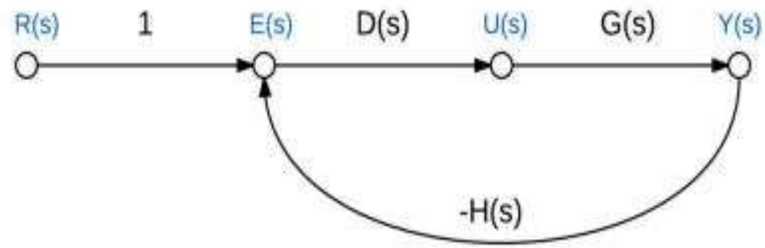
Signal Flow Graphs

- An alternative to block diagrams for graphically describing systems

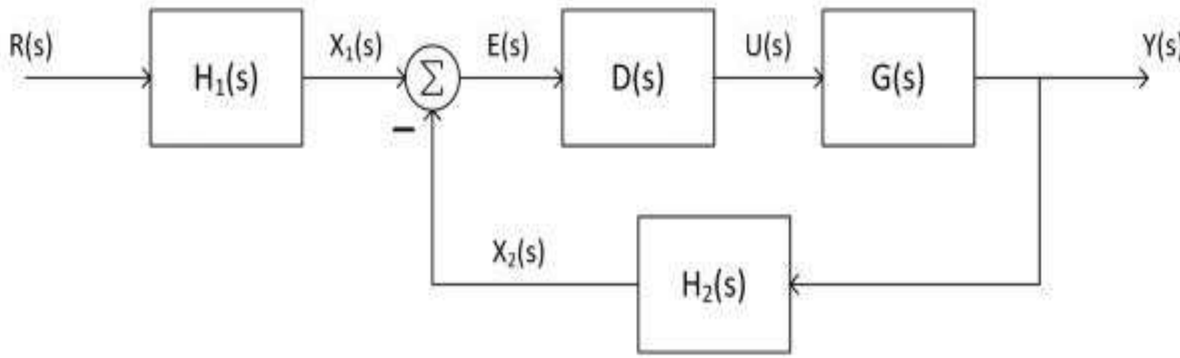


- Signal flow graphs consist of:
  - ***Nodes*** –represent signals
  - ***Branches*** –represent system blocks
- Branches labeled with system transfer functions
- Nodes (sometimes) labeled with signal names
- Arrows indicate signal flow direction
- Implicit summation at nodes
  - Always a positive sum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

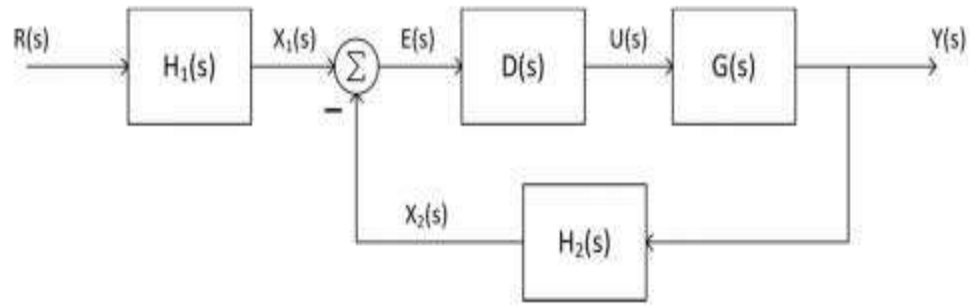
- To convert from a block diagram to a signal flow graph:

