Dynamic Stability and Trajectory Control of Planar Quadrotors: A Study in Control Lyapunov Functions and Quadratic Programming

Joaquin Garcia

1 Abstract

This project introduces an approach for controlling planar quadrotor flight dynamics using a combination of Control Lyapunov Functions (CLFs) and Quadratic Programming (QP) techniques. Quadrotor vehicles have garnered significant interest due to their versatility in various applications, however, achieving precise and robust control of quadrotor motion remains a challenging task. In this work, we propose a control framework that leverages CLFs to design stable control laws for quadrotor flight, ensuring stability and enabling the quadrotor to track desired trajectories while minimizing energy consumption. By formulating the control objectives as constraints in a QP problem it is ensured that the control inputs satisfy both performance specifications and physical constraints, thereby guaranteeing stability and real-time feasibility. Simulations and computations of the trajectories' Lyapunov Exponents on a planar quadrotor platform validate the effectiveness of the proposed approach, demonstrating satisfactory results in flight stability and robustness to perturbations. This methodology holds promise for enhancing the control of planar quadrotors, with potential applications in autonomous navigation.

2 Introduction and Motivation

With the increasing use of drones, particularly quadrotors, in various applications such as surveillance, photography, and logistics, ensuring effective and reliable control has become a topic of much importance. However, navigating these unmanned aerial vehicles (UAVs) through complex environments poses significant challenges, as conventional control techniques often struggle to balance stability, maneuverability, and energy efficiency. In response, this study introduces a novel control framework that combines two simple yet powerful methods: Control Lyapunov Functions (CLFs) and Quadratic Programming (QP). These methods offer a promising solution to enhance the control performance of planar quadrotors by providing stability guarantees, trajectory tracking precision, and adherence to physical constraints. This introduction sets the stage for a detailed exploration of these methods, their implementation, and experimental validation, highlighting their potential to advance the state-of-the-art in quadrotor flight control and facilitate their integration into various real-world applications.

3 Related Concepts

In this section, we break down the key ideas driving the project's development.

3.1 Control Lyapunov Functions (CLFs)

To be able to comprehend CLFs, we first need to define Lyapunov functions. Lyapunov functions are scalar functions that are used to analyze the stability of an equilibrium point of a, usually non-linear, dynamical system. The Lyapunov function $V(x)$ must satisfy the following properties:

- $V(x)$ is continuous and differentiable
- $V(x)$ is positive definite, meaning $V(x) > 0$ for all $x \neq 0$ and $V(0) = 0$
- The time derivative of $V(x)$ along the trajectories of the system is negative, meaning $\dot{V}(x) \leq 0$ for all x, except at the equilibrium points

As defined in [1], we consider the control affine system:

$$\dot{x} = f(x) + g(x)u$$

where $x \in \mathbb{R}^n$ is the state vector and $u \in \mathbb{R}^m$ is the control vector. We can now define a Lyapunov function $V(x)$ and its derivative as:

$$\begin{aligned}\dot{V}(x,u) &= \nabla V(x) \cdot f(x) + \nabla V(x) \cdot g(x)u \\ &= L_f V(x) + L_g V(x)u\end{aligned}$$

such that for all x, we can choose a control which makes the derivative along trajectories negative, meaning

$$\inf_{u \in U} [L_f V(x) + L_g V(x)u] < 0$$

This way we can ensure stability of the dynamical system at the point of equilibrium point of the Lyapunov function. In more general terms, at every state, we will be capable of choosing a control input which will push the system towards a desired state.

Nevertheless, even if we can select a control input to move towards the equilibrium point, we do not have any notion on how fast we are going to reach it, this is why as shown in [2], we can "construct a Lyapunov constraint that impose a condition of minimum rate of decrease in $V(x)$". Such as:

$$\inf_{u \in U} [L_f V(x) + L_g V(x) u + \eta V(x)] < 0$$

for $\eta > 0$. In this way we can control the speed to which we move towards the equilibrium.

