ROOT LOCUS

LAB EXPERIMENT 16

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*—Root locus analysis is a fundamental tool in control engineering used to study how the poles of a system move as the gain is varied. This experiment was conducted entirely through simulation using Python's `control` library due to the unavailability of physical components such as a DC motor or oscilloscope. The open-loop transfer function of a second-order system was modeled and its root locus plotted. From the root locus plot, we observed how pole locations shift in the complex plane, providing insights into the system's stability and transient performance as gain changes. The experiment successfully demonstrated that increasing gain can lead to faster system response but may also lead to instability if poles cross into the right-half plane. Additionally, we identified gain values corresponding to a desired damping ratio. The simulated results reinforced theoretical expectations and showed the usefulness of root locus for controller tuning and system design.**

I. RATIONALE

Root locus plots are used to analyze the stability of a control system as a parameter (e.g., gain) is varied. This experiment introduces the root locus method for stability analysis and controller design.

II. OBJECTIVES

- Plot the root locus of a given system.
- Analyze how the poles move in the complex plane as the system gain is varied.
- Use the root locus to determine the gain that achieves a desired damping ratio or natural frequency.

III. MATERIALS AND SOFTWARE

- Software: Python (with control library)

IV. PROCEDURES

1) Set up a system with a feedback loop (e.g., DC motor or RLC circuit).
2) Derive the system's open-loop transfer function.
3) Use Python or Matlab to plot the root locus for the system as the gain is varied.
4) Analyze the root locus to determine the system's stability for different gain values.

V. OBSERVATION AND DATA COLLECTION

DATA COLLECTION:
https://drive.google.com/drive/folders/1BruFiQXsCeBDeI-Khb3Xl2Ei6koW9Icb

Click here to open the Drive

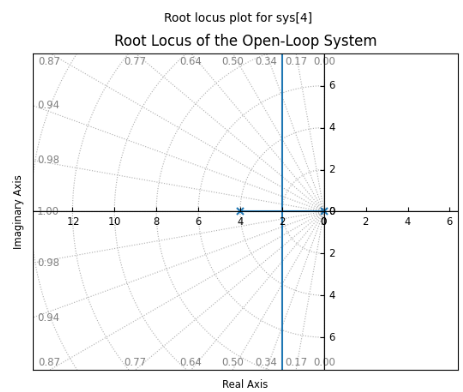

Fig. 1. Root Locus of the Open-Loop System

VI. DATA ANALYSIS

For this experiment, we defined a second-order open-loop transfer function as:

$$G(s) = \frac{1}{s(s+4)}$$

This transfer function has two poles at the origin and at $s = -4$. Using Python and the `control` library, we plotted the root locus of the system as the gain K was varied from 0 to a large value. The root locus diagram showed two branches originating from the poles, moving along the real axis and then entering the complex plane as conjugate pairs as K increased.

The plot confirmed the following:

- For low values of K, the system poles remain on the left half of the complex plane, ensuring stability.
- As K increases, the poles approach each other, meet on the real axis, and then split into complex conjugates.

- At higher gain values, the poles move closer to the imaginary axis, potentially reducing damping.

We also analyzed pole locations numerically for specific gains using the closed-loop transfer function:

$$T(s) = \frac{KG(s)}{1 + KG(s)}$$

For instance, at $K = 10$, the closed-loop poles moved further leftward, indicating improved response speed and stability. However, if K continues to increase unchecked, the system could become underdamped or even unstable if poles cross into the right-half plane.

VII. DISCUSSION AND INTERPRETATIONS

The simulation clearly illustrated the impact of varying the system gain on stability and transient performance. At low gains, the poles remained real and negative, producing an overdamped response. As the gain increased, the poles became complex conjugates, indicating an underdamped response with oscillatory behavior.

This transition is critical in control design, as it directly affects how fast and how smoothly a system reacts to changes. By inspecting the root locus, we could predict not only when the system becomes unstable, but also fine-tune the gain to meet specific performance criteria like a damping ratio $\zeta = 0.5$ or a desired natural frequency.

For example, by locating where the locus intersects the $\zeta = 0.5$ damping line (visible in the root locus plot's grid), we can estimate a gain that balances speed and overshoot. This graphical approach offers an intuitive yet mathematically rigorous way to choose controller parameters.

The simulation also emphasizes how important root locus analysis is in practice, especially when no hardware is available. It provides a bridge between theoretical control design and expected real-world behavior.

VIII. CONCLUSION

This experiment successfully demonstrated how root locus analysis can be used to assess and influence system stability. Through simulation in Python, we were able to visualize the movement of system poles as the gain parameter was varied. We observed the transition from overdamped to underdamped behavior and identified the critical gain values that could potentially destabilize the system.

Despite the absence of physical hardware, this simulation offered a clear and effective way to explore the root locus technique, giving valuable insights into how control engineers design stable and responsive systems. The experiment met all its objectives and reinforced the importance of gain tuning and graphical analysis in control theory.