Control Systems

Programming Assignment

Signal Flow Graphs & Routh Stability Criterion

Team members:

شهد محمد مصطفى محمد - 22010848

منن محمد ابودهب - 22011265

زياد اسلام حسنى-22010778

محمود مجدي محمود محمد محمود منير - 22011195

عبدالرحمن خالد نور - 22010877

Githup repo:

https://github.com/Ziad-Sallam/Signal-Flow-Graph

Phase 1

Problem statement :

- Design a graphical interface to visualize and interact with the signal flow graph.
- Identify and list: All forward paths from input to output, All individual loops within the graph, All possible combinations of non-touching loops.
- Compute the gains
- Calculate the overall system transfer function using Mason's Gain Formula

$$T(s) = \frac{\sum_{i=1}^{m} P_i \ \Delta_i}{\Delta}$$

Where:
P_i : gain of the i-th forward path
Δ_i : determinant excluding loops touching the i-th path
Δ : graph determinant

Main Features :

- Graphical user interface to construct the flow graph (add & delete & connect nodes)
- Result Window which has all the details (forward paths & loops & non-touching loops & global values)
- Additional options : save and load signal graphs

Data Structure :

- Adjacency matrix --> represent graph
- Stack --> for dfs algo
- Array --> save data
- Set --> for non-touching algo

Main modules :

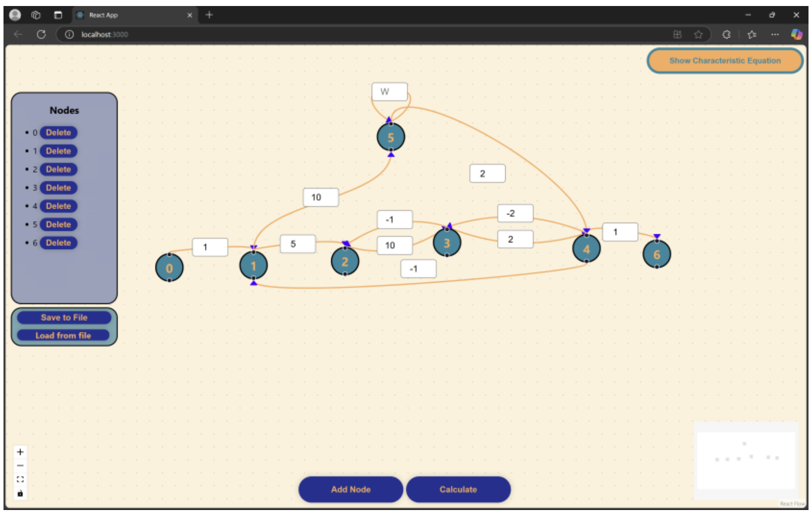
- Input Module
- Graph Construction Module
- Path & Loop Detection Module
- Mason's Gain Calculation Module
- Output & Report Module

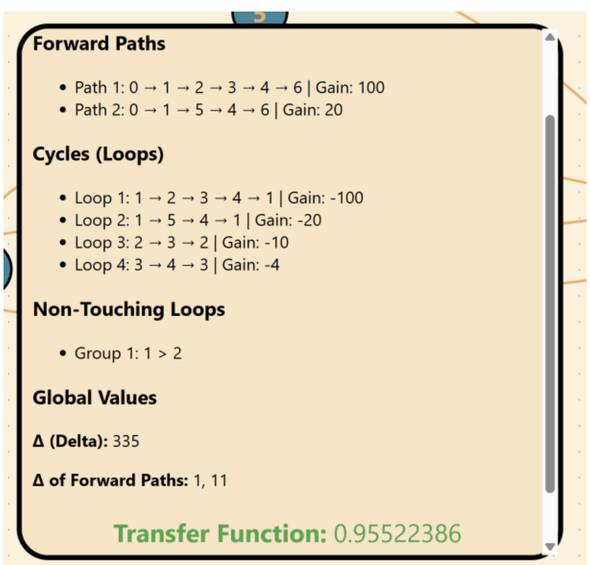
Algorithms used :

