Control Stabilization for Bi-Quadcopter Payload Delivery System

Peter Blumenstein, Srikumar Brundavanam, Hima Hrithik Pamu, Shang Shi, Ben Spin

***Abstract*—With the increasing utilization and interest in unmanned aerial vehicles in both the private and commercial sector, steps to increase payload capacity and flight reliability present significant challenges and opportunities. This paper details the design, implementation, and testing of a novel dual drone system, interconnected via a robust truss structure, with a hanging pendulum for a centralized payload. The focus was on developing and tuning a Cascaded Proportional–Integral–Derivative (PID) controller within Simulink to manage the system's dynamics and ensure stability with variable payloads. The controller was further fine tuned iteratively on the hardware to address the unique dynamics introduced by the flexible truss structure and swinging payload. Field tests were conducted which demonstrated the system's capacity to withstand disturbance, handle variable payload weight, and perform simple turns and maneuvers stably. Our findings indicated that the dual-drone system significantly enhances payload capacity at 43 percent, and improves stability compared to a single drone solution. Therefore, we conclude that this system has potential application in a various fields, including package delivery, remote aid, and search and rescue operations.**

***Index Terms*—swarm-flight, PID, assisted payload, Cascaded Controller, Adaptive Control**

I. Introduction

Drones have emerged as one of the most promising technologies, with significant potential impacts across various sectors. In commercial package delivery, drones promise to substantially reduce labor costs and enhance delivery efficiency [1]. In healthcare, particularly in third-world countries with limited infrastructure, drones can play a pivotal role by delivering essential medical equipment and supplies, potentially saving lives [2]. However, a fundamental challenge in drone utilization is their limited payload capability, which restricts their ability to carry on-board loads [3]. The drone's capacity to carry weights is substantially influenced by its size, power, and design. This limitation becomes more pronounced in scenarios requiring the transport of heavier or bulkier items, as the drone's lifting and carrying capacities are stretched, affecting its viability in a wider range of uses.

In addressing this challenge, our project explores the control stabilization of a Bi-Quadcopter Payload Delivery System. We investigate whether linking two Quadcopters with a rigid structure can enhance their collective payload capacity. This approach may offer a viable solution to the payload limitation issue by distributing the load across multiple drones, potentially increasing the total weight that can be transported without compromising flight stability and efficiency. Our team drew inspiration from previous work in multi-drone payload delivery systems, building upon these foundations to advance the field further.

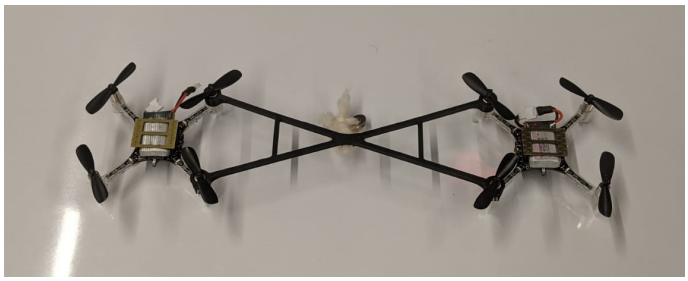

Fig. 1. Final assembly of both Crazyflie quadcopters using custom designed truss and magnet for payload pickup

II. Related Work

Given the widespread interest in drone application and the novel configuration of our system, our team has reviewed several research works related to dual-drone control. There have been significant advancements in controller design for multi-drone systems, with particular emphasis on the complexities of controlling multiple drones in unison and optimizing payload delivery efficiency. In this vein, our investigation has delved into research that address parallel challenges in the realm of multi-drone system control and efficient payload distribution. This exploration has laid the groundwork for an in-depth review of pertinent literature in this field, positioning our project within the broader context of current technological developments in drone control systems.

A. Rigidly Attached Quadcopters

Several papers address the control of rigidly attached quadcopters. [4] proposes a method to estimate, in an online and quick manner, the position and orientation of each quadcopter previous to any flight attempt. First, the method performs an estimation of the physical structure attaching the quadcopters by relying solely on information from the quadcopters' Inertial Measurement Units (IMU). The process involves refining IMU data to minimize noise, correct biases, and reduce drift in position and velocity readings. This refined data is crucial for ensuring the system's stability and precise path following capabilities, especially in scenarios involving quadcopters rigidly attached in different formations. Next, [4] introduces an adaptive controller architecture, where the controller parameters are obtained using Reinforcement Learning. In this system, the controller gains for the Proportional (Kp), Integral (Ki), and Derivative (Kd) aspects of the attitude and angular velocity PID controllers are dynamically adjusted. This adaptation is based on each quadcopter's specific position within the

system's coordinate framework. [4] introduces an alternate method where controller gains are calculated offline through simulations across various system configurations. These gains are then dynamically applied in real-time following the completion of the parameter estimation phase. [4] then utilizes a Motion Capture System (MOCAP) positioning system and the UWB-based Loco Positioning System (LPS) by Bitcraze to adapt their approach for real-time the experiments. Our key takeaways were the use of a rigid structure to join the two CrazyFlie drones and the use of a adaptive control to learn controller gains in real-time.

B. Cooperative Payload Transport

[5] addresses the problem of payload capacity and how the thrust limit of a single robot restricts the weight of the payload it can accommodate. This study explores a method for aerial transport of a suspended payload by two robots, using a leader-follower system, without relying on vision-based guidance or motion capture technologies. [5] employs state-of-the-art methods for transporting suspended payloads using strings, a technique chosen for its low cost and the minimal or negligible need for mechanical design modifications in both the vehicles and the loads. The follower drone uses its onboard inertial sensor to both estimate and maintain the formation, relying on feedback from this sensor and an analysis of the underlying dynamics of the coupled system. This work then introduces a two-stage estimation strategy based on Kalman filters, along with linear control laws, to effectively manage the under-actuated system of a leader, payload, and follower. These vision-less methods have been successfully tested in flight experiments. Their efficiency and robustness allow two lightweight drones, each under 100 grams and with limited computational capacity, to cooperatively transport payloads in both indoor and outdoor settings. While this aspect was not the primary focus of the paper, our key takeaways included the effective use of a string-suspended payload and the employment of passive rigid mechanisms, such as magnets, for payload pickup.

