Quadcopter Load Suspension

1st Zane Meyer
Robotics
Oregon State University
Corvallis, Oregon
meyerzan@oregonstate.edu

2nd Clement Cantil
Robotics
Oregon State University
Corvallis, Oregon
cantilc@oregonstate.edu

3rd Roger Wienaah
Robotics
Oregon State University
Corvallis, Oregon
wienaahr@oregonstate.edu

***Abstract*—In this project, we modeled an autonomous quadrotor with a suspended load as a URDF in the PyBullet simulator and applied a Cascade PID controller to this system to reduce the swinging of the suspended load while ensuring that the quadrotor maintains its position and orientation, preventing any crashes.**

I. Introduction

Modern manned helicopters pick, place, suspend, and otherwise transport payloads to support construction, firefighting, resource extraction, and disaster relief operations [7]. Such loads span the range from ordinary crates of supplies to tranquilized animals. Helicopters are uniquely poised to rapidly deliver oversize cargo to remote areas that are difficult for other vehicles to access [7]. Crane helicopters are expressly designed to fulfill this function, boasting better mobility and, purportedly, payload positioning accuracy compared to ground cranes [7]. Notably, this attribute made crane helicopters the rapid payload delivery vehicle of choice for US military forces in Vietnam, in which crane helicopters securely transported destroyed aircraft, artillery pieces, troops, and various supplies [8].

With respect to delivery tasks, multi-rotor unmanned aerial vehicles have proven to be a scalable alternative to helicopters. Amazon has already begun to capitalize on this technology by using unmanned aerial delivery vehicles, specifically the MK30, to deliver parcels weighing five pounds or less to customers within the United States of America [9]. Currently, the MK30 contains parcels within its main body, simplifying system dynamics and protecting the payload for adverse weather conditions while simultaneously imposing limits on maximum parcel size.

Fig. 1. Amazon's MK30 [9]

Taken in aggregate, this information demonstrates that there is a niche for unmanned aerial vehicles that can transport suspended payloads. One task that demonstrates the existence of this niche is the installation of rooftop HVAC systems, which commonly involves helicopters that lift suspended HVAC units to the installation site at the top of tall buildings [10]. If such technology existed, renting a teleoperated multi-rotor drone capable of picking and lifting suspended HVAC units would present cost savings over renting a helicopter both in terms of transportation (a multirotor unmanned air vehicle might conceivably be shipped from the owner's facility to a worksite) and operation (the operator would not need to work as many hours or possess the same level of expertise [11]).

Unfortunately, when multi-rotor drones or helicopters pick suspended loads, their handling, safety, and agility sharply degrade. [12] Human pilots must focus on compensating for the movements of suspended payloads, resulting in longer transit times, increased costs, and increased workloads [12] [13]. It is not common for a manned helicopter that is traversing a great distance to designate a spotter or otherwise observe the position of the suspended load, which adds complications beyond merely changing the system dynamics [12].

In light of the above, we devote the remainder of the paper to investigating how to model the response of an autonomous quadcopter, a Crazyflie 2.0, with a suspended payload which we have simulated in PyBullet [1] using the gym-pybullet-drones package [2].

Fig. 2. Crazyflie 2.0 by Bitcraze [6]

II. Related Work

Controlling a quadcopter with a suspended payload presents a challenging problem due to the system's underactuated nature. Many researchers have proposed various approaches and developed controllers (e.g., proportional integral derivative controllers, linear quadratic regulators, robust control algorithms, etc.) to stabilize quadcopters and suppress the oscillations of suspended payloads.

An iterative linear quadratic regulator (iLQR) optimal controller was proposed by Alothman et al. [16] to control a single UAV with a cable-suspended heavy rigid body. The controller's outputs were designed to track a specific trajectory for a quadrotor with a load while preventing the load from swinging during transport. The proposed optimal controller was compared with an LQR controller. According to the results obtained from the simulation, the performance of the iLQR controller was faster with a smaller steady-state error than the LQR controller. On the other hand, Sreenath et al. [21] studied how to control the position of a payload when it undergoes large swings, achieving zero tension and finite moments. They designed a feedback controller and demonstrated that the system is differentially flat [22], making it suitable for controlling the UAV and payload system. Huo et al. [23] presented a double closed-loop control strategy that combined PID position control and an input-shaping method. This approach enabled the quadrotor to follow the desired trajectory while simultaneously eliminating load vibrations. This vibration reduction strategy was also integrated with other control methods. For instance, Kusznir et al. [24] incorporated input shaping into a sliding-mode-based controller, while Slabber et al. [25] utilized it in conjunction with an LQR controller. Reinforcement learning (RL) has also been used by some researchers to overcome the challenge of stabilizing a quadcopter with a suspended payload [17]–[19]. In these papers, the researchers utilize RL to track and regulate the trajectory of the suspended payload while aiming to construct swing-free paths. The RL controllers exhibit reduced payload swinging during agile dynamic movements along the target trajectory, indicating that this methodology is effective. However, this approach has the drawback of requiring a significant number of flights to learn the control strategy.

Some researchers have also employed hybrid controllers to solve this problem. For instance, Estevez et al [20] used a combined proportional derivative control and model predictive control to attain optimal trajectory following while achieving a swing free transportation of a double pendulum. Their proposed MPC-PD controller was compared against a cascaded PD controller (i.e., PD-PD controller) for quadcopter path tracking, but not the pendulum swing minimization.

For this project, we employed a cascaded PID controller to minimize the swinging of the load while ensuring the quadcopter maintains a steady hovering position.

III. Dynamical Model of the System

The dynamical model of the quadcopter with cable suspended load used in this project was adapted from the work of Sreenath et al [26]. They developed a coordinate-free dynamic model for the system by using rotation matrices to represent the quadcopter's attitude and the two-sphere for the load attitude representation, as shown in Figure 3 with the various symbols defined in Table I below.