- DFS --> forward paths and loops detection $O(n^2)$
- graph-based approach --> find combinations of non-touching loops $O(2^n)$

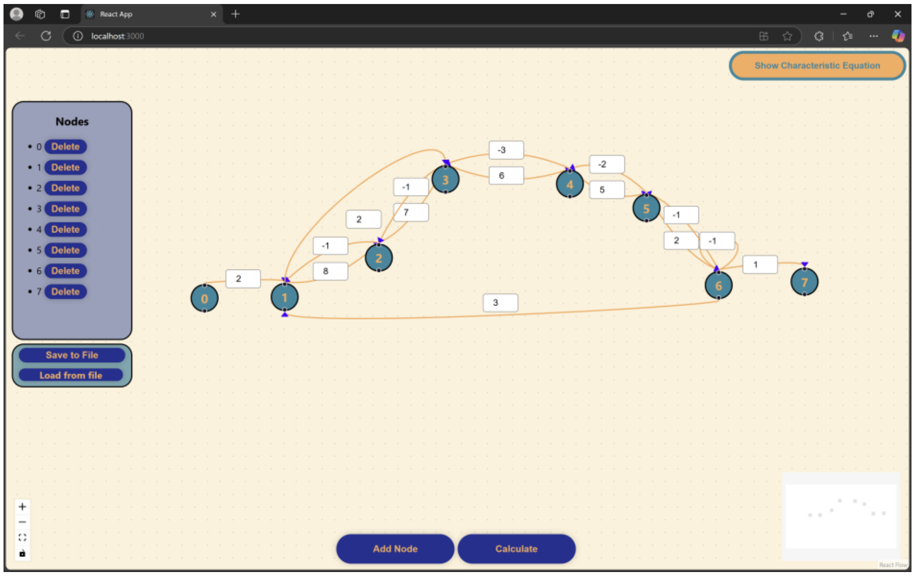
Sample runs :

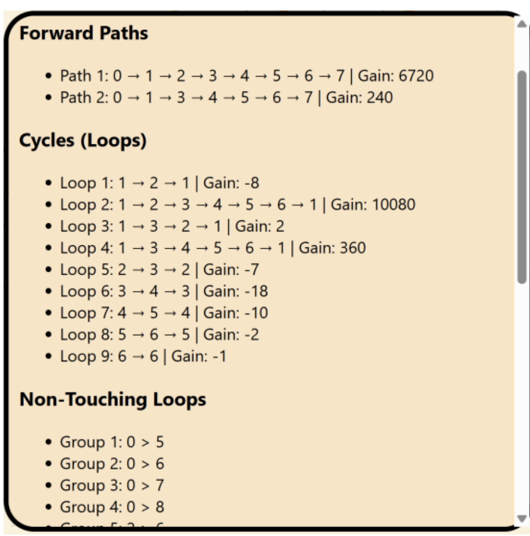
Testcase 1:





Testcase 2:





Forward Paths

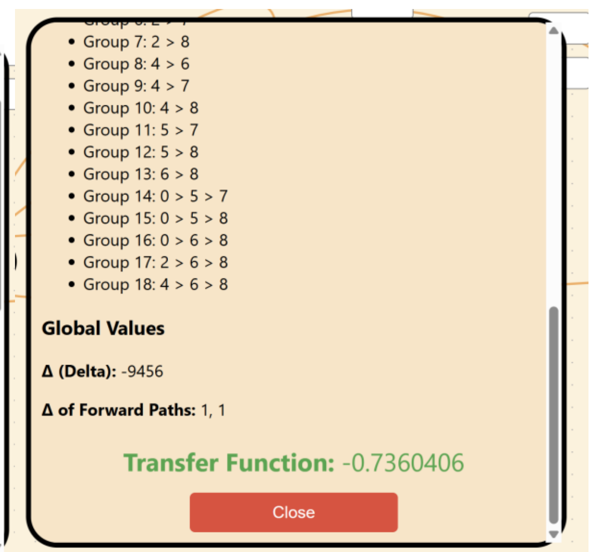
- Path 1: 0 → 1 → 2 → 3 → 4 → 5 → 6 → 7 | Gain: 6720
- Path 2: 0 → 1 → 3 → 4 → 5 → 6 → 7 | Gain: 240