In light of the previous studies, our team recognized the potential for enhancing payload delivery systems. The primary contributions of our project are as follows:

1) Develop a Dual Drone Bi-Copter Payload Delivery System using the CrazyFlies connected using a rigid 3D printed Truss.
2) Stabilize the Control of the Dual Drone system for different flight phases.
3) Achieve a Successful payload pickup and flight using the Dual Drone system.

III. Methodology

A. Truss Design

A pivotal element in enhancing the payload capacity of our drone system was the development of a well-engineered truss. The truss served as a critical link between the drones, necessitating a design that was lightweight yet robust enough to maintain structural integrity and stability. To achieve this, we opted for Polylactic Acid (PLA) for the truss construction, a material known for its light weight and suitability for rapid prototyping through 3D printing. After several iterations, we finalized the truss design depicted in Figure 2.

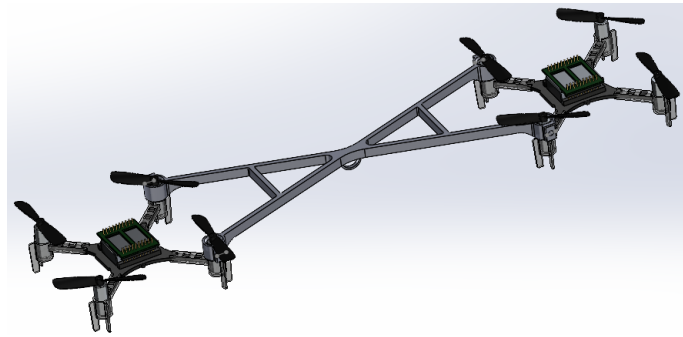

Fig. 2. SolidWorks Assembly of the Dual-Drone System. Note the cross shape of the truss allowing for an unobstructed view for the Flowdecks on the underside

Our design strategically positions the payload at the center, balancing it between the two drones. This positioning, coupled with the symmetry of the CrazyFlie drones' motors and arms, results in a balanced structure. Additionally, we incorporated set screws at all motor contact points to ensure firm and consistent attachment.

Given the CrazyFlie drones' limited thrust capacity, the truss's weight was a more significant factor than its structural properties. The weight at which the truss began to deform was well beyond the drones' lifting capabilities. Incorporating the truss markedly altered the system's dynamics, particularly affecting the roll and yaw movements. These dynamic changes are further explored in the System Modelling section (see Section III-B). The addition of the truss and payload also strained the drones' battery life.

As the truss was designed in SolidWorks, we were able to directly determine its Moment of Inertia (MOI) and weight. The calculated values are as follows:

TABLE I
MOI and Weight Specifications of the Truss

Parameter	Value	Units
Mass	7.931425	grams
I_{xx}	4221.77	g·mm^2
I_{yy}	143365.04	g·mm^2
I_{zz}	139371.19	g·mm^2

In this proof-of-concept phase, our truss met all our primary objectives: maintaining a manageable weight and ensuring flight stability and reliability. For future real-world applications, the truss design would require reassessment to withstand different loads, potentially necessitating a change in material or a redesign. The feasibility of dual-drone flight control in practical scenarios would hinge significantly on these modifications, underscoring their application-specific nature.

B. System Modelling

The following sections detail our approach to modeling the dual-drone system. Our primary objective is to develop a comprehensive model that facilitates precise control of the system linearized about hover. The goal of this effort is the formulation of a state space representation to be used in Simulink for controller tuning . Each subsection incrementally builds towards this final model, detailing the integral components and methodologies employed in the process.

1) State and Control Variables: Our team aligned our coordinate frame with the standard representation onboard the CrazyFlie. This allowed us to utilize standard commands and more easily interpret testing data. The following, Figure 3, is the standard axis and force representation for the CrazyFlie, with the curved arrows representing torques and the Z or Yaw axis orthogonal and pointing outwardly from the page.

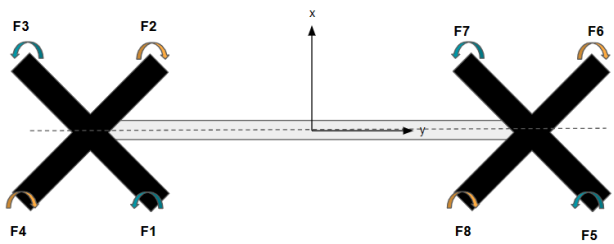

Fig. 3. Axis and Force Representation

We will first model the wrench as a function of these forces and torques as shown in Section III-B2, but will later utilize the relationship described in [6] to condense force and torque. This will allow us to represent our control vector as a function of individual rotor speeds, the standard onboard the CrazyFlie.

$$u = \begin{bmatrix}\Omega_1 & \Omega_2 & \Omega_3 & \Omega_4 & \Omega_5 & \Omega_6 & \Omega_7 & \Omega_8\end{bmatrix}^T \tag{1}$$

where

$$\Omega_i = \text{rotational velocity of rotor } i,$$
$$i = 1, 2, \ldots, 8.$$

where Ω_i represents the rotational velocity of rotor i (i.e., one through eight respectively).

The state variables of our system are standard and encompass both the linear and angular components of the drone's motion. They are defined as follows:

- **X-axis Position** (x) and **Velocity** ($\dot{x}$)
- **Y-axis Position** (y) and **Velocity** ($\dot{y}$)
- **Z-axis Position** (z) and **Velocity** ($\dot{z}$)
- **Roll** (ϕ), **Pitch** (θ), and **Yaw** (ψ)

The complete state vector of the system is defined as:

$$x = \begin{bmatrix}x & \dot{x} & y & \dot{y} & z & \dot{z} & \phi & \theta & \psi\end{bmatrix}^T \tag{2}$$

This state vector x represents the comprehensive set of variables required to describe the system's current state fully, including both its position and orientation in three-dimensional space, along with their respective rates of change.

