ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

SEMESTER PROJECT

Thrust-Assisted Controlled Tree Climbing

Laboratory of Intelligent Systems

Jason Mina

Supervisor: Mohammad Askari

Professor: Dario Floreano

January 5, 2024

EPFL

Laboratory of Intelligent Systems

SEMESTER PROJECT

Title:	Thrust-Assisted Controlled Tree Climbing
Student(s):	Jason Mina (Robotics)
Professor:	Dario Floreano
Assistant 1:	Mohammad Askari

Project description:

Tree or pole climbing is seen in nature in various forms among animals and insects. While some bio-inspired pole climbing solutions have been proposed for different types of robots, there is no existing evidence of a winged robotic platform with such capabilities. At the Laboratory of Intelligent Systems, we are developing a multimodal winged robot capable of flying, perching, and climbing vertical poles. The current robotic platform consists of flight avionics and is capable of crash-perching on trees by hugging them. Yet, it lacks controlled movement and relocation across the perch. The objective of this project is to develop control strategies and implement required hardware and software to showcase climbing up/down trees from a perched configuration with the existing platform. We propose the dual use of existing structures and electronics, such as the propulsion system and flight control surfaces to control thrust and heading for climbing, while keeping the added weight minimum.

The project steps include: (1) Review of the state-of-the-art on climbing, (2) Identify the control strategi(es) best suited for pole climbing and thrust-assisted control, (3) Controller design and embedded software implementation with necessary hardware modification, followed by (4) Experimentation and characterization, and (5) Simple mathematical modeling of the climbing locomotion mode.

Remarks:

You should present a research plan (Gantt chart) to your first assistant before the end of the second week of the project. An intermediate presentation of your project, containing 8 minutes of presentation and 7 minutes of discussion, will be held on November 9, 2023. The goal of this presentation is to briefly summarize the work done so far and discuss a precise plan for the remaining time of the project. Your final report should start by the original project description (this page) followed by a one page summary of your work. This summary (single sided A4), should contain the date, laboratory name, project title and type (semester project or master project) followed by the description of the project and 1 or 2 representative figures. In the report, importance will be given to the description of the experiments and to the obtained results. A preliminary version of your report should be given to your first assistant at the latest 10 days before the final hand-in deadline. A PDF (dated & signed) of your final version should be Sent to your first assistant and to the administrative assistant of the lab before noon January 5, 2024. Failing this deadline will reflect in the final grade. A 20 minute project defense, including 5 minutes for discussion, will take place January 11, 2024. You will be graded based on your results, report, final defense, and working style. All documents, including the report (source and pdf), summary page and presentations along with the source of your programs should be handed-in as a single compressed file on the day of the final defense at the latest.

Responsible professor:	Responsible assistant:
Signature: Dario Floreano Dario Floreano	Signature: M Askari Mohammad Askari

Lausanne, 19 September 2023

Summary

This report presents the development and progression of the Perchug, a multi-modal winged robot capable of perching on vertical poles, which was developed by researchers at the Laboratory of Intelligent Systems. The project aims to advance the design of the climbing capability of the Perchug, with a focus on identifying and implementing the most effective climbing strategy while minimizing additional weight.

It is divided into several sections, each focused on a specific aspect of the project. One section details the design methodology and control strategies implemented, such as the PID controller for stable hovering. Another section discusses the engineering challenges and solutions during the hardware and software development phase. The experimental results and tuning processes are explained in another part, which highlights the successes and areas needing improvement. Finally, the report concludes with a discussion on future work, emphasizing the need for autonomous operation and further refined control mechanisms.

Based on research, it was decided to develop a climbing of the Perchug by making it hover, similar to the concept of prop-hanging, which is traditionally done by a front propeller. Hovering is first achieved by a conventional fixed-wing drone to learn the different techniques to make sure the Perchug is safe when testing. Promising Results were obtained for the Perchug and Further Development strategies are presented.

Figure 1: Figure showing the concept of climbing by unperching

Table of Contents

Table of Figures

1. Introduction

At the Laboratory of Intelligent Systems, a multi-modal winged robot, the Perchug, capable of perching and climbing on vertical poles is being developed. The objective of this project is to develop control strategies and implement the required hardware and software to showcase climbing up/down trees from a perched configuration with the existing platform. Particularly, the project focuses on the strategy of making the Perchug capable of hovering and climbing upwards while doing so.

The Perchug drone concept is motivated by the need for surveillance, environmental monitoring, and telecommunication enhancement. Fixed-wing drones like Perchug have the advantage of longer flight times compared to rotary drones, but they typically require runways for take-off and landing. PerchUg's ability to perch on poles allows for strategic positioning and station-keeping without the need for ground infrastructure.

The motivation for the Perchug drone's climbing ability is to enhance its utility in surveillance, monitoring, and communication by allowing it to reach optimal vantage points on poles or trees. Climbing upwards after perching is critical for several reasons:

1. **Enhanced Visibility and Range**: Gaining altitude can provide a better vantage point for cameras and sensors, and higher elevation can extend communication ranges.
2. **Obstacle Avoidance**: Climbing allows the drone to avoid foliage or other obstructions when perching in environments like forests or urban areas.
3. **Sunlight Exposure:** For solar-powered drones, ascending could mean more exposure to sunlight, thus longer operational times.

2. State of the Art

2.1 Climbing Strategies

Climbing strategies for drones have been explored primarily through various robotic platforms, with a limited number of papers addressing fixed-wing drones specifically. Only one study on aerial climbing by fixed-wing UAVs is present in Literature [1], highlighting a unique approach in this domain.

1. **Legged and Wheeled Climbers:** Several papers discuss climbing robots with legged ([2], [3]-[7]) and wheeled mechanisms [8], emphasizing quasistatic and dynamic gaits for efficient motor operation. While legged and wheeled climbers maintain contact with the

surface, aerial platforms have the advantage of speed due to higher power density ([9]-[12]).

2. **Aerial Climbing:** Aerial platforms do not interact with the surface until perching, affording them much higher climb speeds due to their high power-to-weight ratio. Efficient aerial climbing is contingent on executing low-energy-cost transitions from perching to flight [13].

2.2 Proposed Methods:

Thrust-assisted climbing can be achieved by various methods:

1. **Wheeled Ascension:** Adding wheels to the fuselage to climb with thrust. However, this could add weight and complexity. See Figure 2

Figure 2: Wheeled Ascension developed by Ugo Giudiceanrea

2. **Partial Release and Climb:** Releasing the grip while keeping the wings around the pole. This method may require intricate control to maintain grip and could risk destabilizing the drone. See Figure 3

Figure 3: Partial Release Method Representation

3. **Unperching and Hovering:** Fully unperching to climb. This method is beneficial because it requires less mechanical complexity and exploits the drone's hovering capability, which aligns with fixed-wing drone research where prop-hanging is often demonstrated. See Figure 4

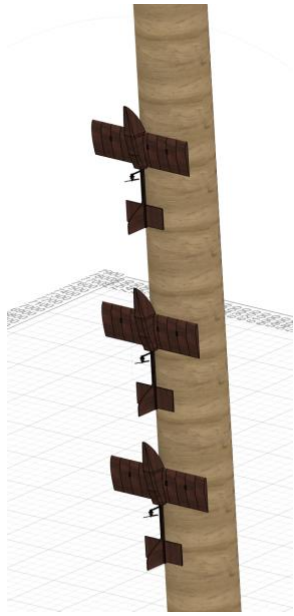

Figure 4: Unperching and hovering method demonstration

Choosing to unperch and hover is advantageous because it simplifies the mechanical design and relies on aerodynamic control, which is generally more robust and easier to model and simulate.

Despite speculating at first that limited airflow passes at the wings, which could complicate roll control, this configuration was chosen for its compact design and to keep the propeller away from obstacles during perching.

Considering the specific context of the Perchug project, the focus on pitch and yaw control over roll control aligns with the initial goal of establishing feasibility. The intention is not to redesign the hardware to incorporate roll control but rather to design and validate a control system that can manage the drone's orientation with the existing configuration.

For future projects, this presents an opportunity for innovation. The PerchUg could explore alternative control mechanisms, such as thrust vectoring, which have not been widely used in conventional fixed-wing UAV hovering but may be feasible with modern advancements in drone technology and control systems.

Table 1 presents a comprehensive comparison between the different methods.