The configuration of the system is defined by the position of the load relative to the inertial frame, the load's attitude, and the quadrotor's attitude. In our project, we utilized a taut cable system. With this setup, the system has eight degrees of freedom with configuration space $Q = SE(3) \times S^2$ and four degrees of underactuation. The quadcopter and load positions are related by

$$x_Q = x_L - l_Q \tag{1}$$

The Lagrange method is used to develop the equations of motion. The Lagrangian for the system, $\mathcal{L} : TQ \to \mathbb{R}$, is defined by $\mathcal{L} = \mathcal{T} - \mathcal{U}$, where $\mathcal{T} : TQ \to \mathbb{R}$ and $\mathcal{U} : Q \to \mathbb{R}$ are the kinetic and potential energies of the mechanism, respectively, and are defined as,

$$\mathcal{T} = \frac{1}{2} m_Q v_Q \cdot v_Q + \frac{1}{2} m_L v_L \cdot v_L + \frac{1}{2} \langle \hat{\Omega}, J_Q \hat{\Omega} \rangle, \tag{2}$$

$$\mathcal{U} = m_Q g e_3 \cdot x_Q + m_L g e_3 \cdot x_L, \tag{3}$$

where v_Q is obtained as the derivative of (1).

The dynamics of the system satisfy the Lagrange-d'Alembert principle.

$$\delta \int_0^\tau \mathcal{L} dt + \int_0^\tau (\langle W_1, \hat{M} \rangle + W_2 \cdot f R e_3) dt = 0, \tag{4}$$

where f is the thrust magnitude, M is the moment vector, and $W_1 = R^T \delta R$, $W_2 = \delta x_Q = \delta x_L - l \delta q$ are variational vector fields [27], with the infinitesimal variations satisfying [28], [29].

TABLE I
Definition of various symbols used in the paper [26].

Symbol	Definition
$m_Q \in \mathbb{R}$	Mass of the quadrotor
$J_Q \in \mathbb{R}^{3\times 3}$	Inertia matrix of the quadrotor with respect to the body-fixed frame
$R \in SO(3)$	Rotation matrix of the quadrotor from body-fixed frame to the inertial frame
$\Omega \in \mathbb{R}^3$	Angular velocity of the quadrotor in the body-fixed frame
$x_Q, v_Q \in \mathbb{R}^3$	Position and velocity vectors of the center of mass of the quadrotor in the inertial frame
$f \in \mathbb{R}$	Magnitude of the thrust for the quadrotor
$M \in \mathbb{R}^3$	Moment vector for the quadrotor in the body-fixed frame
$m_L \in \mathbb{R}$	Mass of the suspended load
$q \in S^2 \subset \mathbb{R}^3$	Unit vector from quadrotor to the load
$\omega \in \mathbb{R}^3$	Angular velocity of the suspended load
$x_L, v_L \in \mathbb{R}^3$	Position and velocity vectors of the suspended load in the inertial frame
$l \in \mathbb{R}$	Length of the suspension cable
$T \in \mathbb{R}$	Tension in the cable.

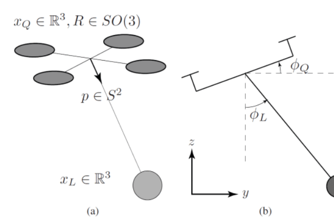


Fig. 3. A quadrotor with a cable suspended load. When the cable is taut, the system evolves on $SE(3) \times S^2$, and has 8 degrees of freedom with 4 degrees of underactuation.

$$
\begin{aligned}
\delta q &= \xi \times q, \quad \xi \in \mathbb{R}^3 \text{ s.t. } \xi \cdot q = 0 \\
\delta \dot{q} &= \dot{\xi} \times q + \xi \times \dot{q} \\
\delta R &= R\hat{\eta}, \quad \eta \in \mathbb{R}^3 \\
\delta \hat{\Omega} &= \hat{\dot{\eta}} + [\hat{\Omega}, \hat{\eta}]
\end{aligned}
$$

with δq being a variation on S^2, and δR a variation on $SO(3)$. Since (4) is satisfied for all possible variations, the equations of motion for the quadrotor with cable-suspended load are obtained as

$$\dot{x}_L = v_L \tag{5}$$

$$(m_Q + m_L)(\dot{v}_L + ge_3) = (q \cdot fRe_3 - m_Q l(\dot{q} \cdot \dot{q}))q \tag{6}$$

$$\dot{q} = \omega \times q, \tag{7}$$

$$m_Q l \dot{\omega} = -q \times fRe_3, \tag{8}$$

$$\dot{R} = R\hat{\Omega}, \tag{9}$$

$$J_Q \dot{\Omega} + \Omega \times J_Q \Omega = M \tag{10}$$

The above dynamics can be written in the standard form, $\dot{X}_n = f_n(X_n) + g_n(X_n)u$ where $X_n = \{x_L, q, R, v_L, \omega, \Omega\}$ is the state, and $u = \{f, M\}$ the input of the system respectively.

The load attitude dynamics (8) can also be written directly in terms of the load attitude, $q \in S^2$, and its derivatives as,

$$m_Q l \ddot{q} + m_Q l(\dot{q} \cdot \dot{q})q = q \times (q \times fRe_3). \tag{11}$$

This equation for the load attitude dynamics is used for control design. Unfortunately, the linearization of these dynamics equations produces a controllability matrix that is not full rank. Therefore, it is not linearly controllable.

IV. Controller Design

To effectively control the quadcopter with a suspended payload, we developed a cascaded PID controller to stabilize the load (i.e., reduce the oscillation of the load) while the quadcopter maintains a certain hovering position without crashing. The controller structure for tracking the load position is shown in Figure 4. We defined the tracking errors for the position and velocity of the load respectively as,

$$e_x = x - x_d \tag{12}$$

$$e_v = v - v_d \tag{13}$$

where $x_d(t) \in R$ is some desired load position, and $v_d = \dot{x}_d$. In this specific case, that position will be defined as directly below the drone. By defining the error relative to the position of the drone, it allows the drone to wander freely, simplifying to a controllable problem.

V. Simulation and Results

A. URDF Generation

We generated a URDF file for a quadcopter with a suspended load in accordance with the Robot Operating System's XML specifications [14]. PyBullet imported the URDF and used it to construct a representation of the desired system out

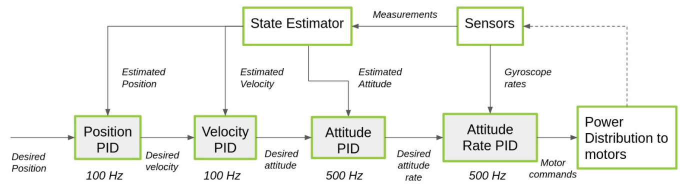


Fig. 4. Cascade PID Controller Block Schematics [5]

of surface meshes and properties, like inertia, quantified in the URDF file.