2) MOI and Wrench Changes: Before fully defining the dynamics of our system, our team decided to construct a simplified model. This model assesses the relative increase in inertia versus the additional force generated by the second drone. These initial ratios provided a baseline for subsequent tuning on the hardware, a process detailed in Section IV-B. This analysis is included to illustrate the impact of the truss on our system.

Initially, two wrenches were defined to represent the states before and after the addition of the second drone:

- **Wrench 1 (Initial)**:

$$\begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \\ F_1\,\mathrm{Ly} - F_2\,\mathrm{Ly} - F_3\,\mathrm{Ly} + F_4\,\mathrm{Ly} \\ F_1\,\mathrm{Lx} + F_2\,\mathrm{Lx} - F_3\,\mathrm{Lx} - F_4\,\mathrm{Lx} \\ \tau_1 - \tau_2 + \tau_3 - \tau_4 \end{bmatrix}$$

- **Wrench 2 (Two Drone)**: The configuration and forces of Wrench 2 are analogous to those of Wrench 1, with the forces doubled to account for the addition of the second drone. The specific values and relationships are derived similarly to Wrench 1 and are omitted here for brevity.

In the wrench definitions, L_x and L_y represent the distances from the center of mass (COM) to the point of application of the forces along the x-axis and y-axis, respectively. These distances are crucial for calculating the torque generated by the forces relative to the COM, which in turn influences the rotational dynamics of the system.

With these wrenches defined, force and torque were then categorized as negative or positive based on their effect (negative or positive torque) along each axis. These values were summed and compared between the pre and post drone addition states to determine a wrench ratio.

- **Wrench Ratios**:
  - **Roll Ratio**: 5.45
  - **Pitch Ratio**: 2.0
  - **Yaw Ratio**: 2.0

Subsequently, the Moments of Inertia (MOIs) along each axis were calculated and compared:

- **MOI Ratios**:
  - **Roll Ratio**: 112.27
  - **Pitch Ratio**: 2.29
  - **Yaw Ratio**: 66.39

Utilizing these two sets of ratios, we derived a final ratio demonstrating the additional effort required from each rotor to achieve the same change in any direction post-modification. These final ratios are as follows:

- **Final Ratios**:
  - **Roll Ratio**: 20.58
  - **Pitch Ratio**: 1.14
  - **Yaw Ratio**: 33.19

3) Dynamics: In our dynamic model, Euler angles and Lagrangian mechanics were employed to formulate the equations of motion. These equations were linearized around the hover

state and then transformed into a state-space representation. Given their complexity, we present here a symbolic description of the key methodologies, rather than detailed matrices.

The state vector is split into $\mathbf{q}$ and $\dot{\mathbf{q}}$, where $\mathbf{q} = [x, y, z, \phi, \theta, \psi]^T$ represents the positional states and Euler angles (roll ϕ, pitch θ, yaw ψ), and $\dot{\mathbf{q}} = [\dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T$ represents their respective linear and angular velocities.

The G transform matrix in our model incorporates both rotation and translation aspects. It is used to apply rotational transformations to the position and orientation states represented by $\mathbf{q}$. This matrix is used to convert the local frame coordinates to the global frame. The G matrix is defined symbolically as:

$$\begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \tag{3}$$

where

$$\begin{aligned} R &= R_z(\psi) \cdot R_y(\theta) \cdot R_x(\phi), \\ R_x(\phi) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}, \\ R_y(\theta) &= \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, \\ R_z(\psi) &= \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\ p &= \begin{bmatrix} x & y & z \end{bmatrix}^T. \end{aligned}$$

Separate G matrices are defined for each CrazyFlie and the truss to account for their individual kinematic properties.

The Jacobian of these matrices plays a crucial role in translating changes in the state vector $\mathbf{q}$ into alterations in the system's velocity and angular momentum. Specifically, it maps the local coordinate tangent space to the global coordinate tangent space, allowing for the conversion of local motion and orientation changes into global dynamics. The Jacobian, which captures these relationships, is defined symbolically as:

$$J(\mathbf{q}) = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} & \cdots & \frac{\partial f_1}{\partial q_n} \\ \frac{\partial f_2}{\partial q_1} & \frac{\partial f_2}{\partial q_2} & \cdots & \frac{\partial f_2}{\partial q_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial q_1} & \frac{\partial f_m}{\partial q_2} & \cdots & \frac{\partial f_m}{\partial q_n} \end{bmatrix} \tag{4}$$

where $\mathbf{q} = [q_1, q_2, \ldots, q_n]^T$ represents the state vector, and $f_i(\mathbf{q})$ denotes the system's motion equations.

To capture the mass and inertial properties, each CrazyFlie and the truss is assigned a mass matrix M, defined symbolically as:

$$M = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_x & 0 & 0 \\ 0 & 0 & 0 & 0 & I_y & 0 \\ 0 & 0 & 0 & 0 & 0 & I_z \end{bmatrix} \tag{5}$$

where

$$\begin{aligned} m &= \text{mass of the object}, \\ I_x &= \text{moment of inertia about its roll axis}, \\ I_y &= \text{moment of inertia about its pitch axis}, \\ I_z &= \text{moment of inertia about its yaw axis}. \end{aligned}$$

With these matrices defined, the total kinetic energy of our system with respect to $\mathbf{q}$ and $\dot{\mathbf{q}}$ at any given time is represented by the equation:

$$T = \frac{1}{2}\dot{\mathbf{q}}^T J^T M J \dot{\mathbf{q}} \tag{6}$$

The potential energy of our system, considering gravitational effects, is simply:

$$U = g \cdot z \cdot (\text{total mass}) \tag{7}$$

where

$$\begin{aligned} g &= \text{acceleration due to gravity}, \\ z &= \text{height above the reference level (the floor)}. \end{aligned}$$

The Lagrangian $\mathcal{L}$ of the system, which is the difference between the kinetic and potential energies, is given by:

$$\mathcal{L} = T - U \tag{8}$$

To solve for a system of equations, we utilize the relationship given by the Euler-Lagrange equation:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \mathcal{Y} \tag{9}$$

where

$$\begin{aligned} \mathcal{Y} &= \text{input forces and torques, i.e., } B\mathbf{u}, \\ z &= \text{height above the reference level (the floor)}. \end{aligned}$$

Therefore, with the Lagrangian $\mathcal{L}$ defined, we will need to derive $\mathcal{Y}$.