Approach	Advantages	Disadvantages
Full unperch	- Flexibility: It can be used on a variety of tree types and sizes without much modification. - Reduced Wear and Tear: By flying upwards and not constantly rubbing against the tree, there may be reduced wear on the drone's components. - Fastest climbing capability - No hardware addition, simpler - mechanical design and lesser weight - low amount of added mass from perching and sensing com- ponents leads to a light and integrated architecture, conserving a high power density while adding versatility	- Full un-pershing of wings requires time and energy - More vulnrable to disturbances - Safety Concerns: If the drone fails during the vertical climb or re-perching, it can fall and get damaged or harm others.
Partial unperch	- Energy Efficiency: Likely consumes less energy than repeatedly taking off and perching. - Safety: Keeps a grip on the tree, which might be safer in certain conditions (e.g., windy conditions). - Faster (opening all the way needs time)	- Wear and Tear: Constant contact with the tree might wear out certain components faster. - Limited Movement: Can be restricted by the tree's surface, branches, or other obstructions.
Wheel-assisted climb	- Controlled Climbing: Provides a controlled way to move upwards, especially if the tree's surface is consistent. - Potential for Energy Efficiency: Can be more energy-efficient than constant flying.	- Mechanical Complexity: Introducing retractable wheels adds more moving parts and potential points of failure. - Tree Surface Dependency: Requires a relatively even tree surface for best operation. - Size and Weight: Introducing wheels might add to the drone's weight and size.

2.3 State of the Art of Hovering for Fixed-Wing Drones

A review of hovering techniques for fixed-wing drones indicates that attitude PID feedback is a favored control strategy, with various approaches and modeling methods explored:

1. **Quaternion Feedback Control:** Most papers show the effectiveness of quaternion feedback control for attitude stabilization papers ([14]-[15]). However, when the rolling error is large, quaternion feedback may fail. For our case, since roll control is limited, this approach is not

preferred. The approach chosen is to use Euler angle feedback, with a transformation that avoids the gimbal lock problem.

2. **Alternative Hovering Strategies:** Other papers [16]**,** delve into thrust vectoring and backstepping control design for more nuanced control. In our case, this is not feasible as the purpose is to work with the available Perchug and not modify its configuration.

3. **Modeling and Simulation:** Papers also discuss the importance of detailed modeling [17], and system identification [18], for the development of effective hovering control. This was initially explored, but after further exploration of the modeling, it became clear that a customized approach was needed to address the specific challenges posed by the Perchug's design, like the absence of ailerons and the unconventional placement of the propeller.

4. **Control Systems Development:** The development of control systems for these maneuvers includes evaluating different controllers (e.g., PD, PID), with many papers suggesting the PID controller due to its simplicity and robustness ([14]-[18]). This control method is the one used to control the drone.

3. Methodology

3.1 Modelling of the Perchug:

In the project, aerodynamic modeling was the first step taken to analyze the aerodynamics of PerchUg, involving the use of Flow5, focusing on its unique wing and tail design. This analysis was essential to determine the lift and drag coefficients, crucial for accurate flight dynamics modeling.

A Simulink model was developed integrating these coefficients. This model simulated PerchUg's flight behaviors, allowing for the testing and refinement of control strategies under various flight conditions. It served as a vital tool in predicting the UAV's performance, especially in complex maneuvers like perching and climbing.

This integration of aerodynamic data into the Simulink model was pivotal in bridging the gap between theoretical analysis and practical application, ensuring that the control strategies developed were grounded in realistic flight dynamics. However, the complexity of modeling, the time constraint, and the effect on the total performance tradeoff were all crucial factors that led to the decision to use experimental model-free tuning instead of based on a model.

Figure 5 shows the aerodynamic coefficients for different angle of attacks and Reynolds numbers, obtained using Flow5 software where it was intended to obtain a detailed model of the Perchug.

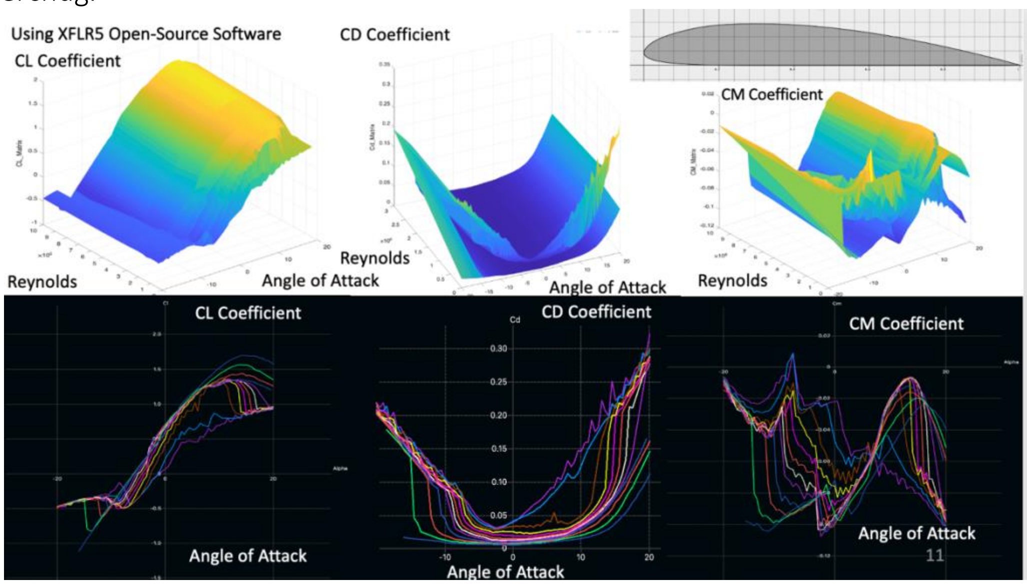

Figure 5: Obtained Aerodynamic Coefficients For Perchug

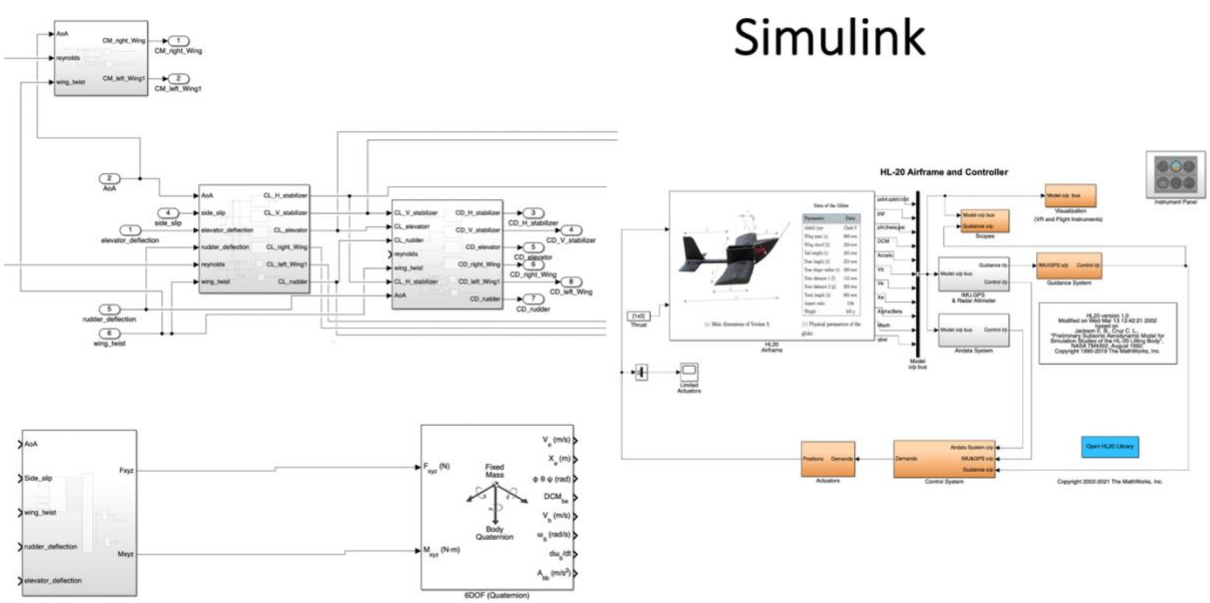

Figure 6: Simulink Model of the Perchug

Figure 6 shows the model implemented in Simulink. Here's a simplified explanation of the model:

1. **Input Section:** The model takes several inputs related to the aircraft's dynamics, such as angle of attack (AoA), side slip, and deflections of various control surfaces like elevators

and rudders. These inputs reflect the conditions and maneuvers the aircraft is undergoing.

2. **Aerodynamics Block:** This section uses the inputs to calculate aerodynamic coefficients, which are crucial for determining how the aircraft will move through the air. These coefficients include lift (CL), drag (CD), and moment (CM) coefficients for different parts of the aircraft like wings, stabilizers, and rudder.

3. **Flight Dynamics Block:** This block further processes the aerodynamic coefficients and other inputs to simulate the forces and moments acting on the aircraft. This is typically done using the 6 degrees of freedom (6DOF) model, which considers three translational and three rotational movements that an aircraft can experience.

4. **Control System Block:** Here, the model processes the forces and moments to generate outputs that would be used to control the aircraft's actuators (like ailerons, elevators, etc.). The control system aims to maintain the aircraft's stability and respond to the pilot's commands or autopilot settings.

