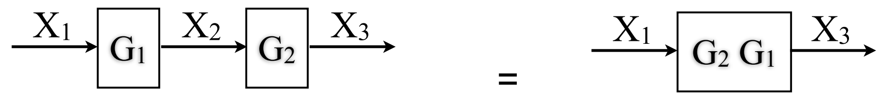
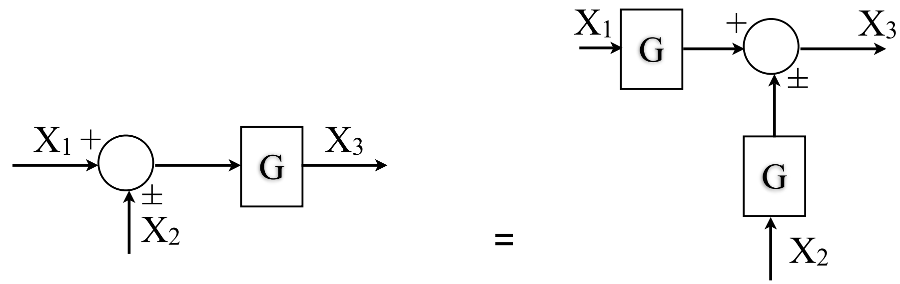
Block Diagram Models, Signal Flow Graphs and Simplification Methods

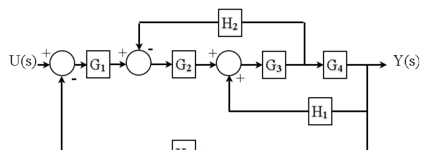
Dynamic Systems and Control
Lavi Shpigelman

Block Diagram Manipulation Rules





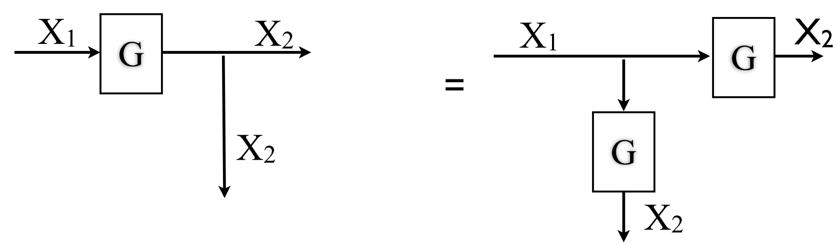
Block Diagram Models

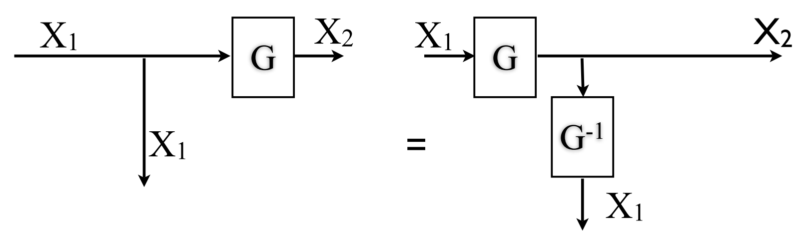


- Visualize input output relations
- Useful in design and realization of (linear) components
- Helps understand flow of information between internal variables.
- Are equivalent to a set of linear algebraic equations

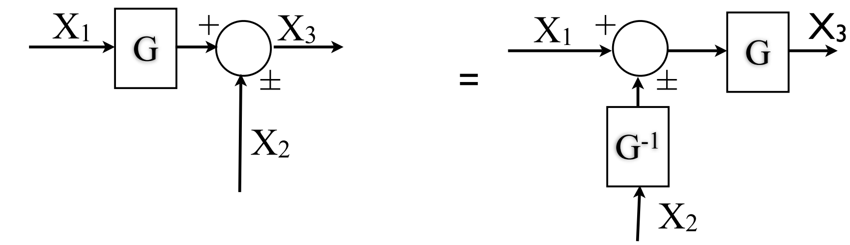
  (of rational functions).
- Mainly relevant where there is a cascade of information flow

