Planar Trajectory Tracking for a Payload-Carrying Quadrotor using Receding Horizon Control

Timmy A. Hussain

timmyh@stanford.edu

June 6, 2021

Abstract

This project develops an optimal controller for a UAV with variable center of gravity.

1 Introduction

Unmanned Autonomous Vehicles (UAVs) have shown and continue to show tremendous value in a number of industries ranging from film and entertainment to agriculture and construction. Part of the value provided by a drone or UAV is in its ability to supplement previously manual capability and to do so with minimal external assistance. In film, entertainment or surveillance, it might be in autonomous tracing out some predefined path and capturing footage over that area where previously this would have been accomplished manually. However, this value add is tied intrinsically to the drone's capability to comfortably carry whatever payload is needed to complete the task be it a camera mounted on a gimbal or infrared or other sensors. For certain applications, a design decision can be made on how the external payload will interact with and be coupled to the drone ahead of time (a gimbal might be mounted to align with the drone's center of gravity) while in other applications this is not entirely possible or variations in the coupling can be expected over the course of a flight (extensions, retractions or rotations of the attachment).

One such domain is the realm of drone delivery. In this case, the only guarantees we can have are those enforced by the carrying mechanism. For example, if we have a box attached to the drone for the purposes of carrying a payload, we can guarantee that (provided nominal operations), our payload will be confined to this box; if we have our payload attached to a tether, we can guarantee that the payload remains attached to the tether. However, we have few guarantees on the specific weight of our payload, or orientation, or how the tether might swing during flight or how the package might move around within the box. For a system such as a drone with a high load capacity, how the payload moves during flight is a disturbance that can't be ignored.

2 Related Work

In setting up the problem, Work has been done in modelling the dynamics of a multi-rotor system with variable center of gravity [1] In solving the problem, Work exists which leverages an external balancer: the control problem there is to control the balancer to minimize deviations from some nominal orientation due to the effect of the shifting center of gravity[2] although this requires additional actuators and, of course, a counter-balance weight which detracts from the drone's payload capability. Development of a control system that is resistant and robust to external disturbances is one that can be solved with classical control techniques. Given sufficient state information, a controller can be developed which is able to control the drone and have disturbance rejection capabilities desired in this problem setup. However, this often does not incorporate knowledge about the source of the disturbance (as we'd like to be robust to a wide class of disturbances).

My work will attempt to incorporate knowledge of the payload dynamics into the development of an optimal control algorithm. By setting this up as an optimal control problem, I am able to included these dynamics in a way that classical control methods cannot, and it gives me a way to leverage knowledge of the problem setting and hopefully achieve a slightly predictive controller rather than a purely reactive one.

In trying to figure out some of the problems that arose during the project, I ultimately delved deeper into some more drone modelling[3],

3 Problem Statement

I will be formulating this project as a planar trajectory tracking problem. The quadrotor will be constrain

The problem setting will be solving an optimal control problem that incorporates the weight-shifting dynamics of a payload being carried by a drone and it will be validated by simulating the drone's ability to complete a trajectory smoothly and with minimal stops. The payload will be modelled as a smooth ball whose motion is coupled with that of the drone as well as influenced by some additive noise.

4 Approach

4.1 Variables

- x_1: x-position of the drone in inertial coordinates
- x_2: y-position of the drone in inertial coordinates
- x_p: x-position of the payload in body-fixed coordinates
- y_p: y-position of the payload in body-fixed coordinates
- ϕ: roll angle
- θ: pitch angle
- U_1: Total upward thrust from props
- U_2: Total roll torque from props
- U_3: Total pitch torque from props
- Q_θ: Roll disturbance
- Q_ϕ: Pitch disturbance
- $\mathbf{x}$: vector of states
- $\mathbf{U} = [U_1, U_2, U_3]^T$: vector of control inputs

4.2 Dynamics

Generally, I will be making a lot of simplifying assumptions including neglecting the effects of drag and more complex effects arising due to rotor inertia . This is primarily because the focus is on the effects of the payload dynamics arising during motion.

4.2.1 Horizontal Dynamics

Ignoring the effects of gravity on the drone, the planar dynamics are influenced purely by the control input U_1 and the orientation of the drone $[\phi, \theta]$.

