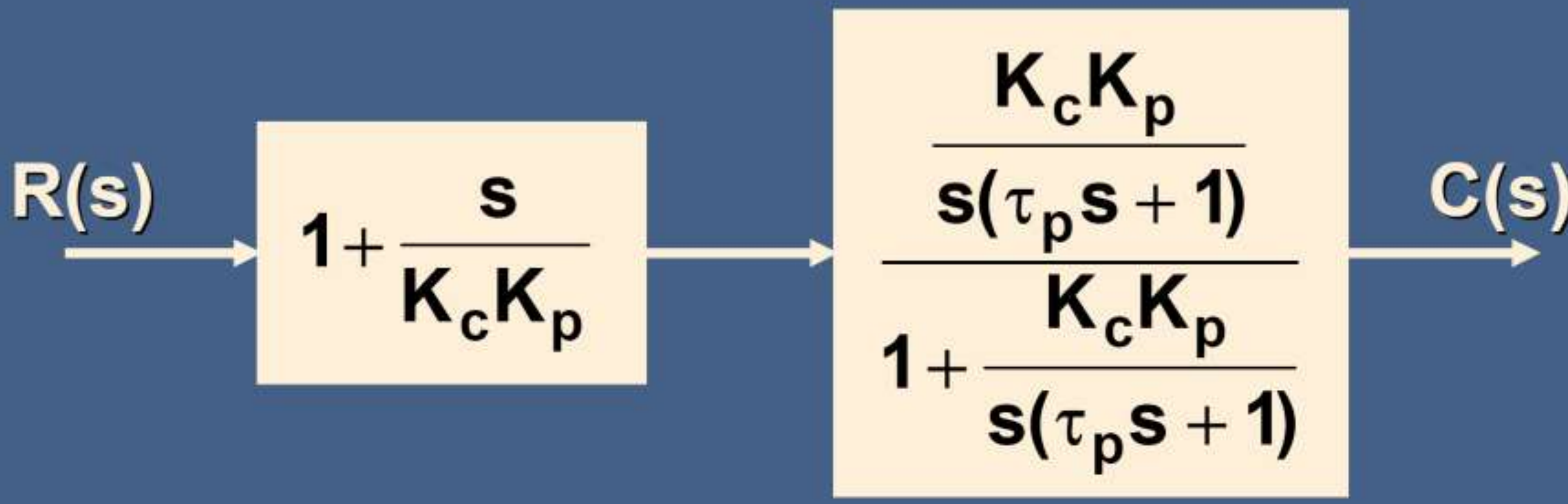


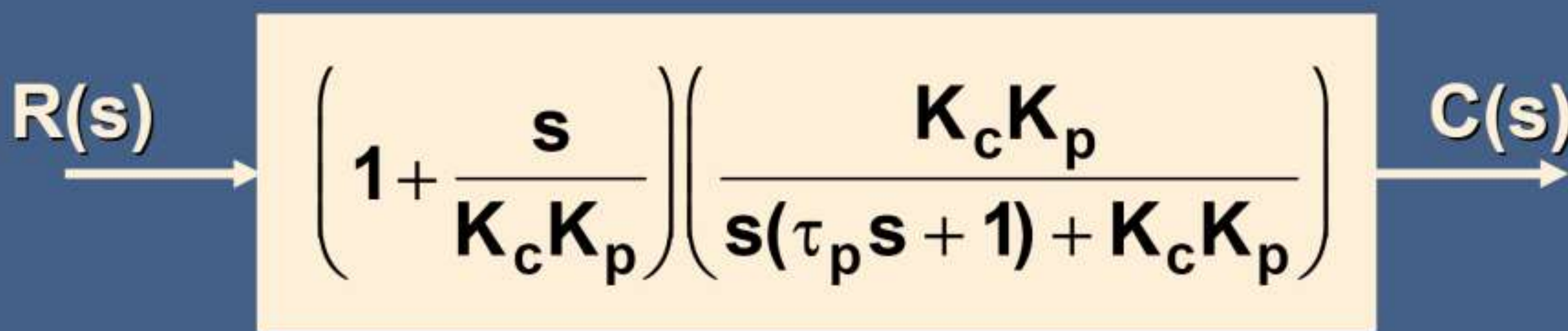
Correct Block Diagram Manipulations



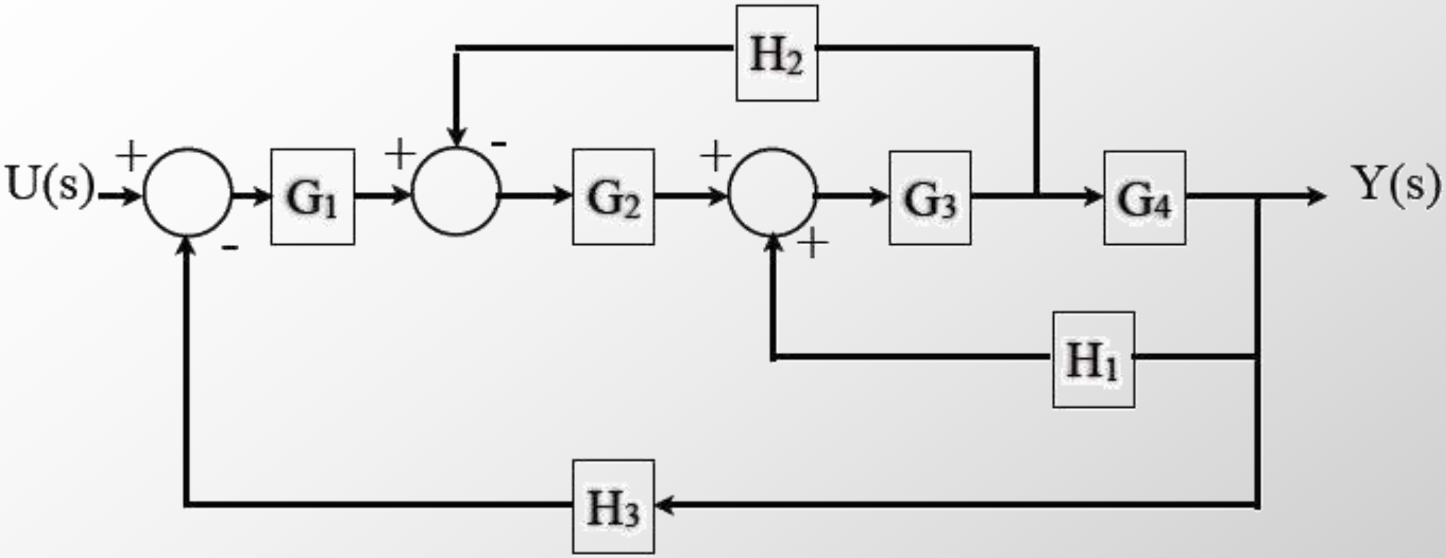


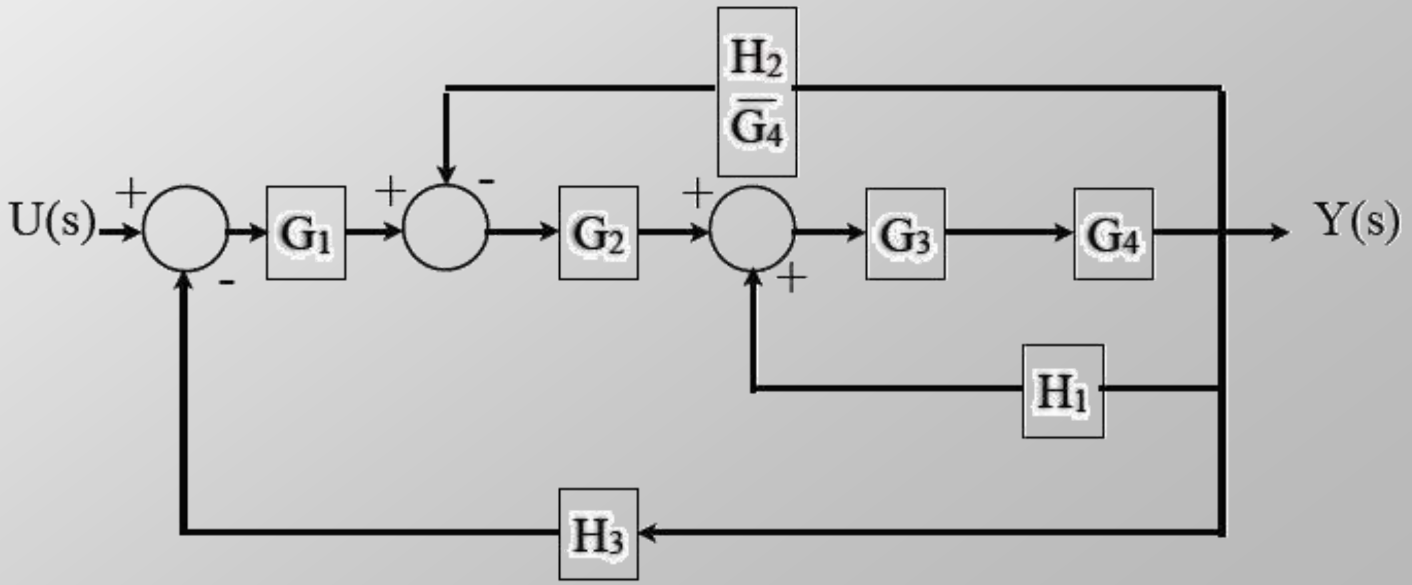
Block Diagram Reduction



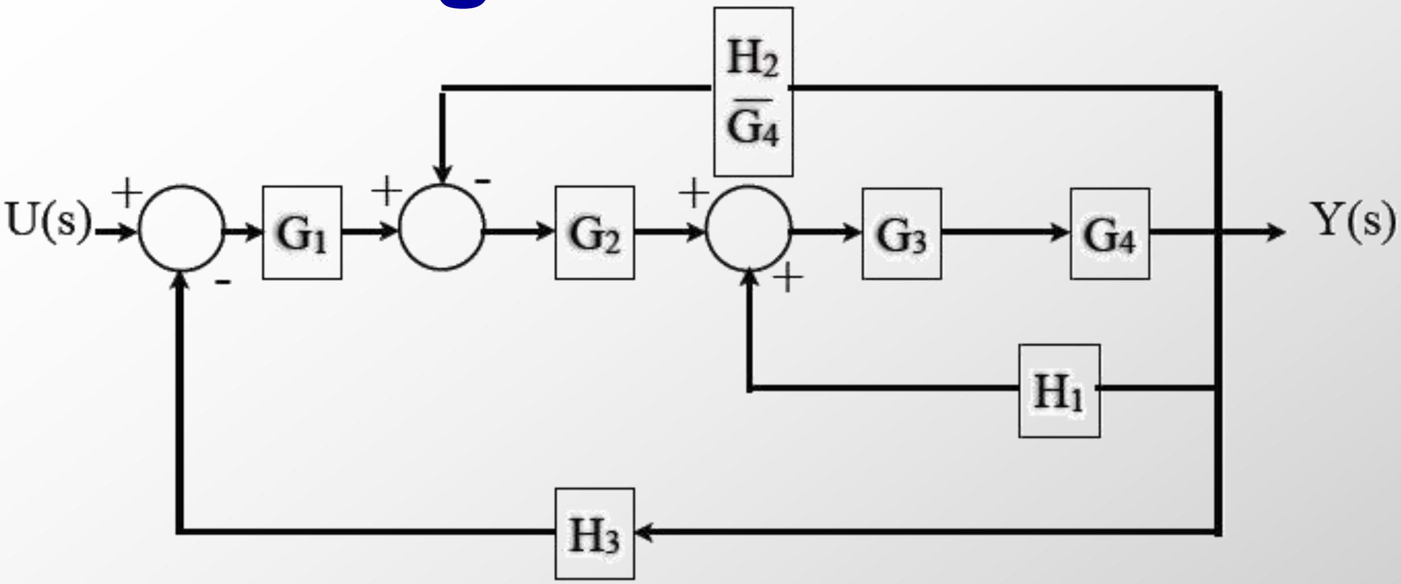


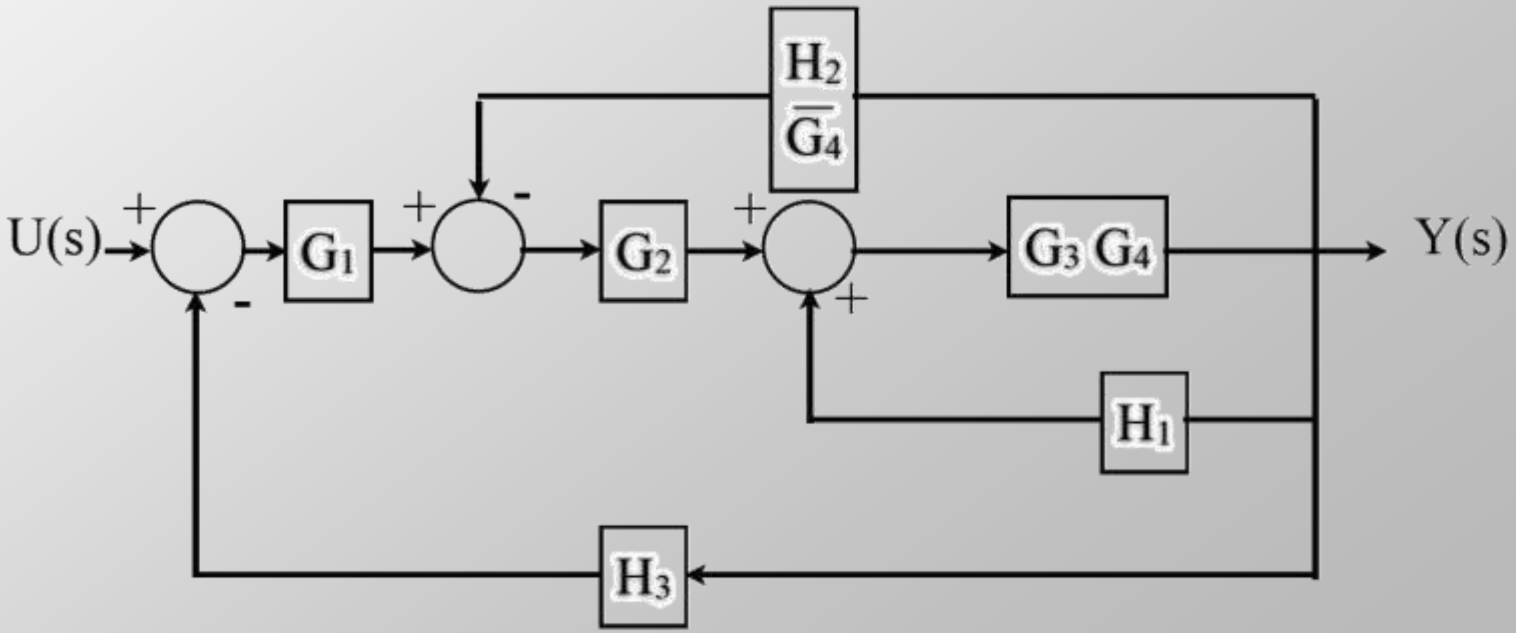
Block Diagram Reduction



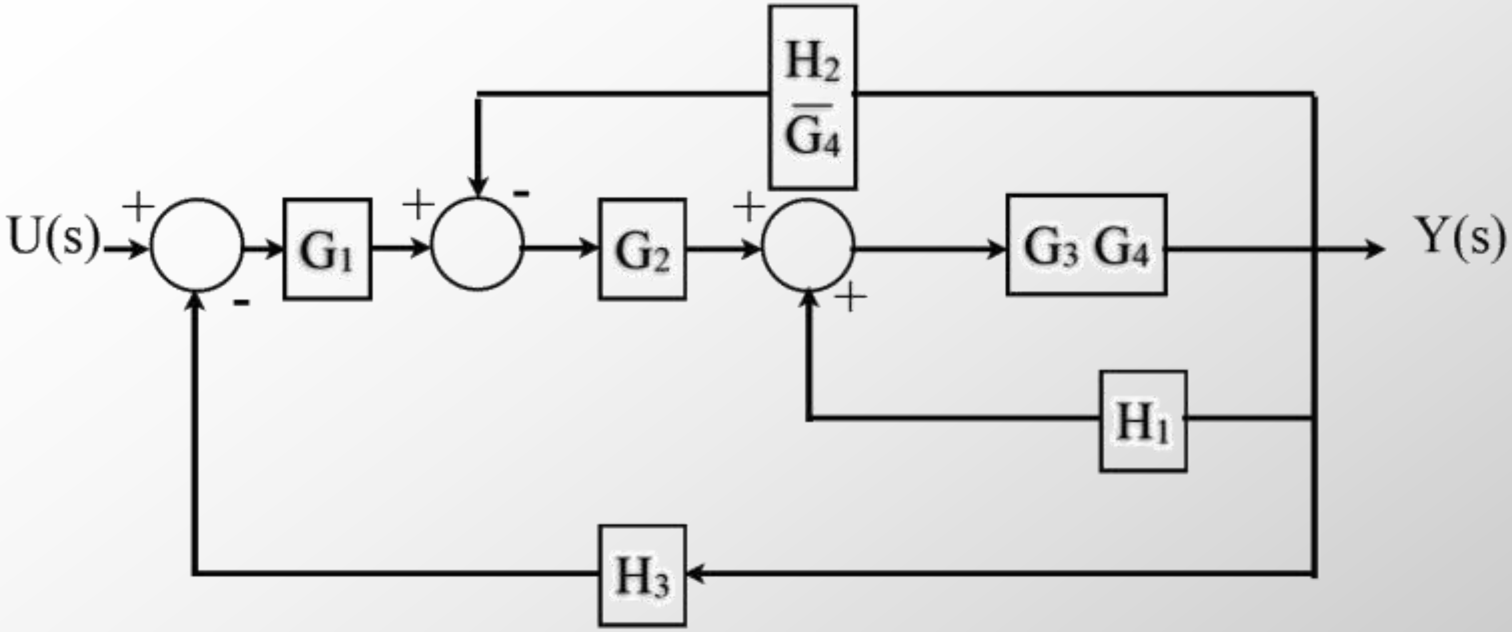


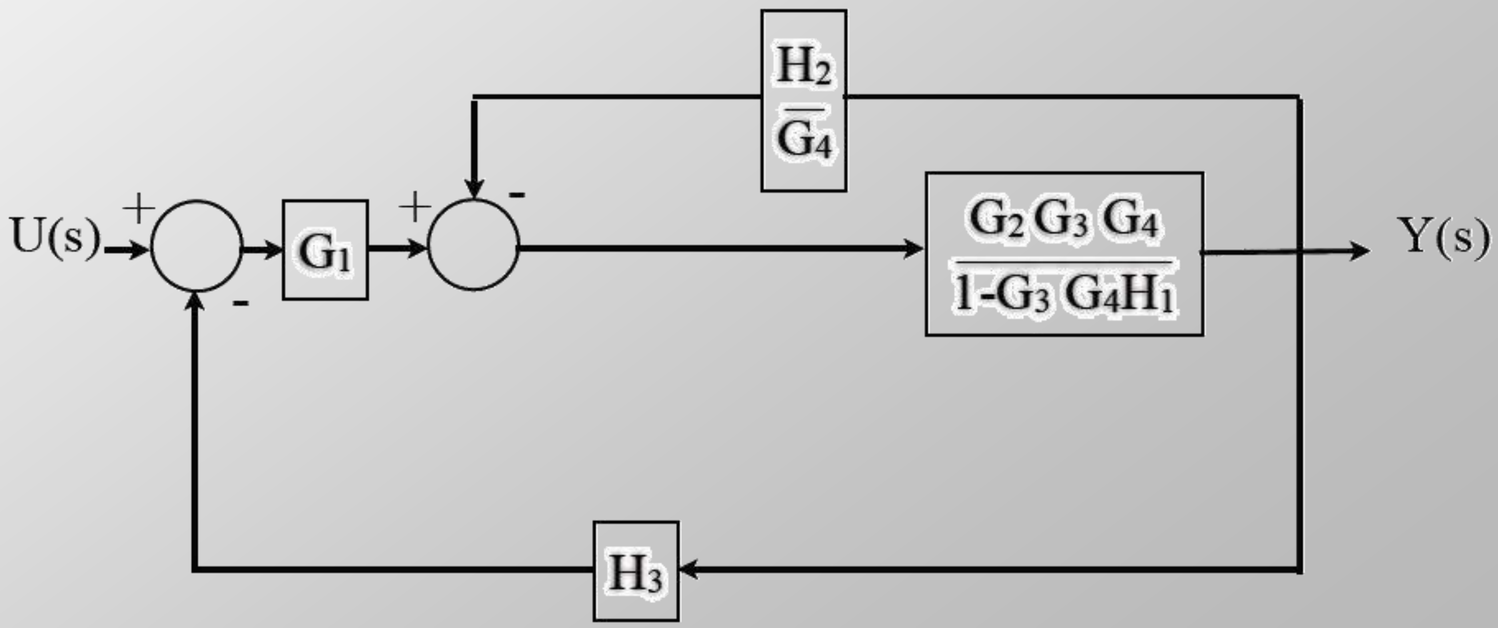
Block Diagram Reduction



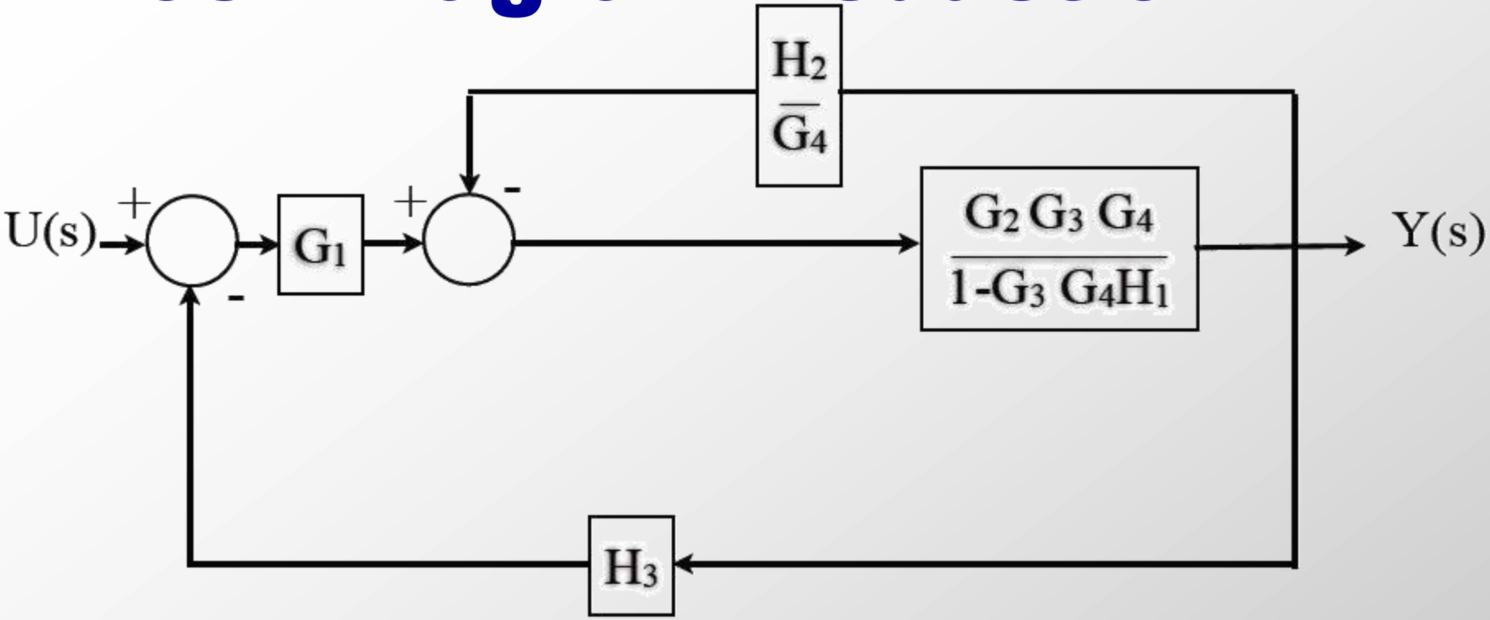