In our model, $\mathcal{Y}$ is defined as the product of the transpose of the Jacobian matrix and the total force-torque vector F_T:

$$\mathcal{Y} = J^T F_T \tag{10}$$

where

$$\begin{aligned} F_T &= \text{sum of wrenches, from each motor, denoted as } F_{Ti}, \\ z &= \text{height above the reference level (the floor)}. \end{aligned}$$

Each F_{Ti} is given by:

$$F_{Ti} = \text{Ad}(G_i)^T \begin{bmatrix} 0 & 0 & F_i & 0 & 0 & \tau_i \end{bmatrix}^T \tag{11}$$

where

$$\begin{aligned} F_i &= \text{force generated by motor } i, \\ \tau_i &= \text{torque generated by motor } i, \\ G_i &= \text{transformation matrix for motor } i. \end{aligned}$$

The adjoint transformation of a matrix G, denoted as $\mathrm{Ad}(G)$, is defined as:

$$\mathrm{Ad}(G) = \begin{bmatrix} R & \hat{P}R \\ 0 & R \end{bmatrix} \tag{12}$$

where

$R =$ rotation matrix part of G,

$\hat{P} =$ skew-symmetric matrix form of the translation vector p.

With $\mathcal{Y}$ now defined, we employ the `solve` function in MATLAB to derive the equations of motion. This step is critical in constructing our state-space representation, as it provides the necessary equations to describe the system's behavior. These derived equations form the backbone of our control system design and analysis, enabling us to simulate and optimize the drone system's response under various conditions.

4) State Space: Our methodology for developing the state-space representation involves linearizing the system dynamics about the hover state. In MATLAB, this process is achieved by defining the system's equations of motion and then symbolically calculating their Jacobians with respect to the state variables and inputs. To linearize about the hover state, we substitute zero for all state derivatives and angular positions (phi, theta, psi) in these equations. This substitution simplifies the dynamics to a point where the system behaves linearly, allowing us to compute the matrices A and B for the state-space model. These matrices represent the linearized system dynamics and input relationships, respectively.

5) Takeaway: In conclusion, the detailed system modeling conducted in this project is of paramount importance to the overall success and viability of the Dual Drone Bi-Copter System. By developing a comprehensive dynamic model, we were able to achieve a nuanced understanding of the system's behavior. This modeling was crucial in designing an effective control strategy, allowing us to accurately predict and manage the system's responses. Furthermore, the precise control achieved through this modeling was instrumental in addressing the initial stability challenges, leading to successful payload pickup and delivery.

C. Controller Design

A single CrazyFlie is controlled by a well tuned cascaded PID controller. This cascaded PID controller consists of four PID layers:

1) Position
2) Velocity
3) Attitude
4) Attitude Rate

Given the reliable performance of the existing controller, we decided to utilize its structure with a couple key modifications and re-tune the PID values to adjust for the new system Dynamics. We also decided that operating in the drone's velocity tracking mode was most efficient, since that is the default mode. In velocity tracking mode, position references are converted to velocity and duration references such that the integral of the velocity gives the desired distance traveled. Utilizing this mode reduces the cascaded controller to only three layers, simplifying the tuning process and reducing error build up.

Within each block, except the PWM (Pulse Width Modulation) block, are three decoupled PID controllers that each act on a single input. For example, in the velocity block, the error in x-velocity is calculated and passed through the PID controller to output the pitch reference, since pitching is how a Quadcopter achieves longitudinal movement. The rest of the inputs/outputs follow the same logic. In the PWM block, the desired rotor speed components for each direction are combined for each rotor depending on its location. A more detailed explanation of this summation is given in the Firmware Alterations section IV-A.

D. Simulation Design

Using the state space representation of our dynamics model derived in section III-B4, we were able to create a simulation of the system using Simulink. The structure of our simulator was designed to replicate the cascaded PID controller onboard the CrazyFlie. The general block diagram for our simulation is shown in Figure 4.

Reference values for linear velocity and yaw were the inputs to the simulation, which aligns with the actual process on the hardware. The response of the dual-drone plant, with default PID values, to a square wave input in y-velocity is shown in Figure 5.

This result suggests that the system is unstable, but the physical system is able to respond to a square input, though not very accurately. Modifying the desired velocity slightly allows the simulation to achieve stability, which suggests the model is on the verge of instability. We attribute this discrepancy between simulation and reality to minor inaccuracies in the modeling and possible damping effects from the flexibility of the truss. As demonstrated later in section IV-B2, tuning the PID values in the simulation leads to successful results on hardware, suggesting that the simulator is adequate for our purposes.

E. Load Pickup

Achieving a successful payload pickup and delivery is a critical part of our project. For this end, we are using rigid "donut" magnets of to pickup a payload comprised of a heap of magnetic paper clips. Each magnet weighs roughly 3g and each paperclips weighs around half a gram (0.5g).