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

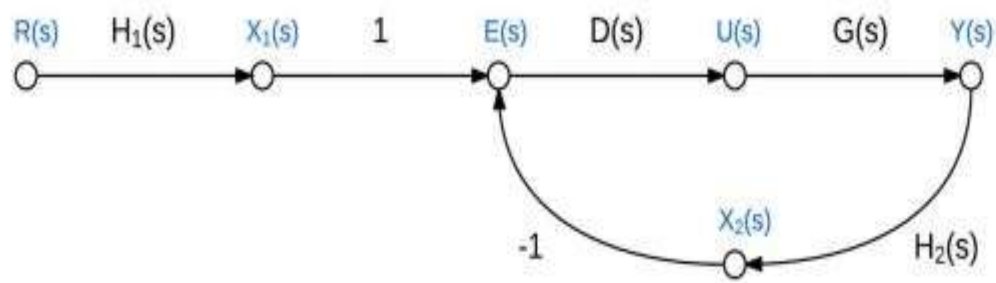
- Convert to a signal flow graph



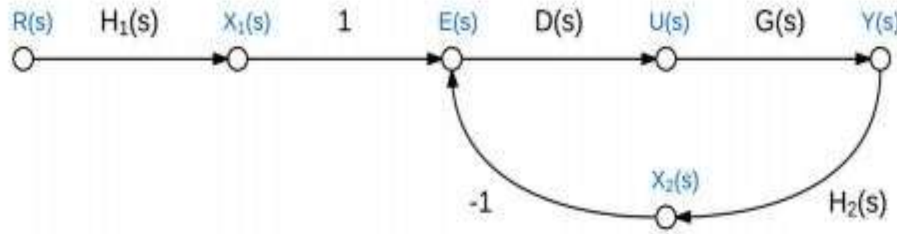
- Label any unlabeled signals
- Place a node for each signal



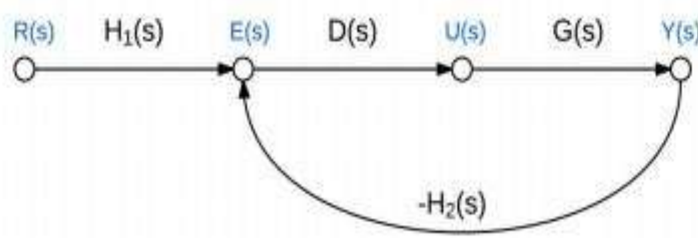
- Connect nodes with branches, each representing a system block



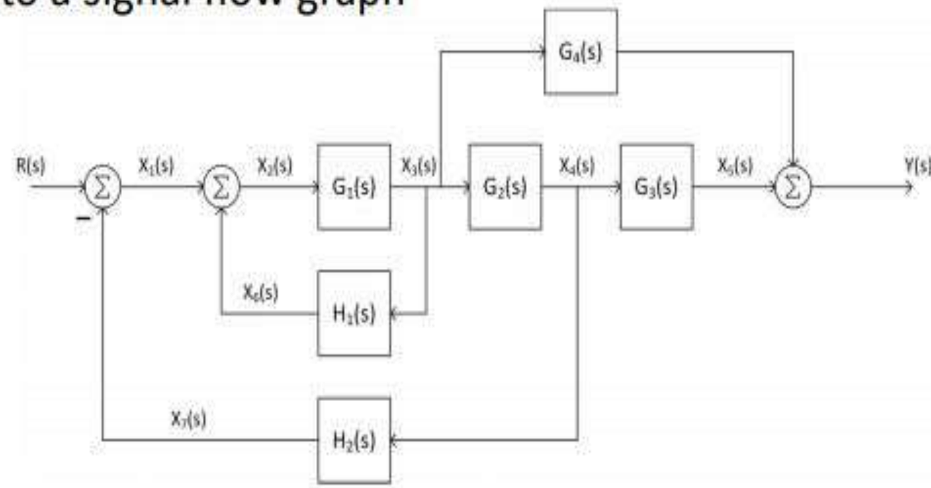
- Note the -1 to provide negative feedback of $X_1(s)$