3.2 Quadratic Programming Formulation

The objective now is to find a controller which satisfies the condition set by the CLF and minimizes the magnitude of the control used at each time-step to account for energy efficiency. In [3], the baseline QP formulation is set as the following:

$$\begin{aligned} \min \quad & \frac{1}{2}(u^T u + \lambda \delta^2) \\ \text{s.t.} \quad & L_f V(x) + L_g V(x) u + \eta V(x) < \delta \\ & u \in U \end{aligned}$$

3.3 Lyapunov Exponents

Lyapunov exponents are mathematical quantities used in dynamical systems theory to measure the rate of divergence or convergence of nearby trajectories in a system's phase space. They provide insights into the stability and predictability of complex systems. Positive Lyapunov exponents indicate chaotic behavior, where small perturbations grow exponentially over time, while negative exponents imply stability, where nearby trajectories converge. These exponents offer a quantitative means of understanding the intricate dynamics of nonlinear systems.

4 Planar Quadrotor Dynamics

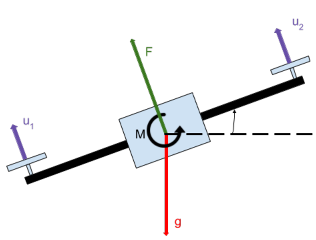

Figure 1. Planar Quadrotor Model

Figure 1 shows the planar quadrotor model that will be used for control simulation, as it can be seen it consists of two propellers which will generate forces F and M, which represent the thrust and torque respectively.

$$F = u_1 + u_2$$

$$M = \frac{L}{2}(u_1 - u_2)$$

The dynamics of the system are given below:

$$\ddot{x} = \frac{F \sin\theta}{m}$$

$$\ddot{y} = \frac{F \cos\theta}{m} - g$$

$$\ddot{\theta} = \frac{M}{J}$$

where g is the gravitational constant, m is the mass and J is the inertia of the quadrotor.

We also define the system as a control affine system in order to make sure it is possible to use CLFs in order to control the system.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ 0 \\ -g \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -\frac{\sin\theta}{m} & 0 \\ \frac{\cos\theta}{m} & 0 \\ 0 & \frac{1}{J} \end{bmatrix} \begin{bmatrix} u_1 & u_2 \end{bmatrix}$$

5 CLF-QP Implementation

As done in [4], we take inspiration by the back-stepping method in geometric control, separating the orientation and transnational dynamics. So our controller consists of two components, a position level QP and an orientation level QP, in which we first set virtual dynamics for the position level QP and use the output controller and use it as input for the orientation level QP from which we will obtain the final controller input for the system.

5.1 Position Level QP

For this step, we disregard the orientation dynamics and make the assumption that the underactuated part is "fully-actuated", meaning that the acceleration in the x and y coordinates is directly given by the control inputs.

$$\ddot{x} = \frac{f_{v_1}}{m}$$

$$\ddot{y} = \frac{f_{v_2}}{m}$$

where f_{v_1}, f_{v_2} are virtual forces. As the next step, in order to be able to formulate the QP problem, we need to select a CLF which satisfies the constraints mentioned before, and additionally, has its point of equilibrium in the desired end position of the planar quadrotor.

$$\hat{V} = \frac{1}{2}(\sqrt{m}e_v + \sqrt{k_1}e_x)^2$$

where $e_v = [\dot{x} - \dot{x_d}, \dot{y} - \dot{y_d}]$ (the difference between actual velocities and desired velocities), $e_x = [x - x_d, y - y_d]$ (the difference between actual position and desired position), and k_1 is a hyperparameter that must be greater than 0. The desired velocities can be chosen arbitrarily, but it must be made sure that they indicate the quadrotor to move towards the desired position and are equal to 0 when the quadrotor has reached the desired position. For this project, the desired velocities are the scaled difference between actual position and desired position. With this set up it is possible to set up the Position Level QP.

Algorithm 1 Position Level QP

$$\begin{aligned} f_v = \arg\min \quad & \frac{1}{2}(f^T Q f) \\ \text{s.t.} \quad & \dot{\hat{V}}(x, y, f) + \eta_1 \hat{V}(x, y) \leq 0 \end{aligned}$$

where $\eta_1 > 0$. We do not impose slack term or constraints for the range of the forces since it is a virtual system.