F. Adaptive Controller

We will design an adaptive controller to dynamically adjust the drone's PID settings in response to different payload weights. This adjustment will be based on the PWM values sent to the motors, which serve as an indicator of the payload's strain on the drone. For instance, a heavier payload necessitates higher PWM values to maintain hover. Our baseline PWM value for the Bi-Quadcopter system without payload is approximately 49500, while with an 8g payload, it increases to around

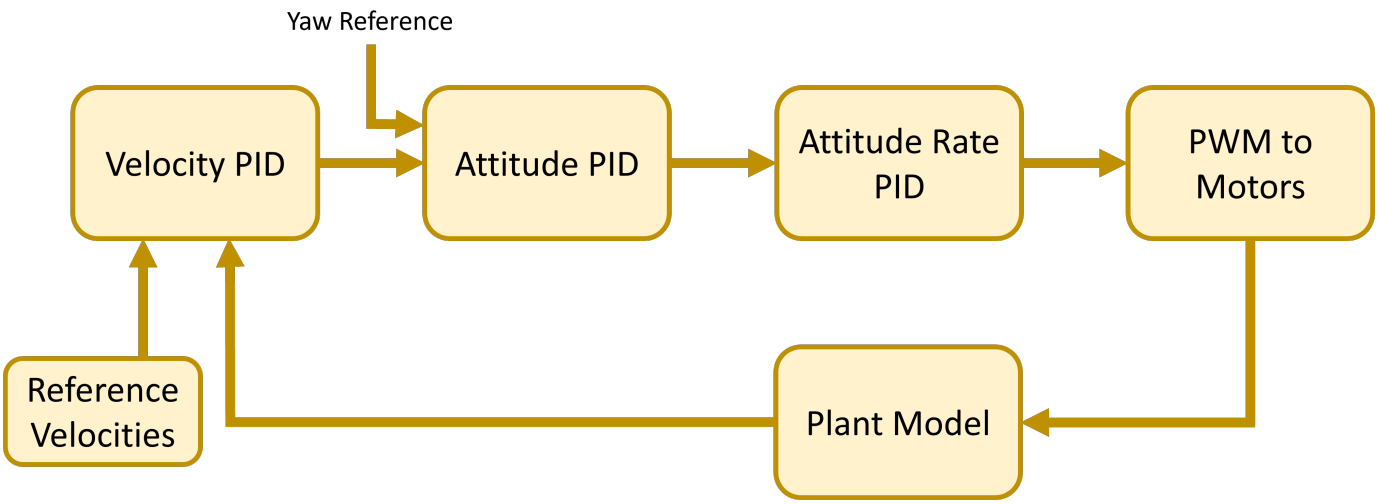

Fig. 4. Block diagram of simulation model

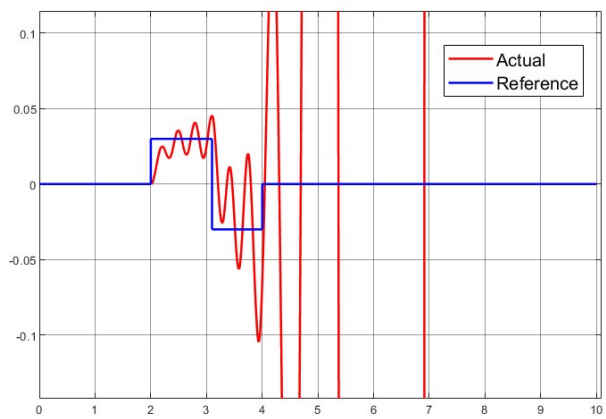

Fig. 5. Simulated response of system to y-velocity square wave reference

52000. After fine-tuning the PID controller for scenarios with and without the 8g payload, we plan to employ linear interpolation to determine appropriate PID values for varying payloads based on their corresponding PWM measurements.

This adaptive control approach is crucial for enabling the system to pick up and stabilize payloads of different masses, fulfilling a key operational requirement of our project.

IV. Experiments

A. Firmware Alterations

Before tuning the onboard Cascaded PID Controller, firmware changes had to be made. Specifically for the roll and yaw directions since the Moment of Inertia (MOI) changed the most by combining two Quadcoptors in these directions (see III-B2).

1) Roll: Without making any changes, the default power distribution follows the below equations where $motor_1$, $motor_2$, $motor_3$, and $motor_4$ are the front right, back right, back left and front left motors respectively:

$$\begin{aligned} motor_1 &= thrust - roll + pitch + yaw \\ motor_2 &= thrust - roll - pitch - yaw \\ motor_3 &= thrust + roll - pitch + yaw \\ motor_4 &= thrust + roll + pitch - yaw \end{aligned} \tag{13}$$

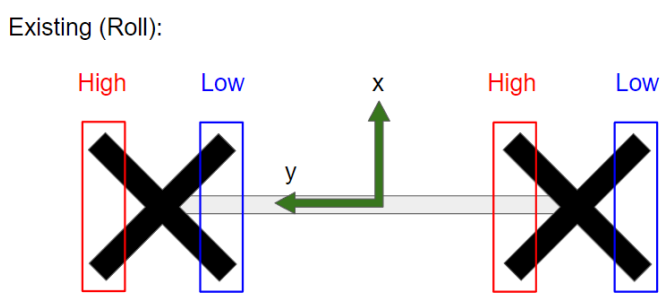

Fig. 6. Default power distribution in roll direction

In a dual drone configuration, the combined roll components are illustrated in Figure 7. However, this arrangement led to a significant issue: when calculating the torque around the center of the truss, we observed that the torques generated by each drone largely canceled each other out. This cancellation effect became evident during our initial tests, where achieving takeoff proved to be impossible due to the excessively slow reaction times resulting from this torque neutralization. To address this challenge, our team developed a new linear power distribution scheme, articulated through the following equations:

$$Left\ Drone$$

$$
\begin{aligned}
motor_1 &= thrust + 0.73 \times roll + pitch + yaw \\
motor_2 &= thrust + 0.73 \times roll - pitch - yaw \\
motor_3 &= thrust + 1.25 \times roll - pitch + yaw \\
motor_4 &= thrust + 1.25 \times roll + pitch - yaw
\end{aligned} \tag{14}
$$

Right Drone

$$
\begin{aligned}
motor_1 &= thrust - 1.25 \times roll + pitch + yaw \\
motor_2 &= thrust - 1.25 \times roll - pitch - yaw \\
motor_3 &= thrust - 0.73 \times roll - pitch + yaw \\
motor_4 &= thrust - 0.73 \times roll + pitch - yaw
\end{aligned} \tag{15}
$$

The ratios are calculated by assuming the roll force required at the center of each drone to be ± 1 and then interpolating for each motor based on its distance from the center of the truss. All the roll components on the left drone are positive since a positive force is required to create a positive moment in the roll direction. For the right drone, the reverse is true. Upon making this change to the power distribution, the drone was able to take off without any adjustments to the PID controller.