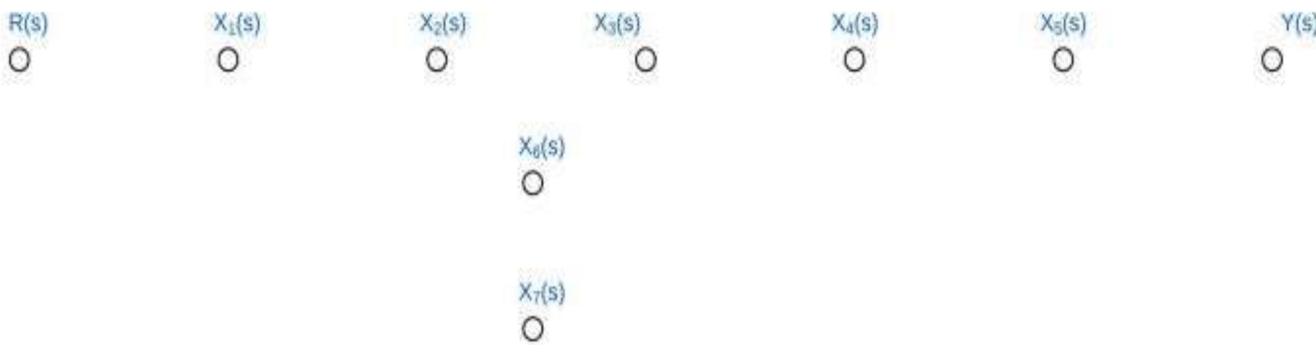
- Nodes with a single input and single output can be eliminated, if desired
  - This makes sense for $X_1(s)$ and $X_2(s)$
  - Leave $U(s)$ to indicate separation between controller and plant

