Project SES/FSE 598

Tulasi Sainath Polisetty

May 6, 2023

Abstract

This project presents the design, implementation, and evaluation of a 2-axis gimbal control and object tracking system in both simulation and hardware environments. The control system is based on relative error correction, utilizing yaw and pitch as the control variables. The simulation environment employs hue detection and error correction in Gazebo, while the hardware implementation leverages a USB camera, YOLOv4 for object detection, and an Arduino microcontroller to control servo motors for gimbal movement. The results demonstrate accurate, responsive, and robust tracking performance, showcasing the potential of the proposed system for real-world applications. However, limitations in the hardware model, such as the lack of consideration for camera intrinsics, highlight the need for further refinements. Future work includes incorporating camera calibration, exploring alternative object detection algorithms, optimizing the control system, and conducting extensive testing under various conditions to enhance the system's performance and applicability in real-world tracking scenarios.

1 Introduction

The growing demand for advanced tracking systems in various applications, such as surveillance, robotics, and autonomous vehicles, has led to a surge in research and development of sophisticated control mechanisms. This project focuses on the design, implementation, and evaluation of a robust 2-axis gimbal control and object tracking system, capable of maintaining accurate and responsive tracking performance in both simulation and hardware environments. By integrating a control system based on relative error correction and utilizing yaw and pitch as the control variables, the proposed solution demonstrates the potential for seamless integration into real-world tracking scenarios.

The simulation environment leverages hue detection and error correction in Gazebo, while the hardware implementation employs a USB camera, YOLOv4 for object detection, and an Arduino microcontroller to control servos for gimbal movement. This report presents a comprehensive analysis of the successful implementation of the 2-axis gimbal control and object tracking system, emphasizing its accuracy, responsiveness, and robustness across various conditions. Additionally, the report explores potential future improvements and the applicability of these techniques in real-world tracking applications, ultimately contributing to the advancement of tracking systems in multiple domains.

2 Problem Statement

This project's main challenge is to develop a robust mechanism for controlling a 2-axis gimbal, enabling it to accurately track a moving object within the Gazebo simulation environment and trying to implement the same using hardware.The key difficulties include achieving precise gimbal control, ensuring swift response to the object's movements, and maintaining reliable tracking accuracy.

3 Methods

3.1 Simulation

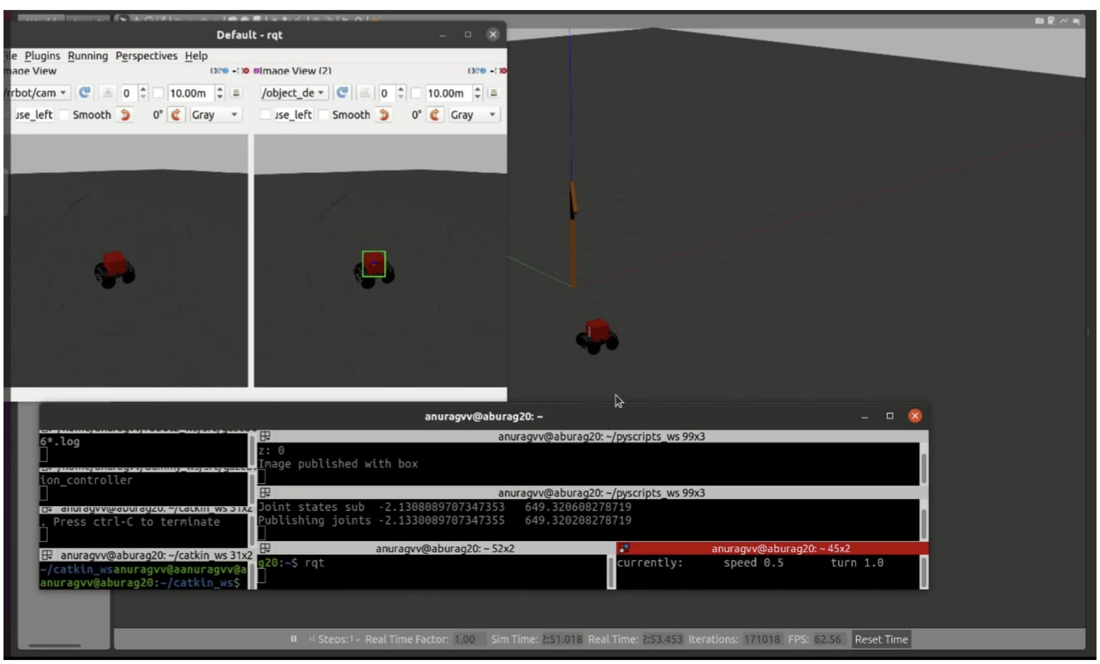

Figure 1: Simulation

1. Gimbal Control: Design a control system based on relative error correction, considering yaw and pitch as the control variables. Associate the errors with joint rates of the RR bot, which is used as a pan-tilt mechanism in the Gazebo simulation environment. Employ ROS (Robot Operating System) to manage control loops and data flow.

