Python GUI for Ball and Beam System

Author:
Kubilay ZENGİN

Supervisor:
Şeniz ERTUĞRUL

Aug 23, 2023

Acrome Inc.
ITU ARI4 Science Park
Maslak, Istanbul
Turkey
info@acrome.net
Phone: +90 532 132 17 22
Fax: +90 212 285 25 94

Printed in Maslak, Istanbul

For more information on the solutions Acrome Inc. offers, please visit the website at:

http://www.acrome.net

Contents

1 OVERVIEW

1.1 Introduction

The Ball and Beam system is a classic example of a control system, where the goal is to maintain the position of a ball on a beam by applying control actions. In this report, we present a Python Graphical User Interface (GUI) designed for the Ball and Beam system by Acrome Robotics. The GUI provides an interactive and intuitive platform to visualize and control the behavior of the Ball and Beam system.

1.2 Objective

The objective of this user-friendly Python GUI is to provide an intuitive platform for students and researchers to easily grasp the fundamentals of control systems. The GUI offers an interactive and visually appealing environment that allows users to experiment with various control inputs, observe real-time responses, and fine-tune control strategies using a PID controller. By offering a hands-on approach to control system analysis and experimentation, this GUI aims to enhance students' understanding of control theory.

1.1 Key Features

1. **User-Friendly Interface:**
    a. Intuitive and easy-to-use graphical interface.
    b. Allows users to interact with the system effortlessly.
2. **Responsive Design:**
    a. Responsive layout design that adapts well to different screen sizes.
    b. Ensures a consistent experience across various devices.
3. **PID Input Section:**
    a. Provides an input section for users to specify PID controller gains (Kp, Ki, Kd).
    b. Allows users to fine-tune the control parameters for optimal system performance.
4. **Real-Time Data Plotting:**
    a. Implements real-time data plotting of ball position versus time.
    b. Provides visual feedback on the behavior of the ball and beam system over time.
5. **Python and Arduino Serial Communication:**
    a. Enables communication between the Python application and an Arduino board.
    b. Facilitates real-time data exchange and control signal transmission.

6. **Automatic Port Detection:**
    a. Automatically detects and lists available COM ports.
    b. Simplifies the process of selecting the appropriate port for serial communication.
7. **Data Saving Functionality:**
    a. Allows users to save entered data (such as PID gains) for future use.
    b. Ensures that user-defined settings are retained even after the application is closed.
8. **Stop Time Input:**
    a. Incorporates an input field for users to set the desired stop time for data collection.
    b. Provides flexibility in controlling the duration of data acquisition.

2 FUNCTIONALITIES

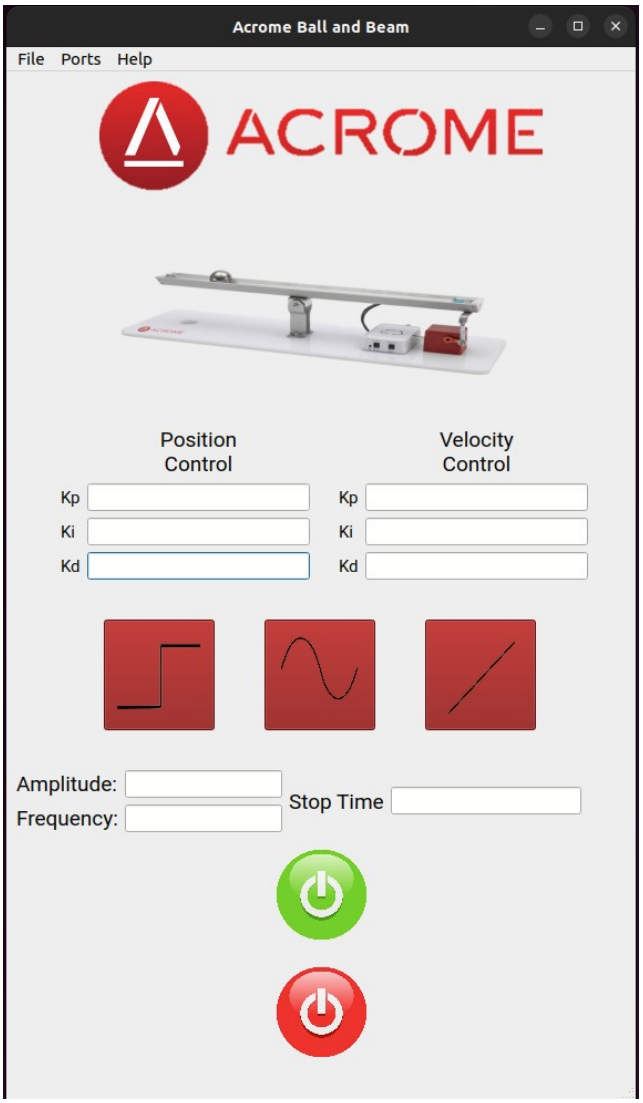

Figure 1. Visual of GUI

The code automatically detects the available serial ports. First of all, the user should select the specific COM Port where the Arduino is connected.