The U_1 control input is defined in the body frame of the drone and is given by $\vec{U_1}_{[b]} = U_1\left(-\hat{z}_{[b]}\right)$. Hence to get how this body-fixed force affects the position of the drone in inertial space, we transform the force from body-fixed to inertial-fixed frames. $\vec{U_1}_{[I]} = [R]\vec{U_1}_{[b]}$ where $[R]$ is the Euler rotation matrix. In the relevant literature, it is appropriate to simplify this relationship by making a small-angle approximation for the rotations. This results in the following relationship that defines our planar dynamics

$$m\begin{bmatrix}\ddot{x}_1\\ \ddot{x}_2\\ \ddot{x}_3\end{bmatrix} = \begin{bmatrix}1 & \psi & \theta\\ \psi & 1 & -\phi\\ -\theta & \phi & 1\end{bmatrix}\begin{bmatrix}0\\ 0\\ -U_1\end{bmatrix} \Longleftrightarrow m\begin{bmatrix}\ddot{x}_1\\ \ddot{x}_2\end{bmatrix} = \begin{bmatrix}-\theta U_1\\ \phi U_1\end{bmatrix} \tag{1}$$

4.2.2 Attitude Dynamics

As the control inputs U_2 and U_3 are already defined as pitching and rolling torques respectively, and by making a few other simplifying assumptions such as unit distance from center of gravity to the rotors and unit moments of inertia, I arrive at the following relationship.

$$\begin{bmatrix}\dot{\theta}\\ \dot{\phi}\end{bmatrix} = \begin{bmatrix}U_2 - Q_\theta\\ U_3 + Q_\phi\end{bmatrix} \tag{2}$$

4.2.3 Payload Dynamics

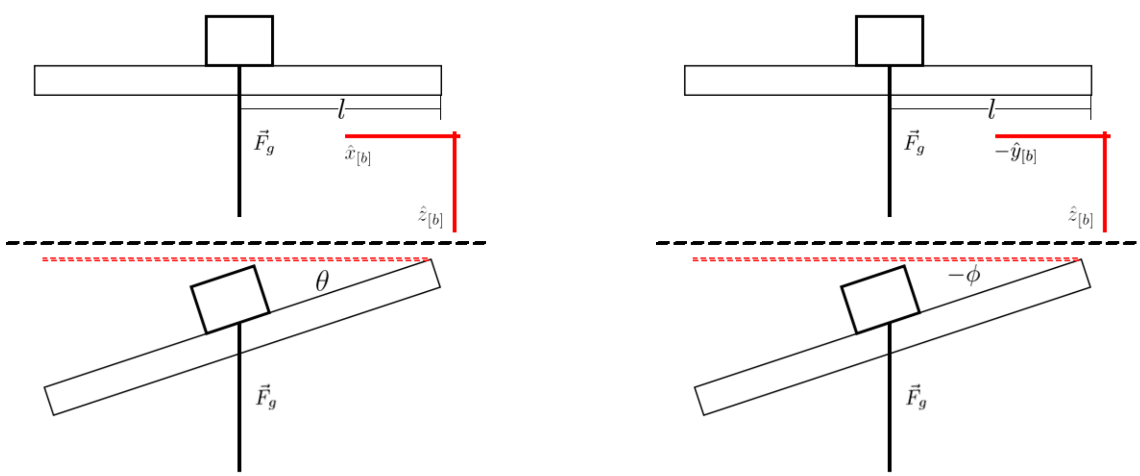


Figure 1: Payload dynamics under the effect of gravity due to roll and pitch

As can be seen from Figure 4.2.3, the payload states x_p, y_p are governed by a mass being acted upon by gravity depending on the current orientation of the drone. Again, the small-angle approximation is made.

$$m_p \begin{bmatrix} x_p \\ y_p \end{bmatrix} = m_p g \begin{bmatrix} sin\theta \\ sin\phi \end{bmatrix} \Longleftrightarrow m_p g \begin{bmatrix} \theta \\ \phi \end{bmatrix} \tag{3}$$

The moment exerted on the drone due to the mass and position of the payload is given by

$$\begin{bmatrix} Q_\theta \\ Q_\phi \end{bmatrix} = \begin{bmatrix} x_p \\ y_p \end{bmatrix} \times \vec{F}_{g_{[b]}} \tag{4}$$

where $\vec{F}_{g_{[b]}} = [R]\vec{F}_{g_{[I]}}$ is the force due to gravity expressed in the body-fixed frame. Following the same process as in Equation (1), we ultimately arrive at the following

$$\begin{bmatrix} Q_\theta \\ Q_\phi \end{bmatrix} = \begin{bmatrix} -F_g x_p \\ F_g y_p \end{bmatrix} \tag{5}$$

Ultimately as this is a trajectory tracking problem, the RHC formulation takes the following form for a horizon of length N

