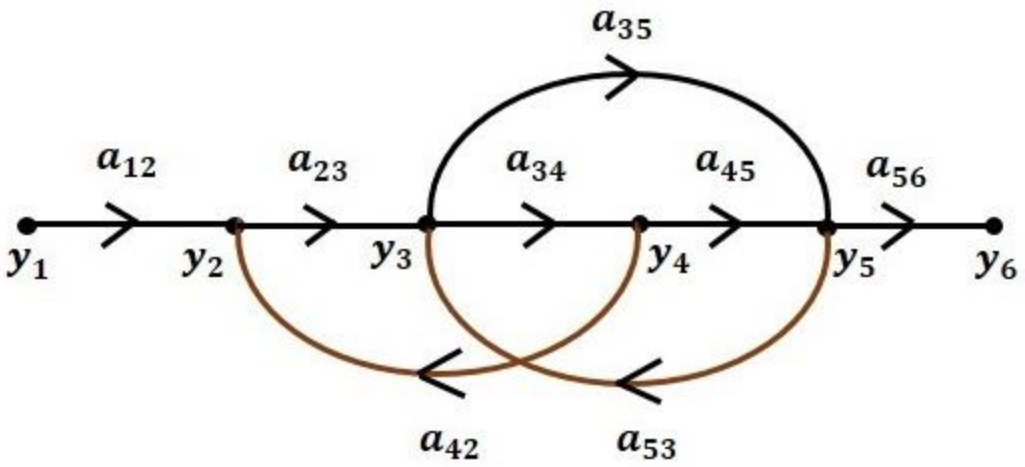
Signal Flow Graph



Names : Aya Ashraf Saber Mohamed (02)
Merit Vector (68)

Table of content

1) Problem Statement:

Signal flow graph is a graphical representation of algebraic equations. In this chapter, let us discuss the basic concepts related signal flow graph and also learn how to draw signal flow graphs.

- **Path.** A path is a continuous set of branches traversed in the direction indicated by the branch arrows.
    - **Open path.** If no node is re-visited, the path is open.
    - **Forward path.** A path from an input node (source) to an output node (sink) that does not re-visit any node.
- **Path gain**: the product of the gains of all the branches in the path.
- **Loop.** A closed path. (it originates and ends on the same node, and no node is touched more than once).
- **Loop gain**: the product of the gains of all the branches in the loop.
- **Non-touching loops.** Non-touching loops have no common nodes.
- **Graph reduction.** Removal of one or more nodes from a graph using graph transformations.
    - **Residual node.** In any contemplated process of graph reduction, the nodes to be retained in the new graph are called residual nodes.
- **Splitting a node.** Splitting a node corresponds to splitting a node into two half nodes, one being a sink and the other a source.
- **Index**: The index of a graph is the minimum number of nodes which have to be split in order to remove all the loops in a graph.
    - **Index node.** The nodes that are split to determine the index of a graph are called *index* nodes, and in general they are not unique
    - We know,
    - $\Delta=1-(\text{sumofallindividualloopgains})$ $\Delta=1-(\text{sumofallindividualloopgains})$
    - $+(\text{sumofgainproductsofallpossibletwonontouchingloops})$ $+(\text{sumofgainproductsofallpossibletwonontouchingloops})$
        - $-(\text{sumofgainproductsofallpossiblethreenontouchingloops})+..$
    -

2) Main Features of the program:

- Our program take the number of the node at first then enable the user to enter the branches between node and their weight.
- Then the user press solve after entering all branches to get overall transfer function .

- We display all analysis calculation to user
    - All Loops in Graph
    - All Forward paths
    - All loops gain
    - All Samll delta to every forward path
    - All non touching Loops.
- The user can return to the main window to slove new problem

3) Data Structure.

- Array List : to store all loops and forward paths
- Map : to use the key of path name and value
- LinkedList : to store all loops and forward paths

4) Main modules.

1) Mode
    a) Analysis
        i) ForwardPathsFinder
        ii) GeneralDelta
        iii) LoopsFinder

iv) Mason
v) NonTouchingLoopsFinder
vi) PathDelta

b) Design
i) Edge
ii) Loop
iii) Forward path
iv) Single flow Graph
v) Node

2) View
a) Start window
i) Take the number of nodes and start new window
b) Graph Representation
i) Take all edges from the use and built the graph
ii) Evalute the tranfster function
c) Result Window
i) Display all analysis to get overall transfer function.
ii) Return to start window.

3) Controller
a) DrawingSignalFlowGraph
i) Tranfer data from model to view

-Design flowChart :

Edge :

Node :

5) Algorithms used :

- Depth First Search :
  We use DFS in Loop finding and to find all forward paths
- And other algorithm will show in flowChart.

Delta :

Non Touching Loops :

General Delta:

Delta for Path :

6) Sample runs.

Test number 7 :

Cases we handle :

7) Simple user guide: