STEADY-STATE ERRORS

LAB EXPERIMENT 14

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 investigates the steady-state error characteristics of a second-order control system using simulation tools. The system's response to step, ramp, and parabolic inputs was evaluated, and steady-state errors were computed using the Final Value Theorem. Error constants K_p, K_v, and K_a were derived to classify system tracking performance. Results showed that the uncompensated system had a finite error for step inputs and infinite errors for ramp and parabolic inputs. To address this, a PI compensator was implemented, significantly improving the system's ability to track the desired input. The study highlights the effectiveness of simulation-based analysis and the role of compensators in minimizing steady-state error in control systems.**

I. RATIONALE

Steady-state error quantifies how accurately a system can track a given input over time. This experiment explores different methods for calculating and minimizing steady-state error in control systems.

II. OBJECTIVES

- Calculate the steady-state error for a system subject to step, ramp, and parabolic inputs.
- Analyze the system's error constant (position, velocity, acceleration) for each input type.
- Implement a compensator to reduce the steady-state error and quantify the improvement.

III. MATERIALS AND SOFTWARE

- Software: Python (with control library)

IV. PROCEDURES

1) Set up a system (e.g., DC motor with position encoder) and apply a step, ramp, and parabolic input.
2) Measure the steady-state error for each input type.
3) Calculate the system's error constants (position, velocity, acceleration).
4) Implement a compensator (e.g., integral control) to reduce the steady-state error and measure the improvement.

V. OBSERVATION AND DATA COLLECTION

DATA COLLECTION:

https://drive.google.com/drive/folders/1O3EupFXqOOlfQQoblI5HVYrhQ3sup8GN

Click here to open the Drive

VI. DATA ANALYSIS

In this experiment, a second-order Type 1 system with the transfer function:

$$G(s) = \frac{1}{s(s+2)}$$

was analyzed to evaluate its steady-state error response to standard test inputs: step, ramp, and parabolic. Using Python and the `control` library, the system's response and corresponding error constants were computed.

1. Steady-State Errors

The steady-state errors for each input were computed using the Final Value Theorem:

- **Step Input**: $e_{ss} = 2.0000$
- **Ramp Input**: $e_{ss} = \infty$
- **Parabolic Input**: $e_{ss} = \infty$

2. Error Constants

The position (K_p), velocity (K_v), and acceleration (K_a) error constants were calculated to classify the system's accuracy for each type of input:

- $K_p = \infty$
- $K_v =$ NaN (undefined)
- $K_a =$ NaN (undefined)

These results indicate that the system can track a step input with a finite error, but cannot track ramp or parabolic inputs effectively.

3. Compensator Implementation

To address the large steady-state errors, a PI (Proportional-Integral) controller was introduced, defined as:

$$C(s) = K_p + \frac{K_i}{s}$$

with gains $K_p = 10$ and $K_i = 5$. After applying the compensator, the system's step response showed a significantly reduced steady-state error. Due to the introduction of the integrator, the system type increased, which theoretically reduces or eliminates steady-state error for step and ramp inputs.

VII. DISCUSSION AND INTERPRETATIONS

The simulation results aligned well with the theoretical expectations of steady-state error in control systems. The uncompensated system, being a Type 1 system (with one pole at the origin), handled a step input with a finite error but failed to track ramp and parabolic inputs, which led to infinite steady-state errors. This behavior is consistent with classical control theory, where the number of integrators (poles at the origin) in a system determines the type and its ability to track inputs of increasing degree.

The error constants further confirmed the system's limitations. The position constant $K_p = \infty$ indicated zero error for step input in ideal Type 1 systems, but due to the structure of our particular transfer function, a small nonzero error still remained. The velocity and acceleration constants returned undefined or NaN, likely due to limitations in the numerical computation or because the system simply does not support those levels of tracking.

Implementing a PI compensator proved effective. The integral component increased the system type to 2, introducing an additional pole at the origin. This allowed the system to eliminate steady-state error for a step input and theoretically track a ramp input with finite error. The visual improvement in the compensated system's step response further validated the compensator's benefit.

This experiment highlights not only how steady-state error is quantified, but also how compensators can be systematically designed to address performance shortcomings. It also reinforces the practical use of simulations to validate theoretical results when hardware is unavailable.

VIII. CONCLUSION

Through simulations in Python, the steady-state error behavior of a second-order control system was successfully analyzed. The system showed a finite steady-state error for step inputs but failed to track ramp and parabolic inputs, as indicated by infinite errors. By analyzing the error constants and applying a PI compensator, the system's ability to follow reference inputs was significantly improved. This experiment demonstrated the importance of system type in steady-state performance and showcased how simulation tools can be used effectively to design and test control strategies without physical hardware.