We built on the Crazyflie 2.x model that gym-pybullet-drones is configured to use out of the box. Our intention was to minimize the time spent modeling the electrical dynamics of the chosen quadrotor while maintaining a high degree of verisimilitude regarding the actuation of the motors. Principally, we did not want to investigate how signal pulse width related to angular velocity in rotations per minute for similar unmanned aerial vehicles as we felt this was getting too far into the weeds for purposes of this project. Using the default Crazyflie 2.x model had the added benefit of allowing us to include drag and ground effect in our simulation for no additional man-hours, leading to a more realistic simulation overall.

The default Crazyflie 2.x model includes a base link (for the frame of the quadrotor), four child links (one for each rotor), and four continuous joints. While each link is defined by standard link property categories (visual, inertial, and collision) and each joint is defined by standard joint properties (name, type, parent link, child link, axis xyz, origin xyz, etc.), the robot as a whole is assigned special properties for modeling specific to gym-pybullet-drones (coefficients for converting rotations per minute to thrust, speed limits, ground effect and drag coefficients, etc.). Here, the entire visual model (mesh reference and scale) of the default Crazyflie 2.x is supplied by a COLLADA file and color specification in the base link description. While the other link descriptions contain inertial (origin, mass, and inertia) property information, they do not contain visual or collision property information meaning those aspects are ignored. In short, the rotors do not spin because they are not modeled separately from the frame except in terms of their inertial properties.

To this model, we add four geometric primitives and four joints. First, we connect a sphere of radius 0.005 meters to the origin of the base link using a continuous joint in the x-plane. Second, we connect this sphere to the end of a cylinder of radius 0.005 meters and a length of 1 meter using a continuous joint in the y-plane. We repeat this process for the other end of the cylinder, using two continuous joints on opposite ends of a 0.005 meter sphere to connect the payload, a box of size 0.5 meters by 0.5 meters by 0.5 meters, to the rest of the model.

By using arbitrarily small spheres with two continuous joints in perpendicular planes normal to the axis of connection (here, the z-axis) as described above, we approximate a ball joint, which the URDF file format does not natively support.

These spheres have no collision properties and very small masses and rotational inertia elements.

The cylinder approximates a cable that is always taut. Again, we assign this primitive very small mass and rotational inertia elements.

We assign the box a mass and compute its 3D rotation tensor in accordance with standard procedures [15].

B. Simulator

We chose to use the PyBullet platform [1] with the gym-pybullet-drones package [2] to simulate the flight of our Crazyflie 2.x model with a suspended payload.

PyBullet is an open-source real-time physics engine designed for VR, games, visual effects, robotics, machine learning, and similar applications [1]. Balancing physics simulation and model visualization, PyBullet allows users to watch processes unfold while record data that is relevant to real-world application. The gym-pybullet-drones package extends this functionality to quadrotor systems by seeking to create an environment in which conventional control theory can be compared to reinforcement learning for both single-agent and multi-agent quadrotor systems. Crucially, PyBullet and gym-pybullet-drones both make use of Python, allowing our entire team to dive into the code base without much spin up time. While we do use the following functionality, it is worth noting that gym-pybullet-drones boasts parallelizablility and multi-agent reinforcement learning capability, further increasing our group's desire to learn how to use the software [2].

To run our code, we made several changes to the gym-pybullet-drones package. For the most part, these changes overrode the drone model restrictions that the original coders put in place to catch errors in model entry. Once that was complete, we identified all code related to controller programming and modified that code to tune our cascaded PID Controller.

C. Experimental Trials and Results

To prove the utility of our cascading PID controllers over an entirely uncontrolled system, we established an initial condition of the drone as stationary and hovering and an initial condition of the load as at an angle of $\pi/2$ rad about the x-axis to provide the maximum disturbance. Since the linearization of this non-linear system has a controllability matrix that is not full rank, it is not controllable with a linear controller. As such, the goal of these cascaded PID controllers is not to stabilize the entire system, but rather to delay the instability and allow the system to maintain a stable position for as long as possible. The PID controllers were tuned by hand and slowly improved to provide the best apparent results.

Looking at the system as the position of the load relative to the drone, as seen in Fig. 5, the system is simply a pendulum and is slowly stabilizing due to the natural damping in the system. If the drone can remain sufficiently stable for this period, the load will eventually come to rest. The instability is more apparent when looking at the true position of the drone over time. In Figs. 6-8, the drone's position is entirely unstable, but it is clear that the PID controllers are having a mitigating effect on the position error, especially in the x and z axes.

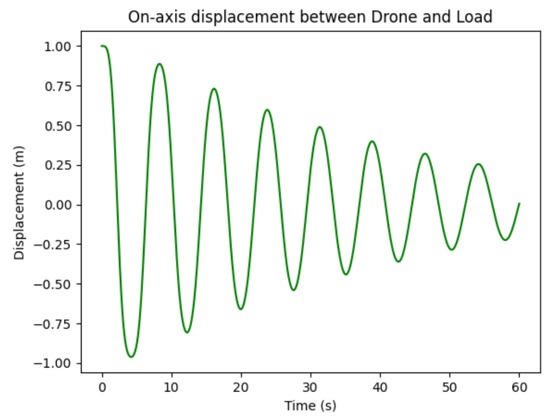


Fig. 5. Displacement of the load relative to the drone

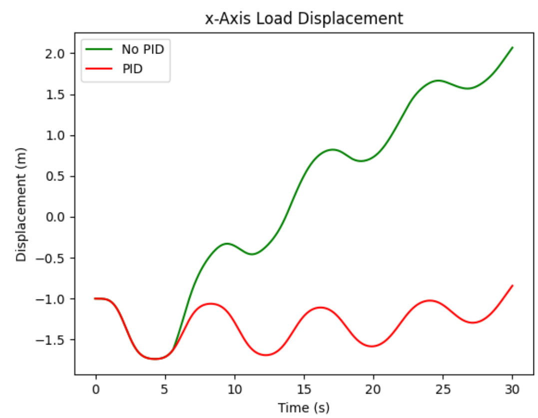


Fig. 6. True displacement of the load along the x-axis (non-motion plane)