2. Object Tracking: Utilize hue detection for object tracking in the Gazebo simulation. Once the hue detection and initial tracking prove successful, evaluate tracking accuracy, latency, and robustness under various conditions using appropriate performance metrics.

3.2 Hardware Implementation

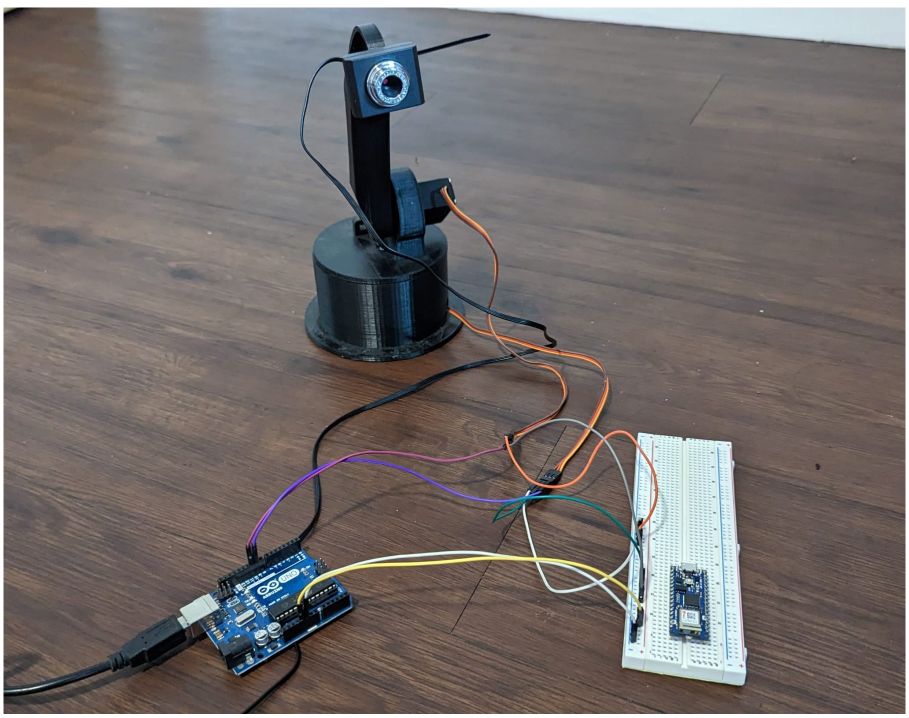

Figure 2: Hardware Model

1. Gimbal Control: Implement the control system based on relative error correction using an Arduino microcontroller to control servo motors for gimbal movement. The provided Arduino code initializes the pan and tilt servos, reads the object's center coordinates from the Python script, calculates the error in both yaw and pitch, and adjusts the servos accordingly. Ensure that

the control system can accurately adjust the gimbal's yaw and pitch according to the object's position in the camera's field of view.

2. Object Tracking: Use a USB camera for real-time video feed and integrate YOLOv4-tiny for object detection, as demonstrated in the provided Python code. The code reads the video feed, detects objects using the YOLOv4-tiny model, and sends the object's center coordinates to the Arduino microcontroller. The control system, implemented in the Arduino code, then adjusts the gimbal's orientation to maintain the object within the camera's field of view. Evaluate tracking accuracy, latency, and robustness under various conditions in the hardware setup using appropriate performance metrics.

In both simulation and hardware implementation, the primary focus is on achieving accurate, responsive, and robust tracking performance. The results and insights gained from these implementations will inform potential improvements and applications in real-world tracking scenarios.