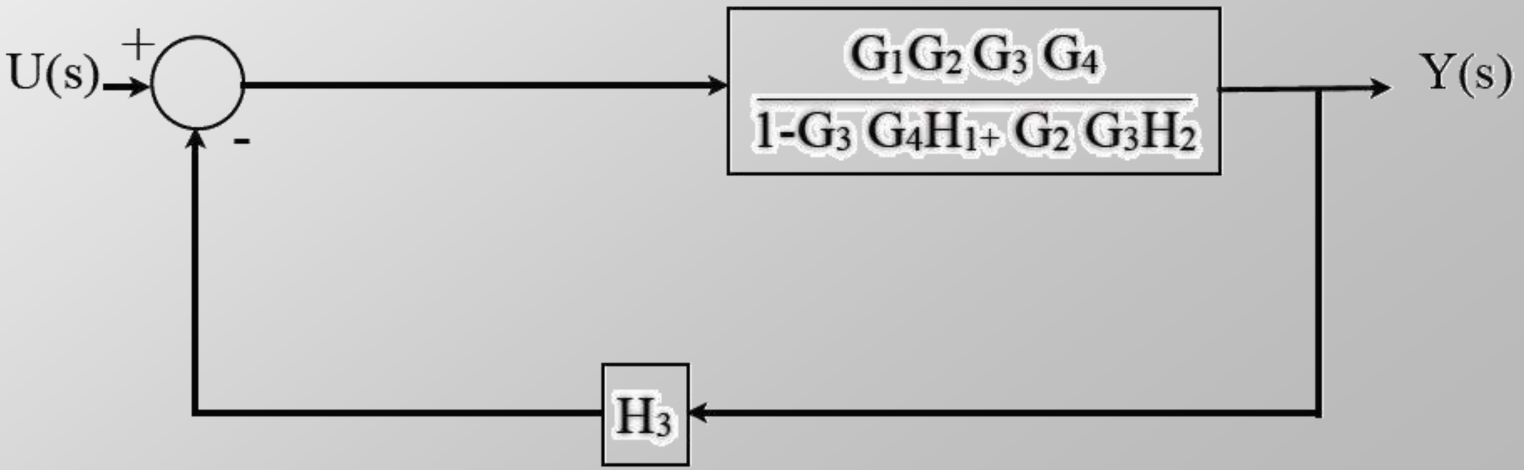
Block Diagram Reduction



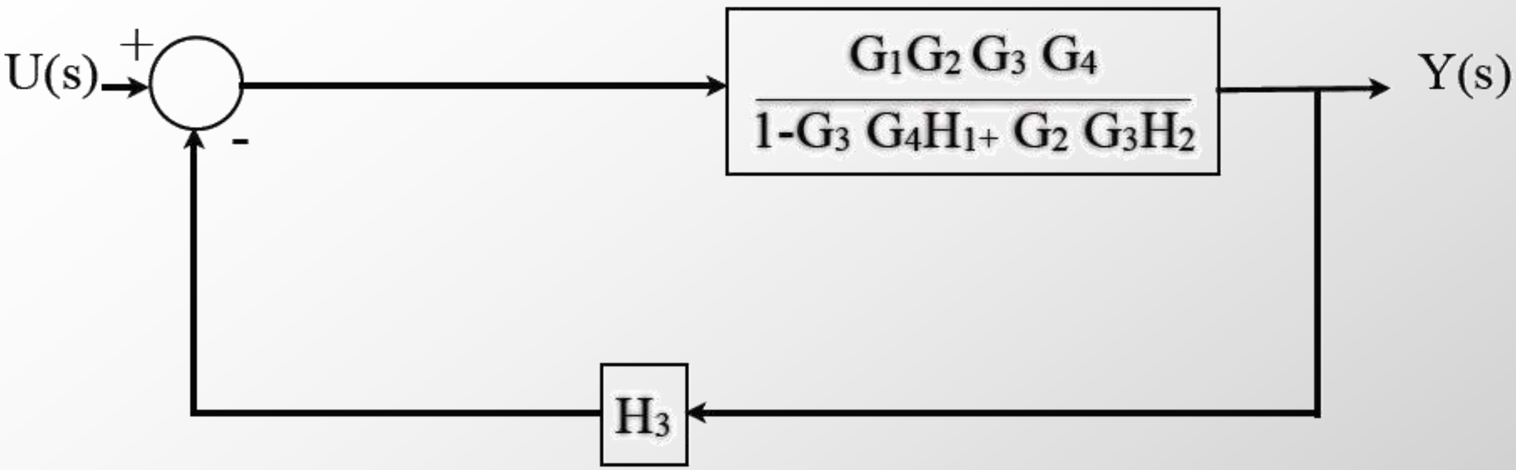


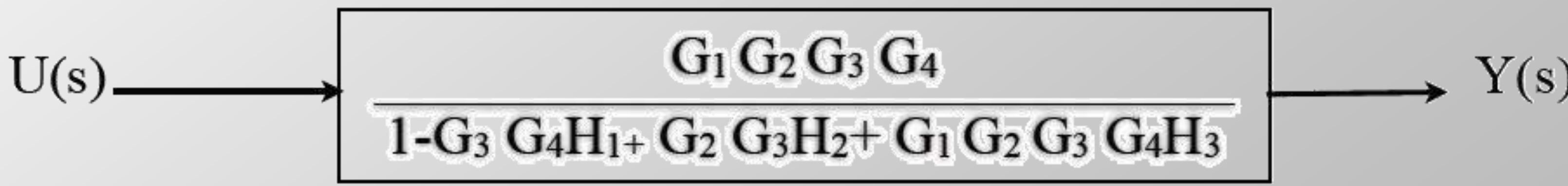
Block Diagram Reduction





Block Diagram Reduction





Signal Flow Graphs

- Alternative to block diagrams
- Do not require iterative reduction to find transfer functions (using Mason's gain rule)
- Can be used to find the transfer function between any two variables (not just the input and output).
- Look familiar to computer scientists (?)