5. **Actuator Dynamics and Output:** The final section simulates the behavior of the aircraft's actuators in response to the commands from the control system. It shows how the aircraft's movements would be adjusted according to the inputs and conditions modeled.

3.2 XIAO Seeed Studio BLE Sense with IMU:

Using the IMU (Inertial Measurement Unit) sensor fusion in the Xiao speed sense board at the beginning of the project was a strategic choice due to several factors:

1. **Outdoor Operation:** The IMU allows for orientation tracking without the need for an external motion capture system, which is essential for real-world applications outside of a controlled environment.

2. **Versatility:** IMUs are less dependent on external references and can provide real-time feedback on the drone's orientation, crucial for dynamic maneuvers such as hovering and climbing.

3. **Integration with Existing Systems:** An IMU can be a more straightforward integration than a system like Pixhawk, especially in the early stages of development when custom solutions may be more appropriate for specific design challenges.

At first, the built-in 6-axis IMU LSM6DS3 was used with a Madgwick filter to obtain the attitude from the accelerometer and gyroscope raw data. The main advantages of the

Madgwick filter are that it has a low error, is computationally inexpensive, and has better performance than the Kalman-based algorithms. However, since no magnetometer was available, yaw drift was dramatic, so an external 9-axis IMU BNO055 was used with the XIAO board. Figure 7 shows the setup, and Figure 8 compares the drift of the yaw using both IMUs on the same board.

However, since motion capture was chosen to be used, this board was replaced by Arduino Nano IOT 33, which has a WIFI module.

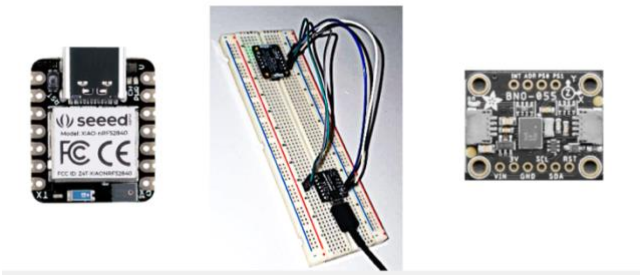

Figure 7: XIAO Seeed Studio with BNO055 IMU

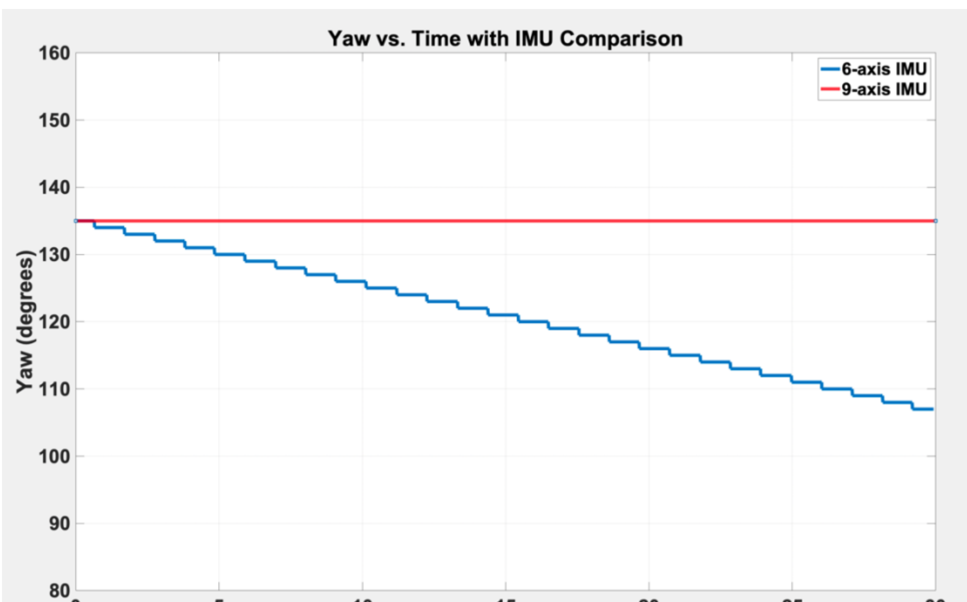

Figure 8: Yaw Drift of the 6-axis and 9-axis IMUs

3.3 Flatty Theory and Methods:

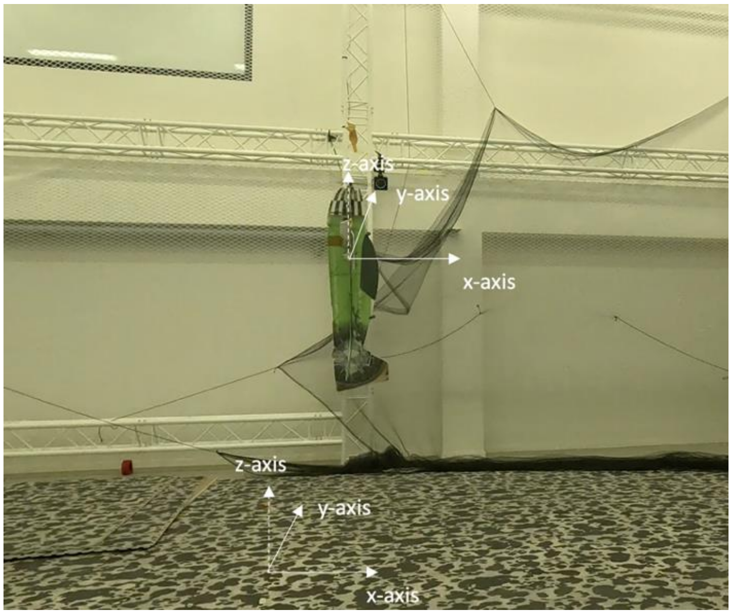

Figure 9: Flatty Setup with Coordinate Frames

The first platform we worked with, the Flatty, was the T-Motor AM20, F3P-A Airplane, a fixed-wing RC-sized drone with all its control surfaces being flat plates. This was used with a motion capture system, moving away from IMU integration. The Flatty drone serves as an intermediary research platform. Its traditional configuration with front-mounted propellers and ailerons allows for a more straightforward application of established control methods. By mastering control with the Flatty, one can develop a nuanced understanding of the PID control necessary for stable hovering.

Working with the Flatty enables to:

1. **Validate Control Algorithms:** test and refine the control algorithms in a more conventional setup before adapting them to the unique challenges of the Perchug.
2. **Iterative Design:** Progressing from a known, controlled environment to one with more variables allows for a methodical approach to problem-solving.

The flatty is symmetric in design, with the propeller being placed in front at its nose. The axis according to which the orientation and position of the flatty is considered (inertial frame) is presented in Figure 9. In a vertical position, at equilibrium, the orientation of the flatty is defined as zero pitch and yaw, see the axis of drone In Figure 9 (explained in setup later). When the flatty is in a vertical position, the propeller is in a horizontal position, the vertical thrust passes through the center of gravity, and thus no moments and compensation due to cog offset is needed as we will see in the Perchug. For the control of the drone to stay hovering in place, the equilibrium point around which to stabilize the system is needed. This equilibrium is defined by:

1. Yaw, pitch, and roll angles define to be zero in our convention
2. The deflection angle of the elevator, rudder, and ailerons.

For the deflection angle of the elevator and rudder, these are defined at zero where no pitching or yawing moment is needed at equilibrium. However, for the aileron, the deflection is not zero but is set to an angle that generates a rolling moment of opposite and equal direction to that generated by the roll moment generated by the propeller rotation. This is found by experimentation, as well as all the other gains needed since the method used to design the controller is a model-free one. The controller used to control the attitude, altitude, and position of the drone is a PID controller.

Before delving into the controller explanation, several assumptions are taken for the controller design:

1. **The couplings** between each of the control surfaces with each other, and with the propeller are neglected. In the work of Bilodeau 2009 [], at the identification step, a multivariable method was used to identify the models. It was found that the cross terms were negligible thus validating the modeling assumptions.
2. **The airflow** generated by the speed of the drone when climbing upwards is not comparable to that generated by the propeller, which can be supported using experimental data and the actuator-disk theory
3. **The system can be decoupled** into three subsystems, one by axis.
4. **Low-speed vertical flight** is assumed, therefore the force generated by the propeller is nearly equal to the gravitational force;
5. **Angular rates** for each considered subsystem are small;
6. **Acceleration terms** resulting from angular rate such as Coriolis acceleration are neglected;

An important thing to consider is the propeller rotational speed, which affects the airflow generated. Since the effect of the control surfaces changes with the airflow, a scaler is a good idea to change the PID gains used for different propeller speeds. However, since the climbing is at a low speed, the fluctuation of propeller speed and thus of the airflow is not of significance, as can be seen by the fluctuation of the throttle command after the drone has been tuned and climbs, with the range between maximum and minimum throttle, is 0.02 on a range of -1 to 1, (0.02/2 = 1% change). See Figure 10.