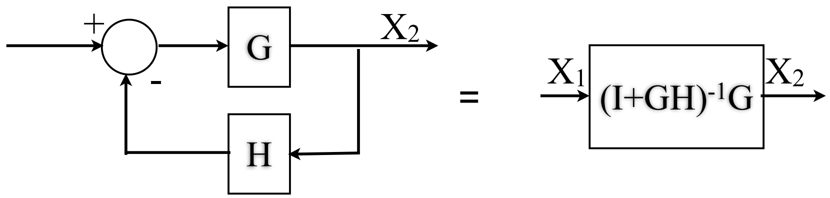
Block Diagram Manipulation Rules





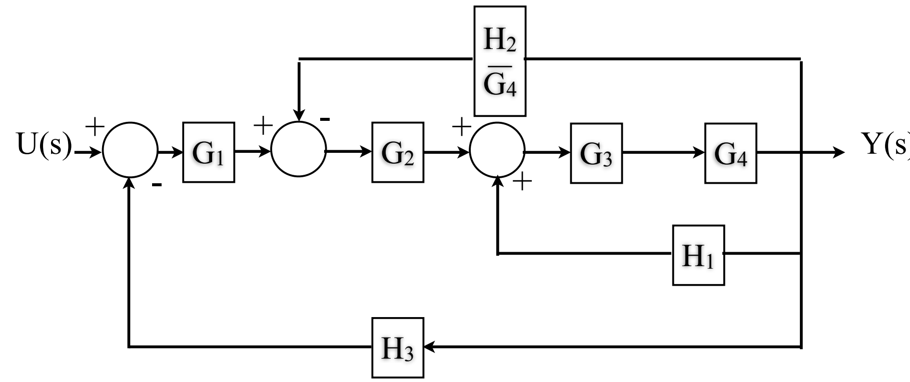
Block Diagram Manipulation Rules





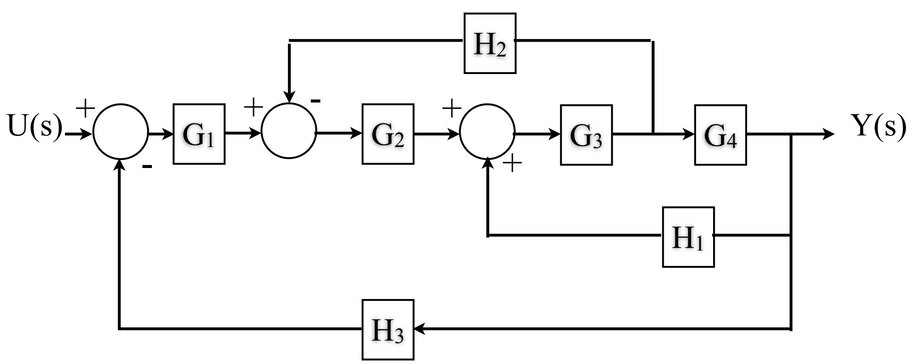
Block Diagram Reduction

(with SISO components)



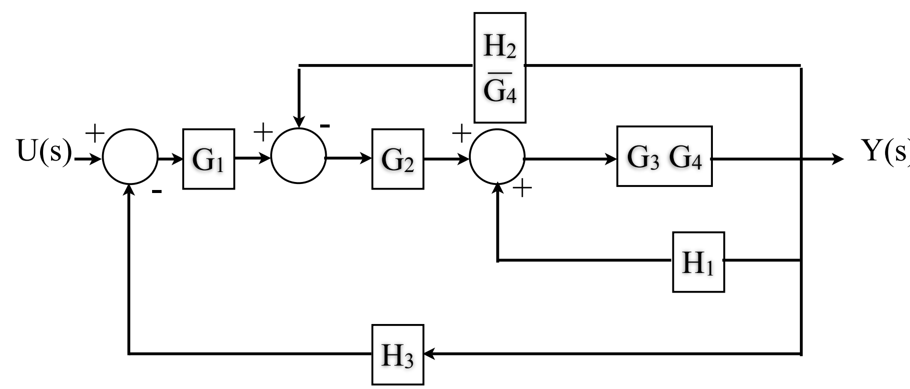
Block Diagram Reduction

(with SISO components)