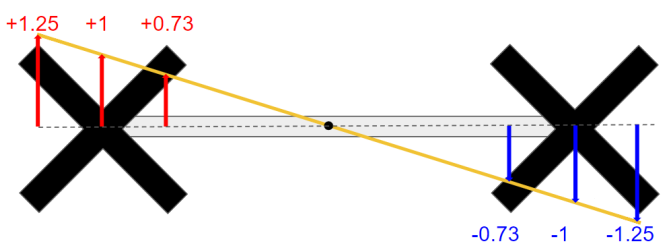

Fig. 7. Default power distribution in roll direction

2) Yaw: When a single drone yaws in place, there is no velocity in any of the x, y, and z directions. However, for the Bi-Quadrotor configuration, this is not the case. When the system yaws about its center, there is actually a velocity in the body x direction for both drones. Although through testing, even without any changes to the yaw direction, the bi-Quadrotor system can perform turning operations, however, it is not quite in place. It appeared that one drone was turning about the second drone, which turned out to be the case. The hypothesis was that both drones were fighting to yaw in place and due to a slight power imbalance, one of the drones always overpowered the other and remained stationary while the other drones revolved around it. This is visualized in Figure 8. The right drone is almost in place while the left drone rotates in a big circle.

To resolve this, we altered the velocity set-points in the x direction for both drones. The alteration follows the below equations where l is the distance from the center of truss to the center of drone:

Left Drone

$$Setpoint_{\dot{X}} = Setpoint_{\dot{X}} - Setpoint_{\dot{\psi}} * l \tag{16}$$

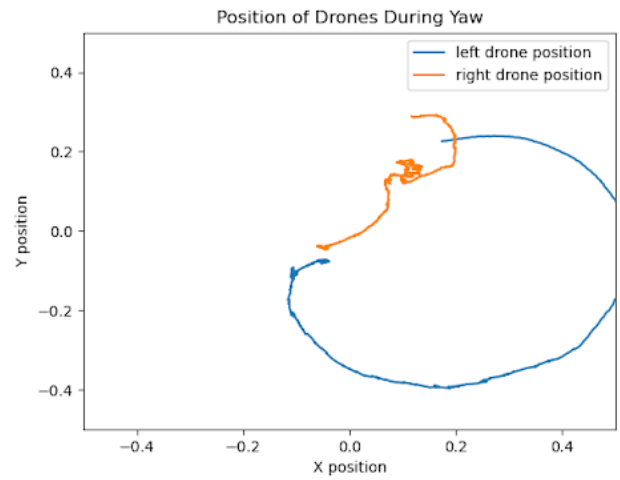

Fig. 8. x and y coordinates of both drones during yaw - default

Right Drone

$$Setpoint_{\dot{X}} += Setpoint_{\dot{X}} + Setpoint_{\dot{\psi}} * l \tag{17}$$

The only difference between the left and right drone equations are that for a positive yaw, the right drone experiences a positive x velocity while the left drone experiences a negative x velocity. Upon making this change, the drones can be seen in Figure 9 following our desired trajectory with the resulting yaw rotation about the center of mass.

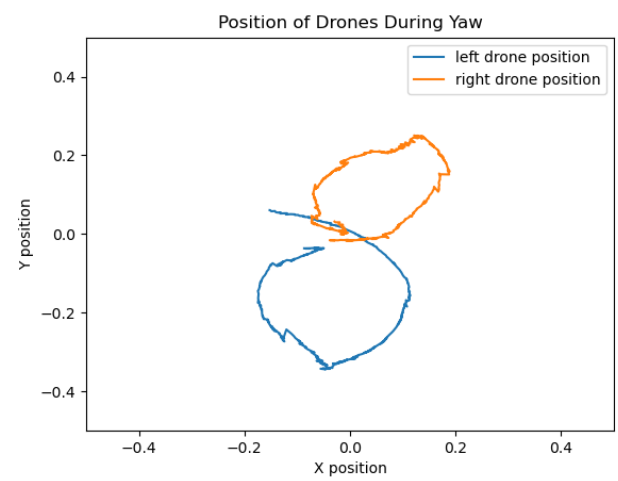

Fig. 9. x and y coordinates of both drones during yaw - post change

B. Testing/Tuning

Prior to initiating the tuning process, the payload capacity of the dual drone configuration was assessed. The truss, weighing approximately 8 grams, is considered an integral component of the dual drone configuration and is thus excluded from the payload calculations. In this context, the payload is comprised

solely of magnets and paper clips, which weigh 3 grams and 0.5 grams respectively. Our testing confirmed the achievement of the targeted increase in payload capacity. A single CrazyFlie drone demonstrated a carrying capability of approximately 7 grams, equivalent to the weight of two magnets and two paperclips. When configured in a dual drone setup, the payload capacity was enhanced to about 10 grams, representing an approximate improvement of 43% in payload capacity.

Tuning of the overall PID controller especially for the roll direction occurred through 4 stages. Firstly, we get a general intuition behind whether to increase or decrease the values based on the new mass moment of inertia (MOI) values of the Bi-Quadcoptor system. Next, we tune the simulated controller based on the intuition and get a set of PID values. With these values, we validate and finetune them on the hardware. Finally, with a well tuned baseline system, we start attaching payloads to the drone and tuning based on payloads.

As discussed in previous sections, both intuitive assessment and empirical testing identified roll and yaw as the primary contributors to unstable flight operations. While the yaw instabilities were mitigated by adjusting the setpoints, stabilizing roll required tuning the PID gains for roll rate within the Attitude rate block, as depicted in Figure 4. This tuning process was informed by our in-depth understanding of the firmware. Utilizing a Cascaded PID model in Simulink as discussed in Subsection III-D, we methodically tuned the roll rate PID gains to stabilize both pre and post payload pickup scenarios. Following these adjustments, the modified gains were rigorously tested and fine-tuned on our hardware. Additionally, we implemented a delay between flight operations to rectify instability caused by abrupt directional changes in flight.