The convention of the x,y,z axis of the environment and the roll, pitch, yaw conventions will be presented in the setup section.

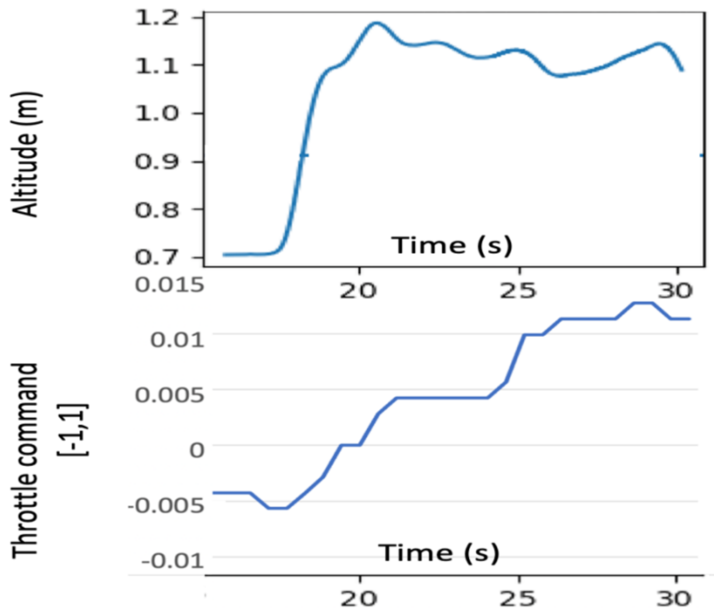

Figure 10: Throttle Command and altitude variation as a function of time

3.4 PID controller logic and Architecture:

In total, 6 PID controllers (2 cascaded PIDs consisting of two loops, and 2 classical PIDs) are used for the control of the drone:

1. A PID is used to keep a constant roll of the drone. The input is the aileron deflection, the reference is the initial roll of the drone, and the output is the roll. See the figure below.

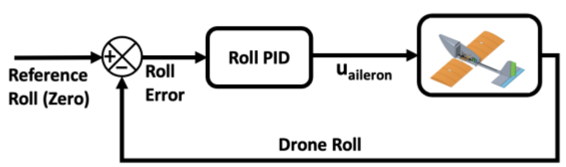

Figure 11: Roll and Aileron Feedback Loop

1. A PID is used to keep a desired altitude. This is done by taking the input as the throttle, the reference as the z position of the drone – the desired position, while the output is the z position of the drone. This is done to make sure the current z tracks the desired z. The throttle is then converted so that the vertical part of the thrust generated by the throttle is equal to the required input. See the figure below.

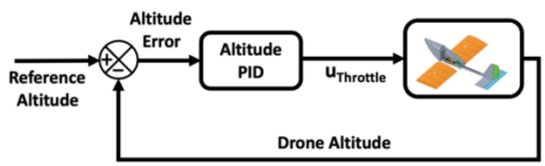

Figure 12: Altitude and throttle Feedback loop

2. A cascaded PID is used for the x-position control of the drone. This PID has an outer loop whose output is the x-position, the reference is the desired x-position, and the input is the reference pitch for the inner pitch control PID. This can be assumed since with an assumed fixed roll, the pitch mainly affects the x-position of the drone. The input than in the inner loop is the commanded elevator deflection. See the figure below.

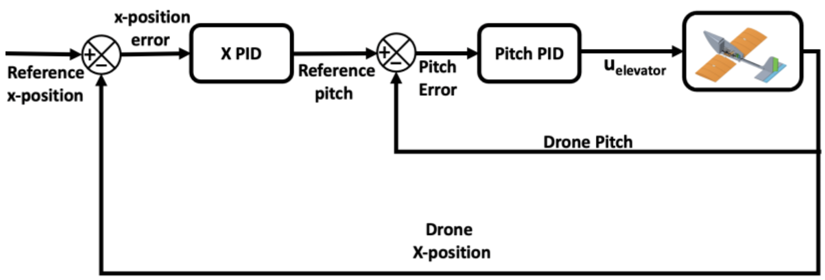

Figure 13: Cascaded x-position Feedback loop (pitch)

3. Similar to the x-cascaded PID, a similar Cascaded PID for the y-position, yaw, rudder deflection is constructed. See the figure below.

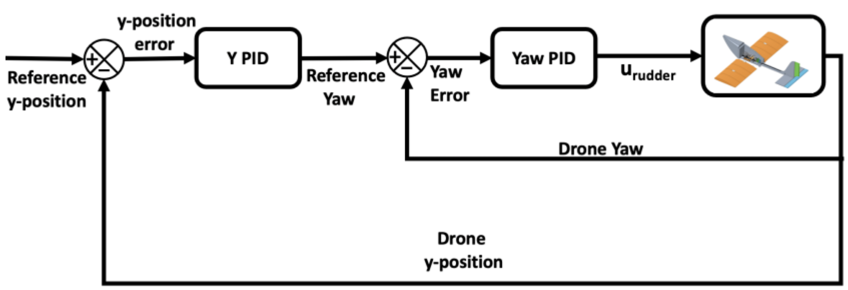

Figure 14: Cascaded y-position Feedback loop (Yaw)

4. The tuning of the PID gains for each of the loops is done using the classical method of studying the effect of the different PID gains. The main challenge is to choose the iterations to follow in tuning the different gains to reach a good hovering result.

3.5 Perchug Theory and Methods:

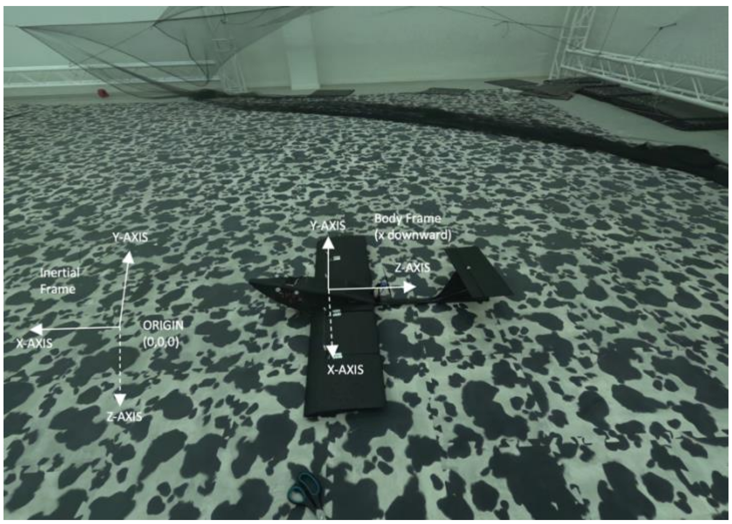

Figure 16: Perchug Frame definition

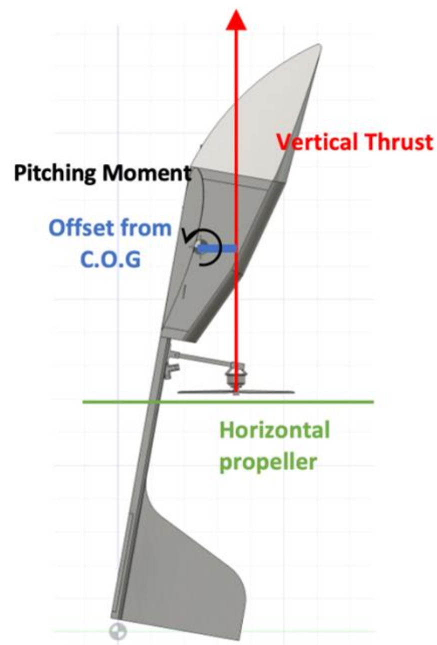

Figure 15: COG offset causing pitching moment

The theory and methods used for the Perchug are very similar to the Flatty, with some needed modifications that are very important.

Compared to the flatty, the Perchug has the propeller in the back of its fuselage. For the flatty, the pulling force is directed from upwards, which is like the way a normal pendulum acts, with a changing pivot point, but is an inherently stable system, even if the angle deflects too much, the system stays controllable. However, the Perchug, with the pushing force from below, is similar to an inverted pendulum configuration, similar to balancing a stick by hand, which makes the system inherently unstable. However, it is controllable and stabilizable as shown in experimentations.

As the thrust needs at equilibrium to consist of only a vertical component, the propeller ideally should be at a horizontal level. However, for the Perchug, at the horizontal propeller position, the vertical component does not pass through the center of gravity, as can be seen in Figure 15.

This causes a pitch moment which makes the Perchug rotate and lose stability, deviation from the equilibrium point. To compensate for this moment, an elevator offset is needed to give a counter moment, by generating a force that causes a moment of opposite and almost equal magnitude as the one caused by the offset of the center of gravity. Therefore, stabilization happens at this equilibrium point, with a deflected elevator angle. For the yaw, this offset doesn't affect them since the thrust happens to be in the center of the plane passing through the wings.