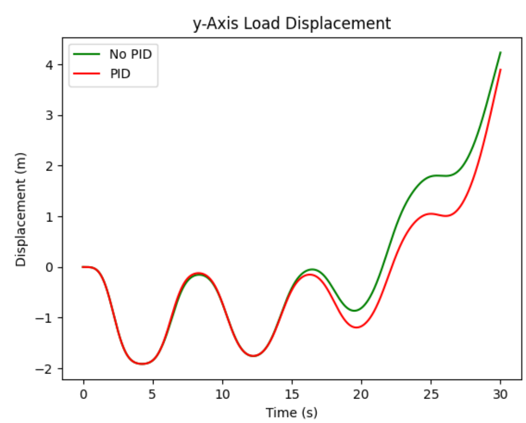


Fig. 7. True displacement of the load along the y-axis (motion plane)

VI. Conclusion

This project successfully implemented a Cascade PID controller in an attempt to control an autonomous quadrotor with a suspended load. The results were simulated in PyBullet and accurately reflected both an unstable system and the amount that these PID controllers were able to stabilize the system.

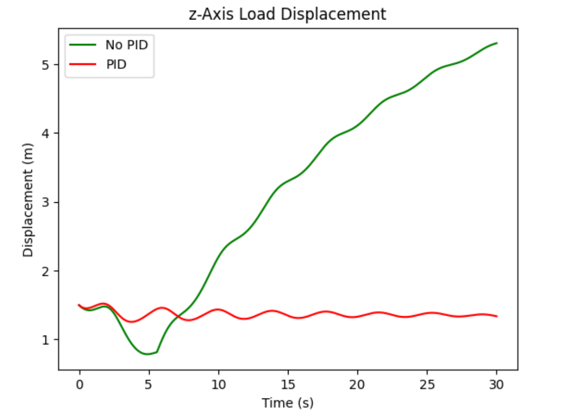


Fig. 8. True displacement of the load along the z-axis (motion plane)

Future work would extend the use of this particular PyBullet library to implement non-linear control schemes in an attempt to fully control this system.

REFERENCES

[1] E. Coumans et al., ”PyBullet, a Python module for physics simulation for games, robotics, and machine learning,” 2016-2021

[2] J. Panerati et al., ”Learning to Fly - a Gym Environment with PyBullet Physics for Reinforcement Learning of Multi-agent Quadcopter Control,” arXiv Preprint, 2021, doi: 10.48550/arXiv.2103.02142

[3] K. Sreenath et al., ”Geometric control and differential flatness of a quadrotor UAV with a cable-suspended load,” 52nd IEEE Conference on Decision and Control, Firenze, Italy, 2013, pp. 2269-2274, doi: 10.1109/CDC.2013.6760219.

[4] Y. Alothman et al., ”Quad-rotor lifting-transporting cable-suspended payloads control,” 2015 21st International Conference on Automation and Computing (ICAC), Glasgow, UK, 2015, pp. 1-6, doi: 10.1109/IConAC.2015.7313996.

[5] Bitcraze AB, ”Controllers in the Crazyflie,” Bitcraze Documentation, [Online]. Available: https://www.bitcraze.io/documentation/repository/crazyflie-firmware/master/functional-areas/sensor-to-control/controllers/. [Accessed: 08-Jun-2025].

[6] Bitcraze, ”Crazyflie 2.0,” Bitcraze.io, [Online]. Available: https://www.bitcraze.io/products/old-products/crazyflie-2-0/. [Accessed: Jun. 8, 2025].

[7] ”Why Helicopter Crane Services Are Essential for Remote and Urban Projects,” FairLifts, Apr. 24, 2025. [Online]. Available: https://www.fairlifts.com/cranes/why-helicopter-crane-services-are-essential-for-remote-and-urban-projects/

[8] ”History of the Sikorsky Skycranes,” Siller Helicopters, Inc., Accessed Jun. 8, 2025. [Online]. Available: https://sillerhelicopters.com/history-of-the-sikorsky-sky-cranes/

[9] T. Greenawalt, ”How Amazon delivers packages using its new drone MK30,” About Amazon, Dec. 11, 2024. [Online]. Available: https://www.aboutamazon.com/news/operations/mk30-drone-amazon-delivery-packages

[10] Corporate Helicopters, ”HVAC Lifts,” Corporate Helicopters, [Online]. Available: https://corporatehelicopters.com/helicopter-services/helicopter-utility/hvac-lift/. [Accessed: Jun. 8, 2025].

[11] Federal Aviation Administration, ”Become a Certificated Remote Pilot,” FAA.gov, [Online]. Available: https://www.faa.gov/uas/commercial_operators/become_a_drone_pilot. [Accessed: Jun. 8, 2025].

[12] Serpas Siguenza, J. (2013). Model-Following Control for a Helicopter with Suspended Load in Low Velocity Forward Flight (Master’s thesis, University of Calgary, Calgary, Canada). Retrieved from https://prism.ucalgary.ca. doi:10.11575/PRISM/26934

[13] Cheetham, J. L. and Buckingham, A. J., Aircraft Accident Report, Civil Aviation Authority of New Zealand, 1998

[14] Open Source Robotics Foundation, ”urdf,” ROS Wiki, [Online]. Available: https://wiki.ros.org/urdf. [Accessed: Jun. 8, 2025].

[15] https://en.wikipedia.org/wiki/List_of_moments_of_inertia#List_of_3D_inertia_tensors

[16] Alothman, Yaser, and Dongbing Gu. “Quadrotor transporting cable-suspended load using iterative linear quadratic regulator (ilqr) optimal control.” 2016 8th Computer Science and Electronic Engineering (CEEC). IEEE, 2016.

[17] Faust, Aleksandra, et al. ”Learning swing-free trajectories for UAVs with a suspended load.” 2013 IEEE International Conference on Robotics and Automation. IEEE, 2013.

[18] Faust, Aleksandra, et al. ”Automated aerial suspended cargo delivery through reinforcement learning.” Artificial Intelligence 247 (2017): 381-398.