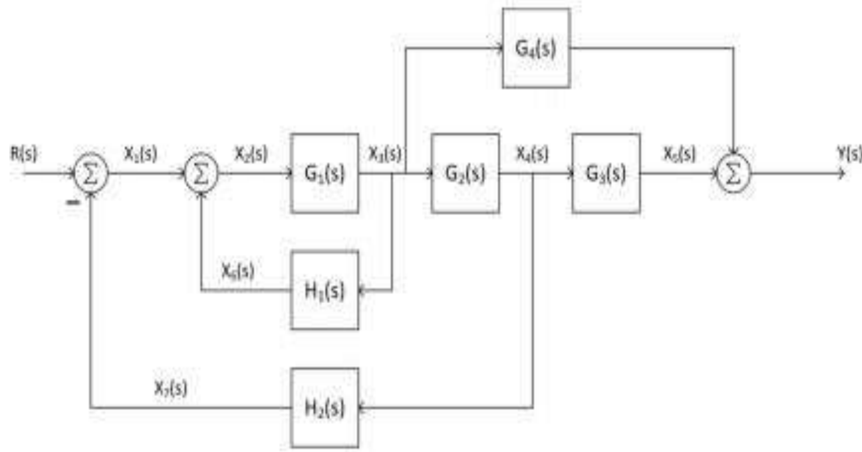


- Revisit the block diagram from earlier
  - Convert to a signal flow graph

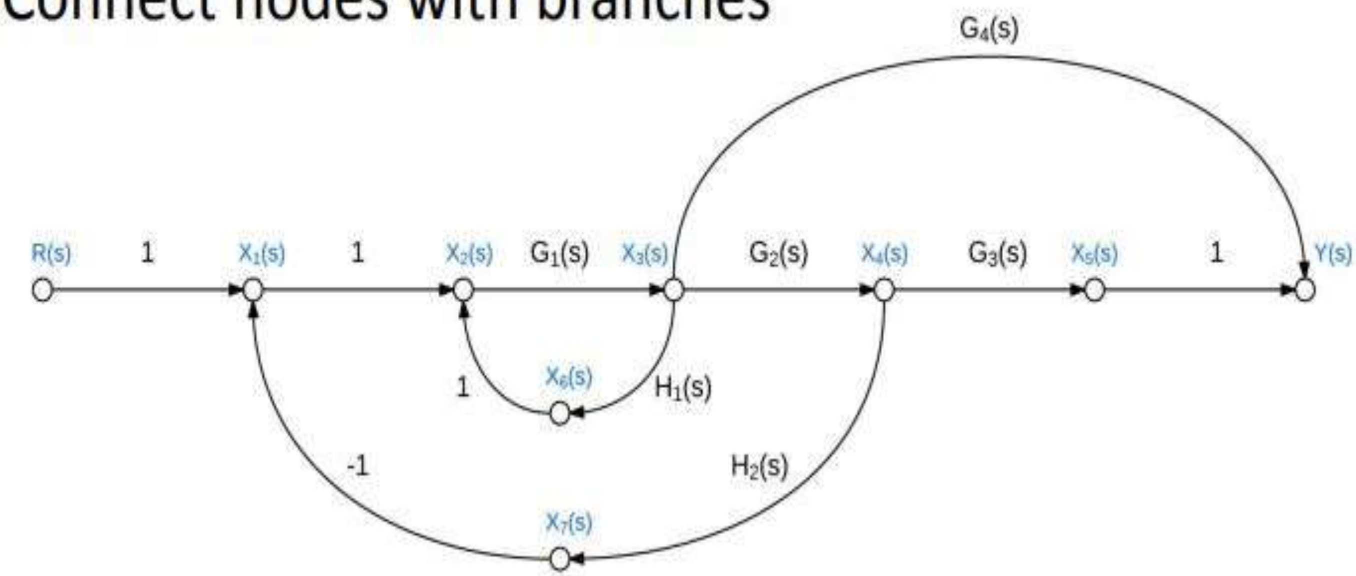


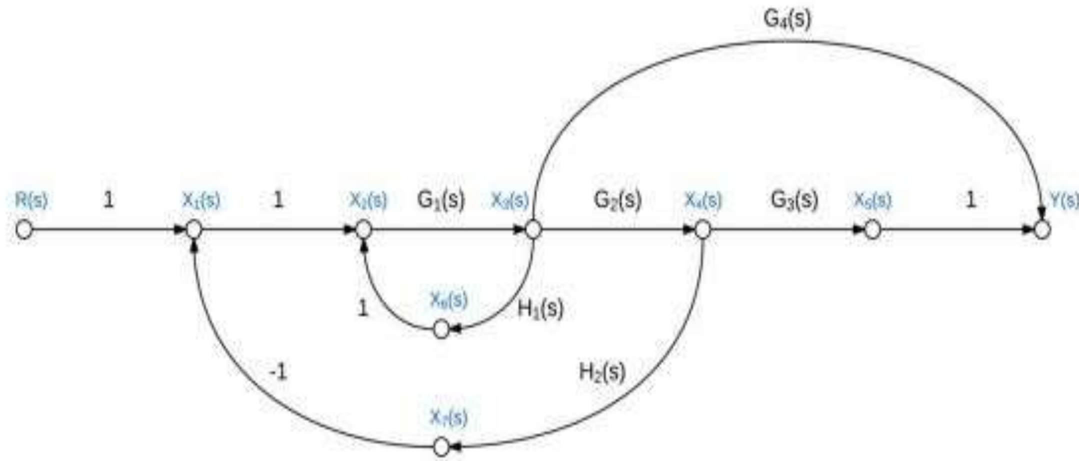
- Label all signals, then place a node for each