Signal-flow graph

From Wikipedia, the free encyclopedia

A signal-flow graph (SFG) is a special type of block diagram, constrained by rigid mathematical rules, that is a graphical means of showing the relations among the variables of a set of linear algebraic relations. Nodes represent variables, and are joined by branches that have assigned directions (indicated by arrows) and gains. A signal can transmit only in the direction of the arrow.

Utility of Signal Flow Graphs

- Alternative to block diagram approach

  - may be better for complex systems

  - good for highly interwoven systems

  - system variables represented as nodes

  - branches (lines) between nodes show relationships between system variables

- The "flow graph gain formula" (Mason) allows the system transfer function to be directly computed without manipulation or reduction of the diagram.

Example 1 Simple amplifier

$$y_1 = a_{11}x_1$$

$$x_1 = y_1 / a_{11}$$

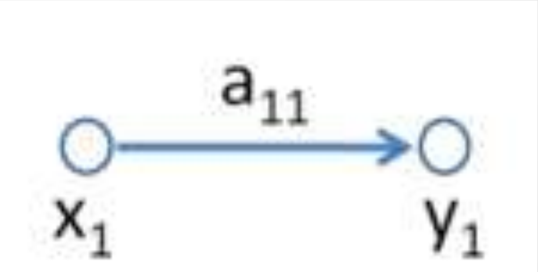


Figure 1: Simple SFG

Amplification of a signal x_1 to become a larger output y_2 by an amplifier with gain a_{11} is described mathematically by:

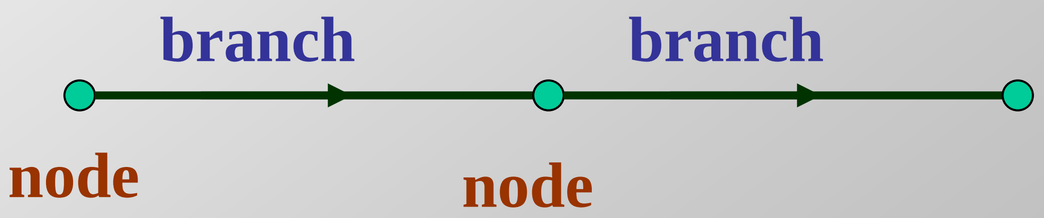
which becomes the signal flow graph of Figure 1.

Although this equation is represented by the SFG of Figure 1, the algebraically equivalent relation

is *not* considered to be implied by Figure 1. That is, the SFG is unilateral, sometimes emphasized by calling x_1 a "cause" and y_1 an "effect", or by calling x_1 an "input" and y_1 an "output".

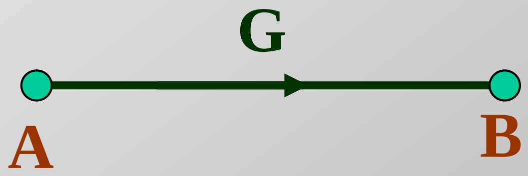
Basic Elements of Signal Flow Graph

- **A** **Signal flow graph** is a diagram consisting of **nodes** that are connected by several directed **branches**.



Basic Elements of Signal Flow Graph

- A **node** is used to represent a variable (inputs, outputs, other signals)
- A **branch** shows the dependence of one variable (**node**) on another variable (**node**)
  – Each **branch** has GAIN and DIRECTION
  – A signal can transmit through a **branch** only in the direction of the arrow
  – If gain is not specified gain =1



B = G A

Example 2 Two-port network

$$y_1 = a_{11}\, x_1 + a_{12}\, x_2$$

$$y_2 = a_{21}\, x_1 + a_{22}\, x_2\ ,$$

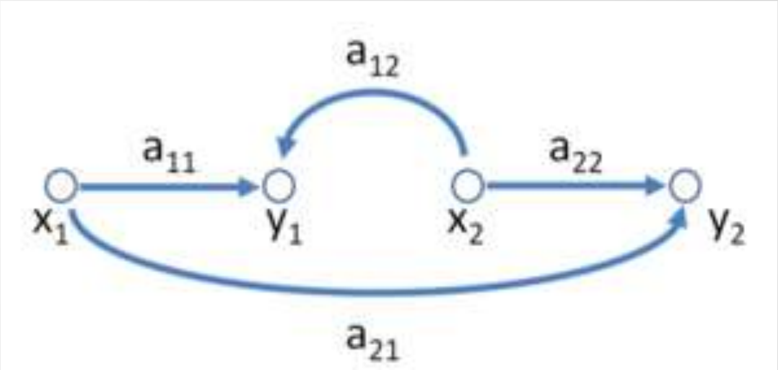


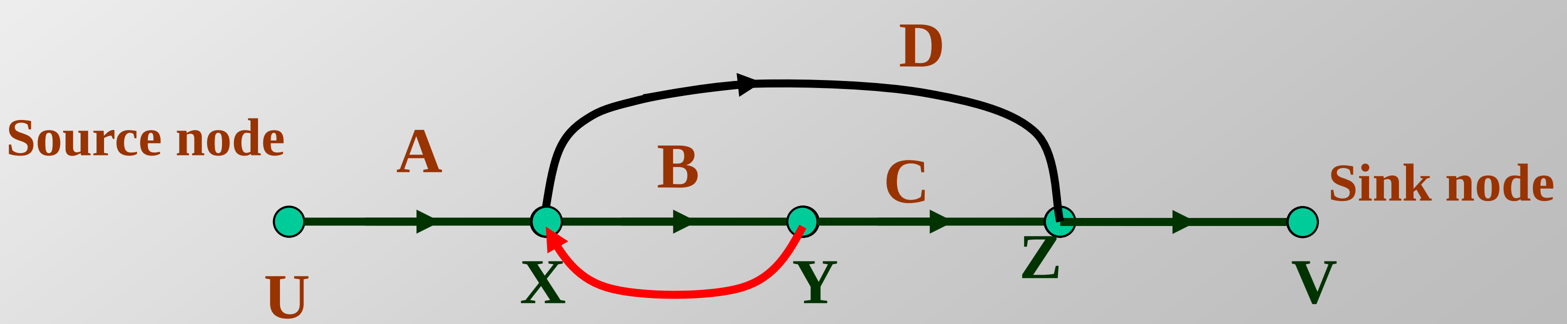
Figure 2: Two-port SFG

The two coupled equations below can represent the current-voltage relations in a two-port network:

which equations become the signal flow graph of Figure 2.

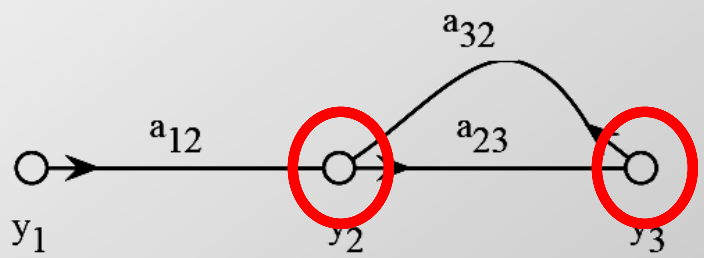
Nodes

- A **node** is used to represent a variable
- Source **node** (input node)
  - All braches connected to the node are leaving the node
  - Input signal is not affected by other signals
- Sink **node** (output node)
  - All braches connected to the node are entering the node
  - output signal is not affecting other signals

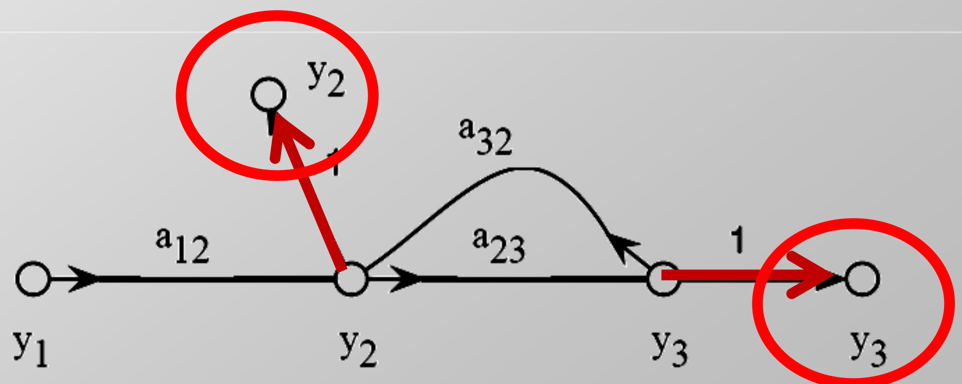


Input / Output

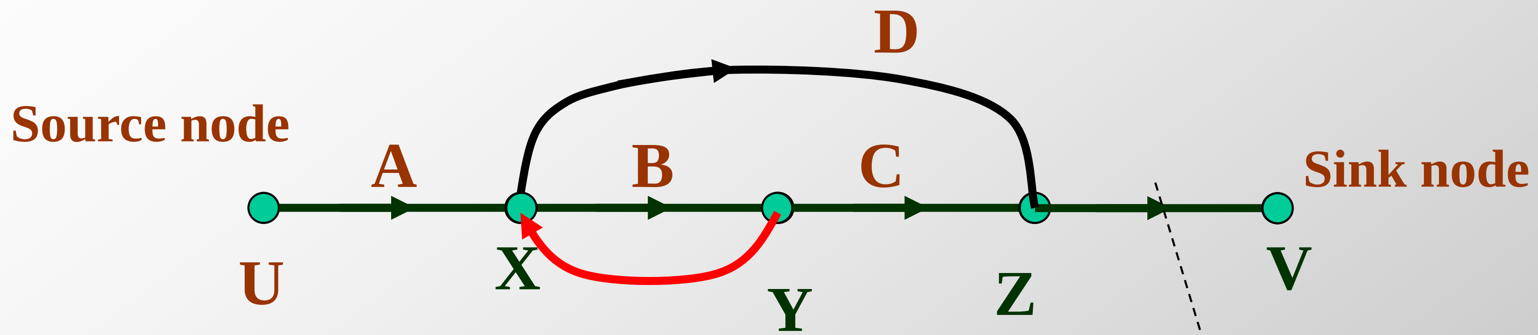
- Input (*source*) has only outgoing edges
- Output (*sink*) has only incoming edges



- *any* variable can be made into an output by adding a sink with "1" edge



Relationship Between Variables



Gain is not shown means gain=1

U (input)

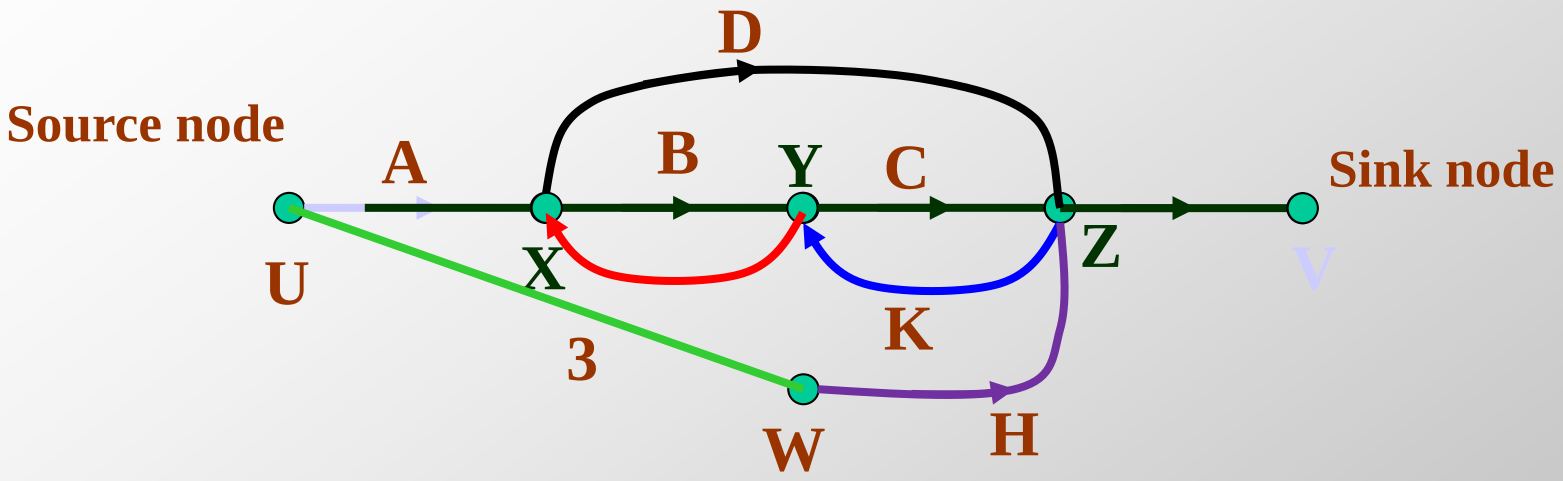
X=AU+Y

Y=BX

Z=CY+DX

V=Z (output)