[19] Palunko, Ivana, Rafael Fierro, and Patricio Cruz. ”Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach.” 2012 IEEE international conference on robotics and automation. IEEE, 2012.

[20] Estevez, J.; Lopez-Guede, J.M.; Garate, G.; Graña, M. A Hybrid Control Approach for the Swing Free Transportation of a Double Pendulum with a Quadrotor. Appl. Sci. 2021, 11, 5487. https://doi.org/10.3390/app11125487

[21] Sreenath, K.; Michael, N.; Kumar, V. Trajectory generation and control of a quadrotor with a cable-suspended load—A differentially-flat hybrid system. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 4888–4895.

[22] Fliess, M.; Lévine, J.; Martin, P.; Rouchon, P. Flatness and defect of non-linear systems: Introductory theory and examples. Int. J. Control 1995, 61, 1327–1361.

[23] Huo, Xin, et al. ”Vibration elimination for quadrotor slung system based on input shaping and double closed-loop control.” 2019 12th Asian Control Conference (ASCC). IEEE, 2019.

[24] Kusznir, Tom, and Jaroslaw Smoczek. ”Sliding mode-based control of a UAV quadrotor for suppressing the cable-suspended payload vibration.” Journal of Control Science and Engineering 2020.1 (2020): 5058039.

[25] Slabber, J. F., and Hendrik W. Jordaan. ”Vision-Based Control of an Unknown Suspended Payload with a Multirotor.” 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2021.

[26] K. Sreenath, T. Lee and V. Kumar, ”Geometric control and differential flatness of a quadrotor UAV with a cable-suspended load,” 52nd IEEE Conference on Decision and Control, Firenze, Italy, 2013, pp. 2269-2274, doi: 10.1109/CDC.2013.6760219.

[27] J. Milnor, Morse Theory. Princeton: Princeton University Press, 1963.

[28] T. Lee, M. Leok, and N. H. McClamroch, “Geometric Tracking Control of a Quadrotor UAV on SE (3),” in IEEE Conference on Decision and Control, Atlanta, GA, 2010, pp. 5420–5425

[29] F. Bullo and A. D. Lewis, Geometric Control of Mechanical Systems. New York-Heidelberg-Berlin: Springer-Verlag, 2004.

Quadcopter Load Suspension

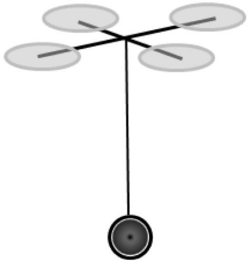
Clement Cantil, Zane Meyer, Roger Wienaah

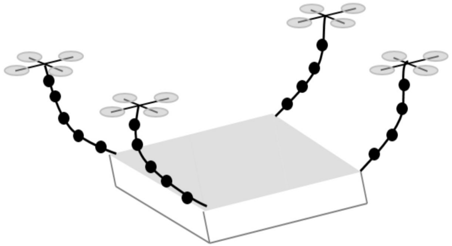
Motivation

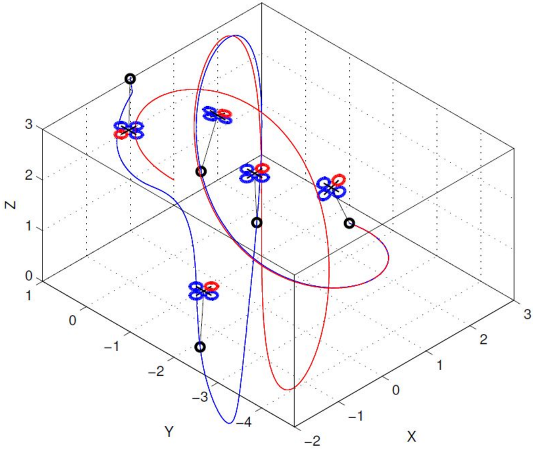
- UAV payload transportation receiving a lot of attention from military and industry.
- Difficulty in stabilizing UAVs with suspended payloads due to the underactuated nature of the system.
- Need for robust control systems
- Our motivation is use this project to explore pybullet

Relevant Prior Work: Problem Modeling

- Single UAV [5] vs Multiple UAV systems [7]
- Taut Cable Model [5] vs Flexible Cable Model [6]
- Quadcopter trajectory tracking [1] vs Load trajectory tracking [5]







Relevant Prior Work: Controllers

- Cascaded PID Controller [1]
- Feedforward-Feedback adaptive control (Notch Filter + LQG) [2]
- Reinforcement Learning [3]
- Model Predictive Control [4]

Lesson Learned: Don’t Trust Conference Papers (entirely)

- Not all papers in good journals / conferences are accurate:
  - Paper: Quad-rotor Lifting-Transporting Cable-Suspended Payloads Control, Alothman et al, 2015
  - Publisher: IEEE, 21st International Conference on Automation and Computing (ICAC)

- Paper proposed a linear model that was uncontrollable
  - We didn’t think to check it until we couldn’t reproduce their results.
- Wasted time and energy trying to implement their proposed model

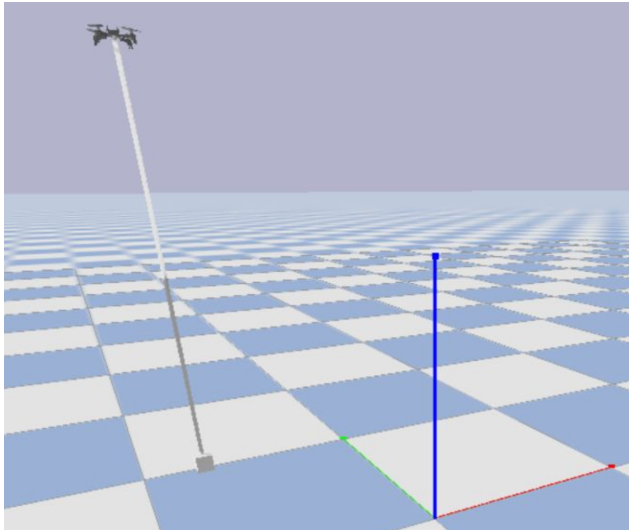
Our Model

Single UAV

Quadcopter Load Balancing

Taut cable model

URDF



Dynamical Model of UAV with suspended load

System has 8 DOF with 4 degree underactuation.