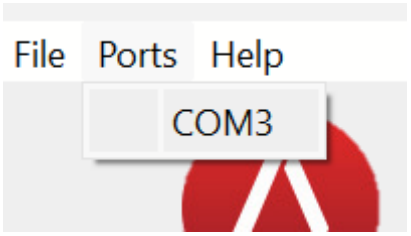

Figure 2. Arduino Serial Port Selection Menu

The user can enter amplitude and frequency values for various input types. The amplitude value is fixed between -250 and 250 mm since the beam is 500 mm long. The center point is assumed at 0 mm. The frequency value is fixed between 1 and 1000 hertz.

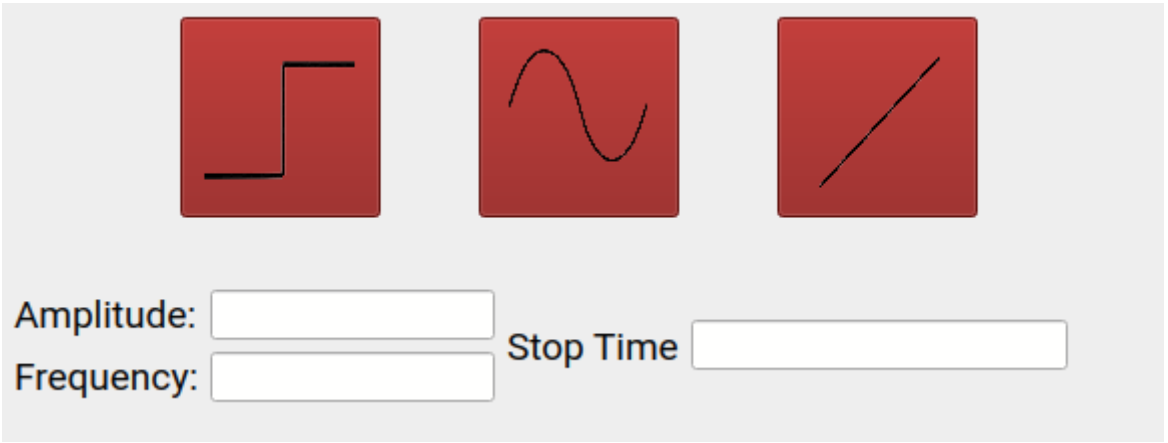

Figure 3. Amplitude and Frequency Input Part

GUI provides two PID controllers with adjustable gains (Kp, Ki, Kd). The first PID is for position control and the second is for velocity control.

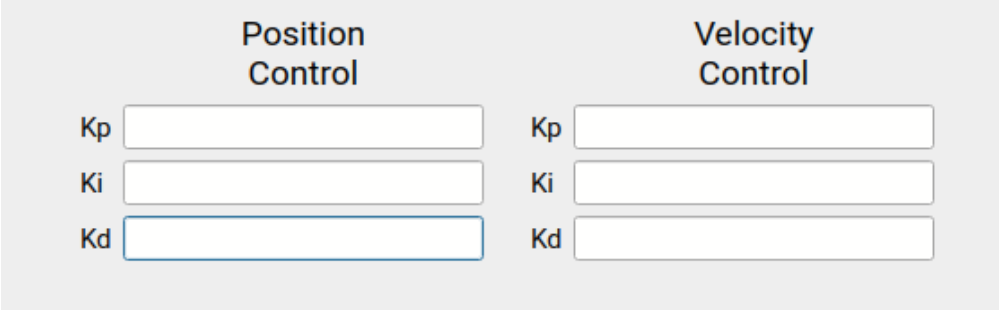

Figure 4. PID Input Part

Position and velocity controllers have different structures. Their corresponding structures are stated below.

Position Control: $G(s)_{PID} = \frac{U(s)}{E(s)} = Kp + \frac{KI}{S} + Kds$

Velocity Control: $\frac{dU(s)}{E(s)} = Kp + \frac{KI}{S} + Kds\,,\ U(s) = U(s) + dU(s)$

GUI has start and stop buttons, which enable real-time position data plotting and terminating the code, respectively.

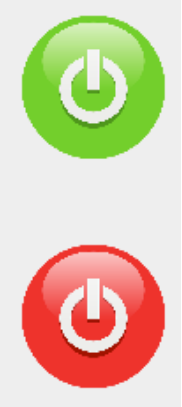

Figure 5. Start and Stop Buttons.

When the user presses the start button, real-time position data plotting begins. The y-axis was selected as Position in terms of mm and the x-axis was selected as time in terms of seconds.