Cycles (Loops)

- Loop 1: 1 → 2 → 1 | Gain: -8
- Loop 2: 1 → 2 → 3 → 4 → 5 → 6 → 1 | Gain: 10080
- Loop 3: 1 → 3 → 2 → 1 | Gain: 2
- Loop 4: 1 → 3 → 4 → 5 → 6 → 1 | Gain: 360
- Loop 5: 2 → 3 → 2 | Gain: -7
- Loop 6: 3 → 4 → 3 | Gain: -18
- Loop 7: 4 → 5 → 4 | Gain: -10
- Loop 8: 5 → 6 → 5 | Gain: -2
- Loop 9: 6 → 6 | Gain: -1

Non-Touching Loops

- Group 1: 0 > 5
- Group 2: 0 > 6
- Group 3: 0 > 7
- Group 4: 0 > 8



- Group 7: 2 > 8
- Group 8: 4 > 6
- Group 9: 4 > 7
- Group 10: 4 > 8
- Group 11: 5 > 7
- Group 12: 5 > 8
- Group 13: 6 > 8
- Group 14: 0 > 5 > 7
- Group 15: 0 > 5 > 8
- Group 16: 0 > 6 > 8
- Group 17: 2 > 6 > 8
- Group 18: 4 > 6 > 8

Global Values

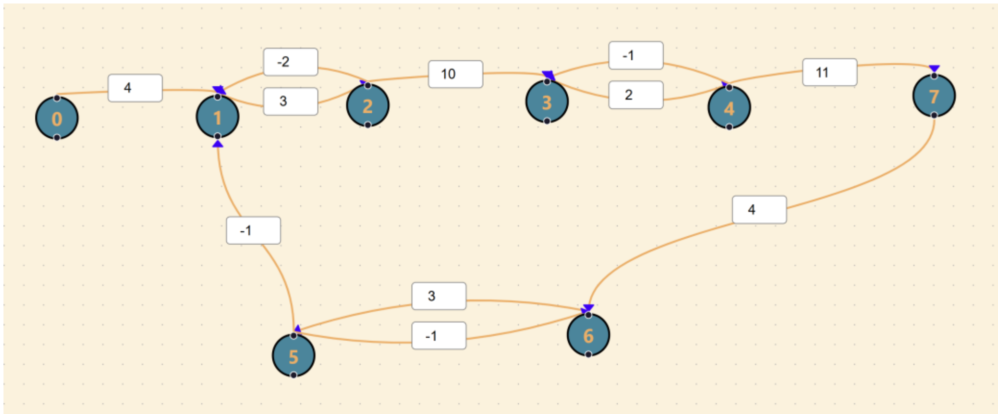
Δ (Delta): -9456

Δ of Forward Paths: 1, 1

Transfer Function: -0.7360406

Close

Testcase 3:



Forward Paths

- Path 1: 0 → 1 → 2 → 3 → 4 → 7 | Gain: 2640

Cycles (Loops)

- Loop 1: 1 → 2 → 1 | Gain: -6
- Loop 2: 1 → 2 → 3 → 4 → 7 → 6 → 5 → 1 | Gain: -7920
- Loop 3: 3 → 4 → 3 | Gain: -2
- Loop 4: 5 → 6 → 5 | Gain: -3

Non-Touching Loops

- Group 1: 0 > 2
- Group 2: 0 > 3
- Group 3: 2 > 3
- Group 4: 0 > 2 > 3

Global Values

Δ (Delta): 8004

Δ of Forward Paths: 4

Global Values

Δ (Delta): 8004

Δ of Forward Paths: 4

Transfer Function: 1.3193403

Simple user guide:

1. Click “Add Node” to the signal flow graph
2. Drag and drop the move it
3. Click and drag on the point on the node to create a connection between the nodes
4. To delete the node, click “Delete” button beside the node number or select the required node and click “Backspace”
5. To delete an edge, select the edge and click “Backspace”
6. To solve the signal flow graph, Click on “Calculate”

Phase 2

Problem statement :

- The goal of code to implement the Routh-Hurwitz stability criterion, a mathematical method to determine the stability of a linear time-invariant (LTI) system
- First: Write the Characteristic Polynomial, ensure coefficients are ordered from highest to lowest power of s.
- Second: Check Necessary (but Not Sufficient) Conditions which are All coefficients must be non-zero and All coefficients must have the same sign.
- Third: Construct the Routh Array and it has The Routh array has $n+1$ rows and $n/2$ columns. Using this formula

$$Routh[i][l] = \frac{(ab)-(bc)}{a}$$

- Handle Special Cases

  - Case 1: Zero in the First Column

    -> Replace 0 with ϵ (small positive number) and proceed

  - Case 2: Entire Row is Zero

    -> Indicates purely imaginary roots (critically unstable).

    -> Replace the zero row with coefficients of the derivative of an auxiliary polynomial (formed from the previous row).

- Determine Stability: Count sign changes in the first column of the Routh array

  - No sign changes: All poles in LHP ⇒ Stable.
  - Sign changes: Number of poles in RHP = Number of sign changes ⇒ Unstable.
  - Zero row: Critically unstable (poles on imaginary axis)

Main Features :

- Graphical user interface to construct the Polynomial Input (add coefficient & remove coefficient & submit)
- Result table which has (Routh Array Visualization & Stability answer & Root Calculation)

Data Structure :

- 2D array → Routh array
- Array list<double> → Polynomial Coefficients

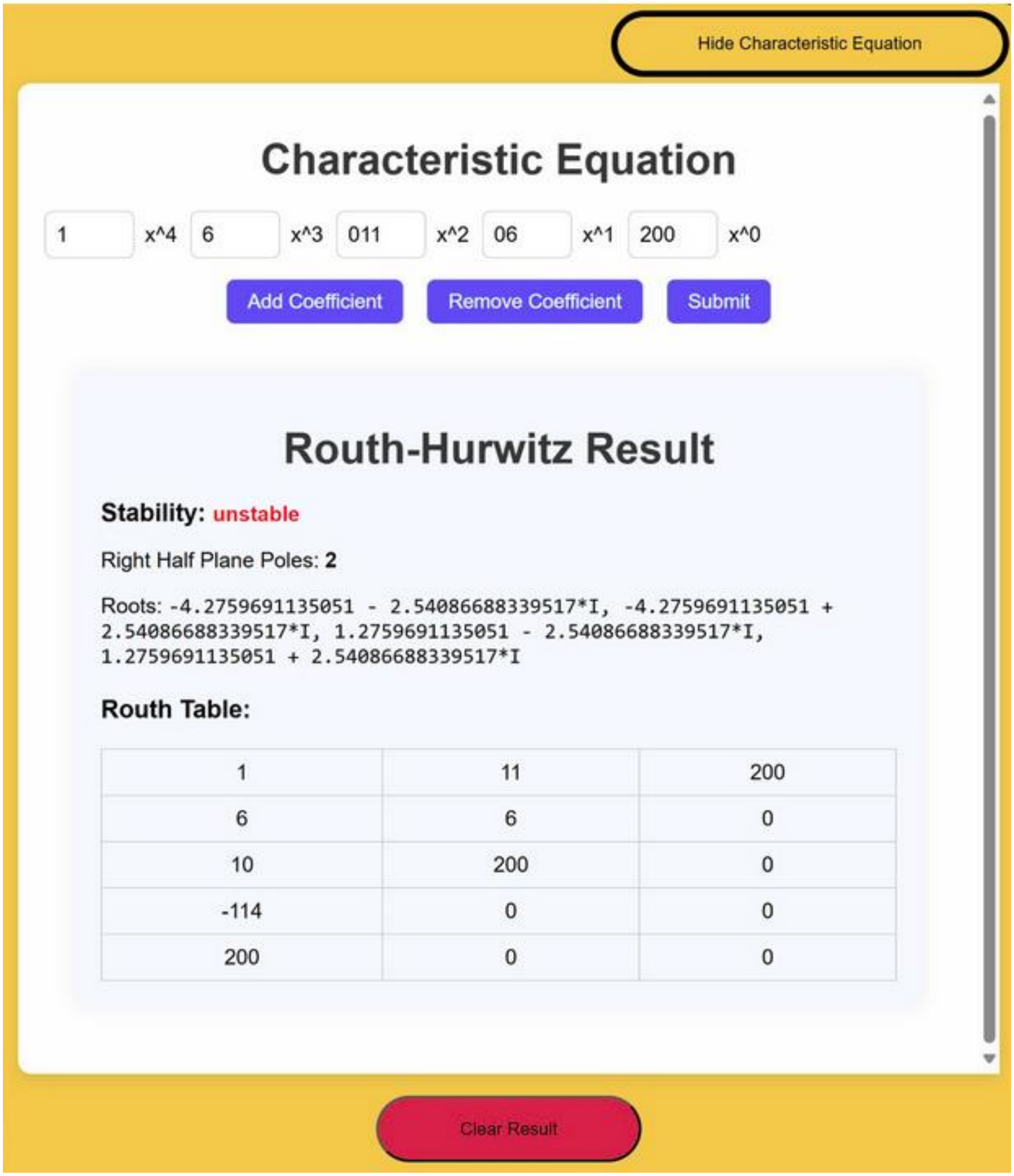
Main modules :

