Design, Implementation, and Control of a Bicopter System: System Identification and Motor Speed Optimization

Sourish Tetakayala
Department of Aerospace and Mechanical Engineering
University of Arizona
Email: sourish@arizona.edu

***Abstract*—This report presents the design, implementation, and control of a bicopter system with a focus on system identification and motor speed optimization. Using system identification techniques, dynamic models of the bicopter were derived from experimental input-output data, enabling accurate controller design. The project aims to stabilize and control the bicopter's tilt angles while progressively increasing motor speed for improved maneuverability. A combination of PID and LQR controllers was designed and implemented to achieve precise angle maintenance and responsiveness. Experimental data was collected using Simulink models and Arduino integration, facilitating the identification of system dynamics and the evaluation of control performance. Results demonstrate the system's ability to maintain stability under varying motor speeds. Challenges, such as initial motor response delays and controller tuning, are discussed, along with recommendations for future improvements.**

***Index Terms*—Bicopter, System Identification, PID Control, LQR Control, Motor Speed Optimization, Aerial Robotics, Simulink.**

I. Introduction

The increasing demand for efficient and agile aerial vehicles has spurred the development of innovative designs, such as bicopters, which utilize two rotors for lift and control. Bicopters, with their compact structure and minimal actuator requirements, present significant advantages for applications in reconnaissance, inspection, and precision delivery.However, their underactuated dynamics and inherent instability pose unique challenges, particularly in achieving precise control and responsiveness under varying conditions [1]–[3].

System identification plays a major role in addressing these challenges by deriving mathematical models that capture the bicopter's dynamic behavior. Accurate models are essential for designing robust controllers capable of maintaining stability and optimizing performance. This project utilizes system identification techniques to extract dynamic models from experimental input-output data, serving as the foundation for control system development.

The primary objectives of this project include stabilizing the bicopter at specific tilt angles, maintaining these angles under varying motor speeds, and gradually ramping up motor speeds to evaluate the system's limits. Proportional-Integral-Derivative (PID) and Linear Quadratic Regulator (LQR) controllers were implemented to achieve precise tilt angle control and motor speed optimization. The integration of Simulink with Arduino enabled real-time data acquisition, model validation, and controller deployment, bridging the gap between simulation and physical implementation.

This report presents a comprehensive approach to bicopter control, starting from system modeling and identification to advanced controller design and experimental validation. Insights gained during the project highlight the challenges of real-time control, including initial motor response delays and the trade-offs between stability and performance. These findings underscore the importance of system identification in modern aerial robotics and offer a pathway for future research and development.

II. Literature Review

System identification and control design play a critical role in addressing the challenges faced by underactuated aerial vehicles, such as bicopters. Previous studies have explored dynamic modeling, controller design, and practical implementation strategies, providing a foundation for this work.

Enikov et al. [4] developed an educational bicopter setup utilizing black-box modeling through Auto-Regressive with eXogenous inputs (ARX) methods. Their work emphasized the importance of selecting appropriate input test sequences for system identification, which greatly affects model accuracy. This project adopts a similar ARX-based identification approach but extends it to include motor speed optimization and real-time control deployment. Furthermore, the open-source GitHub repository by Enikov [5] serves as a key resource for integrating Simulink and Arduino, offering practical tools for system identification and feedback controller design.

Uddin et al. [6] presented a pitch stabilization system for bicopters using Lyapunov-based control techniques. While their study focuses on theoretical stability analysis, it aligns with this project's goal of maintaining precise tilt angles under varying conditions. The robustness of Lyapunov-based control complements the implementation of PID and LQR controllers in our work.

In contrast, He and Wang [7] introduced a novel bicopter design featuring trajectory tracking control with servomotors. Their work highlights the integration of mechanical design and control systems, showcasing the importance of achieving dynamic stability during trajectory tracking. The principles from their study are adapted here for analyzing system stability during motor speed ramp-up and control optimization.

The use of PID control for bicopter systems has been demonstrated effectively by Jayadi et al. [8], who implemented a one-axis balancing system. Their results validate the applicability of PID controllers for steady-state performance, a method similarly utilized in this project for initial stabilization before advancing to LQR-based optimization. Further, Fahmizal et al. [3] proposed a low-cost bicopter design focusing on attitude control. Their emphasis on balancing cost and performance resonates with our approach of leveraging affordable components like Arduino and Simulink for real-time experimentation.