Another Example



X=AU+Y

Y=BX+KZ

Z=CY+DX+HW

W=3U

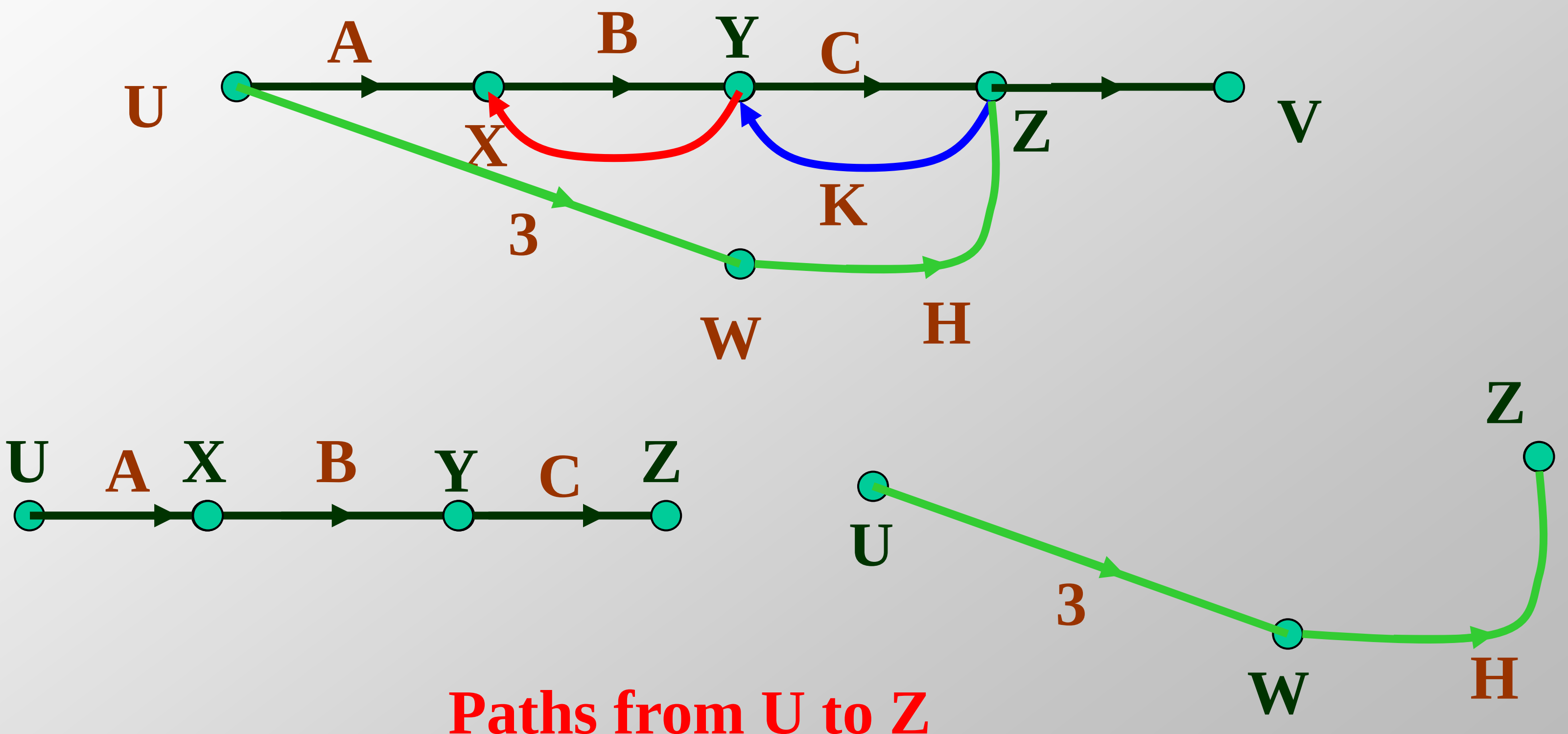
V=Z

Basic Properties

- Signal flow graphs applies to linear systems only
- Nodes are used to represent variables
- A branch from node X to node Y means that Y depends on X
- Value of the variable (node) is the sum of gain of branch * value of node
- Non-input node cannot be converted to an input node
- We can create an output node by connecting unit branch to any node

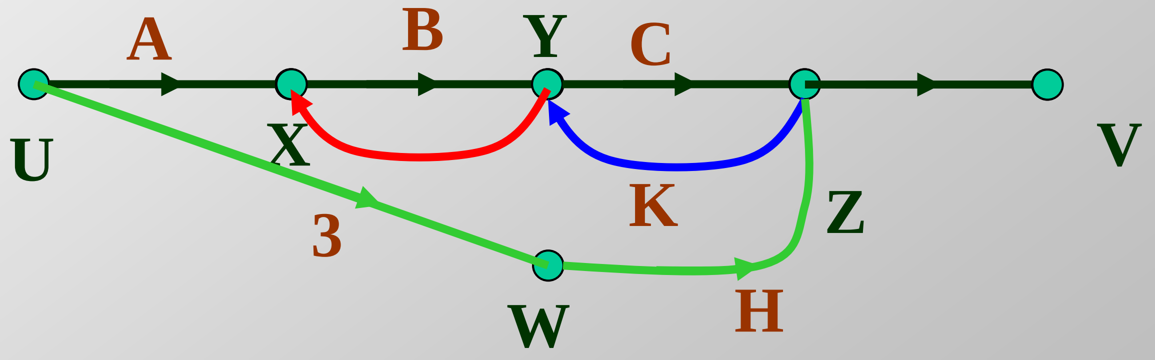
Terminology: Paths

- **A path:** is a branch or a continuous sequence of branches that can be traversed from one node to another node



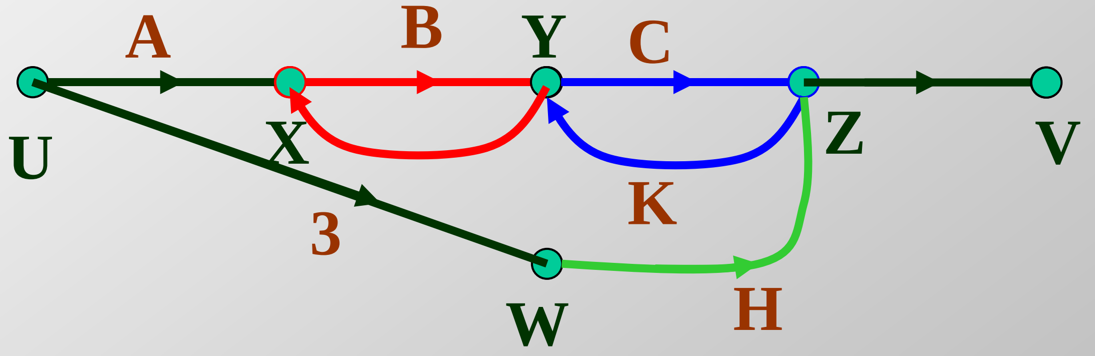
Terminology: Paths

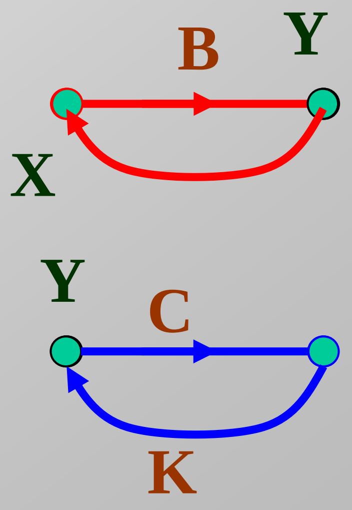
- **A path:** is a branch or a continuous sequence of branches that can be traversed from one node to another node
- **Forward path**: path from a source to a sink
- **Path gain**: product of gains of the braches that make the path



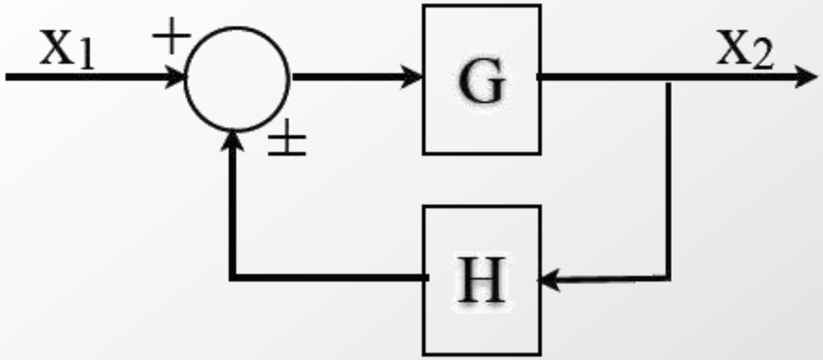
Terminology: loop

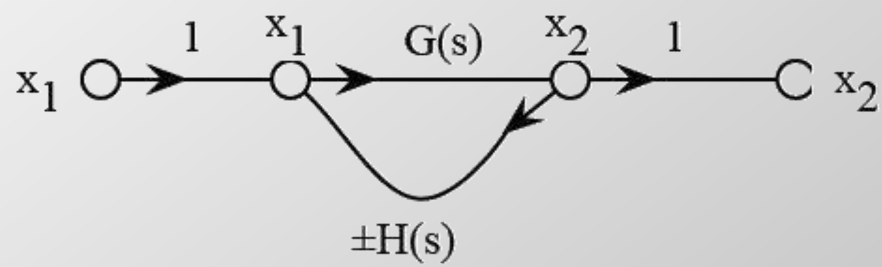
- A **loop:** is a closed path that originates and terminates on the same node, and along the path no node is met twice.
- **Nontouching loops**: two loops are said to be nontouching if they do not have a common node.





Block Diagram vs Signal Flow Graph





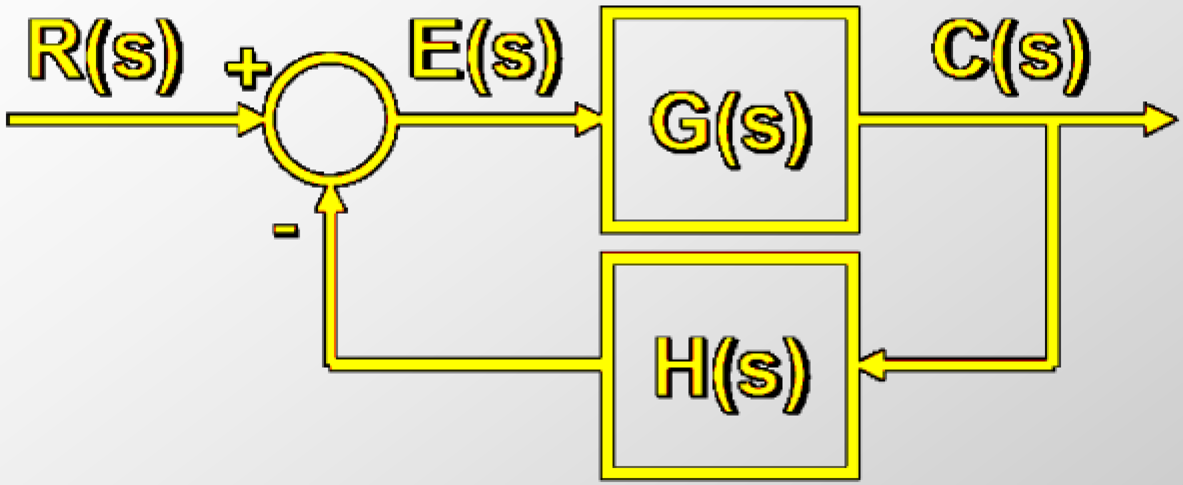
- Blocks ⇒ Edges (branches)

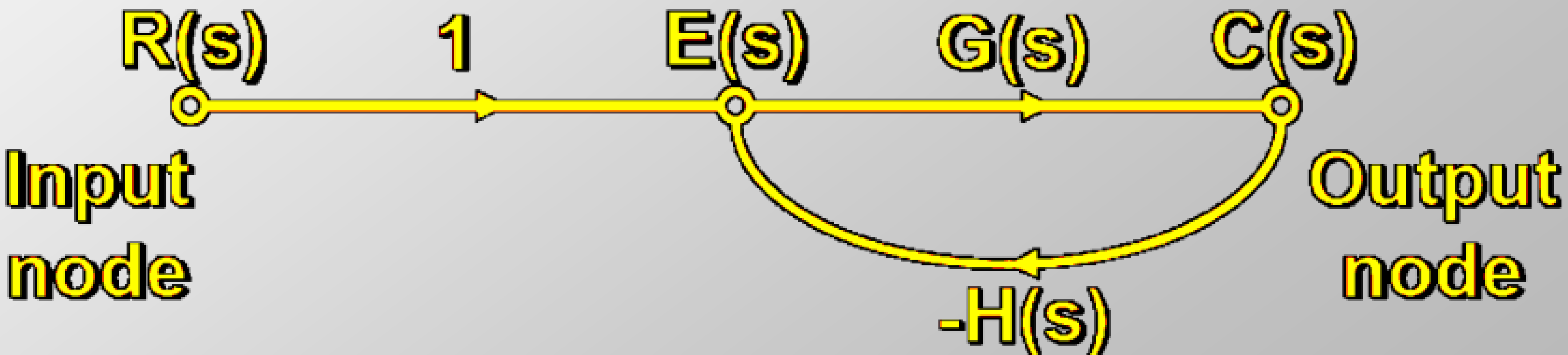
  (representing transfer functions)

- Edges + junctions ⇒ Vertices (nodes)

  (representing variables)