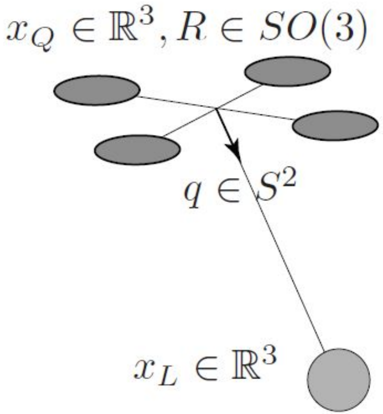
Quadrotor and load positions: $x_Q = x_L - lq,$ (1)

Equations of motion (Lagrange method):

$$\mathcal{L} = \mathcal{T} - \mathcal{U}, \text{ where } \mathcal{T} : TQ \to \mathbb{R} \text{ and } \mathcal{U} : Q \to \mathbb{R}$$

$$\mathcal{T} = \frac{1}{2} m_Q v_Q \cdot v_Q + \frac{1}{2} m_L v_L \cdot v_L + \frac{1}{2} \langle \hat{\Omega}, \widehat{J_Q \Omega} \rangle, \qquad (2)$$

$$\mathcal{U} = m_Q g e_3 \cdot x_Q + m_L g e_3 \cdot x_L, \qquad (3)$$



Satisfies, Lagrange d'Alembert principle;

$$\delta \int_0^T \mathcal{L}\, dt + \int_0^T \left(\langle W_1, \hat{M} \rangle + W_2 \cdot f R e_3 \right) dt = 0, \qquad (4)$$

Dynamical Model of UAV with suspended load

$$\dot{x}_L = v_L, \tag{5}$$

$$(m_Q + m_L)(\dot{v}_L + ge_3) = (q \cdot fRe_3 - m_Q l(\dot{q} \cdot \dot{q}))q, \tag{6}$$

$$\dot{q} = \omega \times q, \tag{7}$$

$$m_Q l \, \dot{\omega} = -q \times fRe_3, \tag{8}$$

$$\dot{R} = R\hat{\Omega}, \tag{9}$$

$$J_Q\dot{\Omega} + \Omega \times J_Q\Omega = M. \tag{10}$$

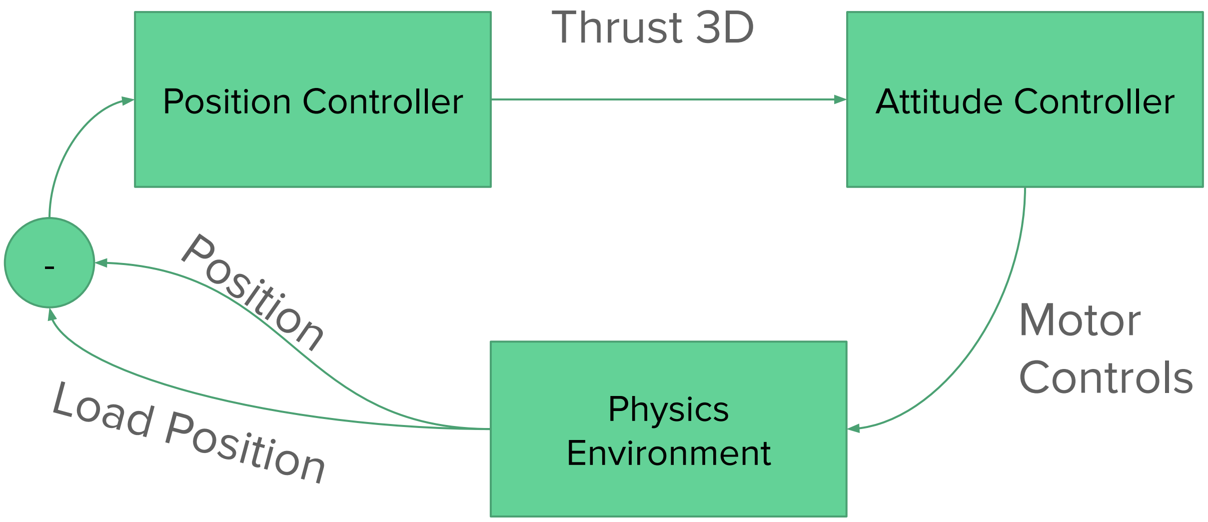
Above dynamics can be written in the standard form: $\dot{X}_n = f_n(X_n) + g_n(X_n)u$

Where, $X_n = \{x_L, q, R, v_L, \omega, \Omega\}$ is the state and u = $\{f, M\}$ the input to the system

Updated Question:

With a swinging load, can we just stabilize the load so that we don’t crash?