Complementing these studies, works by Dong et al. [1] and Zhao et al. [2] addressed nonlinear control methods for bicopters, utilizing feedback linearization and robust control techniques, respectively. Their insights highlight the importance of handling system nonlinearities to achieve reliable performance, an area considered in our future work. Additionally, Tang et al. [9] explored adaptive PID control for quadrotors, further underscoring the value of combining system identification with adaptive control strategies.

These studies collectively emphasize the significance of system identification and robust control methods in bicopter systems. While prior works have individually addressed dynamic modeling, stabilization, or trajectory tracking, this project integrates these aspects into a unified approach, focusing on motor speed optimization, system stability, and real-time control performance. By building upon existing frameworks and incorporating validated methods, this work aims to contribute a comprehensive solution for bicopter dynamics and control.

III. System Overview

The bicopter system is a compact and efficient platform designed for both stabilization and dynamic control experiments. It integrates hardware and software components to achieve precise control of angular positions and motor speeds. The main components and their roles are as follows:

A. Hardware Components

- **DC Motors and Propellers:** Two small DC motors with propellers are mounted on opposite ends of a rigid beam. These motors generate thrust, which induces angular motion around the pitch axis. The motor speed directly influences the tilt angle and overall stability.
- **Chassis and Wheels:** A lightweight chassis supports the motors, electronics, and sensors. Two large wheels are mounted to the structure to allow ground-based balancing experiments. The wheels are designed to reduce rolling resistance and ensure stability.
- **Arduino Nano 33 IoT:** The Arduino serves as the central microcontroller, managing motor actuation, sensor data acquisition, and communication with the Simulink environment.
- **Inertial Measurement Unit (IMU):** An LSM6DS3 accelerometer and gyroscope module measures the bicopter's angular position and acceleration. This data is crucial for real-time feedback control.
- **Motor Driver:** A dual-channel motor driver controls the PWM signals sent to the motors, enabling precise speed regulation.
- **Power Supply:** The system is powered using a compact battery pack, ensuring portability and continuous operation.

B. Integration of Simulink and Arduino

The bicopter system leverages Simulink for controller design, simulation, and data visualization. Real-time communication between Simulink and Arduino is established using TCP/IP protocols, allowing seamless data acquisition and control implementation. The workflow includes:

1) **Controller Design in Simulink:** The PID and LQR controllers are designed and tuned in Simulink. These controllers generate motor commands based on real-time IMU feedback.
2) **Data Exchange:** Input signals (e.g., setpoints) are sent from Simulink to the Arduino, which executes motor control logic. IMU sensor readings are transmitted back to Simulink for monitoring and analysis.
3) **Real-Time Execution:** Simulink enables real-time visualization of the bicopter's response, facilitating controller tuning and performance evaluation.

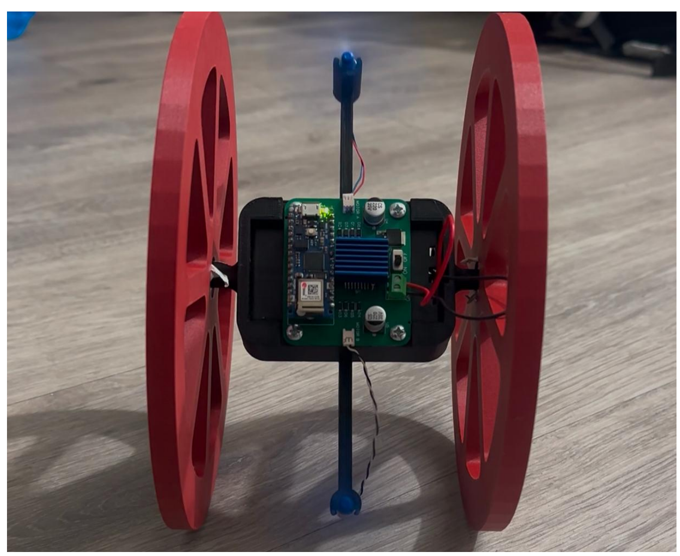

Fig. 1. Hardware implementation of the bicopter system, showing motors, Arduino, IMU, and chassis [5].