Basic Signal Flow Graph



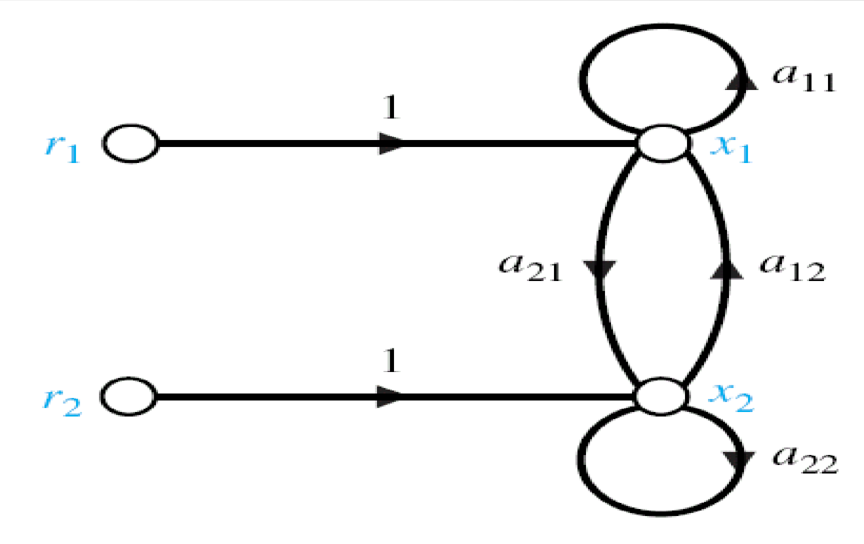


Algebraic Eq representation

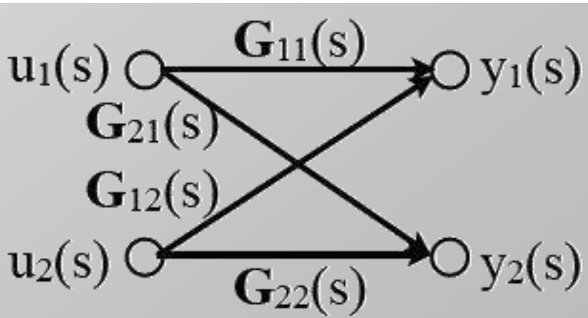
- $\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{r}$

  $x_1 = a_{11}x_1 + a_{12}x_2 + r_1$

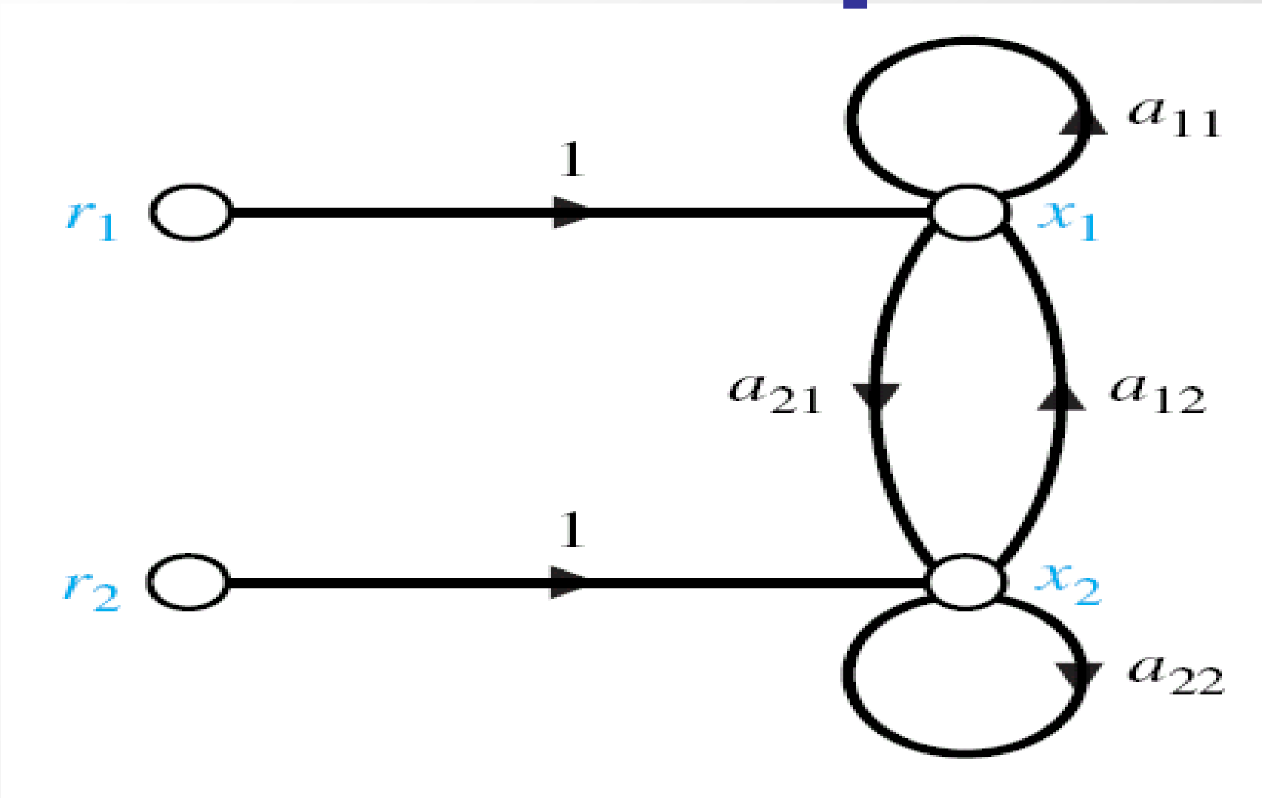
  $x_2 = a_{21}x_1 + a_{22}x_2 + r_2$



- $\mathbf{y}(s) = \mathbf{G}(s)\mathbf{u}(s)$



An example



$$a_{11} x_1 + a_{12} x_2 + r_1 = x_1$$

$$a_{21} x_1 + a_{22} x_2 + r_2 = x_2$$

An example...

$$(1 - a_{11})x_1 + (-a_{12})\ x_2 = r_1$$

$$(-a_{21})\ x_1 + (1 - a_{22})x_2 = r_2$$

- This have the solution

$$x_1 = (1 - a_{22})/\Delta\, r_1 + a_{12}/\Delta\, r_2$$

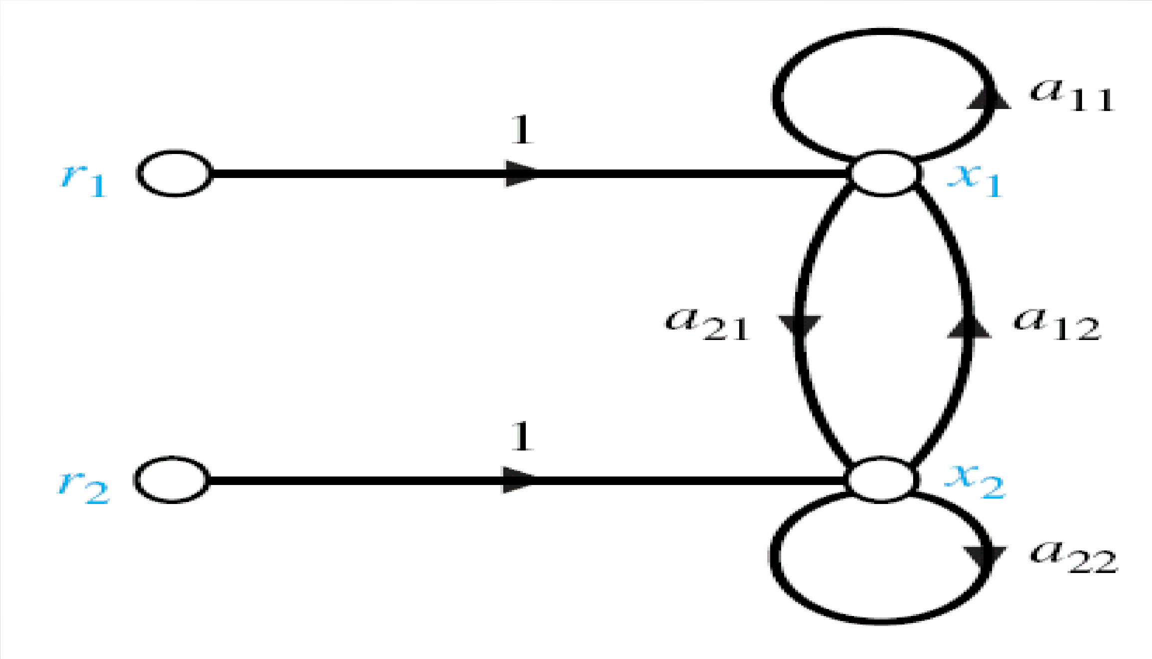
$$x_2 = (1 - a_{11})/\Delta\, r_2 + a_{21}/\Delta\, r_1$$

$$\Delta = 1 - a_{11} - a_{22} + a_{22}\, a_{11} - a_{12}\, a_{21}$$

An example

$$\Delta = 1 - a_{11} - a_{22} + a_{22}a_{11} - a_{12}a_{21}$$

- Self loops a_{11} , a_{22} , $a_{12}a_{21}$
- Product of non touching loops $a_{22}a_{11}$



SGF : in general

- The linear dependence (T_{ij}) between the independent variable x_i (input) and the dependent variable (output) x_j is given by Mason's SF gain formula
- $$T_{ij} = \frac{\sum_k P_{ijk} \Delta_{ijk}}{\Delta}$$
  $$P_{ijk} = k^{\text{th}} \text{ path from } \mathrm{x_i} \text{ to } \mathrm{x_j}$$
  $$\Delta = \text{ determinant of the graph}$$
  $$\Delta_{ijk} = \text{cofactor of the path } P_{ijk}$$

The determinant Δ

$$\Delta = 1 - \sum_{n=1}^{N} L_n + \sum_{m=1,q=1}^{M,Q} L_m L_q - \sum L_r L_s L_t \ldots$$

$$L_q \text{ is the loop gain}$$

- Or

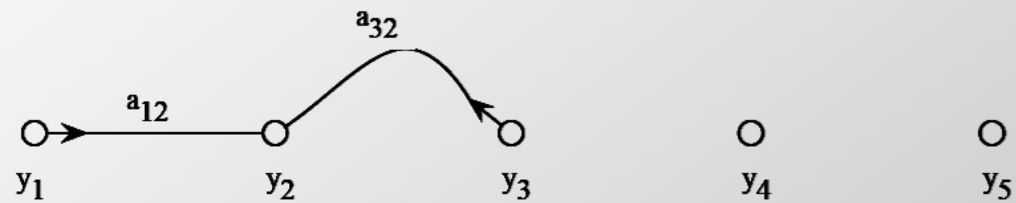
Δ=1 –(sum of all different loop gains) +(sum of the gain products of all combinations of 2 non-touching loops)

-(sum of the gain products of all combinations of 3 non-touching loops)…

- The cofactor Δ_{ijk} is the determinant with loops touching the kth path removed

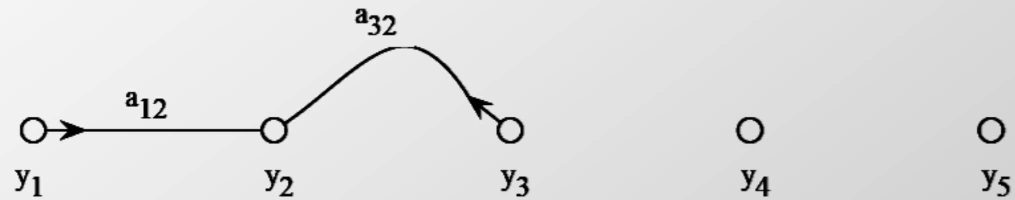
Another SFG Example

$y_2 = a_{12}y_1 + a_{32}y_3$

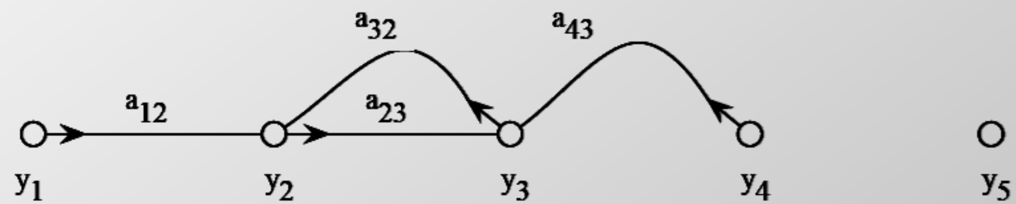


Another SFG Example

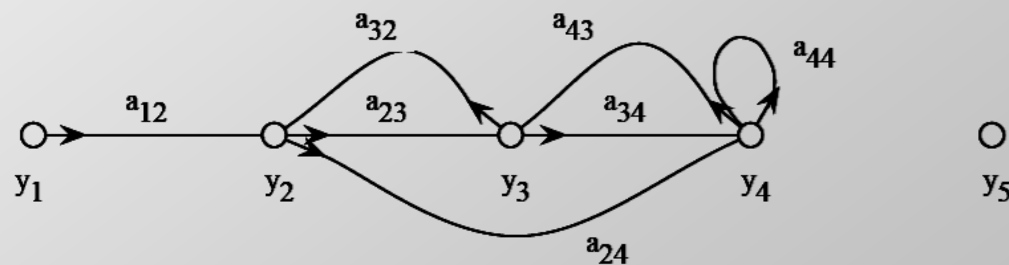
$y_2 = a_{12}y_1 + a_{32}y_3$



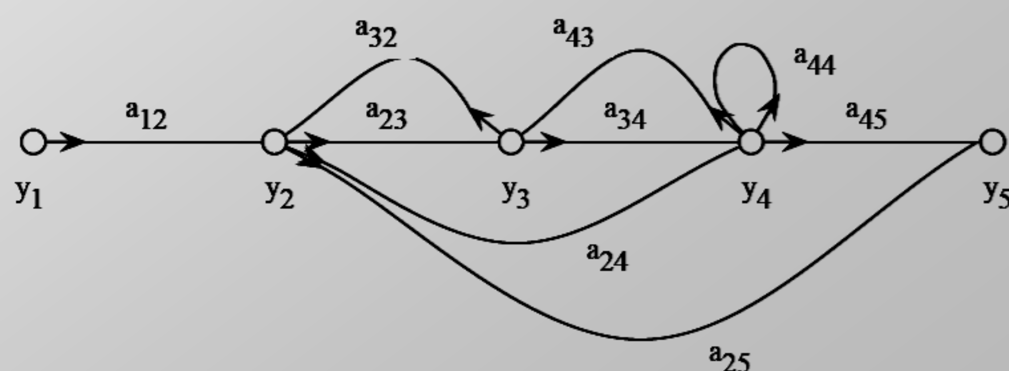
$y_3 = a_{23}y_2 + a_{43}y_4$



$y_4 = a_{24}y_2 + a_{34}y_3 + a_{44}y_4$



$y_5 = a_{25}y_2 + a_{45}y_4$

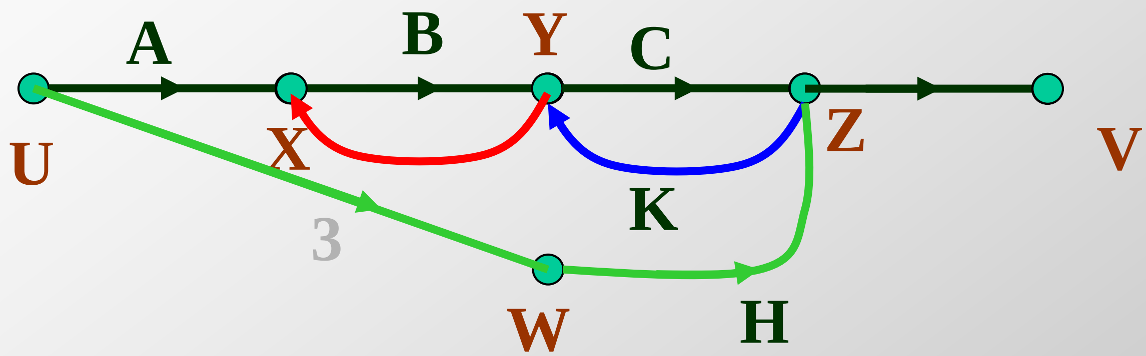


Definitions

- **Input:** (*source*) has only outgoing branches
- **Output:** (*sink*) has only incoming branches
- **Path:** (from node i to node j) has no loops.
- **Forward-path:** path connecting a source to a sink
- **Loop:** A simple graph cycle.
- **Path Gain:** Product of gains on path edges
- **Loop Gain:** Product of gains on loop
- **Non-touching Loops:** Loops that have no vertex in common (and, therefore, no edge.)