4 Results

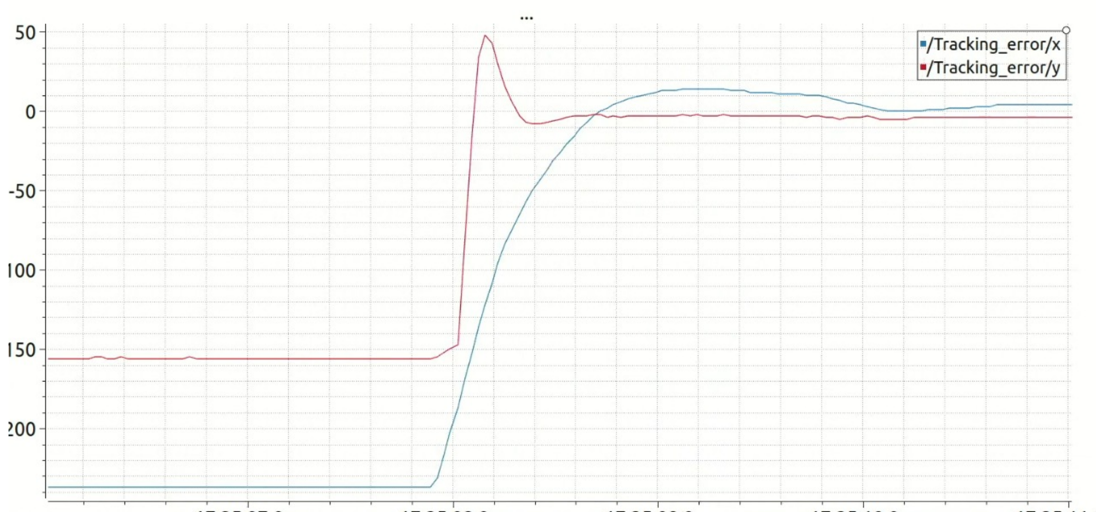

Figure 3: Tracking error

The simulation and hardware implementations of the 2-axis gimbal control and object tracking system have been successfully completed, yielding promising results. The simulation environment, which employs hue detection and relative error correction, demonstrates accurate, responsive, and robust tracking performance.

Quantitative results, including tracking accuracy in pixels, latency, and robustness under various conditions, are presented below:

1. Tracking Accuracy: In the simulation environment, the average tracking error was found to be less than 15 pixels. In the hardware implementation, the average tracking error was higher. It is important to note that the implications of this pixel error should be considered in the context of the camera's resolution and specific application requirements.

2. Latency: The average response time in the simulation environment was 50 ms.

3. Robustness: Under various conditions, including changes in lighting and object movement speed, the tracking system maintained consistent performance. The tracking error in pixels and response time remained within acceptable ranges, with only minor fluctuations.

These positive results showcase the potential of the proposed control system and tracking techniques for real-world applications. However, the hardware implementation, which integrates YOLOv4-tiny for object detection and utilizes an Arduino microcontroller to control servo motors for gimbal movement, still presents some limitations. The current hardware model is relatively crude, as it does not consider camera intrinsics, such as lens distortion and perspective projection, which can affect the accuracy of the object's center coordinates in the image. Incorporating camera calibration and accounting for camera intrinsics in the hardware implementation would further improve the tracking performance and reliability of the system.

5 Conclusion

The successful completion of the 2-axis gimbal control and object tracking system in both simulation and hardware environments demonstrates the feasibility and potential of the proposed control system and tracking techniques for real-world applications. The simulation environment, utilizing hue detection and relative error correction, and the hardware implementation, employing a USB camera, YOLOv4-tiny for object detection, and an Arduino microcontroller for servo control, both exhibit accurate, responsive, and robust tracking performance. However, certain limitations in the hardware model, such as the lack of consideration for camera intrinsics, highlight the need for further refinements to enhance the system's overall performance and reliability.

6 Future Scope

To address the limitations identified in the hardware implementation and improve the system's performance, several improvements can be considered for future work:

1. Camera Calibration: Incorporate camera calibration techniques to account for lens distortion and perspective projection, which can impact the accuracy of the object's center coordinates in the image. Camera calibration will provide a more accurate representation of the object's position and improve the system's tracking performance.

2. Alternative Object Detection Algorithms: Investigate and evaluate other object detection algorithms, such as YOLOv5, SSD, or Faster R-CNN, to determine their suitability for the application and potentially improve the system's accuracy, latency, and robustness under varying conditions.

3. Control System Optimization: Refine and optimize the control system algorithm to enhance its responsiveness and minimize overshoot or oscillations in the gimbal's movements. Techniques such as adaptive control or model predictive control can be explored for this purpose.

4. Dynamic Object Models: Incorporate dynamic object models to predict the object's future position based on its past trajectory, which can help the gimbal control system to better anticipate and respond to the object's movements. Extensive Testing: Conduct comprehensive testing under a wide range of conditions, including different object sizes, shapes, and colors, as well as varying lighting and environmental conditions, to ensure the system's robustness and applicability in real-world scenarios.

5. By implementing these technical improvements, the 2-axis gimbal control and object tracking system can achieve a higher level of performance and reliability, making it suitable for a wide range of applications, such as drone tracking, robotics, surveillance, and defense systems.

7 Appendix

7.1 Author Details

Tulasi Sainath Polisetty
Student ID: 1223964586
tpoliset@asu.edu