A challenge that arises at this point is that at different propeller speeds, the airflow changes, and the effect of the elevator thus changes, which necessitates the change of the offset in the elevator deflection. However, since the commanded thrust has a small range around the hovering thrust, especially for slow ascending of the drone, a constant offset can be used. For better results, gain scheduling is done, where for different thrusts, different offsets are given, with a good enough number of offsets used to be found by experiments.

Another approach is to use the flat plate model be used to estimate the force generated, with the actuator disk theory used to calculate the airflow. For the Perchug, we can't have big movements in xy position, since we can't have high deflection angle like flatty, so just small changes in xy, preferably if just z control.

4. Setup and Tuning:

The setup for the Perchug and the Flatty used is the same, so figures from both drones will be used interchangeably. The following main parts the consist the setup are the hardware, software, and physical setup.

4.1 Hardware and Software:

To obtain the position and orientation of the drone, the motion capture system at the LIS lab, uses the motive Opti-track software. For the MOCAP system to detect the drone, several steps are taken:

1- The source of information comes from several cameras at different positions which communicate with the motive software Figure 17. A calibration of the cameras is required. This was already available in the LIS lab computer.

2- To define a coordinate frame, an origin axis (inertial frame) should be defined. This is done by a 3-marker calibration rod, shown in Figure 17. The long axis is defined as the x axis, the shorter axis is define as the y-axis, and the z-axis is deduced.

3- Markers are used to detect certain point locations in the area. Figure 17.

4- To define the drone, 5 markers are stuck to the drone so that a rigid body is defined in the motive software. Figure 17.

5- The center of gravity should be explicitly defined, which is done by identifying it through placing a marker at the center of gravity. Figure 17.

With this setup, the MOCAP system can identify the position and orientation of the drone.

To obtain the information from the mocap to the local machine, ROS1 is used, where motive creates a topic “vrpn_client_node/rigid_body/pose” which publishes the pose of the drone Figure 17. The local machine Figure 17 obtains the pose by subscribing to this topic.

To make the process of tuning easier where I can observe the roll, pitch, and yaw angles and tune accordingly, and to avoid the problem of a gimbal lock, instead of using quaternions which is not easily interpreted, a transformation is done where the pitch is considered zero when the drone is vertically upwards (the body axis is rotated 90 degrees about the y-axis). (Fig. 5 and 12)

After obtaining the positions and Euler angles, these are used to calculate the commands from the machine.

The main board used for both drones is the Arduino Nano IOT 33 Figure 17, which is equipped with a wifi module. The Arduino listens to a socket on a certain port on its own IP address, to which the local machine sends the commands. These commands are mapped and sent to the 3 different servos to control elevator, rudder, and aileron. For the throttle, a servo is used to send a signal to a 30 Amp brushless ESC which converts the signal to the motor. The whole loop is shown in Figure 17. Python is used to calculate the commands on the local machine, while C++ is used to program the Arduino.

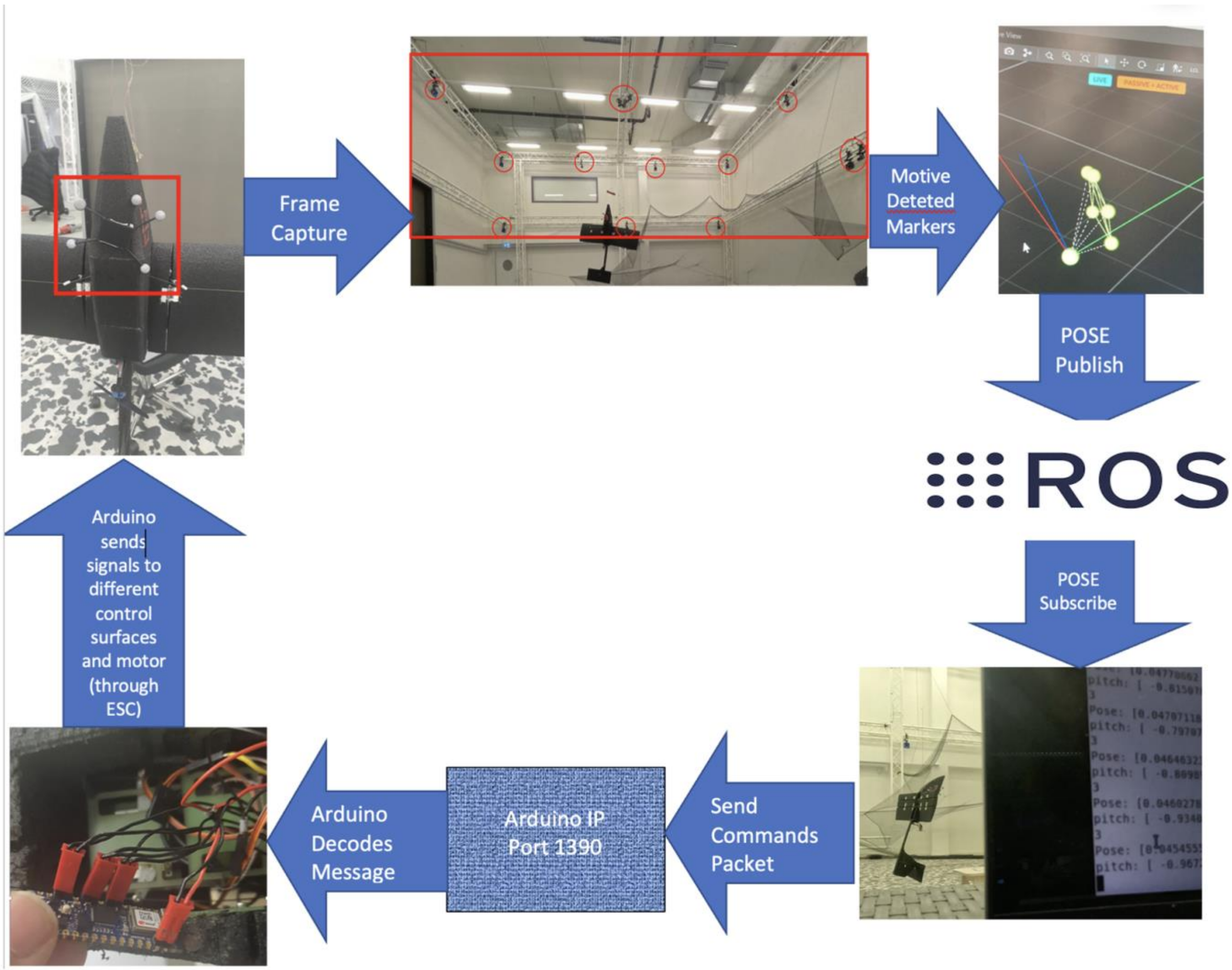

Figure 17: Software and Hardware Communication Setup

4.2 Flatty Setup for Experimentation:

To tune the PID gains for each of the different loops mentioned in section 3.4, a setup is needed where the Flatty is safe and at the same time has the freedom when it gains altitude to have all its degrees of freedom.

Because of the propeller, cannot be hanged using a single rope. Therefore, two ropes are used. These ropes are hung at the two wings symmetrically to the ceiling railing.
The way they are hung is that the rope hanging at the railing has an offset from both the wings, where it has an angle with the wing.

This is done for the following reasons:

1- This hanging is useful for tuning the pitch while restricting the other rotational degrees of

2- By hanging the drone from front and back similar to how the wings are hung, the yaw now can be tuned independently while the other rotational degrees of freedom are restricted.

The following is the process of tuning the different PIDs:

1- After hanging the wings, a +-10 degrees square wave is given as the reference for the pitch, and the gains are tuned to reach a reasonable response. See Figure 18 and Figure 19. The big jump at first is due to the initial hanging.
3- For the yaw feedback PID, a similar process is done as the pitch tuning. See Figure 20 and Figure 21.
4- After the pitch and yaw are tuned, the throttle and z position can now be tuned, a desired z position is given, and tuning is carried out.
5- While in the air and hovering, with the wires being slack, all degrees of freedom are now free, and therefore the roll and elevator PID is tuned.
6- Finally, the x and y position PIDs are tuned by changing the zero pitch and yaw references to the output generated by the x and y feedback loops.