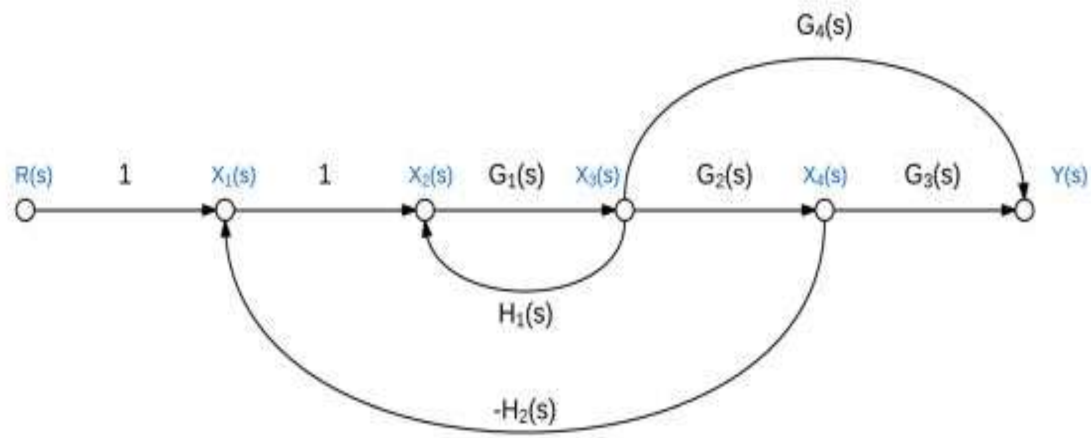


- Connect nodes with branches





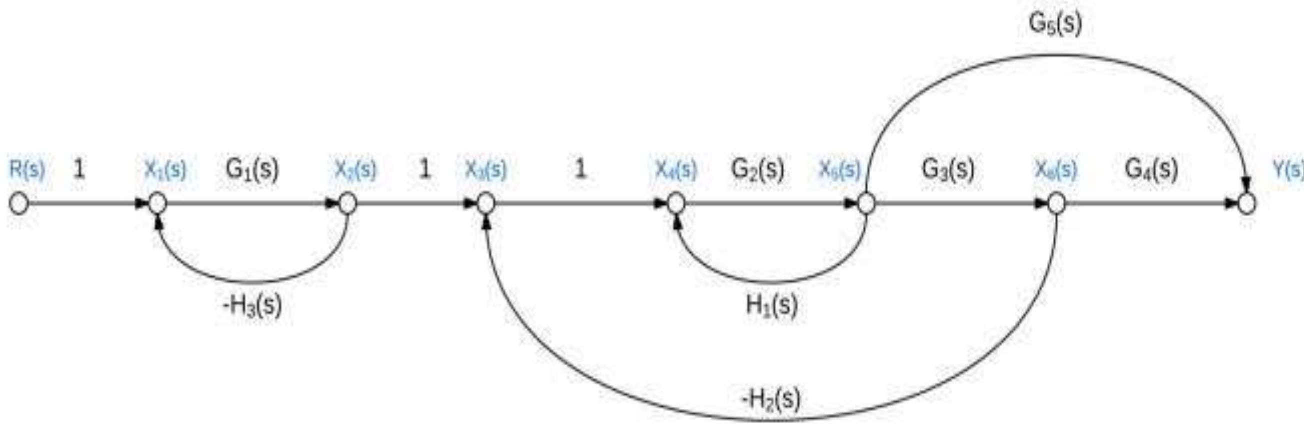
- Simplify – eliminate $X_5(s)$, $X_6(s)$, and $X_7(s)$



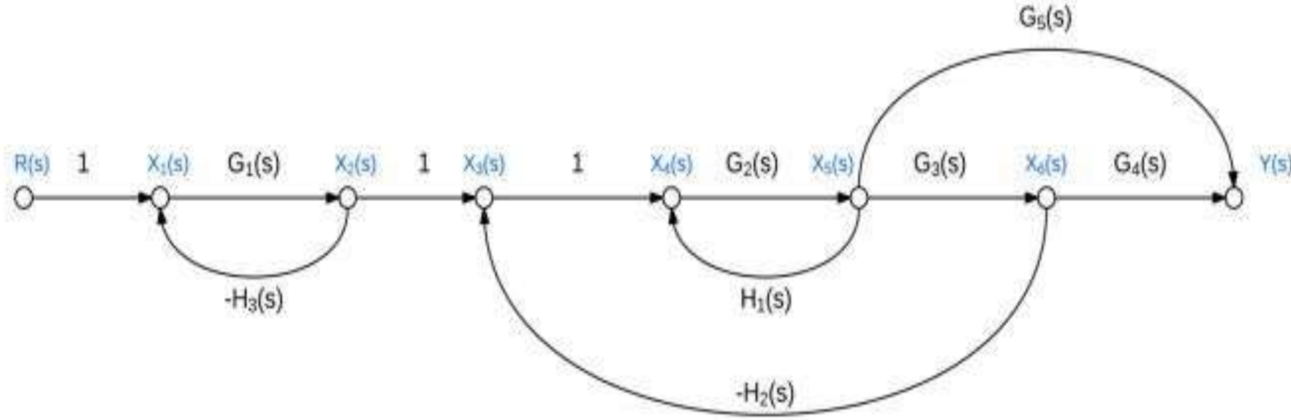
- Signal flow graphs and block diagrams are ***alternative***, though ***equivalent***, tools for graphical representation of interconnected systems
- A generalization (not a rule)
  - ***Signal flow graphs*** – more often used when dealing with ***state-space*** system models
  - ***Block diagrams*** – more often used when dealing with ***transfer function*** system models