The integration of these components allows for a robust experimental platform capable of testing advanced control strategies and validating system identification models. By combining hardware experimentation with Simulink-based simulations, the system bridges the gap between theory and practical implementation.

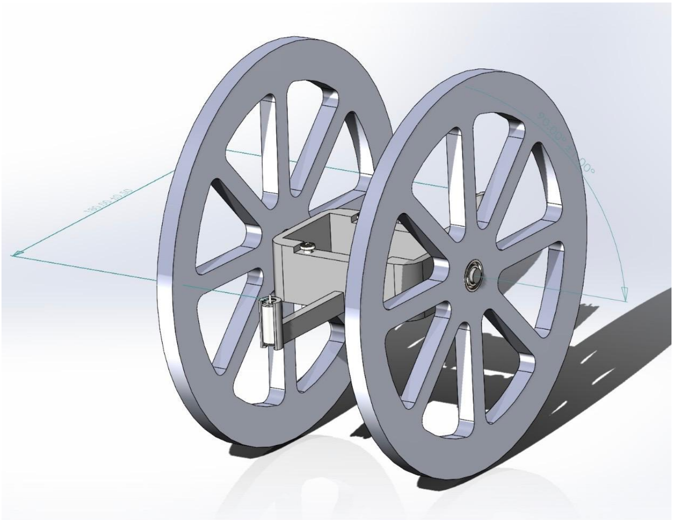

Fig. 2. CAD model of the bicopter system highlighting mechanical design and wheel assembly [5].

IV. Methodology

The bicopter control project follows a systematic approach involving experimental testing, system identification, and control implementation. The overall methodology is divided into the following tasks:

A. Task 1: Open-Loop Test and Data Acquisition

The open-loop test serves as the foundation for system identification. The objective is to analyze the bicopter's response to an input signal and extract meaningful input-output data.

- **Input Signal Generation:** A Random Binary Signal (RBS) was generated using the `idinput()` function in MATLAB. The signal excites the system over a specified bandwidth:

$$\text{Input} = \text{idinput}(\text{dur}, \text{'rbs'}, [0, \frac{1}{30}], [-7, 7]), \tag{1}$$

  where $\text{dur} = 2000$ represents the number of data points, and the amplitude range is set between $[-7, 7]$.
- **Data Logging:** The input signal was transmitted to the bicopter via TCP/IP, and the tilt angle response was recorded using the onboard IMU. The collected data was saved in MATLAB's `iddata` format for analysis:

$$\text{data} = \text{iddata}(\text{Output}', \text{Input}', \text{Ts}), \tag{2}$$

  where $\text{Ts} = 0.01$ is the sampling time.
- **Detrending the Data:** Offsets and low-frequency drifts were removed to ensure accurate system identification:

$$\text{data} = \text{detrend}(\text{data}). \tag{3}$$

The open-loop Simulink model, illustrated in Fig. 3, was used to transmit the input signals and log the tilt angle responses.

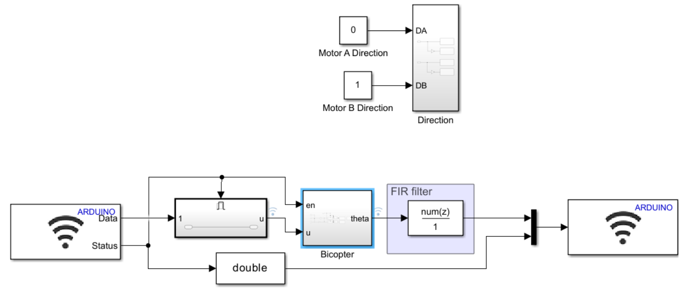

Fig. 3. Open-loop Simulink model for data acquisition.

B. Task 2: System Identification

System identification aims to develop a mathematical model that characterizes the bicopter's dynamic behavior, enabling accurate controller tuning and validation [1], [4], [9]. Two approaches were implemented:

1) ARX Model: The Auto-Regressive with eXogenous inputs (ARX) model was chosen due to its simplicity. The general form of the ARX model is:

$$y(t)+a_1y(t-1)+\cdots+a_ny(t-n) = b_1u(t-1)+\cdots+b_mu(t-m), \tag{4}$$

where $y(t)$ is the output (tilt angle), $u(t)$ is the input (PWM signal), and a_i, b_i are model parameters.