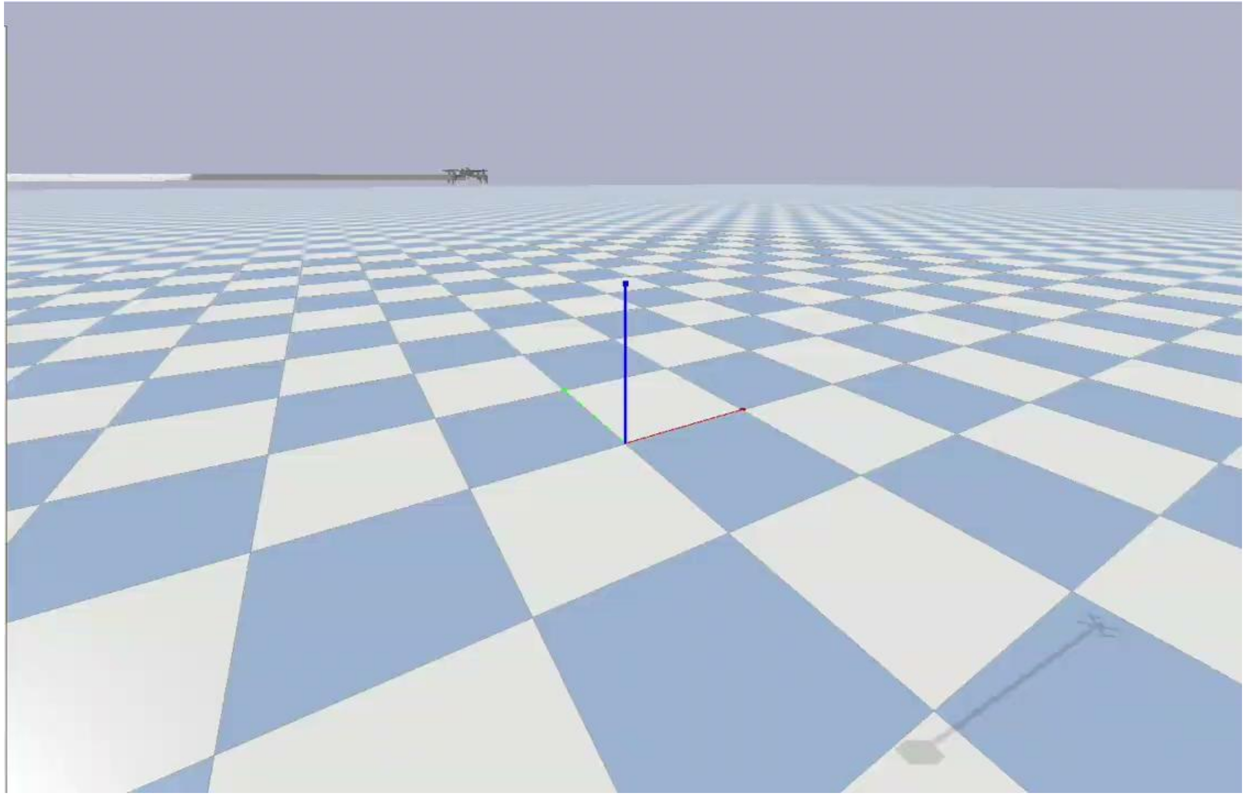
PID Controller(s) Design



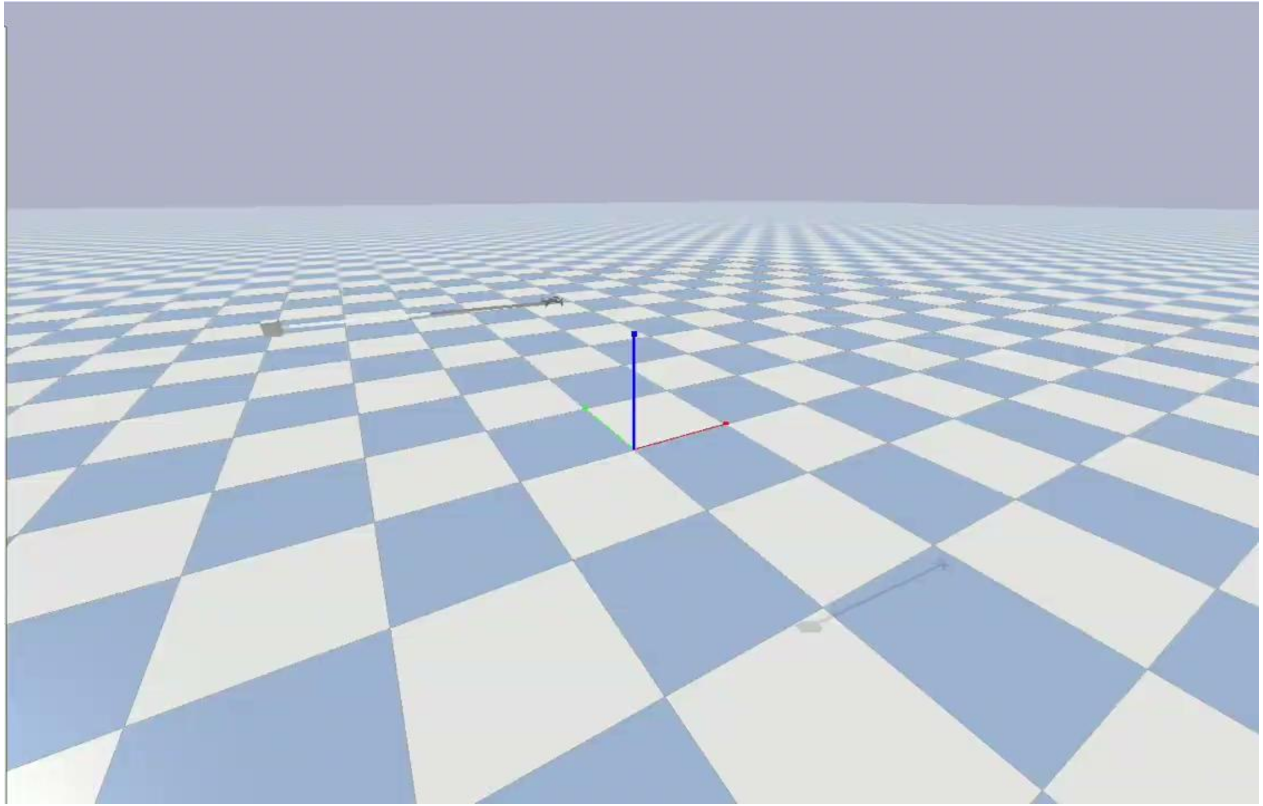
Pos	P	I	D
X	0.4	0.05	0.2
Y	0.4	0.05	0.2
Z	1.25	0.05	0.5