Mason’s Rule

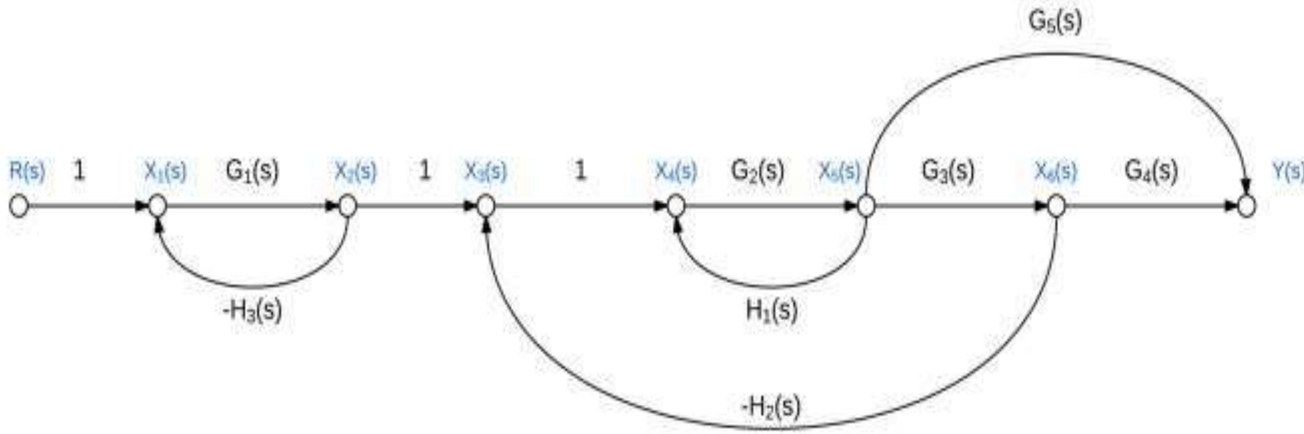
- We've seen how to reduce a complicated block diagram to a single input-to-output transfer function
  - Many successive simplifications
- ***Mason's rule*** provides a formula to calculate the same overall transfer function
  - Single application of the formula
  - Can get complicated
- Before presenting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



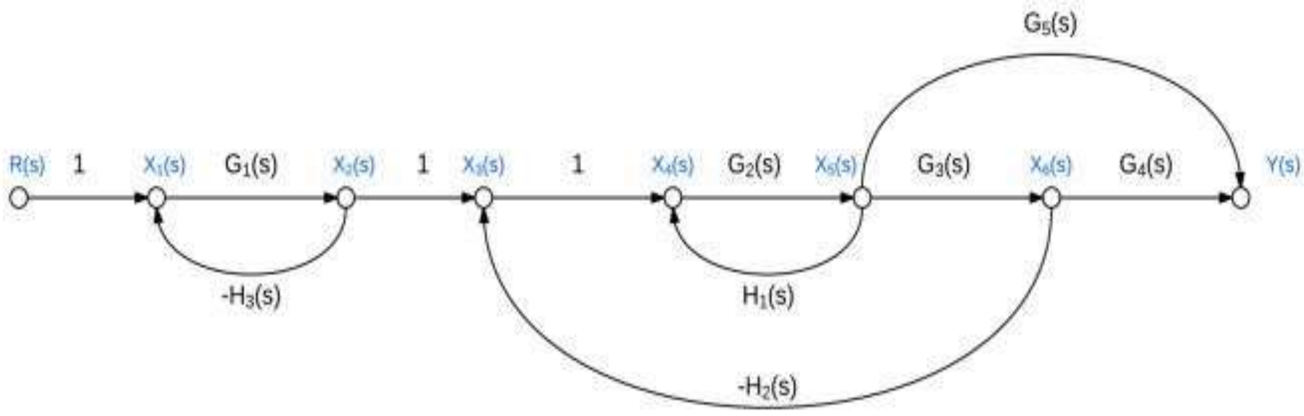
- ***Loop gain*** – total gain (product of individual gains) around any path in the signal flow graph
  - Beginning and ending at the same node
  - Not passing through any node more than once
- Here, there are three loops with the following gains:
  1. $-G_1H_3$
  2. G_2H_1
  3. $-G_2G_3H_2$