The ARX model was estimated using the MATLAB command:

$$G = \text{arx}(\text{data}, [16\ 16\ 0]), \tag{5}$$

where 16 poles and zeros were used to approximate the system dynamics. The identified model was validated by comparing the simulated and measured outputs using MATLAB's `compare()` function (Fig. 4).

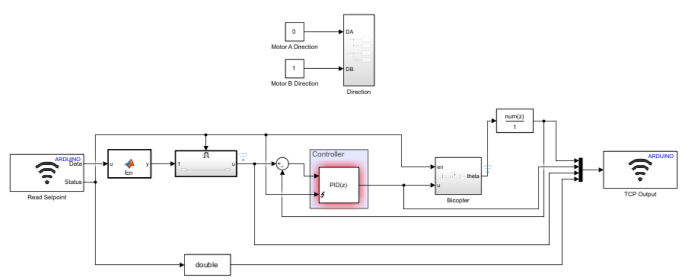

Fig. 4. ARX model validation: Comparison of measured and simulated responses.

2) State-Space Model (N4SID): The state-space model was obtained using the N4SID (Subspace Identification) algorithm, which derives a discrete-time representation of the system. The general state-space equations are:

$$x(k+1) = Ax(k) + Bu(k), \tag{6}$$
$$y(k) = Cx(k) + Du(k), \tag{7}$$

where $x(k)$ is the state vector, $u(k)$ is the input, $y(k)$ is the output, and A, B, C, D are system matrices.

The state-space model was identified using:

$$G_{\text{ss}} = \text{n4sid}(\text{data}, 16), \tag{8}$$

where the model order was set to 16. The performance of the identified state-space model was validated by comparing it with the ARX model and experimental data (Fig. 5).

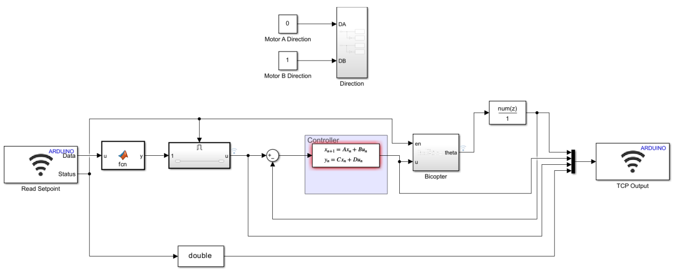

Fig. 5. State-space model validation: Simulated versus measured responses.

C. Task 3: Controller Design and Implementation

Once the dynamic models were validated, controllers were designed to stabilize the bicopter and optimize its performance.

1) PID Controller: The PID controller was implemented to regulate the tilt angle based on the error between the desired setpoint and the measured response. The controller's transfer function is:

$$C(s) = K_p + \frac{K_i}{s} + K_d s, \tag{9}$$

where K_p, K_i, and K_d are the proportional, integral, and derivative gains, respectively.

The gains were tuned using Simulink's `pidtune()` function:

$$C = \text{pidtune}(G, C_0, 1), \tag{10}$$

resulting in optimized values for K_p, K_i, and K_d.

The closed-loop Simulink model, shown in Fig. 6, was used to implement and validate the PID controller.

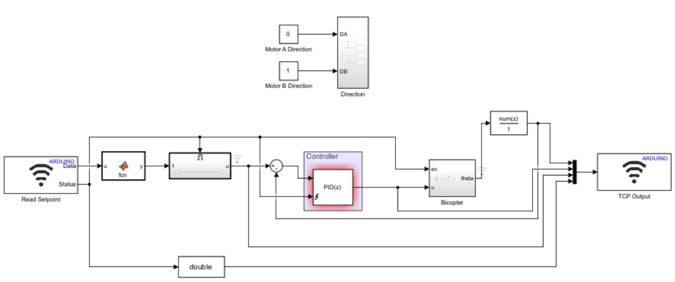

Fig. 6. Closed-loop Simulink model with PID controller.