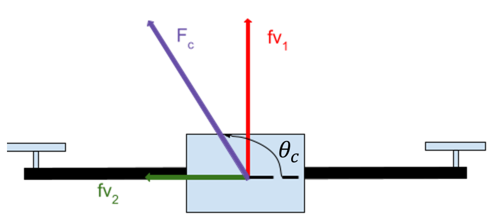

Figure 2. Position Level QP output

As it can be seen in Figure 2, the position level QP output, will represent virtual forces of how much we want to move in the x and y coordinates, nevertheless, in order for the planar quadrotor to move, we require to change its orientation towards the desired direction. For this reason, we compute the desired thrust $F_c = \max\{f_v \cdot [\sin\theta, \cos\theta], 0\}$ and the desired angle $\theta_c = \arctan(f_{v_1}/f_{v_2})$.

5.2 Orientation Level QP

Using the results obtained in the Position Level QP, we can define the Control Lyapunov function:

$$V_\theta = \frac{1}{2}(\sqrt{J}(\dot{\theta} - \dot{\theta}_c) + \sqrt{k_2}(\theta - \theta_c))$$

Then, we can construct the Orientation Level QP to obtain the thrust and torque control inputs:

Algorithm 2 Orientation Level QP

$$\begin{aligned} [F^*, M^*]^T = \arg\min \quad & \frac{1}{2}\lambda_1(F - F_c)^2 + \frac{1}{2}(M)^2 + \frac{1}{2}\lambda_2\delta^2 \\ \text{s.t.} \quad & \dot{V}_\theta(M) + \eta_2 V_\theta \leq \delta \\ & 0 \leq F \leq F_{MAX} \\ & |M| \leq M_{MAX} \end{aligned}$$

where $\lambda_1, \lambda_2, \eta_1$ are hyperparameters.

6 Results & Discussion

In this section, we validate the CLF-QP controller on the planar quadrotor using numerical simulation. The parameters for the planar quadrotor are $m = 1.0(kg)$, $J = 0.25(kg \cdot m^2)$, $F_{MAX} = 40N$, and $M_{MAX} = 20N \cdot m$. We set the initial postion to be at the origin $(0, 0)$ and the desired position at $(50, 50)$. Simulations are performed in Python using the Runge-Kutta method of order 5 with $\delta t = 0.01$ and simulated for $t = 10$ seconds. In order to solve both QP problems we utilize the **cvxpy** library and use the CLARABEL method.

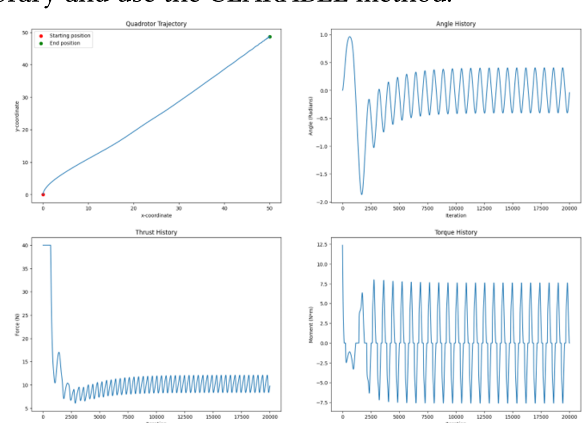

Figure 3. Simulation Results

The controller is capable of making the planar quadrotor reach its desired destination at least on the $x, y-$plane.

Nevertheless, even when the quadrotor has reached the desired position, we can observe that there is a wave behavior in the angle of rotation which is even greater when looking at the control input forces. This type of behavior is undesirable since it can contribute to instability if the system is ever affected by external forces such as wind. However, looking at the positives, we can observe that the average of the angle of rotation and the torque is 0, which would be the desired end state, and the average thrust converges to a value equal to gravity, which will keep the quadrotor maintain its position. Extra hyperparameter tuning will be needed in order to negate the "unstable" behavior presented on this parameters, but this is a complex process since there is not much intuition on which parameters create this behavior and there is a significant amount of hyperparameters.