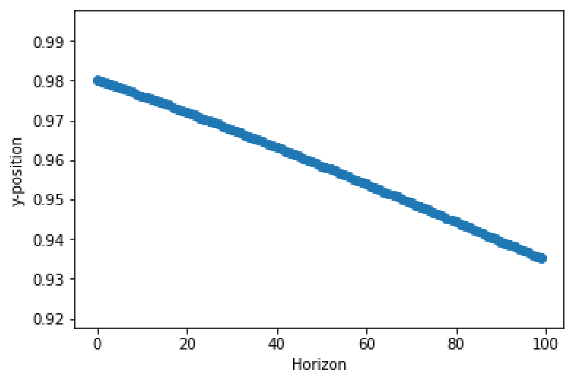
$$\min_{u \in \mathcal{U}} \left(\mathbf{x}^N - \mathbf{x}_{\text{ref}}^N\right)^T P \left(\mathbf{x}^N - \mathbf{x}_{\text{ref}}^N\right) + \sum_{i=0}^{N-1} \left(\mathbf{x}^i - \mathbf{x}_{\text{ref}}{}^i\right)^T Q \left(\mathbf{x}^i - \mathbf{x}_{\text{ref}}{}^i\right) + \sum_{i=0}^{N-1} \left(\mathbf{u}^i - \mathbf{u}_{\text{ref}}{}^i\right)^T R \left(\mathbf{u}^i - \mathbf{u}_{\text{ref}}{}^i\right)$$

subject to $\begin{cases} \mathbf{x} \in \mathcal{X} \\ x_{k+1} = Ax_k + Bu_k, k \geq 1 \end{cases}$

where $\mathcal{U}$ and $\mathcal{X}$ represent feasible and admissible state and control inputs respectively.

5 Experiments

I wound up getting somewhat hamstrung on this problem for a few reasons. Most important of those are the non-linearities of the problem. Even with small-angle approximations made and simplifying assumptions included, there is ultimately a coupling between state variables which are the angle and the control inputs in how they control the lateral dynamics. This non-linearity makes the RHC problem implicitly non-convex. I ultimately attempted linearizing around an operating point $\bar{\mathbf{x}}, \bar{\mathbf{u}}$ and developing an iLQR type approach where I would linearize and obtain state matrices A_k, B_k which govern the dynamics, using these matrices for one iteration of the RHC problem, obtaining a feasible next step and feasible control, and then linearizing around those operating points and proceeding. This worked to some extent but the dynamics seemed to be really "slow". I needed to run several iterations of have an extremely long horizon to see any real changes in the position of the drone.



Subsequently, I ended up taking a different approach which simplified the dynamics even further by decoupling the Upward thrust control input into two component inputs: x-direction thrust and y-direction thrust. Combined with the small-angle approximation, this resulted in reasonably linear dynamics. However, I was unable to get that coded and tested in time.

6 Conclusions and Future Work

The inclusion of the payload into the dynamics model works well as it shows up simply as some external disturbance that the controller needs to compensate for. Although we have knowledge of the dynamics of the payload and can model its evolution through time, the drone does not have this knowledge and can only feel the effect of the package moving around: the moment arm caused by its weight. This is fundamentally the same as the drone feeling a gust of wind or some other disturbance and figuring out how best to compensate for it. A receding horizon controller gives us the opportunity to get closed-loop behaviour by solving an open-loop optimization problem. However, as this project demonstrates, the problem must be set-up properly to guarantee a solution.

I still think the linearization approach should have worked and likely, there is some issue in my code which I was unable to suss out in time. The linearization and subsequent linearizations around our operating points, combined with the minimization of a loss function which penalizes the error to our nominal trajectory, coupled with the evolving dynamics of the drone and package and the effect of moment due to the package weight arising as a disturbance in the dynamics should all combine to make a working RHC controller.

Video project link: `https://tinyurl.com/aa203-proj`

References

[1] Xingjian Xu, Chang Liu, and Bobo Ye. "Modeling and applicition of multi-rotor with variable center of gravity". In: *2016 8th International Conference on Modelling, Identification and Control (ICMIC)*. 2016, pp. 348–353. DOI: 10.1109/ICMIC.2016.7804135.

[2] Kamolwat Chaisena, Kontorn Chamniprasart, and Suradet Tantrairatn. "An Automatic Stabilizing System for Balancing a Multi-Rotor Subject to Variations in Center of Gravity and Mass". In: *2018 Third International Conference on Engineering Science and Innovative Technology (ESIT)*. IEEE. 2018, pp. 1–5.

[3] Samir Bouabdallah and Roland Siegwart. "Full control of a quadrotor". In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems.* 2007, pp. 153–158. DOI: 10.1109/IROS.2007.4399042.