1) Intuition with MOI: As discussed in Subsection III-B2, our initial step in controller tuning involved an approximate assessment of the dynamic changes experienced by the system in a hover state. This assessment was based on ratios that compared the increased inertia, which needed to be overcome, against the enhanced force capacity provided by the dual-drone system. Although these estimations were not sufficient to completely fine-tune our controller, they provided valuable insights into the relative increases in dynamics along each axis. These insights served as a foundational starting point for more precise tuning within our simulation environment.

2) Simulation: Since changing moments of inertia primarily affect the force or torque needed to move the drone, we focused our tuning efforts on the attitude rate PID controllers, since force and torque result directly from rotor speed. Using the intuition gathered from the MOI calculations, we tuned the roll, pitch, and yaw rate controllers until adequate performance was achieved in simulation. The response of the simulated system to a square wave in y-velocity after tuning is shown in Figure 10

These results show a time constant of 0.5 seconds for a square wave of 1 Hz. Though this is somewhat of slow response, we deemed it a sufficient starting point to implement on hardware. The final PID parameters for the attitude rate controller can be found in Table II

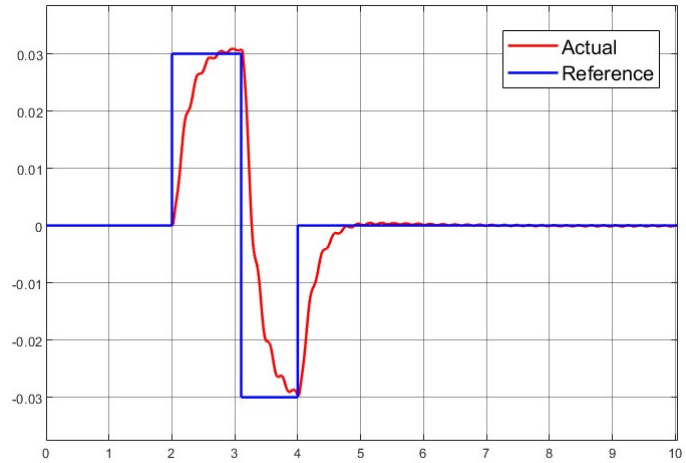

Fig. 10. Simulated response of tuned PID to square wave y-velocity

TABLE II
TUNED ATTITUDE RATE PID PARAMETERS

Direction	Kp	Ki	Kd
Roll	600	500	6
Pitch	250	500	2.5
Yaw	120	16.7	0

3) No Mass Tuning: Initially, the drone was calibrated without additional mass to minimize steady-state error prior to payload pick-up flight operations. Pre-tuning analysis revealed a mean error of approximately 8.5%, accompanied by notable flight oscillations and general instability as shown in 11. The operational parameters involved executing left and right translation movements over a distance of 2 meters at a velocity of 0.3 meters per second. It is important to note that the incorporation of a delay between directional changes in translation contributed to overall stability. As seen in Figure 12 after tuning, the mean error was reduced to 6.4%, alongside noticeable improvements in stability and a lower steady state error.

4) With Mass Tuning: Subsequently, the drone was calibrated with an approximate mass of 8 grams to reduce steady-state error during payload transport operations. The pre-tuning analysis, as referenced in 13, indicated a mean error of approximately 7.4%, along with flight instability comparable to the no-mass scenario prior to tuning. The control commands implemented were consistent with the no-mass tuning phase, involving lateral movements to the left and right over a distance of 2 meters at a speed of 0.3 meters per second. The implementation of a delay in the transition between directional changes was maintained, as this was found to enhance the stability of the drone. As illustrated in Figure 14, tuning successfully reduced the mean error to 5.8% and enhanced stability.

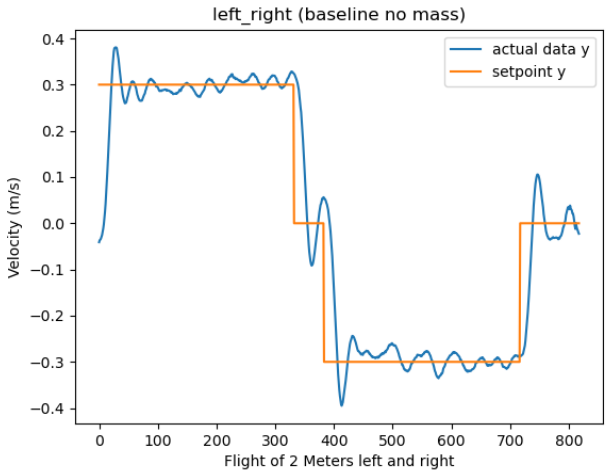

Fig. 11. Baseline left-right flight test (8.5% error)

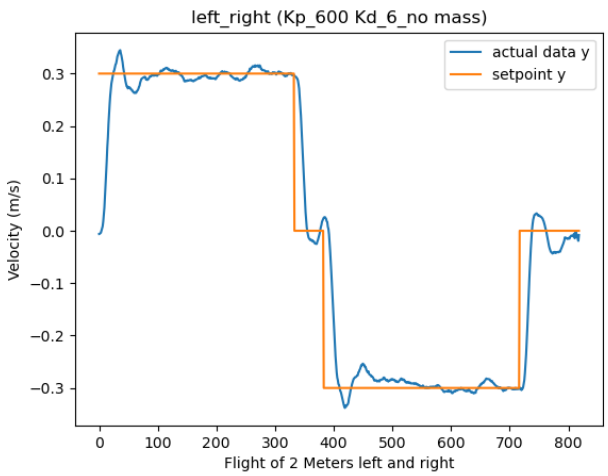

Fig. 12. Left-right flight test after PID tuning (6.4% error)

