SIGNAL FLOW GRAPHS

LAB EXPERIMENT 12

Harel Ko
BS Electronics Engineering, Samar State University (of Aff.)
Institute of Electronics Engineers of the Philippines (of Aff.)
Catbalogan City, Philippines
Harelko999@gmail.com

Luis Nacional
BS Electronics Engineering, Samar State University (of Aff.)
Institute of Electronics Engineers of the Philippines (of Aff.)
Catbalogan City, Philippines
Obeywes7@gmail.com

***Abstract*—This experiment explores the application of Signal Flow Graphs (SFG) in control system analysis and their effectiveness in determining a system's transfer function using Mason's Gain Formula. By simulating a second-order system composed of two transfer function blocks with unity feedback, we constructed the corresponding SFG, derived the system's transfer function analytically, and compared it to the one derived using traditional block diagram algebra. The results confirmed the accuracy and consistency of SFG-based analysis, demonstrating its practicality as an alternative approach for analyzing control systems, especially in simulation-based settings.**

I. RATIONALE

Signal flow graphs are an alternative method for representing and analyzing control systems. This experiment explores how signal flow graphs can be used to simplify and analyze the system's transfer function..

II. OBJECTIVES

- Construct a signal flow graph for a given system.
- Use Mason's gain formula to calculate the system's transfer function from the signal flow graph.
- Compare the transfer function obtained using the signal flow graph with that from block diagram algebra.

III. MATERIALS AND SOFTWARE

- Materials: DC motor or RLC circuit, Power supply, Oscilloscope
- Software: Python (with control library), Octave

IV. PROCEDURES

1) Set up a system (e.g., DC motor or RLC circuit) and create its block diagram representation.
2) Convert the block diagram into a signal flow graph by identifying nodes and branches.
3) Apply Mason's gain formula to calculate the system's transfer function from the signal flow graph.
4) Compare the transfer function obtained using the signal flow graph with the one derived from block diagram algebra.

V. OBSERVATION AND DATA COLLECTION

DATA COLLECTION:

https://drive.google.com/drive/folders/16dxOl4dSZIlG35Z4KrBmlgwQEDiXV80X

Click here to open the Drive

VI. DATA ANALYSIS

In this experiment, we considered a system with two cascaded transfer functions:

$$G_1(s) = \frac{10}{s+2}, \quad G_2(s) = \frac{5}{s+1}$$

connected in a unity feedback configuration.

Transfer Function from Block Diagram Algebra

The closed-loop transfer function derived using block diagram algebra is:

$$T(s) = \frac{G_1(s)G_2(s)}{1+G_1(s)G_2(s)} = \frac{50}{(s+2)(s+1)+50} = \frac{50}{s^2+3s+52}$$

Transfer Function from Mason's Gain Formula

The corresponding Signal Flow Graph includes:

- One forward path: $P = G_1 \cdot G_2$
- One loop: $L = -G_1 \cdot G_2$

Using Mason's Gain Formula:

$$T(s) = \frac{P}{1-L} = \frac{50}{s^2+3s+52}$$

Simulation Comparison

Both systems were simulated using Python's `control` library. The resulting step responses from the block diagram and SFG-derived transfer functions were visually identical. Below is the plot comparing both methods:

VII. DISCUSSION AND INTERPRETATIONS

The simulation confirms the theoretical equivalence of the two approaches. Both the block diagram algebra and Mason's Gain Formula yielded the same transfer function. This alignment validates the reliability of signal flow graph analysis in control systems.

Analyzing the system's time-domain behavior also provided insight into the characteristics of second-order systems. The simulated response was stable, with minimal overshoot and a smooth rise to the steady-state value. This aligns with expectations from a moderately damped second-order system, indicating good controllability and predictable dynamics.

Moreover, applying Mason's formula reinforced understanding of forward paths, loops, and how they influence system behavior. While block diagram algebra is straightforward for simple systems, SFG becomes especially useful in more complex systems with multiple feedback loops and interconnections.

VIII. CONCLUSION

Through this simulation-based experiment, we demonstrated that signal flow graphs are a powerful and accurate method for analyzing control systems. By deriving the transfer function using Mason's Gain Formula and validating it against traditional block diagram simplification, we confirmed their equivalence. The ability to conduct this entirely via Python simulation further emphasizes the value of software tools in control system analysis when physical hardware is unavailable. This method not only supports learning but also deepens conceptual understanding.