Att	P	I	D
X	70	50	50
Y	70	50	50
Z	6e4	500	12e3

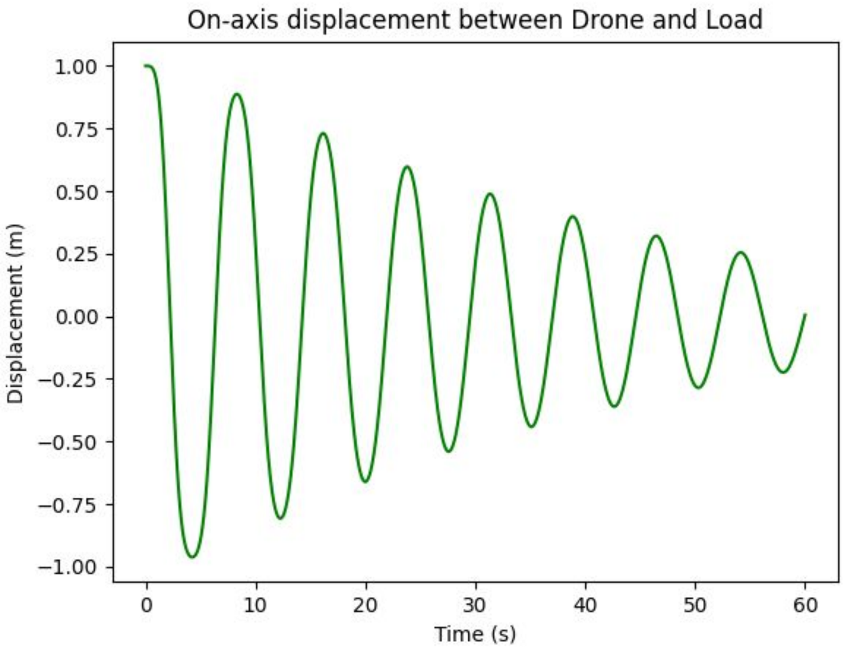
Simulation: Without PID controllers



Simulation: With both PID controllers

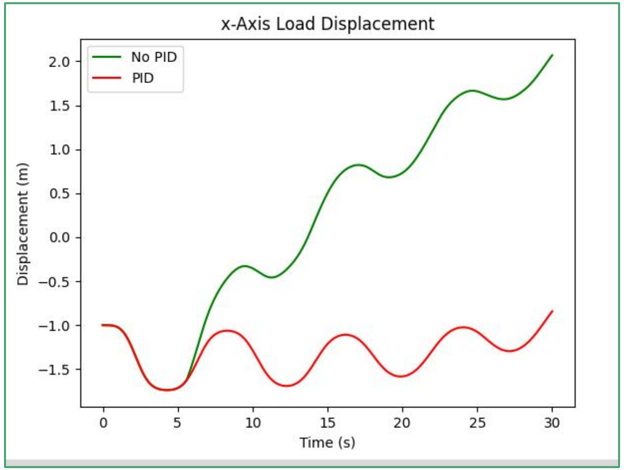


Results

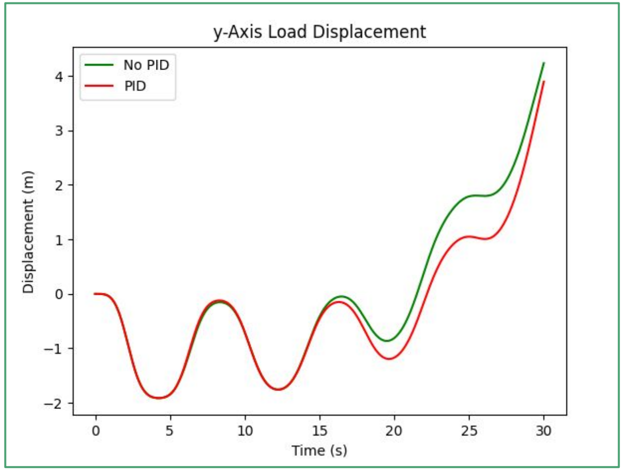


On-axis displacement between Drone and Load
Displacement (m)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
10
20
30
40
50
60
Time (s)

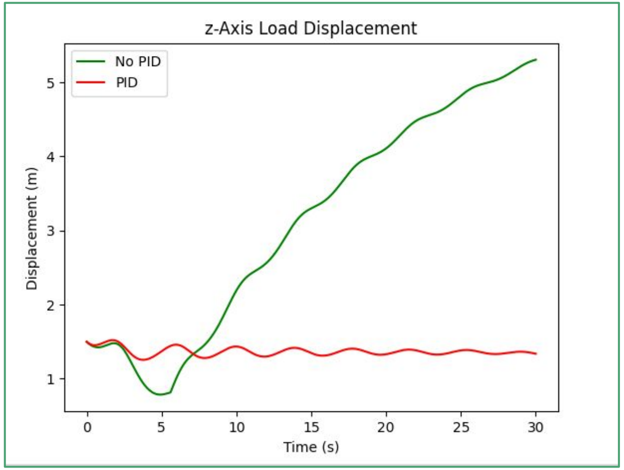
Results



x-Axis Load Displacement
No PID
PID
Displacement (m)
Time (s)
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0
5
10
15
20
25
30



y-Axis Load Displacement
No PID
PID
Displacement (m)
Time (s)
4
3
2
1
0
−1
−2
0
5
10
15
20
25
30



z-Axis Load Displacement
No PID
PID
Displacement (m)
Time (s)
5
4
3
2
1
0
5
10
15
20
25
30

Conclusion

- Designed a PID controller to stabilize a suspended load
- Learned a new simulator (i.e., Pybullet)
- Stay away from bad papers

THANK YOU

Questions ?

References

1. X. Huo, J. Chen, Q. Liu and X. He, "Vibration Elimination for Quadrotor Slung System Based on Input Shaping and Double Closed-loop Control," 2019 12th Asian Control Conference (ASCC), Kitakyushu, Japan, 2019, pp. 492-497.
2. J. F. Slabber and H. W. Jordaan, "Vision-Based Control of an Unknown Suspended Payload with a Multirotor," 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 2021, pp. 4875-4880, doi: 10.1109/IROS51168.2021.9636648.
3. Faust, A.; Palunko, I.; Cruz, P.; Fierro, R.; Tapia, L. Automated aerial suspended cargo delivery through reinforcement learning. Artif. Intell. 2017, 247, 381–398
4. Urbina-Brito, N.; Guerrero-Sánchez, M.E.; Valencia-Palomo, G.; Hernández-González, O.; López-Estrada, F.R.; Hoyo-Montaño, J.A. A predictive control strategy for aerial payload transportation with an unmanned aerial vehicle. Mathematics 2021, 9, 1822.
5. Sreenath, Koushil, Nathan Michael, and Vijay Kumar. "Trajectory generation and control of a quadrotor with a cable-suspended load-a differentially-flat hybrid system." 2013 IEEE international conference on robotics and automation. IEEE, 2013.
6. Bisgaard, Morten, Jan Dimon Bendtsen, and Anders la Cour-Harbo. "Modeling of generic slung load system." Journal of guidance, control, and dynamics 32.2 (2009): 573-585.
7. Michael, Nathan, Jonathan Fink, and Vijay Kumar. "Cooperative manipulation and transportation with aerial robots." Autonomous Robots 30 (2011): 73-86.