2) LQR Controller: To further optimize performance, a Linear Quadratic Regulator (LQR) controller was designed. The LQR minimizes the cost function:

$$J = \int (x^T Q x + u^T R u)\, dt, \tag{11}$$

where Q is the state cost matrix and R is the control effort matrix.

The state-feedback gain K was computed using:

$$K = \text{dlqr}(A, B, Q, R). \tag{12}$$

The LQR controller was validated using the state-space model in the closed-loop Simulink model. Results are shown in Fig. 7.

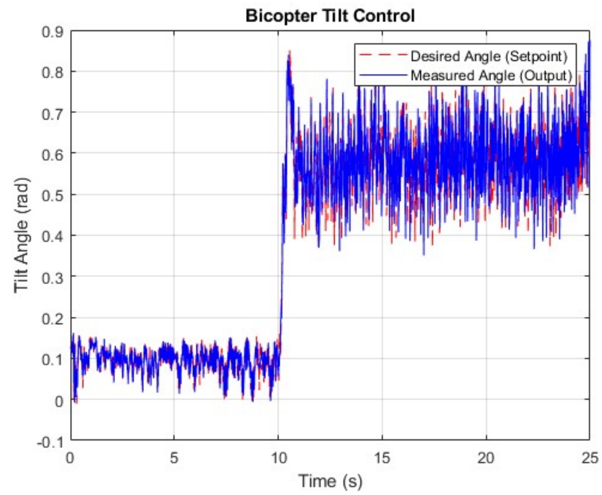

Fig. 7. LQR Controller: Closed-loop response.

D. Task 4: Experimental Validation

The controllers were deployed on the bicopter via Simulink's real-time interface with the Arduino. Step inputs were used to test the closed-loop response for both the PID and LQR controllers. Experimental results demonstrated the following:

- PID control provided good stabilization but exhibited small steady-state errors.
- LQR control achieved improved performance with reduced steady-state error and better robustness.

V. Results

A. System Identification Results

To analyze the dynamic behavior of the bicopter system, two system identification approaches were implemented: **ARX Model** and **State-Space Model**.

- **ARX Model:**
  - The Auto-Regressive with eXogenous input (ARX) model was selected with an order of [2 2 0].
  - **Fit to Data:** 64.41%
  - The ARX model captured the system dynamics with acceptable accuracy considering the noise and measurement variations.
- **State-Space Model:**
  - A state-space representation using the **N4SID algorithm** was employed with an order of 2.
  - **Fit to Data:** 58.15%
  - The lower model fit was expected due to the system's inherent nonlinearities and noise.

- **Transfer Function Estimation:** The discrete-time transfer function extracted using **tfest** provided the following representation:

$$\text{sys\_tf} = \frac{-0.003011z + 0.005255}{z^2 - 1.979z + 0.9811} \tag{13}$$

This transfer function demonstrates the system's response with a sample time of **0.01 seconds**.

B. Performance Metrics

The performance of the system was evaluated using the following metrics:

- **Steady-State Error:**
  The steady-state error was computed to quantify the deviation from the desired angle once the response stabilized.
  Value: 0.0094 radians.
- **Settling Time:**
  The settling time was determined based on the **2% tolerance criterion** relative to the final setpoint.
  Value: 17.99 seconds.
- **Corrected Average Overshoot:**
  The average overshoot was calculated after ignoring the first **5 seconds** to account for transient startup effects.
  Value: 17.45%.

C. Visual Results

- **Measured vs. Desired Angle (Raw Response):**
  The raw measured response demonstrates significant oscillations during the transient period, particularly before settling near the desired angle.
  Refer to **Figure 8**.
- **Measured vs. Desired Angle (Filtered Response):**
  The filtered response, obtained using a low-pass filter (cutoff frequency: **1 Hz**), shows smoother behavior and aligns closely with the desired angle.
  Refer to **Figure 9**.
- **Motor Speed Scaling:**
  The motor speed input signal reflects variations in control effort, particularly during transitions. Peaks observed near **10 seconds** highlight the control's reaction to sudden changes.
  Refer to **Figure 10**.
- **Comparison: Desired, Raw, and Filtered Responses:**
  A combined comparison plot highlights the improvements achieved through filtering. The filtered response exhibits better alignment with the desired angle while reducing noise and overshoot.
  Refer to **Figure 11**.