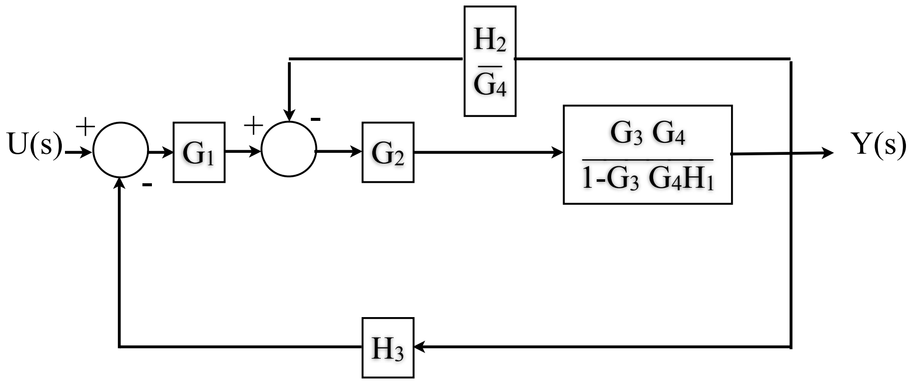


Block Diagram Reduction

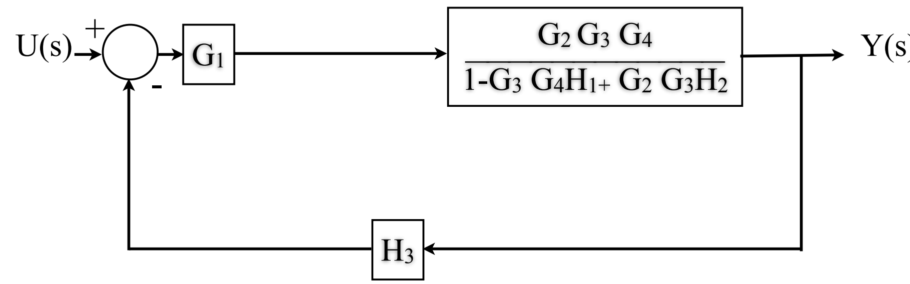
(with SISO components)



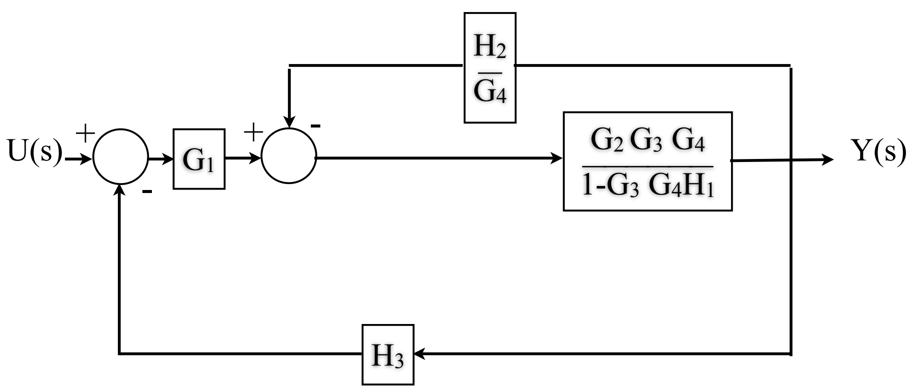
Block Diagram Reduction
(with SISO components)



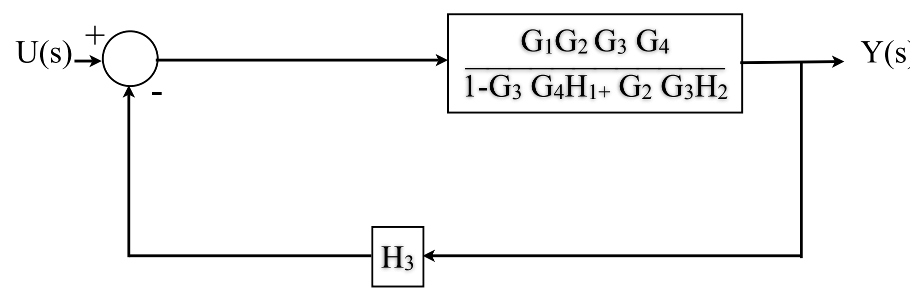
Block Diagram Reduction
(with SISO components)



Block Diagram Reduction
(with SISO components)

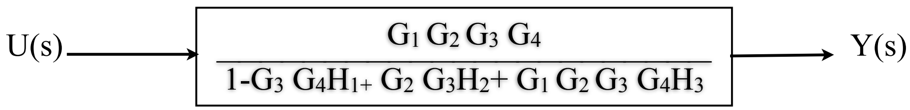


Block Diagram Reduction
(with SISO components)

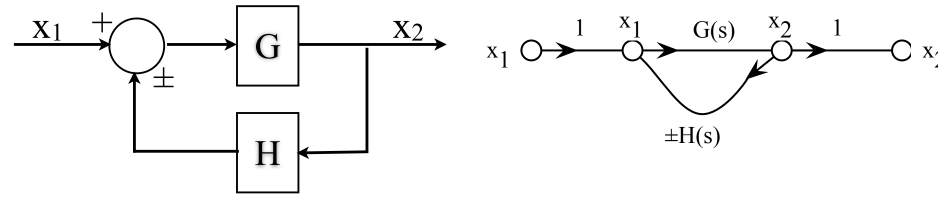


Block Diagram Reduction

(with SISO components)