- ***Forward path gain*** – 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. $G_1 G_2 G_3 G_4$
  2. $G_1 G_2 G_5$



- ***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*** – 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. $[-G_1H_3] \cdot [G_2H_1]$
  2. $[-G_1H_3] \cdot [-G_2G_3H_2]$

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

where

P = # of forward paths

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

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

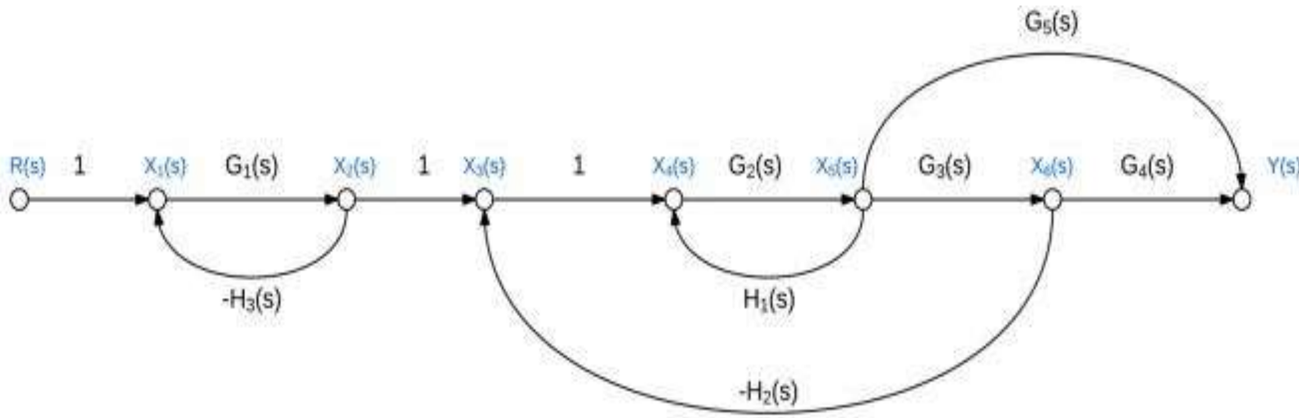
$+\Sigma$(non-touching loop gains taken two-at-a-time)

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

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

$-\Sigma$ ...

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



- # 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):