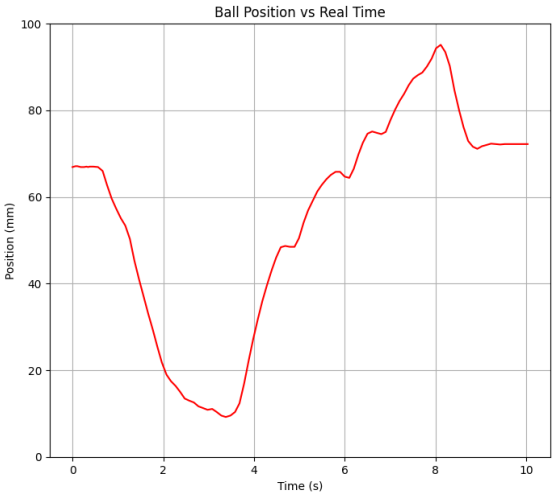

Figure 6. Real-Time Position Data Plotting

When the user presses the stop button, the program terminates itself and prints a message box.

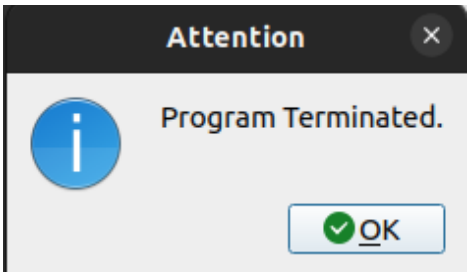

Figure 7. Terminating Program

When there is no detected available port, the program prints an attention message.

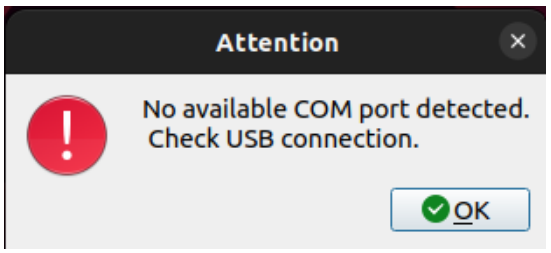

Figure 8. When There Is No Serial Port Available

The user can now enter stop time input for data plotting. That allows the user to specify a desired duration for the real-time data plot. The user can enter a stop time in seconds, and the program will continue plotting the real-time data until the specified stop time is reached.

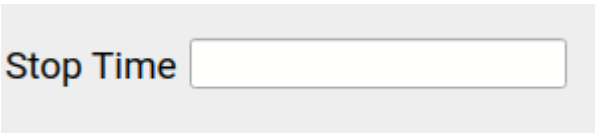

Figure 9. Stop Time User Input

For example, if the user enters 5 as an input, data plotting will end after 5 seconds. The example is given in Figure 10.

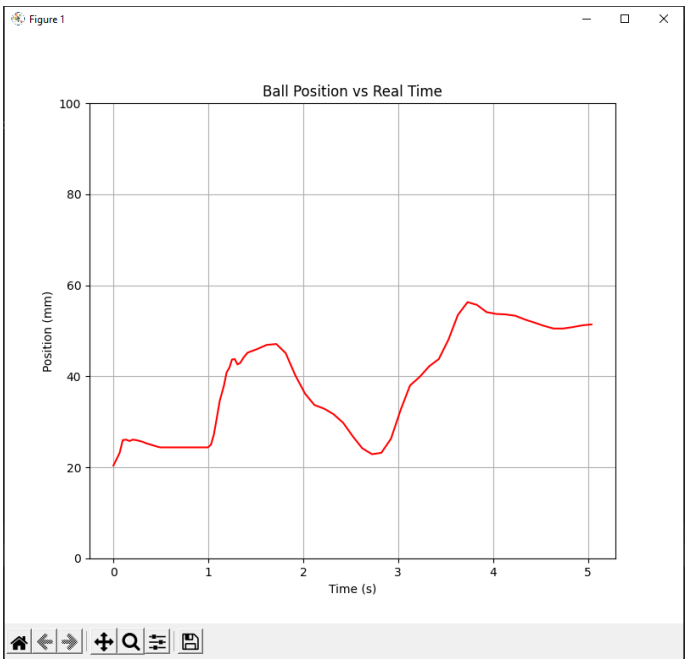

Figure 10. Data Plotting for 5 Seconds

3 CONCLUSION

The Python Graphical User Interface for the Ball and Beam system offers an efficient and user-friendly platform for controlling and visualizing the Ball and Beam system's behavior. Its intuitive layout, PID control capabilities, and real-time data plotting make it a valuable tool for control system analysis, education, and experimentation. With this GUI, users can explore different control strategies, fine-tune PID parameters, and gain valuable insights into the dynamic behavior of the Ball and Beam system.

4 REFERENCES

1. Acrome Robotics Ball and Beam System: https://www.acrome.net/products/ball-and-beam

ACROME Inc.

ITU Science Park, ARI4 Building
No: B204 Koruyolu

Maslak - Istanbul Turkey

For further information on ACROME equipment please contact.

Website: http://www.acrome.net/

e-mail: info@acrome.net

Telephone: 0212 807 0456