Quadrotor Control Using RL

Stanford University AA 203 Spring 2022

Name: Albert Chan
SUNet ID: albertc2
Department of Mechanical Engineering
Stanford University
albertc2@stanford.edu

Name: Pei-Chen Wu
SUNet ID: pcwu1023
Stanford University
pcwu1023@stanford.edu

Name: Kevin Lee
SUNet ID: kelelee
Department of Electrical Engineering
Stanford University
kelelee@stanford.edu

1 Motivation

While the problem of quadrotor control has been greatly explored, more work is still needed to expand the payload carrying capability of the quadrotor. If the payload is unstable or fragile, a more powerful controller is needed to ensure the quadrotor is able to deliver the payload safely. To model this, we now consider the problem of balancing an inverted pendulum on top of a quadrotor.

2 Technical Contribution

The main contribution of this report is the development of a learning based controller that is able to swing up an inverted pendulum on top of a quadrotor, and track desired trajectories, while being robust enough to withstand external disturbances acting on the body such as external wind while achieving good tracking performance.

3 Related Work

The work of Hintz et. al.[1] uses a LQR controller to stabilize an inverted pendulum on top of a drone and track trajectories, which results in our first baseline. While MPC has previously been used for quadrotor control[2] and inverted pendulum balancing [3], to the best of our knowledge, there are no journal or conference papers directly applying MPC to the problem of pendulum balancing on a quadrotor. MPC has been found to have better tracking performances than LQR and be more robust since the system dynamics are re-linearized and the control input optimized at each time step [3], making it a compelling baseline to compare our learning based controller against. Learning based control has also been applied to the problem of balancing an inverted pendulum on top of a drone with some success. The work of Figueroa, et al [4] uses a Continuous Action Fitted Value Iteration (CAFVI) algorithm to approach this specific problem, allowing an optimal policy to be computed for continuous states, while also having consistent results and being efficient with training data. Other approaches to balancing inverted pendulums in general include using Advantage Actor Critic, Proximal Policy Optimization, and Policy gradient methods as proposed by Bates et al .[5]

4 Problem Formulation

To simplify the dynamics of quadrotor and allow us to focus on developing control algorithms, a planar quadrotor will be considered for the project. The baseline case will include the implementation of LQR based and MPC based control algorithims. A gaussian noise with mean $\mu = 0$ and $\Sigma = \mathrm{diag}(0, 0, 0, 0, 10^{-3}, 10^{-3}, 10^{-3}, 10^{-3})$ will be added to the state to simulate process noise in the system. The baseline control algorithms will then be evaluated against an RL based controller, by comparing the robustness, trajectory tracking error, setting time, and required control effort.

To evaluate the robustness, a constant velocity wind will be added in the y direction until the drone is unable to stabilize around a desired target state. To evaluate the tracking error, the drone will be given a circular trajectory with a radius $r = 2$m to track. Lastly, the average control effort and settling time will be compared.

5 Model Dynamics

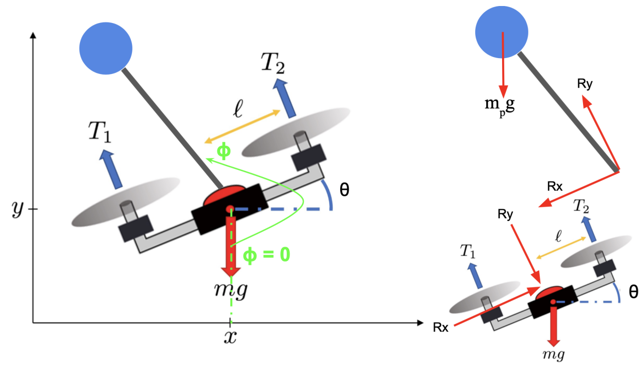


Figure 1: Free body diagram of drone with pendulum

R_x and R_y are components of a force R acting on the pendulum as a result of being attached to the drone as a revolute joint. Based on Figure 1, the acceleration of the pendulum $(\ddot{x}_p, \ddot{y}_p)$ is given by:

$$\begin{aligned} \ddot{x}_p &= \ddot{x} - L\ddot{\phi}\sin(\phi - \pi/2) - \dot{\phi}^2 L\cos(\phi - \pi/2) \\ \ddot{y}_p &= \ddot{y} + L\ddot{\phi}\cos(\phi - \pi/2) - \dot{\phi}^2 L\sin(\phi - \pi/2) \end{aligned} \tag{1}$$