D. Summary of Results

E. Figures

VI. Conclusion

This study successfully demonstrates the design, implementation, and control of a bicopter system using system identification techniques and advanced control strategies. Dynamic models were identified from experimental input-output data

TABLE I
SUMMARY OF PERFORMANCE METRICS

Metric	Value
ARX Model Fit	64.41%
State-Space Model Fit	58.15%
Steady-State Error	0.0094 radians
Settling Time	17.99 seconds
Average Overshoot	17.45%

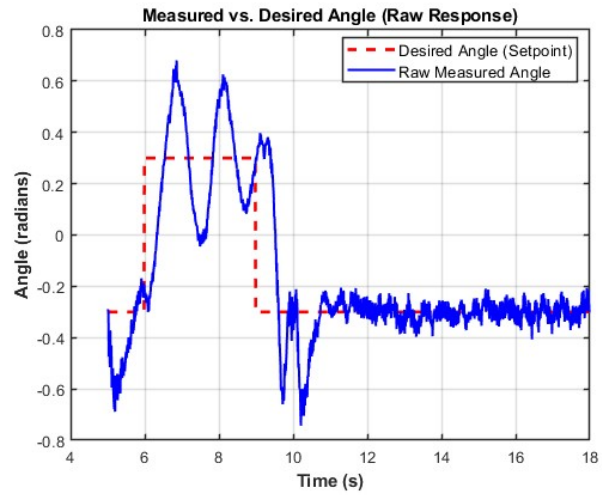

Fig. 8. Measured vs. Desired Angle (Raw Response)

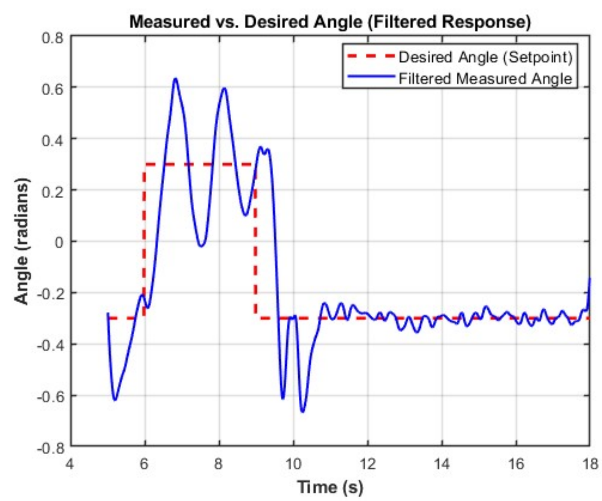

Fig. 9. Measured vs. Desired Angle (Filtered Response)

using ARX and State-Space methods, achieving model fits of 64.41% and 58.15%, respectively. The discrete-time transfer function provided further insight into the system's dynamic response, enabling the design of precise control systems.

The implementation of PID and LQR controllers enabled stabilization and control of the bicopter's tilt angles. Key performance metrics were achieved during experimental validation:

- **Steady-State Error:** 0.0094 radians,

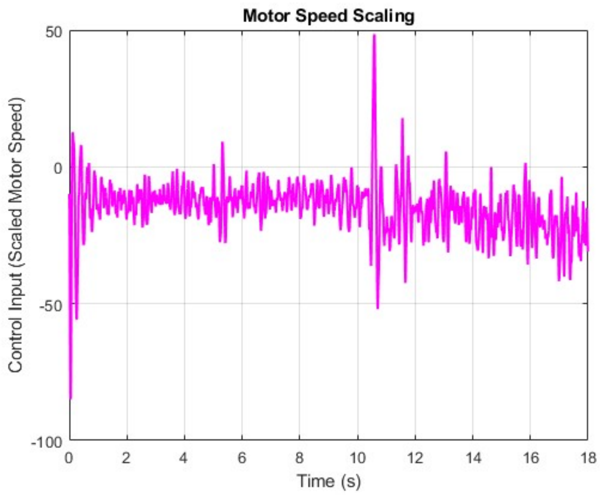

Fig. 10. Motor Speed Scaling

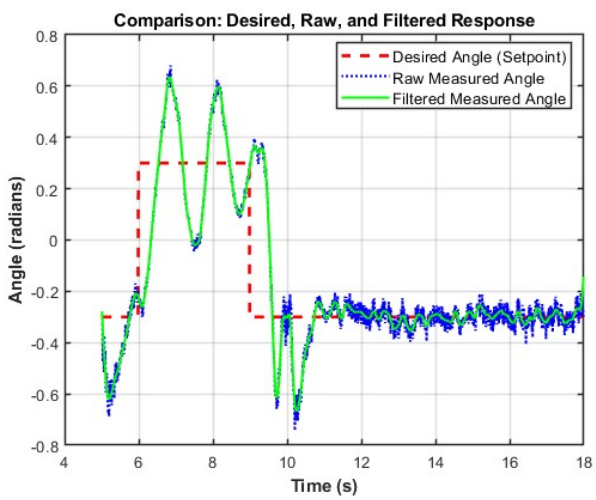

Fig. 11. Comparison: Desired, Raw, and Filtered Responses

- **Settling Time:** 17.99 seconds,
- **Corrected Average Overshoot:** 17.45%.

The filtered response effectively reduced noise and improved accuracy when compared to the raw measurements, aligning closely with the desired setpoints. However, challenges such as overcorrection during large tilt angles (exceeding 60-70 degrees) and occasional motor-induced rollovers were observed.

Future Work

Future efforts will focus on addressing the limitations identified during this study. Specifically:

- Redesigning the mechanical structure to prevent rollovers when the system overcorrects beyond 60-70 degrees [2], [3].
- Implementing advanced control strategies, such as adaptive or nonlinear controllers, to handle aggressive maneuvers and enhance stability [6], [10].
- Further tuning of PID and LQR controllers to improve accuracy and responsiveness under varying conditions.
- Incorporating higher-order models or hybrid identification techniques to capture nonlinear dynamics more effectively.
- Reduce the weight of the wheels.

By addressing these challenges, the bicopter system can achieve improved stability, greater robustness, and enhanced performance for practical applications.

REFERENCES

[1] Y. Dong, L. Han, and Y. Luo, "Nonlinear control of a bicopter based on feedback linearization," *IEEE Access*, vol. 7, pp. 35 877–35 885, 2019.

[2] H. Zhao, W. Li, and Z. Wang, "Bicopter dynamic modeling and robust control based on backstepping approach," *International Journal of Robotics and Automation*, vol. 36, no. 4, pp. 1–10, 2021.

[3] Fahmizal, H. A. Nugroho, A. I. Cahyadi, and I. Ardiyanto, "Attitude control and low cost design of uav bicopter," 2023. [Online]. Available: https://arxiv.org/abs/2309.08209

[4] *System Identification and Controller Design of Propeller Driven Pendulum (Bi-Copter)*, vol. Volume 8: Engineering Education, October 2023. [Online]. Available: https://doi.org/10.1115/IMECE2023-109757

[5] E. T. Enikov, "Arduino-based bi-copter experiments," 2024, gitHub Repository. Retrieved December 14, 2024. [Online]. Available: https://github.com/eenikov/Arduino-based-bi-copter-experiments

[6] N. Uddin, H. G. Harno, A. Manurung, and M. Nasucha, "A pitch control system design for bicopter uavs," in *2021 8th International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE)*, 2021, pp. 57–62.

[7] X. He and Y. Wang, "Design and trajectory tracking control of a new bi-copter uav," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9191–9198, 2022.

[8] A. Jayadi, F. D. Adhinata, J. P. Sembiring, N. U. Putri, E. Pranita, and A. Setiawan, "Implementation of pid control as a one axis balancing control system on the bicopter vehicle," in *2023 International Conference on Converging Technology in Electrical and Information Engineering (ICCTEIE)*, 2023, pp. 113–117.

[9] Y. Tang, H. Zhang, and C. Li, "System identification and adaptive pid control of a quadrotor uav," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 6, pp. 5176–5186, 2020.

[10] X. Cheng and J. Chen, "Adaptive control of underactuated uavs with external disturbances," *Aerospace Science and Technology*, vol. 128, p. 107670, 2022.