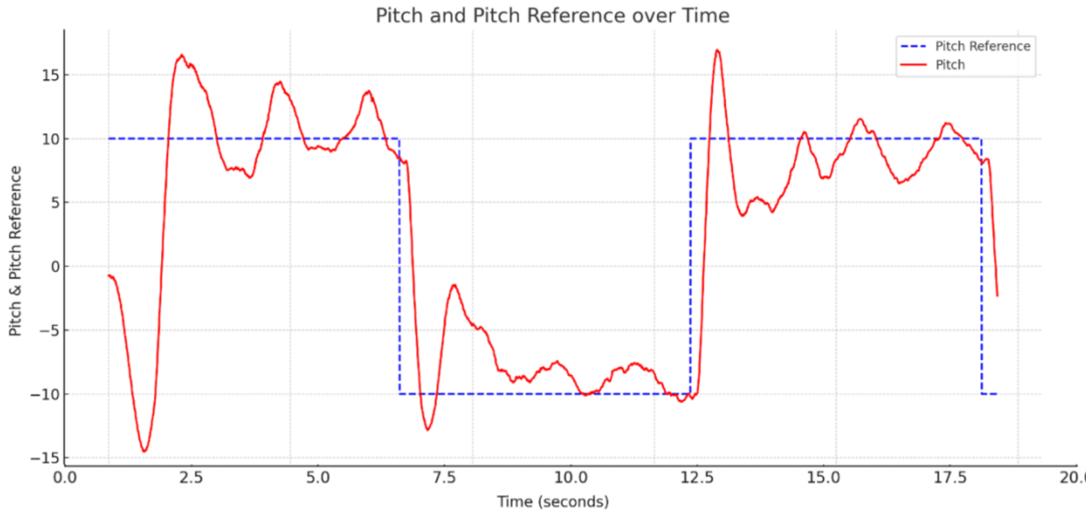

Figure 18: Pitch Tuning Response

Figure 19: Pitch Tuning

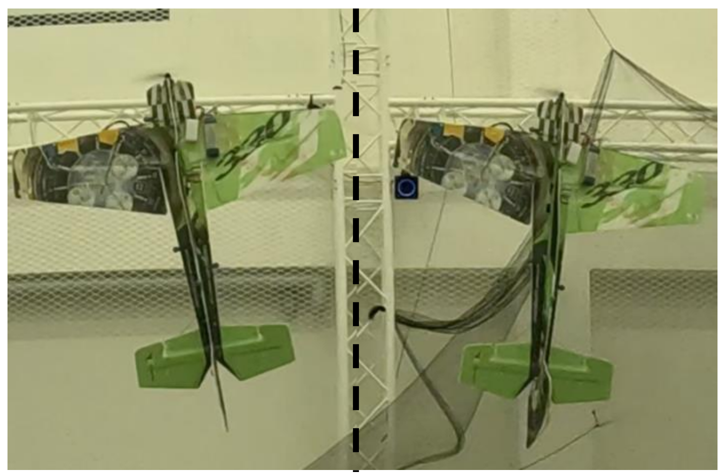

Figure 20: Yaw Tuning

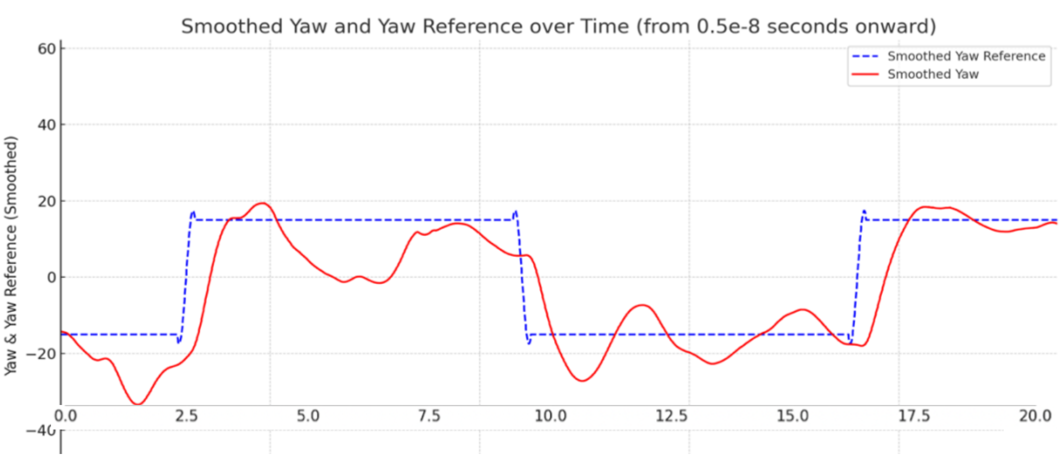

Figure 21: Yaw Tuning Response

4.3 Perchug Setup for Experimentation:

Figure 24: Yaw Tuning

Figure 22: Pitch Tuning

Figure 23: Perchug Setup

For the Perchug, several attempts were made to follow the same hanging and tuning process. However, there was no way to hang it at a point where the drone is initially at a position where the propeller is leveled horizontally and where the axis of rotation is about the C.O.G. This is necessary due to the Perchug being similar to Inverted pendulum, and it is initially required to be close to the equilibrium point to tune the PIDs. Therefore, it was chosen to tune the PIDs by holding the drone by hand and tuning it. See Figure 22 and Figure 24.

The process is very similar to that of the Flatty, but for the Perchug the offset of the elevator is to be also tuned. This is done by giving the hovering thrust command while holding the drone with a horizontal level of the propeller, and then changing the elevator deflection and checking at which threshold the drone starts flying backward instead of forward due to the pitching moment generated by the COG offset.

After it is found, the tuning is done in a similar way to that of the flatty (see Figure 23), but without a square wave, just a step response. The reference is the angle at which the thrust is horizontal, which depends on how the Perchug Is initially defined in the motive mocap. In this case, it is at -8 degrees pitch and 4 degrees of yaw. (See Figure 25 and Figure 26)