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

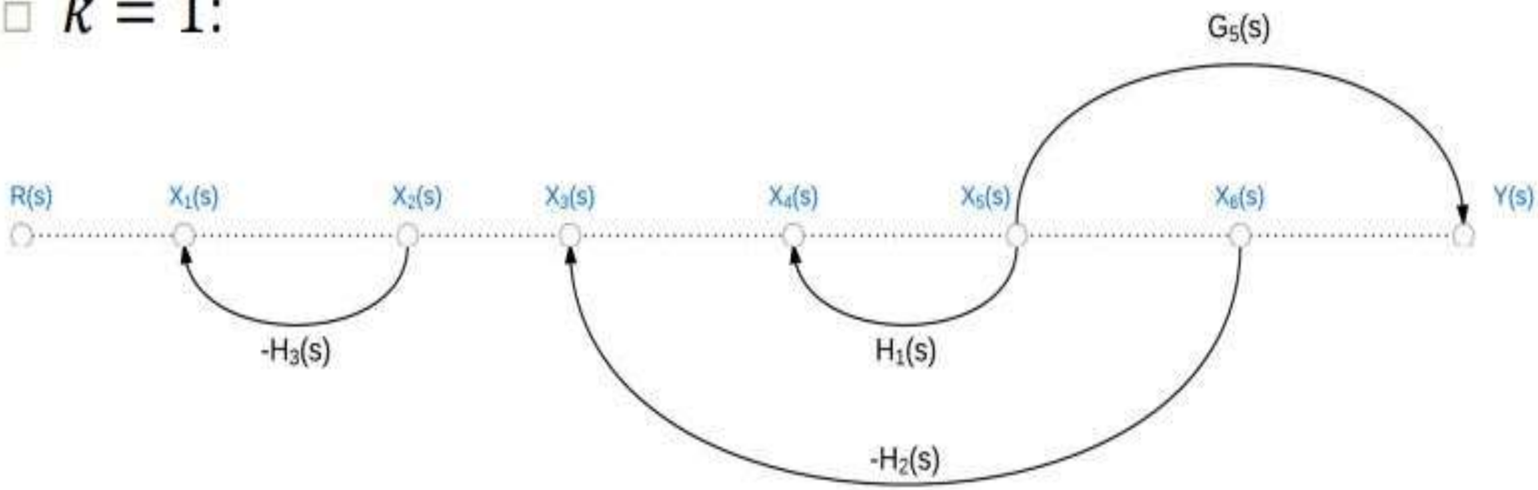
- Σ(NTLGs taken two-at-a-time):

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

- Δ:

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

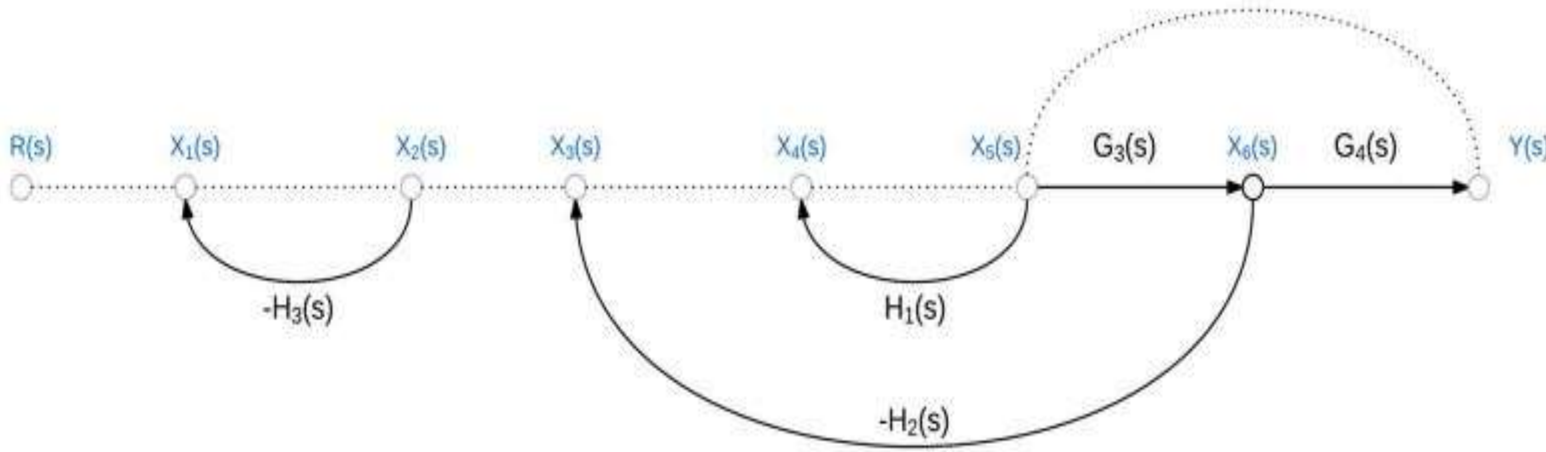
- Simplest way to find Δ_k terms is to calculate Δ with the k^{th} path removed – must remove *nodes* as well
- $k = 1$:



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

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

- $k = 2$:



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

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

- 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:

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

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