- Input Module
- Routh Array Construction Module
- Stability Analysis Module
- Root Calculation Module
- Output & Report Module

Algorithms used :

- Routh-Hurwitz Algorithm

Sample runs:


Hide Characteristic Equation
Characteristic Equation
1 x^4 6 x^3 011 x^2 06 x^1 200 x^0
Add Coefficient
Remove Coefficient
Submit
Routh-Hurwitz Result
Stability: unstable
Right Half Plane Poles: 2
Roots: -4.2759691135051 - 2.54086688339517*I, -4.2759691135051 + 2.54086688339517*I, 1.2759691135051 - 2.54086688339517*I, 1.2759691135051 + 2.54086688339517*I
Routh Table:
1 11 200
6 6 0
10 200 0
-114 0 0
200 0 0
Clear Result

Hide Characteristic Equation

Characteristic Equation

1 x^5 2 x^4 3 x^3 6 x^2 5 x^1 3 x^0

Add Coefficient Remove Coefficient Submit

Routh-Hurwitz Result

Stability: **unstable**

Right Half Plane Poles: **2**

Roots: `-1.66808883897419, -0.508833141633747 - 0.701995131769539*I, -0.508833141633747 + 0.701995131769539*I, 0.342877561120843 - 1.50829016116663*I, 0.342877561120843 + 1.50829016116663*I`

Routh Table:

1	3	5
2	6	3
0.00001	3.5	0
-699993.9999999999	3	0
3.5000000000428577	0	0
3	0	0

Clear Result

Hide Characteristic Equation

Characteristic Equation

1 x^5 7 x^4 6 x^3 42 x^2 8 x^1 56 x^0

Add Coefficient Remove Coefficient Submit

Routh-Hurwitz Result

Stability: **stable**

Right Half Plane Poles: **0**

Roots: `-7.00000000000000, -2.0*I, -1.4142135623731*I, 1.4142135623731*I, 2.0*I`

Routh Table:

1	6	8
7	42	56
28	84	0
21	56	0
9.333333333333334	0	0
56.00000000000001	0	0

Clear Result

Simple user guide:

1 - Enter the Characteristic Equation

2 -click submit to View the Results

3 - click Clear result to try another example