

Alexandria University,  
Faculty of engineering,  
Computer and systems engineering department,  
Linear control systems.

## **Signal flow graph assignment**

### **Team members**

- Mohamed Salah Osman. (41)
- Mohamed Mohamed Abdlhakem.(43)

## **Problem statement:**

Given a signal flow graph representation of the system. Assume that total number of nodes and numeric branches gains are given. The program provide the forward paths, individual loops and all combinations of non-touching loops and overall transfer function.

## **Main features:**

- The program provides a web page with a friendly user interface which allows the user to draw the signal flow graph, add nodes and edges with its gain.
- User has the ability to modify edges and nodes by adding or deleting.
- User can move nodes and edges in a friendly way and reshape the graph in an acceptable form.
- The program provides the steps of solving the signal flow graph which includes all forward paths, loops, non-touching loops combinations, gains and deltas corresponding to each loop or path and overall system transfer function.
- User can choose any two nodes in signal flow graph to solve them and get the corresponding data to them.

## **Used Data structures:**

- Stack: to implement depth first search to get all forward paths and loops.
- List of nodes to store any forward path or loop.
- VIS.js dataset to store all edges and nodes that the user has drawn.
- List of list of objects to store all non-touching loops and forward paths.

## Main modules:

### 1. Solver.js:

It is the module corresponding to the returning all forward paths, loops, non-touching loops combinations, gains and deltas corresponding to each loop or path and overall system transfer function.

### 2. Drawer.js:

It is the module corresponding to drawing the nodes, the edges and sending vis.js dataset which contains the data corresponding to the drawn graph to the solver.js module which in role returns the solution.

GitHub repository: <https://github.com/muhammads97/SigFG>

## Algorithms used:

- Depth first search algorithm to find all forward paths and loops.

## Sample runs:





## User guide:

- Firstly, the user can draw all the nodes and connect them with edges and append a gain to the edge.
- Secondly, the user fire solve button to print a well-organized solution with steps.
- The user can move the edges and nodes by selecting and dragging the mouse.