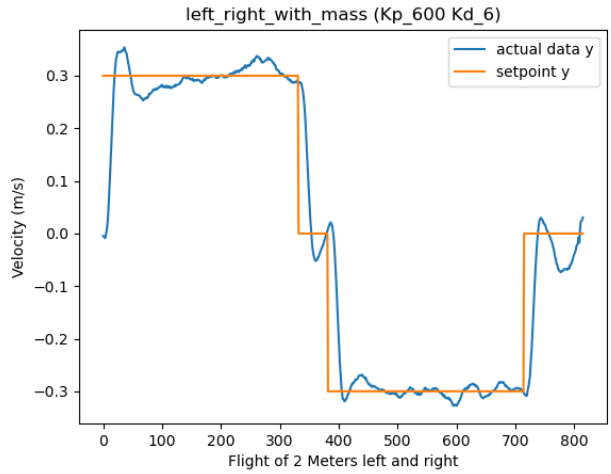

Fig. 13. Left-right flight test with 8g payload using no mass PID (7.4% error)

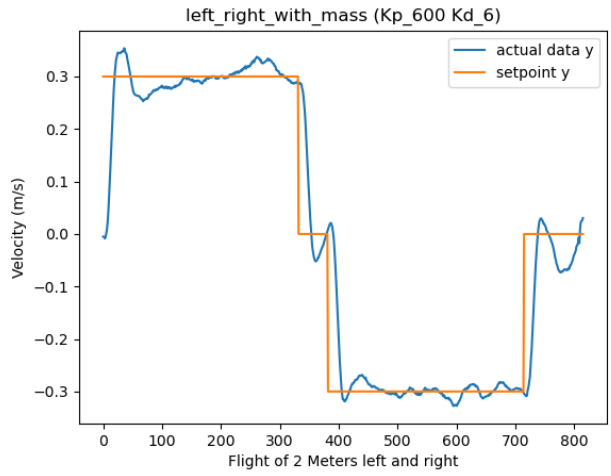

Fig. 14. Left-right flight test with 8g payload tuned (5.8% error)

V. Results

Based on the Firmware Alterations and Tuning sections, it can be clearly seen that the dual drone system has been stabilized and can support a higher payload than a single drone. The overall summary can be seen in table III.

In addition, upon applying the adaptive controller, the drone was indeed able to fly to the load, pickup the load, compare the baseline PWM to the measured PWM and scale the PID values. With the new scaled PID values, the drone was still able to complete the left-right motion test. Therefore the overall controller system successfully completed all the team's planned tasks.

TABLE III
Single vs Dual Drone Performance Overview

	Single Drone	Dual Drone	% Change
Payload Capacity	7g	**10g**	43% increase
Left-right Error%	7.8%	**5.8%**	26% decrease

CODE BASE: GitHub repository.

DEMO VIDEOS: YouTube.

VI. Conclusion

In conclusion, this paper has presented our project: a Dual Drone Bi-Copter System designed for efficient payload pickup and delivery. The system, integrating two CrazyFlie drones connected by a rigid 3D printed truss, features a payload delivery mechanism utilizing a rigid magnet for picking up items such as paperclips. We developed a comprehensive dynamic model, incorporating State Space Representation and the drones' onboard Cascaded PID controllers, to achieve precise control. Initial stability challenges were addressed through firmware modifications and the implementation of an Adaptive controller, which dynamically compensates for variations in payload mass. These enhancements led to the

successful stabilization and smooth control of the Dual-Drone system, showcasing its practical applicability. This project not only demonstrates technical innovation but also emphasizes the crucial role of iterative testing and adaptation in unmanned aerial vehicle development.

Looking ahead, our future work aims to expand this concept by integrating additional drones into the system. This expansion will explore how multiple drones can collaboratively manage larger or more complex payloads, thereby enhancing operational capabilities. Our focus will include the development of advanced rigid joints to connect multiple drones and the refinement of control systems for maintaining stability and efficiency as the system scales up. By broadening the scope of the system, we intend to delve deeper into the challenges and opportunities of multi-drone operations, potentially setting new benchmarks in aerial transport and delivery across diverse environments.

ACKNOWLEDGMENTS

We would like to express our sincere gratitude to Dr. Mark Bedillion, whose expertise, understanding, and patience, added considerably to our experience and understanding of control systems and their implementation. We deeply appreciate his willingness to spend time discussing concepts and guiding us through the complexities of this project.

We would also like to thank Dr. Aaron Johnson for his assistance in developing the dynamics model of our project. His willingness to share his profound knowledge and experience has been a great source of inspiration for us.

Additionally, we are indebted to Khai Nguyen, the teaching assistant for 24-774 Special Topics: Advanced Control Systems Integration for Fall 2023, for his invaluable assistance throughout this project. His insights and advice have been fundamental in the successful completion of this report.

REFERENCES

[1] J.-P. Aurambout, K. Gkoumas, and B. Ciuffo, “Last mile delivery by drones: An estimation of viable market potential and access to citizens across European cities,” European Transport Research Review, vol. 11, no. 1, 2019. doi:10.1186/s12544-019-0368-2

[2] J. E. Scott and C. H. Scott, “Models for drone delivery of medications and other healthcare items,” Unmanned Aerial Vehicles, pp. 376–392, 2019. doi:10.4018/978-1-5225-8365-3.ch016

[3] Mohsan, S.A.H., Othman, N.Q.H., Li, Y. et al. Unmanned aerial vehicles (UAVs): practical aspects, applications, open challenges, security issues, and future trends. Intel Serv Robotics 16, 109–137 (2023). https://doi.org/10.1007/s11370-022-00452-4

[4] D. G. Morín, J. Araujo, S. Tayamon and L. A. A. Andersson, ”Autonomous Cooperative Flight of Rigidly Attached Quadcopters,” 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 2019, pp. 5309-5315, doi: 10.1109/ICRA.2019.8794266.

[5] XIE, Heng ; DONG, Kaixu ; CHIRARATTANANON, Pakpong. / Cooperative Transport of a Suspended Payload via Two Aerial Robots with Inertial Sensing. In: IEEE Access. 2022 ; Vol. 10. pp. 81764-81776.

[6] Luis, C., & Le Ny, J. (August, 2016). ”Design of a Trajectory Tracking Controller for a NanoQuadcopter”. Technical report, Mobile Robotics and Autonomous Systems Laboratory, Polytechnique Montreal.