Example

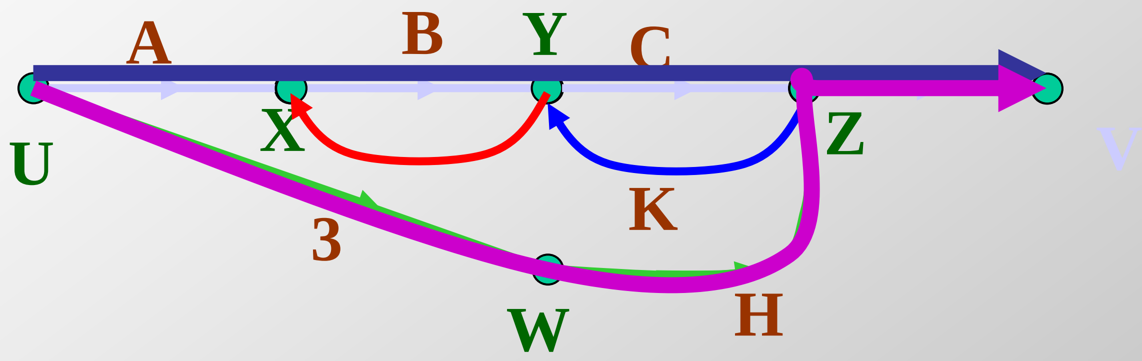
Determine the transfer function between V and U



- The number of forward paths from U to V = ?
- Path Gains ?
- Loops ?
- Determinant ?
- Cofactors ?
- Transfer function ?

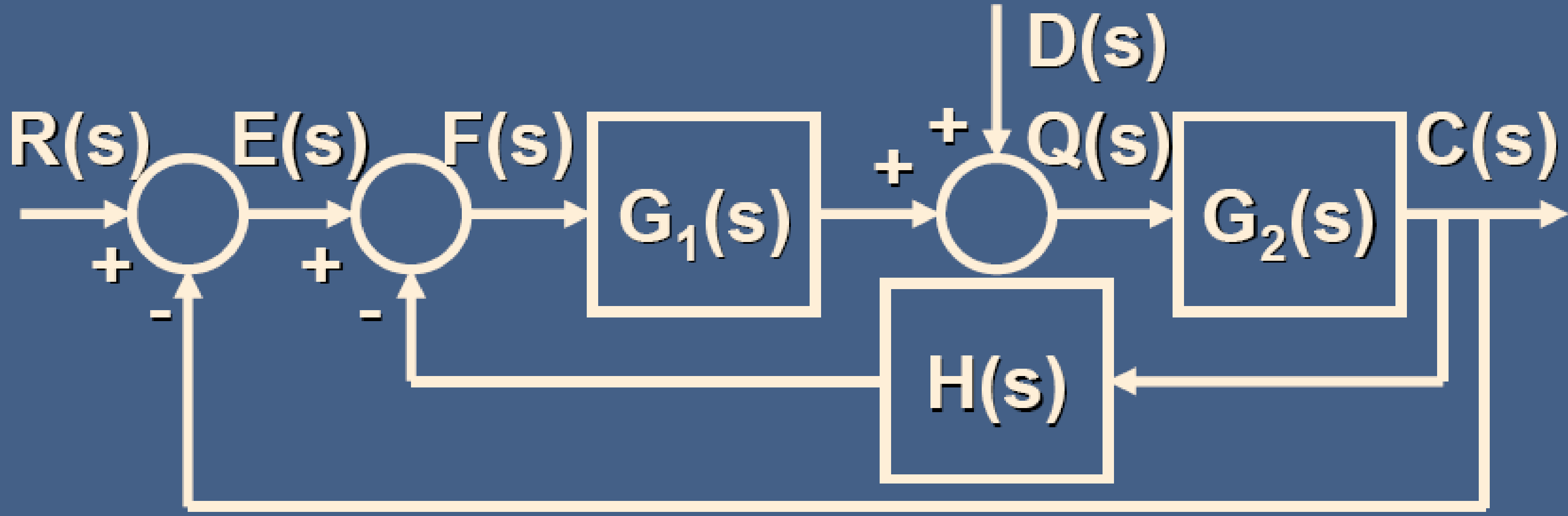
Example

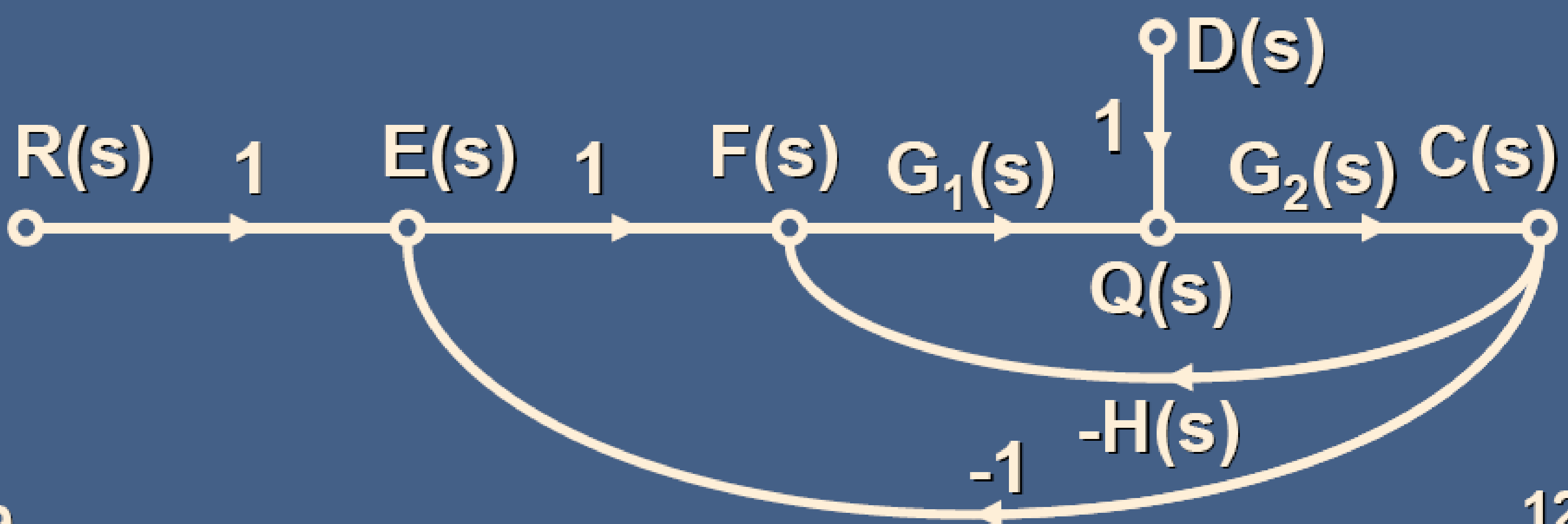
Determine the transfer function between V and U



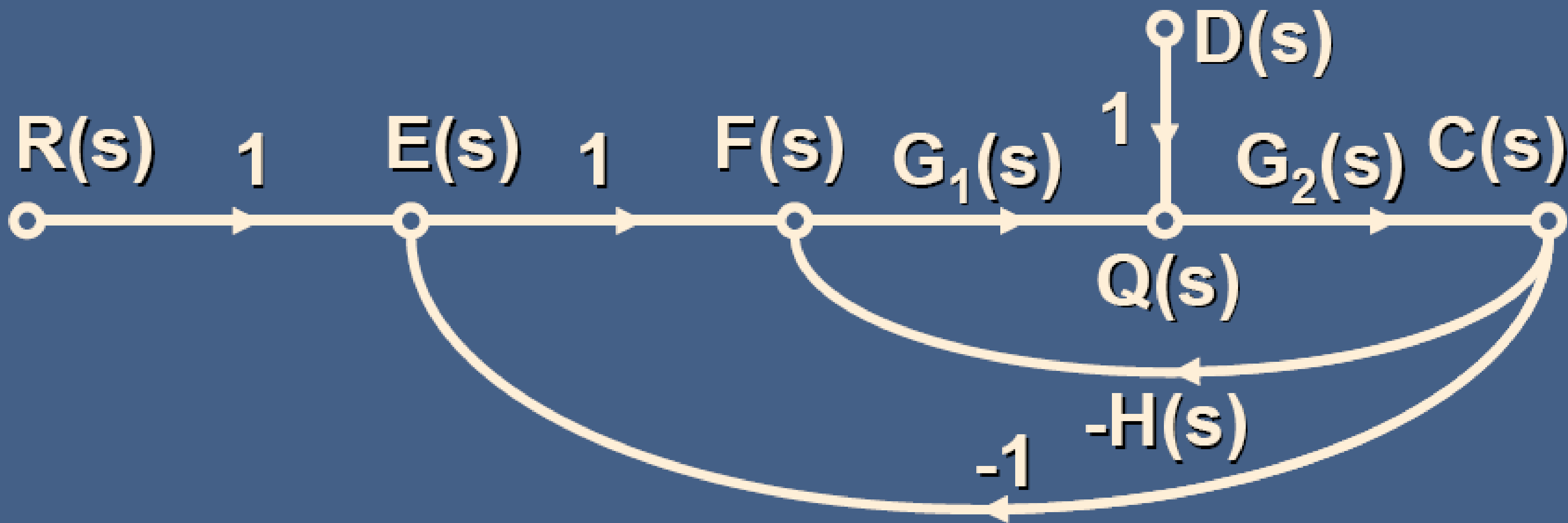
- The number of forward paths from U to V = 2
- Path Gains ABC, 3H
- Loop Gains B, CK
- Transfer function (ABC+3H-3HB)/(1-B-CK)

Signal Flow Graph Example



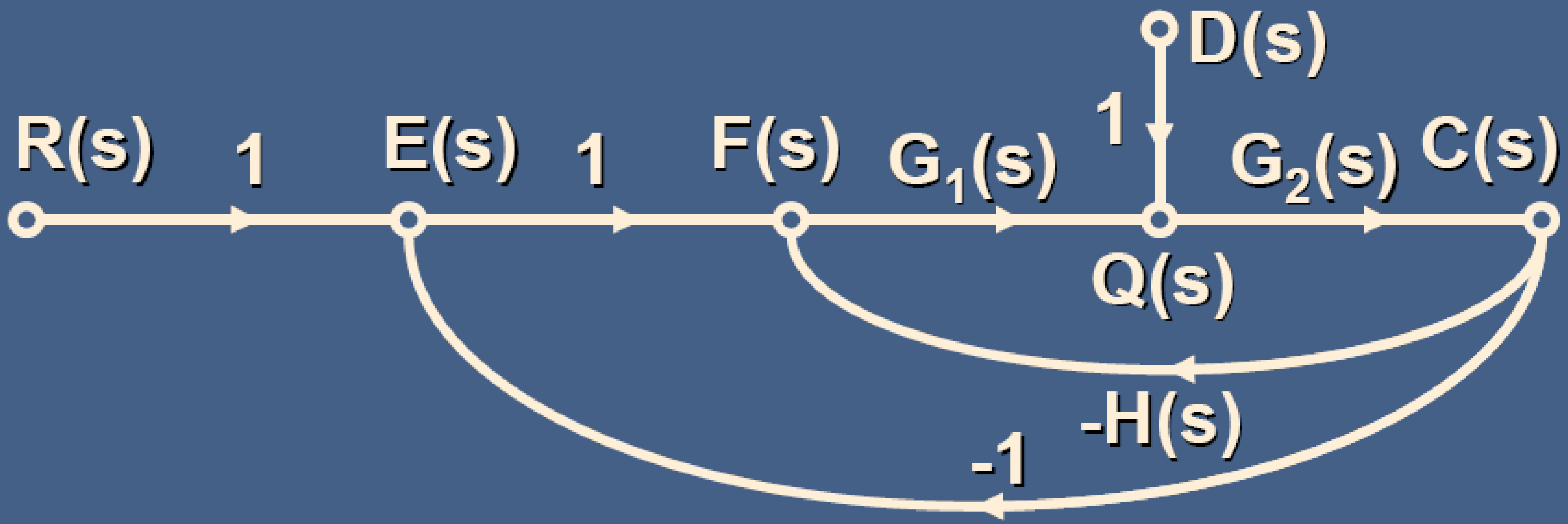


Example of Gain Formula Use



- Assume R(s) = 0, desire to find the transfer function C(s)/D(s).
- There is only one forward path between D(s) and C(s), therefore k = 1.
- There are two loops. They are touching.