Block Diagram Vs. SFG



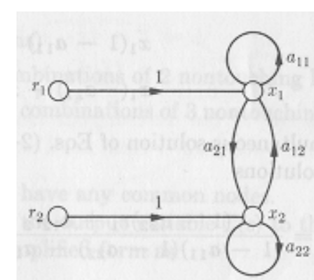
- Blocks ⇒ Edges (aka branches) (representing transfer functions)
- Edges + junctions ⇒ Vertices (aka nodes) (representing variables)

Signal Flow Graphs

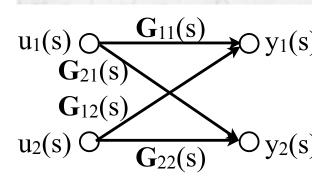
- Alternative to block diagrams
- Do not require iterative reduction to find transfer functions (using Mason's gain rule)
- Can be used to find the transfer function between any two variables (not just the input and output).
- Look familiar to computer scientists (?)

Algebraic Eq representation

- $\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{r}$
  $x_1 = a_{11}x_1 + a_{12}x_2 + r_1$
  $x_2 = a_{21}x_1 + a_{22}x_2 + r_2$



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



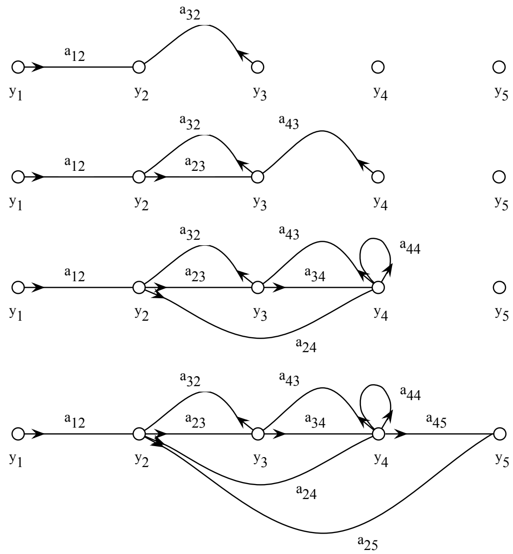
Another SFG Example

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

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

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

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

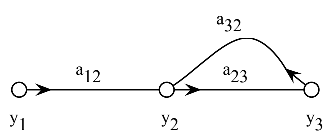


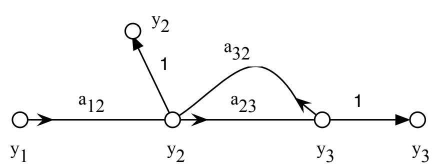
Definitions

- **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.)

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





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)

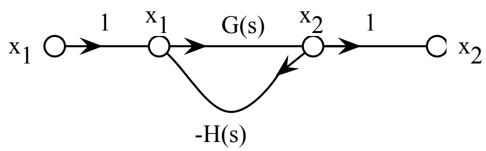
$+ \ldots$

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

Mason's Rule for Simple Feedback loop

$P_1 = G(s)$

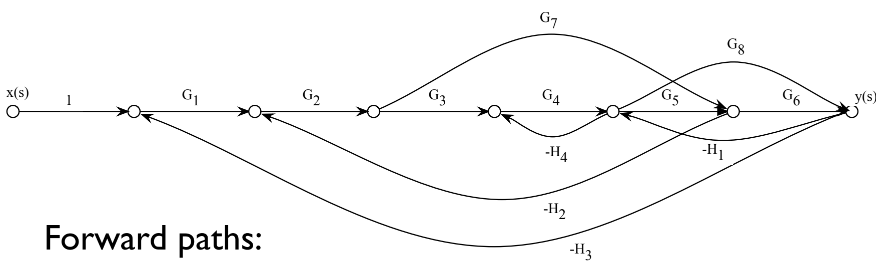
$L_1 = -G(s)H(s)$



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

$\Delta_1 = 1$

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



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

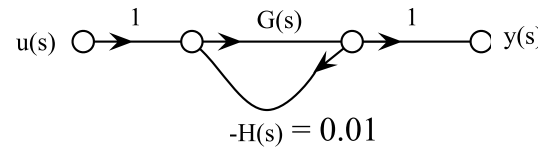
$\Delta = 1-(L_1+L_2+L_3+L_4+L_5+L_6+L_7+L_8)+(L_3L_4+L_4L_5+L_5L_7)$

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

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

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