

# Signal Flow Graph Simulator Mason's Formula Evaluator *Report*

*Karim Waguih El Azzouni - 51*

*Mohamed AbdelRahman El Fekki - 60*

- Problem Statement:

Given:

Signal flow graph representation of the system. Assume that total number of nodes and numeric branches gains are given.

Required:

1. Graphical interface.
2. Draw the signal flow graph showing nodes, branches, gains, ...
3. Listing all forward paths, individual loops, all combination of n non-touching loops.
4. The values of  $\Delta_1, \Delta_2, \dots, \Delta_m$  where m is number of forward paths.
5. Overall system transfer function.

- Introduction

Basic graph data structure was used in this application to represent the signal flow graph. We implemented our own Graph class along with its basic methods (found in *graphElements* package). Moreover, basic graph traversal algorithms were used to retrieve different requirements from the given signal flow graph; such as Depth First Search.

A class FindPaths was constructed to manipulate the graph and deduce all the forward paths, loops, corresponding non-touching loops and overall non-touching loops; those requirements were then used in CalculateMasons class to evaluate the given graph's Mason's formula and transfer function.

Package *main* contains two classes; GUI and DrawingArea. Those classes are mainly responsible for building the graphical user interface to interact with the user. Elements from Java Swing library was used in this application.

- Main Features

- Simulation of Signal Flow Graphs
- Construction of Signal Flow Graphs
- Calculation of Overall Transfer Function of an SFG using Mason's Formula

- Bonus Features

- Supports self loops.
- Supports editing the graph and recalculating the new Overall Transfer Function.
- Select any forward path to view its corresponding nontouching loops.
- Choose any two nodes in the graph as input and output nodes.

- Data Structures and Algorithms

Mainly graphs and linked lists, along with basic graph traversal algorithms (DFS).

- User guide

- In this application, the window is divided into three panels; The Control Panel (On the left with a grey background), The Drawing Panel (In the center with a white background) and the Data Panel (On the bottom with four lists).
- The "Calculate!" button is only enabled after the user is done creating the SFG and choosing its input and output nodes.
- Please make sure to keep an eye on the space below the button that says "Calculate!". Whenever you click a button, button description will be printed there to guide you through the process.
- To add a node, please press the button that says "Add Node", and press on any location on the drawing panel. Note that the node are named alphabetically.

- ➔ To remove a node, please press the button that says “Remove Node”, and press on any node in the drawing panel.
- ➔ To add an edge, please choose the type of edge you want to add (Straight or Curved), then press the button that says “Add Edge”, select the source node from the drawing panel, select the target node from the drawing panel, then input the required weight. If a self-loop is required, please select the same node for the source and the target nodes. The default value of any edge is “0.0”. Thus, if the user doesn’t type a weight for the edge, it’s weight will be set with the default weight.
- ➔ To remove an edge, please press the button that says “Remove Edge”, and select the desired edge from the drawing panel.
- ➔ After you are done from creating your graph, and whenever you’re ready to view its data, please press the button that says “Set Bounds”, choose the input and output nodes from the drawing panel, then press the button that says “Calculate!”.
- ➔ After viewing the data, you can always edit the graph again and recalculate the overall transfer function for the new graph. Simply, start adding/removing components in the drawing panel, set new bounds for the edited graph and click “Calculate” to view data about your new graph.

- Screenshots