Example of Gain Formula Use

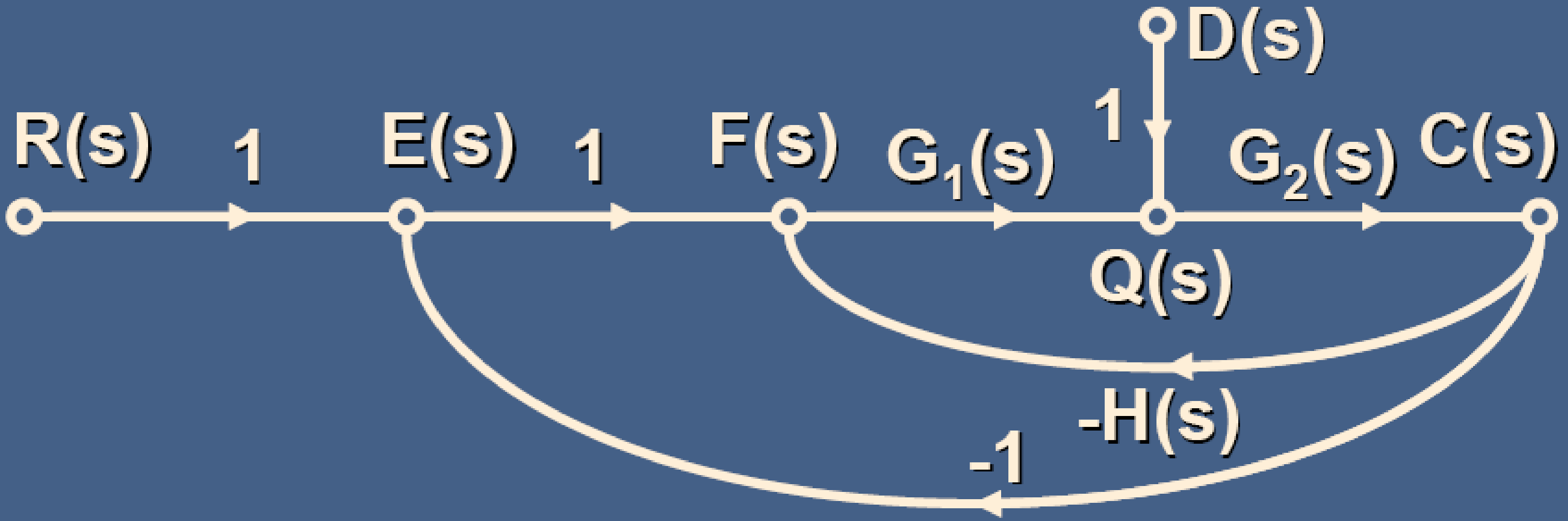


$P_1 = G_2(s)$

$\Delta = 1 - [-G_1(s)G_2(s)H(s) - G_1(s)G_2(s)]$

$\Delta_1 = 1$ (Both loops touch the k^{th} path)

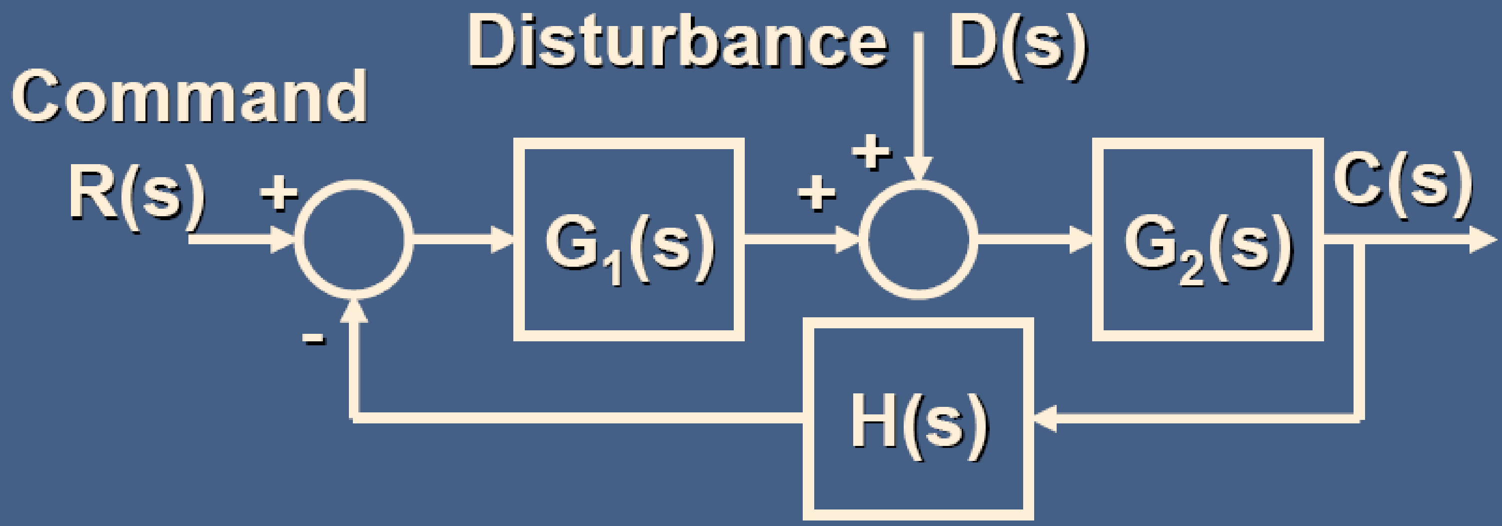
Example of Gain Formula Use



$$\frac{C(s)}{D(s)} = \sum_k \frac{P_k \Delta_k}{\Delta}$$

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

Block Diagram with Disturbance Input

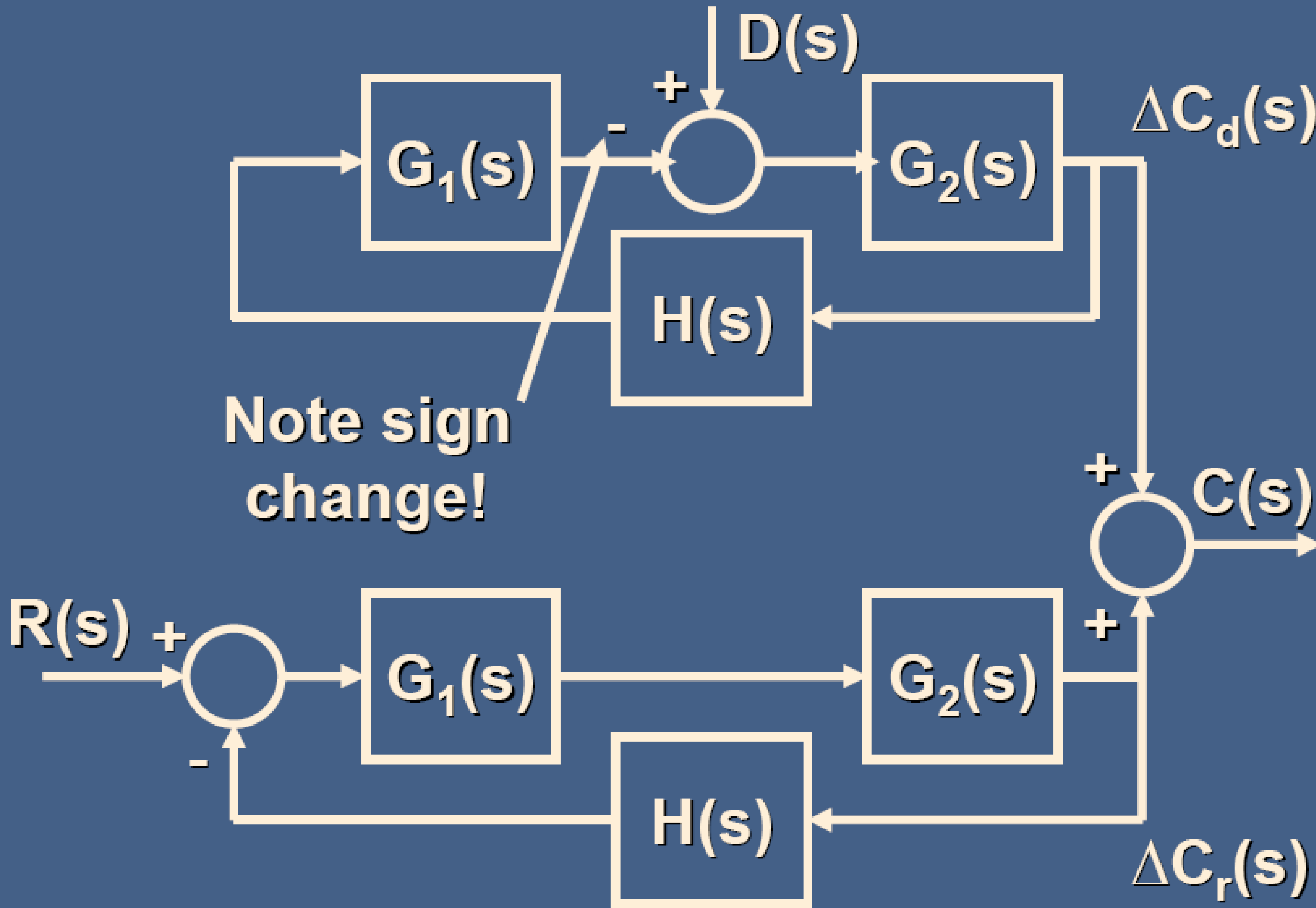


- A disturbance input is an unwanted or unavoidable input signal that affects a system's output. Examples:

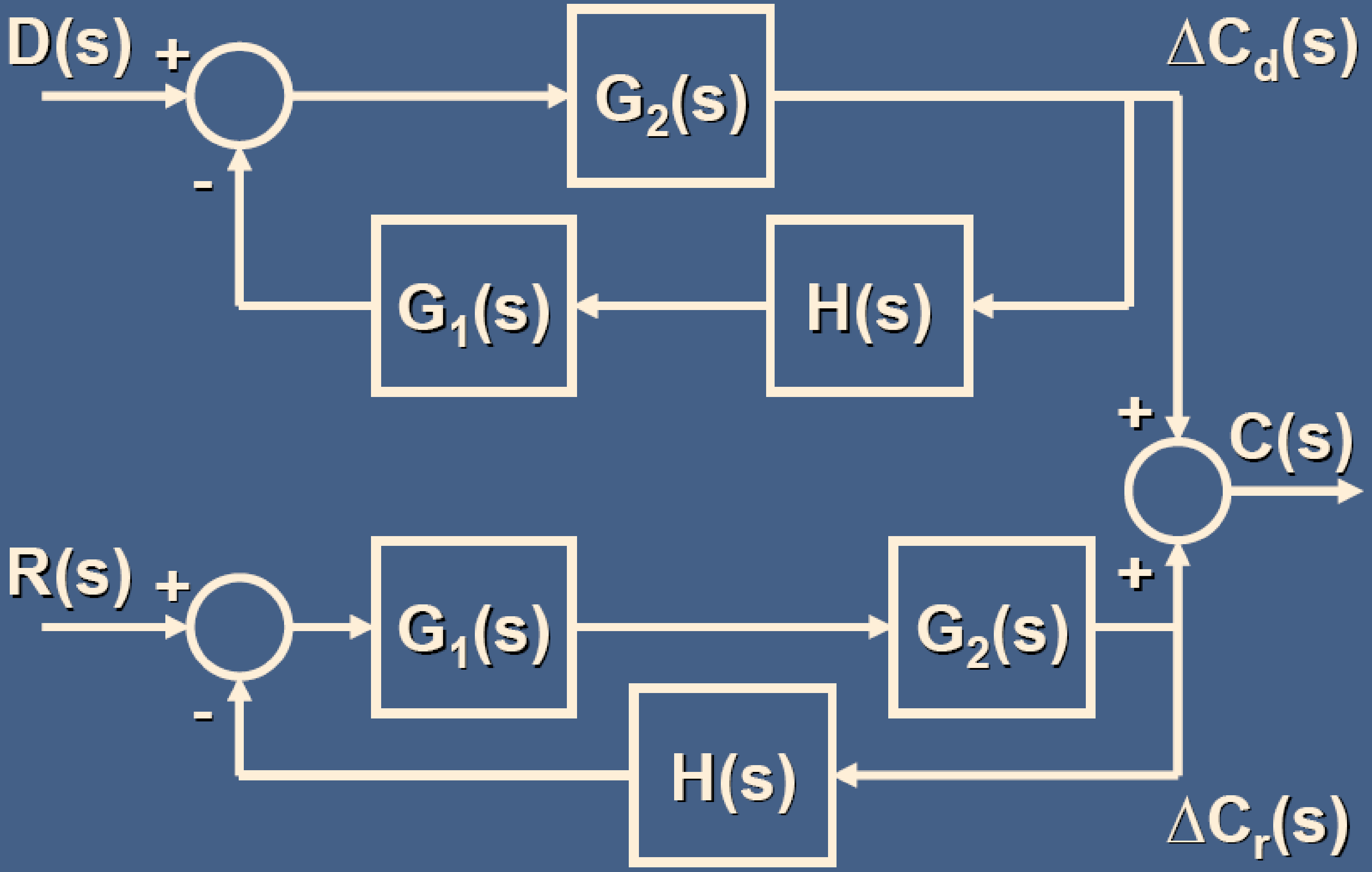
  - load torque in motor control

  - open door in room climate control

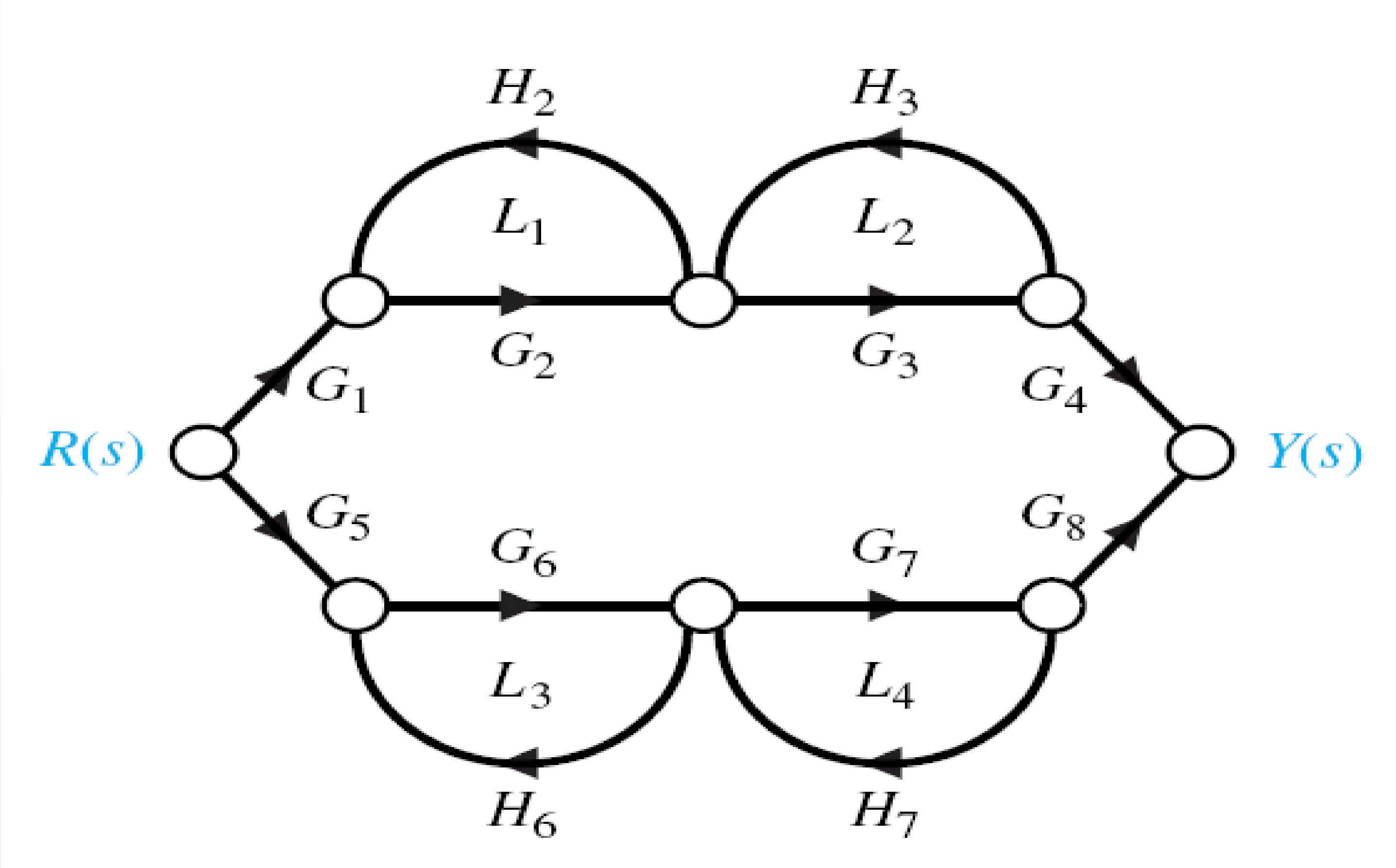
Superposition: $C(s) = \Delta C_r(s) + \Delta C_d(s)$



Disturbance Portion Redrawn



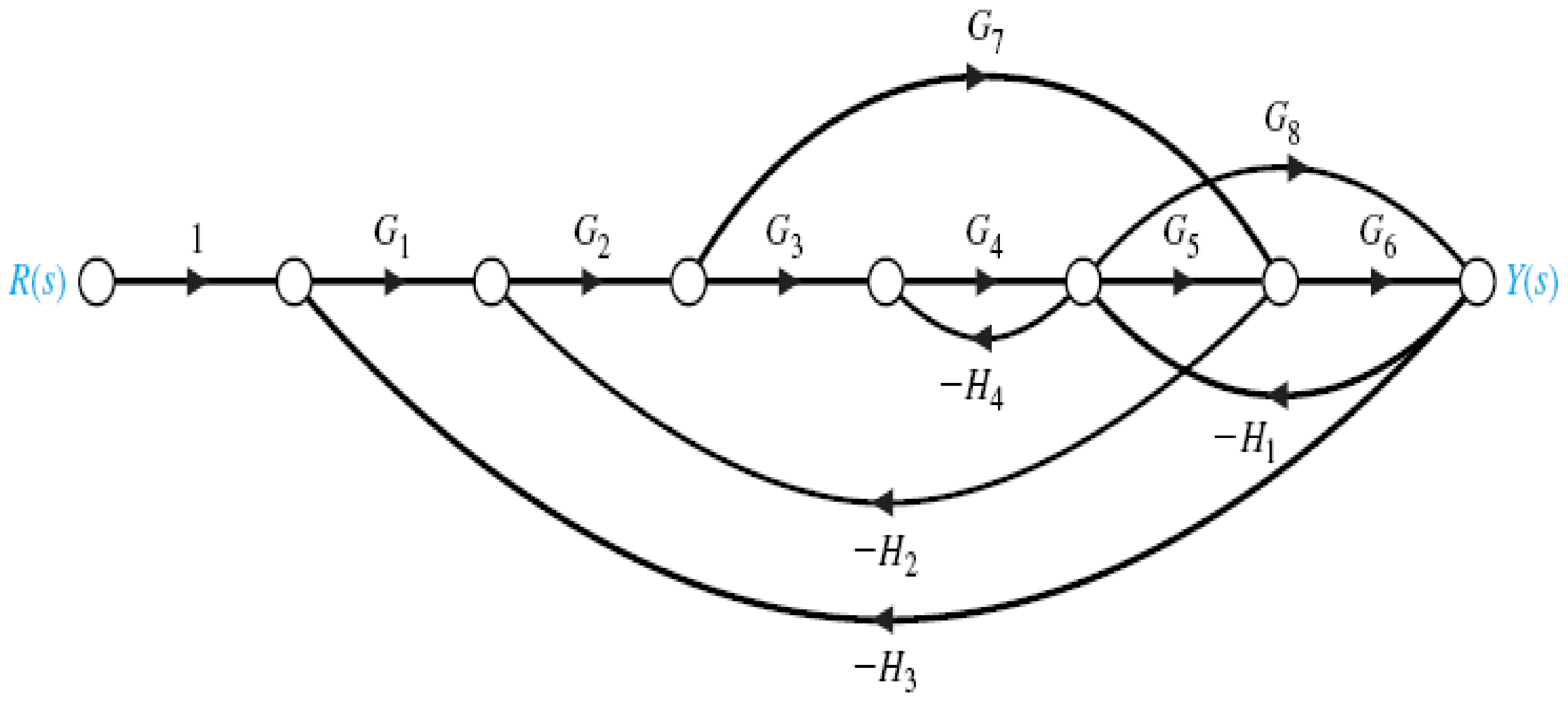
An example



An example:

- Two paths :P1, P2
- Four loops
- $P_1 = G_1G_2G_3G_4$, $P_2 = G_5G_6G_7G_8$
- $L_1=G_2H_2$ $L_2=G_3H_3$ $L_3=G_6H_6$ $L_4=G_7H_7$
- $\Delta = 1 - (L_1+L_2+L_3+L_4)+(L_1L_3+L_1L_4+L_2L_3+L_2L_4)$
- Cofactor for path 1: $\Delta 1 = 1 - (L_3+L_4)$
- Cofactor for path 2: $\Delta 2 = 1-(L_1+L_2)$
- $T(s) = (P_1\Delta_1 + P_2\Delta_2)/\Delta$

Another example



- 3 Paths
- 8 loops

Mason's Gain Rule (1956)

Given an SFG, a source and a sink, N forward paths between them and K loops, the gain (transfer function) between the source-sink pair is

$$T_{ij} = \frac{\sum P_k \Delta_k}{\Delta}$$

P_k is the gain of path k, Δ is the "graph determinant":

$\Delta = 1 - \sum$(all loop gains)

$+ \sum$(products of non-touching-loop gain pairs)

$- \sum$(products of non-touching-loop gain triplets)

$+ \ldots$

$\Delta_k = \Delta$ of the SFG after removal of the k_{th} forward path

Terms for Mason's Gain Formula

- Path: A branch or sequence of branches that can be traversed from one node to another.
- Loop: A closed path, along which no node is met twice, that originates and terminates in the same node.
- Nontouching: Two loops are nontouching if they do not share a common node.
- Gain: Refers, in this case, to the product of transfer functions.

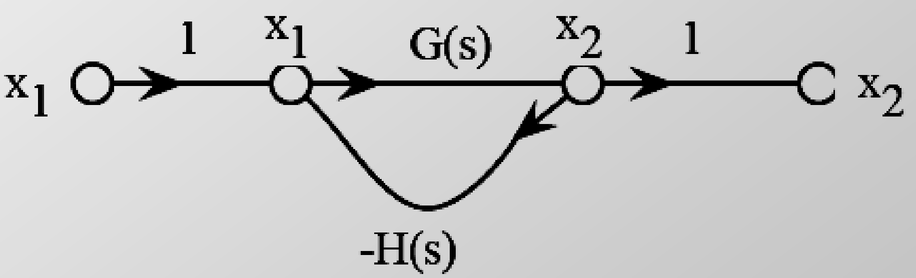
Mason's Gain Formula: $\frac{O(s)}{I(s)} = \sum_k \frac{P_k \Delta_k}{\Delta}$

- P_k = the gain of the k^{th} forward path between I(s) and O(s).
- Δ = 1 - (sum of all individual loop gains)
  + (sum of gain products of all combinations of 2 nontouching loops)
  - (sum of gain products of all combinations of 3 nontouching loops)
  + …
- Δ_k = value of Δ for that part of graph nontouching the k^{th} forward path.

Mason's Rule for Simple Feedback loop

$P_1 = G(s)$

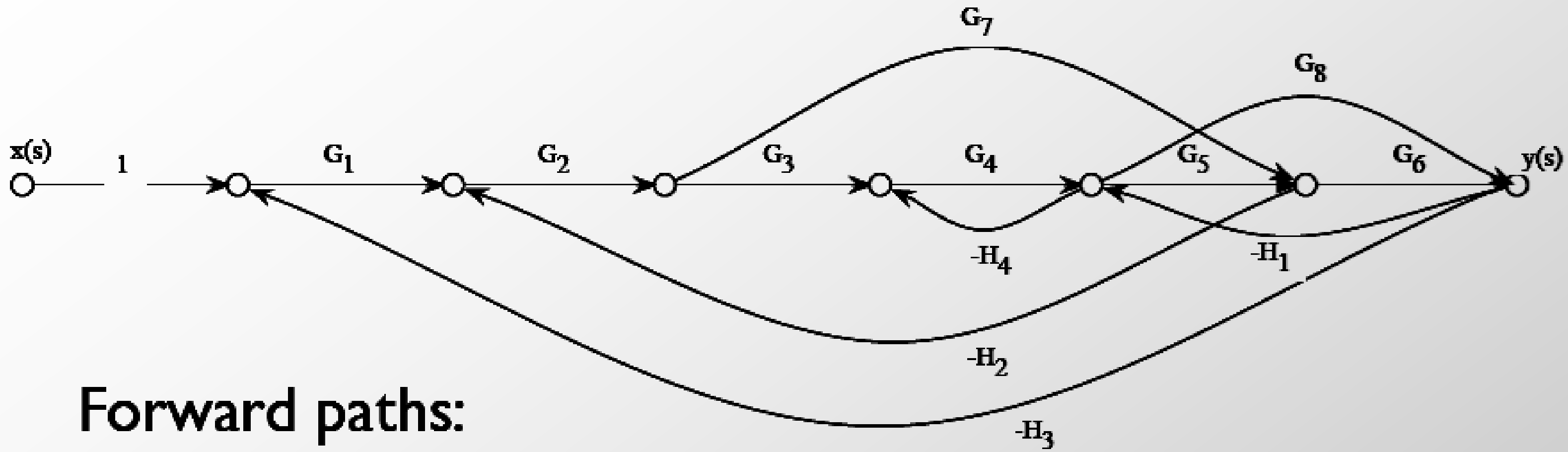
$L_1 = -G(s)H(s)$



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

$\Delta_1 = 1$

$$T(s) = \frac{P_1 \Delta_1}{\Delta} = \frac{G(s)}{\Delta} = \frac{G(s)}{1+G(s)H(s)}$$



Forward paths:

$P_1 = G_1G_2G_3G_4G_5G_6$ $P_2 = G_1G_2G_7G_6$ $P_3 = G_1G_2G_3G_4G_8$

Feedback loops:

$L_1 = -G_2G_3G_4G_5H_2$ $L_2 = -G_5G_6H_1$ $L_3 = -G_8H_1$

$L_4 = -G_7H_2G_2$ $L_5 = -G_4H_4$ $L_6 = -G_1G_2G_3G_4G_5G_6H_3$

$L_7 = -G_1G_2G_7G_6H_3$ $L_8 = -G_1G_2G_3G_4G_8H_3$

Loops $\{3,4\},\{4,5\}$ and $\{5,7\}$ don't touch

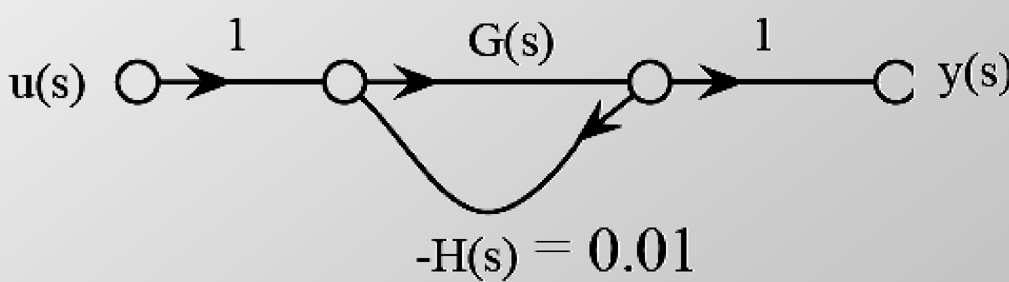
$\Delta = 1-(L_1+L_2+L_3+L_4+L_5+L_6+L_7+L_8)+(L_3L_4+L_4L_5+L_5L_7)$

$\Delta_1 = \Delta_3 = 1$, $\Delta_2 = 1-L_5 = 1 - G_4H_4$

$$T(s) = \frac{y(s)}{x(s)} = \frac{P_1+P_2\Delta_2+P_3}{\Delta}$$

A Feedback Loop Reduces Sensitivity To Plant Variations

y(s)/u(s)=G/(1+GH)



G=10000

y(s)/u(s)=10000/(1+10000*0.01)=99.01

G=20000

y(s)/u(s)=20000/(1+20000*0.01)=99.50