With the state $s := (x, y, \theta, \phi, \dot{x}, \dot{y}, \dot{\theta}, \dot{\phi}) \in \mathbb{R}^8$, the following equations represent the dynamics of our drone with an inverted pendulum:

$$\ddot{x}_p = \frac{R_x\cos(\theta) - R_y\sin(\theta)}{m_p} \tag{2}$$

$$\ddot{y}_p = \frac{R_x\sin(\theta) + R_y\cos(\theta) - m_p g}{m_p} \tag{3}$$

$$\ddot{\phi} = \frac{L}{2I_p}\left[R_x\sin(\phi - \frac{\pi}{2} - \theta) - R_y\cos(\phi - \frac{\pi}{2} - \theta) - m_p g\cos(\phi - \frac{\pi}{2})\right] \tag{4}$$

$$\ddot{x} = \frac{-R_x\cos(\theta) + R_y\sin(\theta) - (T_1 + T_2)\sin(\theta)}{m_Q} \tag{5}$$

$$\ddot{y} = \frac{-R_x\sin(\theta) - R_y\cos(\theta) + (T_1 + T_2)\cos(\theta) - m_Q g}{m_Q} \tag{6}$$

$$\ddot{\theta} = \frac{(T_2 - T_1)l}{I_Q} \tag{7}$$

6 Preliminary Baseline Results

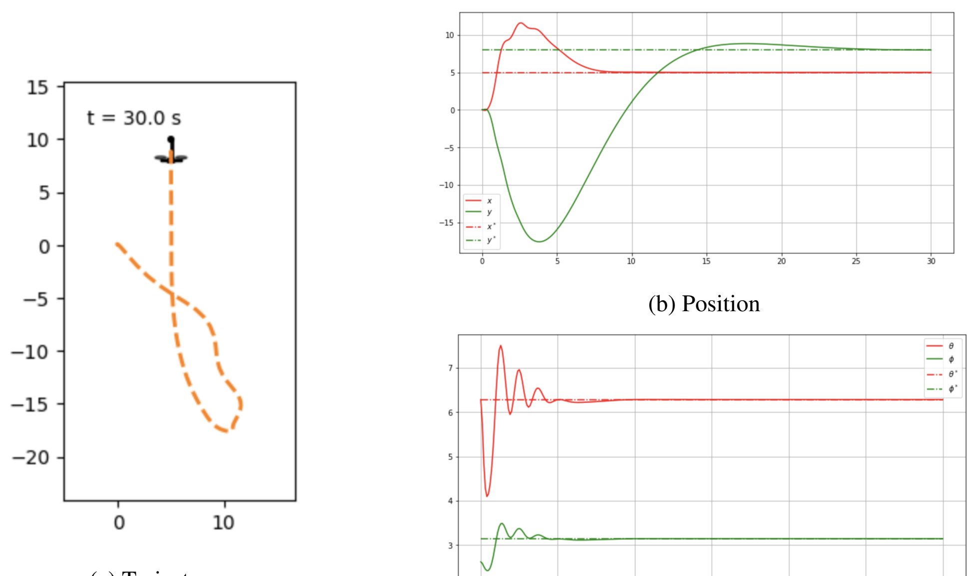
We have used two different approaches to make the drone to balance inverted pendulum, which are LQR and MPC. We initialized the system at s = [0, 0, 2π, $5\pi/6$, 0, 0, 0, 0] to test our controller and make the drone to reach the final state at s = [5, 8, 2π, π, 0, 0, 0, 0] (upright state).

6.1 LQR

We derived A and B matrices based on the above model dynamics. Then we applied Riccati recursion to calculate $K \in \mathbb{R}^{2\times 8}$. From Figure 2, we can see that the LQR controller can balance pendulum successfully.

$$K[0] = [0.38, -0.69, 40.15, -104.50, 1.05, -3.05, 6.28, -29.45]$$
$$K[1] = [-0.38, -0.69, -40.15, 104.50, -1.05, -3.05, -6.28, 29.45]$$

$s^* := (5, 8, 2\pi, \pi, 0, 0, 0, 0)$ with $u^* := [\frac{(m_Q+m_p)g}{2}, \frac{(m_Q+m_p)g}{2}]$



(a) Trajectory

(b) Position

(c) Angles

Figure 2: LQR w. intial state: [0, 0, 2π, $5\pi/6$, 0, 0, 0, 0]

6.2 Disturbance Rejection

We add a gaussian noise with mean $\mu = 0$ and $\Sigma = \text{diag}(0, 0, 0, 0, 10^{-3}, 10^{-3}, 10^{-3}, 10^{-3})$ to investigate the disturbance rejection ability of the controller. The results are shown in Figure 3.

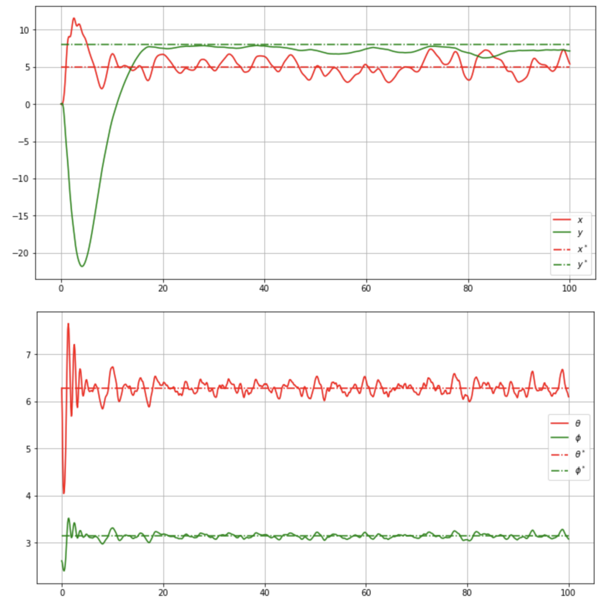


Figure 3: Disturbance Rejection using LQR at intial state: [0, 0, 2π, $5\pi/6$, 0, 0, 0, 0]