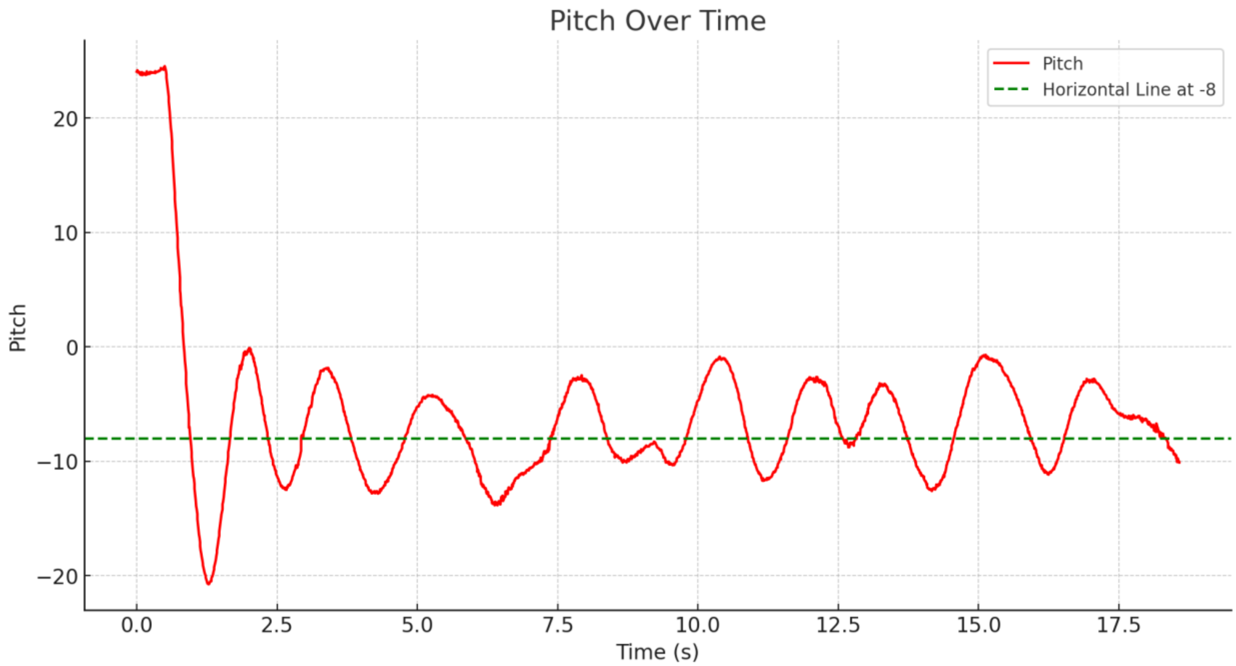

Figure 25: Tuned Pitch Response over time

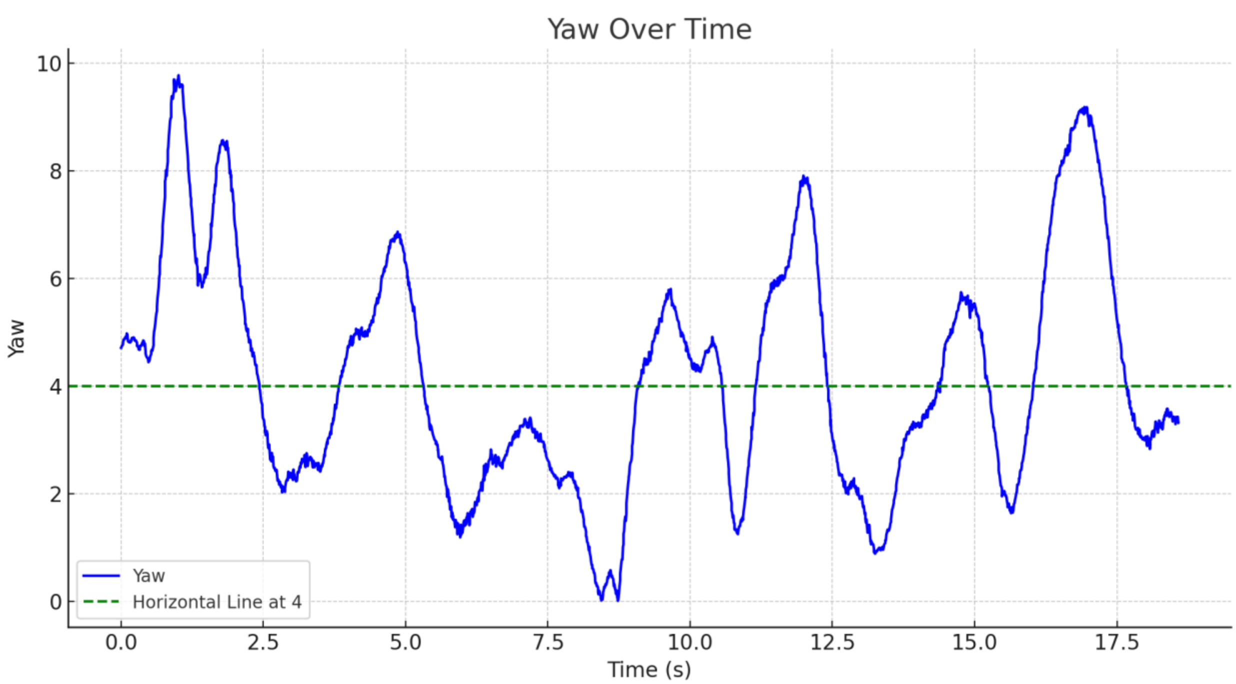

Figure 26: Tuned Yaw Response Over Time

One constraint is the effective control of the roll of the drone, due to the absence of generated airflow by the propeller to the wings. This can be tackled later by placing motors on the wings to compensate for the roll, which is discussed in the Future Work section.

5. Experimental Results:

5.1 Flatty Experimental Results

For the flatty, altitude control has been achieved, as well as position control. The results of an effective hovering state can be seen in Figure 27. where position control is achieved. Figure 28. shows the variation of the Roll, Pitch, and Yaw for this hovering. A video of the hovering can be seen here.

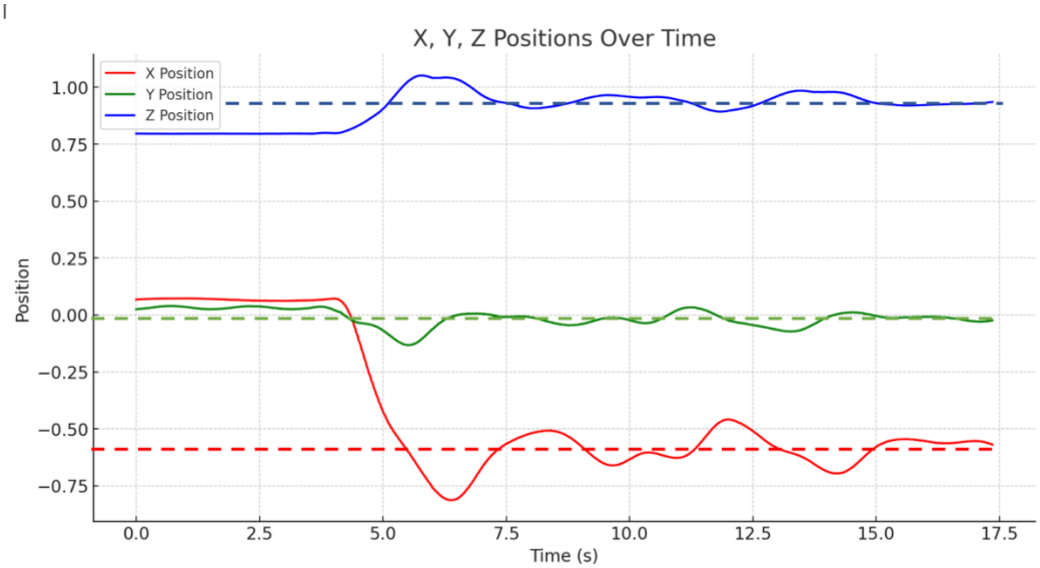

Figure 27: Position Response for the Hovering Flatty

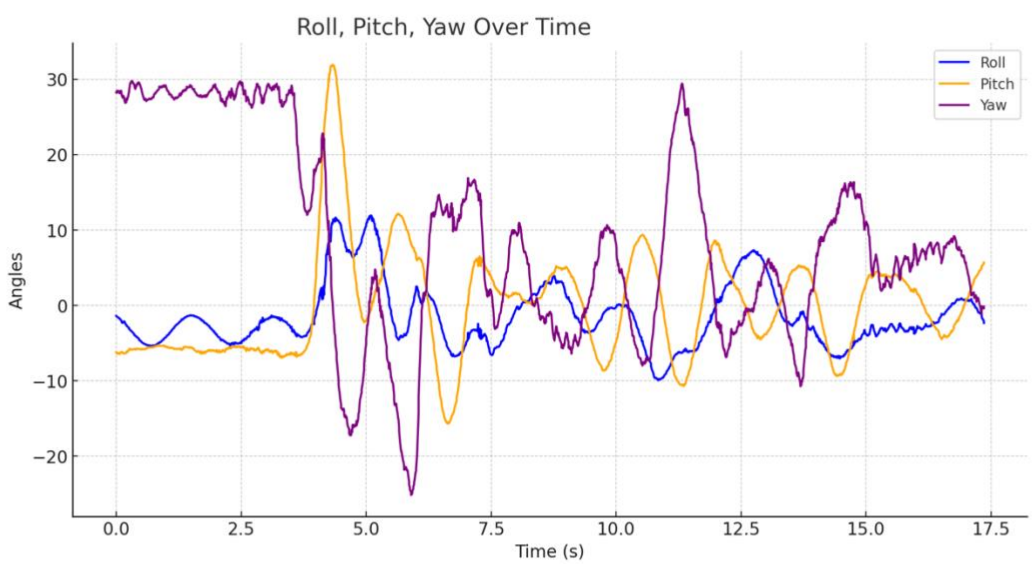

Figure 28: Attitude evolution during Flatty Hovering

5.2 Perchug Experimental Results:

For the Perchug, while pitch and yaw have been tuned, there is limited effect on the roll. Is it still possible to control the altitude, ideally with the drone being at equilibrium and spinning about the axis of symmetry. Testing has been carried out for pitch and yaw, but altitude control has not yet been tested. However, the results so far have been promising and hopefully, a full autonomous hovering Perchug will be shown in the presentation, although with a spinning due to limited control of roll.

The results show that the pitch and yaw control have an effect on the altitude and position, thus making the Perchug approach stability. The Video showing Perchug without active controller is shown [here](). The improved response with intermediately tuned PIDs is shown [here](). The plots without control are shown in Figure 31 and Figure 32. and the plots with control feedback are shown in Figure 29. and Figure 30. The improvement is evident.