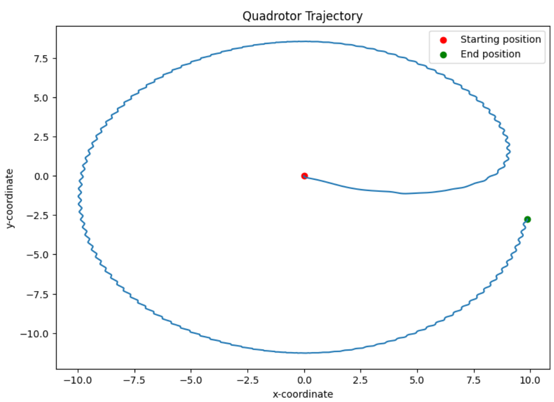

Figure 4. Simulation on trajectory

Since we are capable of making the planar quadrotor reach a desired position, it is also possible to set a series of desired positions in order to make the planar quadrotor follow a trajectory. For experimentation, we set the trajectory to be a circle with a given radius $r = 10$. As we can observe, the trajectory is followed with a certain accuracy but it is possible to see also a wavy behavior which could cause sensibility to outside disturbances. Nevertheless, all of the results obtained until now show significant progress towards developing a robust controller for planar quadrotors.

6.1 Planar Quadrotor Lyapunov Exponents

Lyapunov Exponents (LEs) as mentioned before will tell us how nearby trajectories, will converge or diverge over time, which will give us valuable information about the stability of the controlled dynamical system. When trying to use the power iteration method in order to compute the highest LE, we ran into a problem to define the Jacobian matrix of the system. This is mainly because there was no clear way of defining the control input components of the Jacobian due to how the control input where obtained through QP, but if we left them out, it would lead to the Jacobian matrix to be singular. For this reason, I made the assumption that the control input u could be defined as a function of the state x, ($u = u(x)$), so that at each time-step, the Jacobian could be approximated using the central difference quotient. The results was that the highest LE came to be $\lambda_1 \approx -27.6$, which means that the controlled dynamical system is greatly stable, and that nearby trajectories converge extremely fast. As a way to demonstration, two trajectories starting from different points ($(0, 0)$ and $(5, 5)$) are simulated to reach the same final destination. Figure 4, shows how fast both trajectories converge.

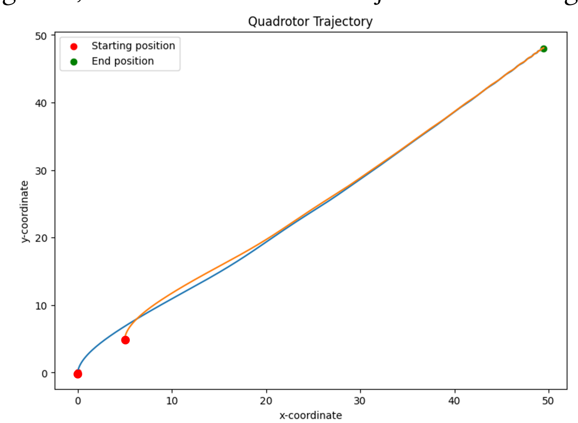

Figure 4. Two Trajectories Convergence

References

[1] R.A. Freeman and J.A. Primbs. Control lyapunov functions: new ideas from an old source. In *Proceedings of 35th IEEE Conference on Decision and Control*, volume 4, pages 3926–3931 vol.4, 1996.

[2] Jason Choi. Jason choi – introduction to control lyapunov functions and control barrier functions, March 2021.

[3] Mrdjan Jankovic. Combining control lyapunov and barrier functions for constrained stabilization of nonlinear systems. In *2017 American Control Conference (ACC)*, pages 1916–1922, 2017.

[4] Guofan Wu and Koushil Sreenath. Safety-critical control of a planar quadrotor. In *2016 American Control Conference (ACC)*, pages 2252–2258, 2016.

[5] Praveen Venkatesh, Sanket Vadhvana, and Varun Jain. Analysis and control of a planar quadrotor. *arXiv preprint arXiv:2106.15134*, 2021.

[6] Bike Zhang Jason Choi, Prasanth Kotaru. Cbf-clf-helper, 2021.