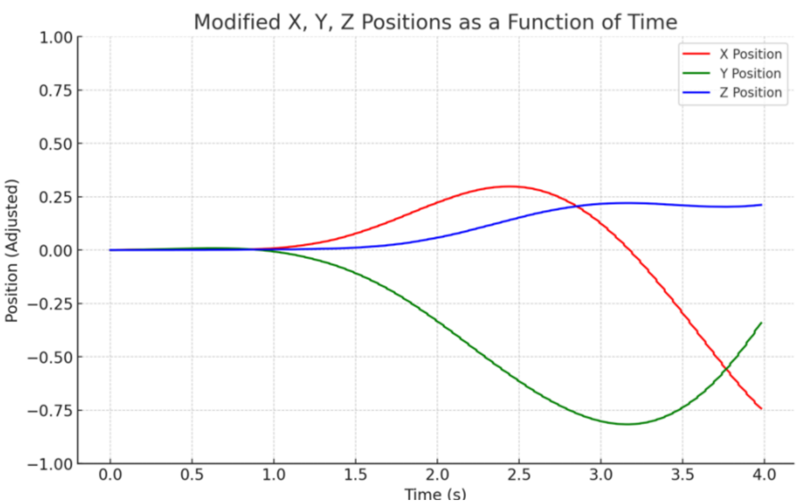

Figure 32: Position Response for Perchug without feedback control

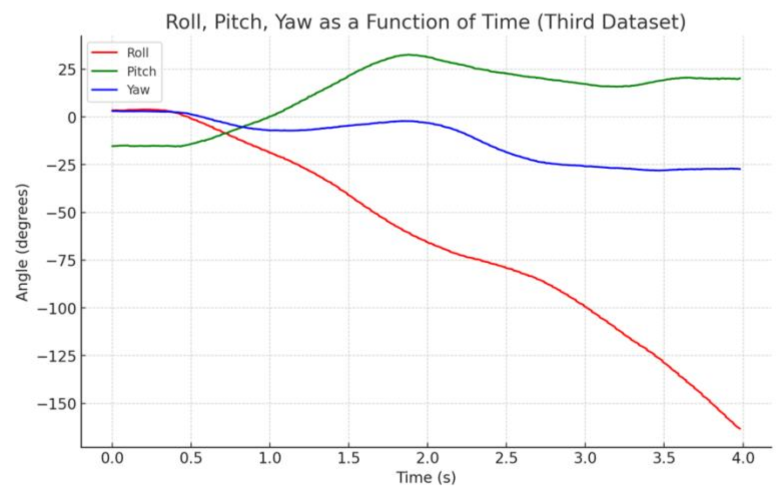

Figure 31: Attitude Response for Perchug without feedback control

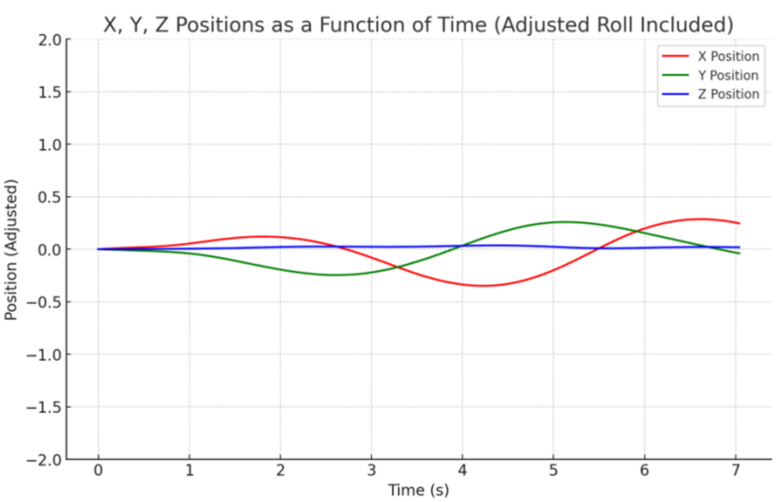

Figure 30: Position Response for Perchug with feedback control

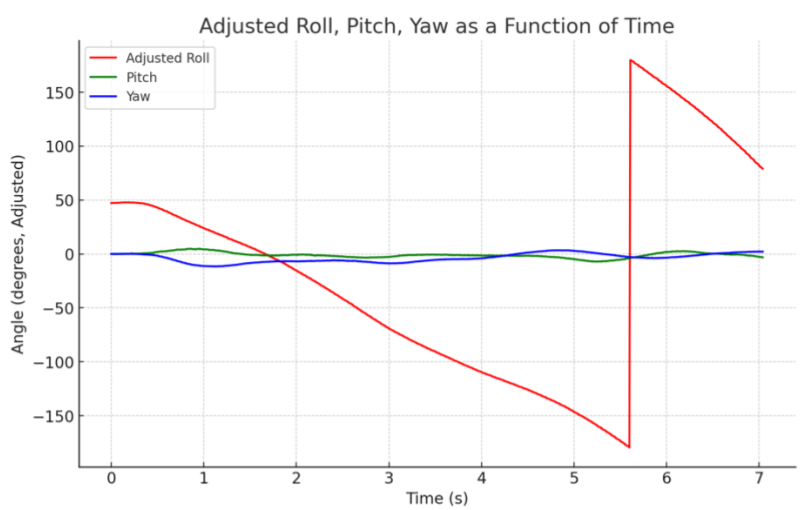

Figure 29: Attitude Response for Perchug with feedback control

6. Conclusions and Future Work

The project has successfully developed control strategies for the Perchug robot. Going forward, efforts will concentrate on enhancing roll control to achieve stable hovering. Further development aims to fully realize an autonomous hovering Perchug that demonstrates improved altitude and positional stability, contributing significantly to fields requiring elevated vantage points for surveillance and communication.

To address the lack of roll control, several innovative solutions could be explored:

1. Differential Thrust: Implementing motors on the wings that can provide differential thrust could actively control roll without traditional ailerons. By varying the power between these motors, the drone could induce a rolling motion to correct its orientation.

2. Vector Thrust: Modifying the main propeller to allow for vector thrust could provide another means of influencing the drone's roll. By directing the thrust in various directions, it could help in stabilizing or generating roll when needed.

3. Controlled motors on the wings in opposite directions, these motors can cause rolling moments and counteract the initial undesired moments.

7. References

[1] D. Mehanovic, D. Rancourt and A. L. Desbiens, "Fast and Efficient Aerial Climbing of Vertical Surfaces Using Fixed-Wing UAVs," in IEEE Robotics and Automation Letters, vol. 4, no. 1, pp. 97-104, Jan. 2019, doi: 10.1109/LRA.2018.2881433.

[2] D. Schmidt and K. Berns, "Climbing robots for maintenance and inspec- tions of vertical structures: A survey of design aspects and technologies," Robot. Auton. Syst., vol. 61, no. 12, pp. 1288–1305, 2013.

[3] M. A. Estrada et al., "Perching and vertical climbing: Design of a multimodal robot," in Proc. IEEE Int. Conf. Robot. Autom., 2014, pp. 4215–4221.

[4] G. C. Haynes et al., "Rapid pole climbing with a quadrupedal robot," in Proc. IEEE Int. Conf. Robot. Autom., 2009, pp. 2767–2772.

[5] S. Kim et al., "Whole body adhesion: Hierarchical, directional and dis- tributed control of adhesive forces for a climbing robot," in Proc. IEEE Int. Conf. Robot. Autom., 2007, pp. 1268–1273.

[6] M. P. Murphy, C. Kute, Y. Mengu ̈c ̧, and M. Sitti, "Waalbot ii: Adhesion recovery and improved performance of a climbing robot using fibrillar adhesives," Int. J. Robot. Res., vol. 30, no. 1, pp. 118–133, 2011.

[7] E. W. Hawkes et al., "Vertical dry adhesive climbing with a 100 × bodyweight payload," in Proc. IEEE Int. Conf. Robot. Autom., 2015, pp. 3762–3769.

[8] P.Beardsleyetal.,"Vertigo-awall-climbingrobotincludingground-wall transition," Disney Res., 2015.

[9] W. R. Provancher et al., "Rocr: An energy-efficient dynamic wall- climbing robot," IEEE Trans. Mechatronics, vol. 16, no. 5, pp. 897–906, Oct. 2011.

[10] G.A.Lynch,J.E.Clark,P.-C.Lin,andD.E.Koditschek,"Abioinspired dynamical vertical climbing robot," Int. J. Robot. Res., vol. 31, no. 8, pp. 974–996, 2012.

[11] B. D. Miller, P. R. Rivera, J. D. Dickson, and J. E. Clark, "Running up a wall: The role and challenges of dynamic climbing in enhancing multi- modal legged systems," Bioinspiration Biomimetics, vol. 7, no. 1, 2015, Art. no. 20160094.

[12] D. Ruffatto, "Discussion on climbing performance of a wheeled micro- climber from JPL," NASA Jet Propulsion Laboratory (JPL), Anytown, MN, USA, 2018.

[13] D. Brunner, "Extreme high power drone with highest climb rate," 2018. [Online]. Available: http://www.research-drone.com/en/ extreme_climb_rate.html.

[14] Green, W. E., & Oh, P. Y. (2009). A Hybrid MAV for Ingress and Egress of Urban Environments. IEEE Transactions on Robotics, 25(2), 97-104.

[15] Matsumoto, T., Kita, K., Suzuki, R., Oosedo, A., Go, K., Hoshino, Y., Konno, A., & Uchiyama, M. (2010). A Hovering Control Strategy for a Tail-Sitter VTOL UAV that Increases Stability Against Large Disturbance. In Proceedings - IEEE International Conference on Robotics and Automation.

[16] Wang, W., Zhu, J., & Kuang, M. (2017). Design Modelling and Hovering Control of a Tail-sitter with Single Thrust-vectored Propeller. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

[17] Frank, A., McGrew, J. S., Valenti, M., Levine, D., & How, J. P. (2007). Hover Transition and Level Flight Control Design for a Single-Propeller Indoor Airplane. In AIAA Guidance Navigation and Control Conference and Exhibit.

[18] De Blauwe, H., Bayraktar, S., Feron, E., & Lokumcu, F. (2007). Flight Modeling and Experimental Autonomous Hover Control of a Fixed Wing Mini-UAV at High Angle of Attack. In AIAA Guidance Navigation and Control Conference and Exhibit.