6.3 Circle Trajectory

We attempt a circular trajectory while maintaining the pendulum upright, and the results are shown in Figure 4. The drone can follow the circular path pretty well, but θ and ϕ still oscillate around 2π and π, respectively.

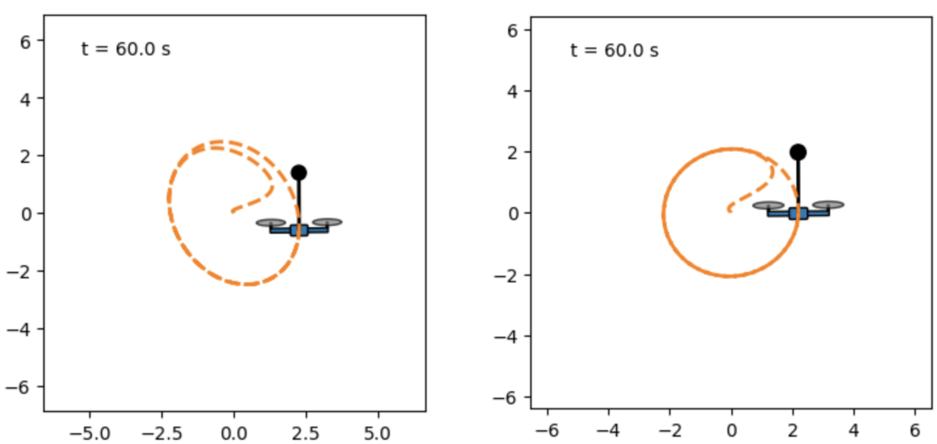


Figure 4: Circle trajectory using LQR controls with $Q = I$ (left) $Q = 100I$ (right)

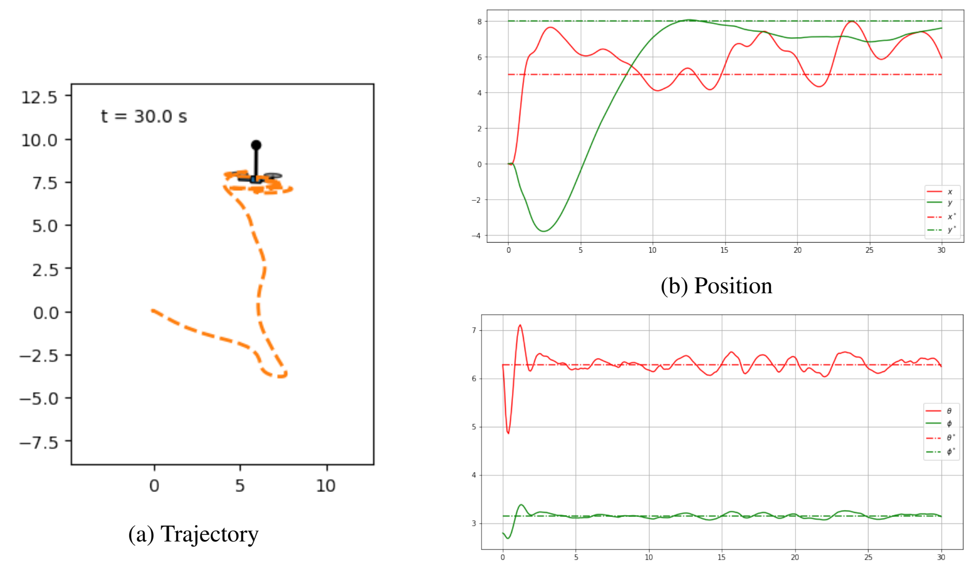
6.4 MPC

MPC is an control technique that solves an optimal control problem online at each individual time step. At each time step, the problem is linearized in the form $\Delta s_{k+1} = A\Delta s_k + B\Delta u_k$ based on the derived dynamics. A constrained convex program is solved that minimizes the quadratic cost across a finite prediction horizon of 10 seconds, subject to dynamics constraints and actuator limits.

The optimization problem solved is:

$$
\begin{aligned}
\min_{\Delta u_k, \Delta s_k} \sum_{k=1}^{hf} \Delta s_k^T Q \Delta s_k + \Delta u_k^T R \Delta u_k \\
\text{subject to } \Delta s_{k+1} = A\Delta s_k + B\Delta u_k \text{ for } k = 1, 2, \ldots hf \\
u_k \leq T_{max}
\end{aligned}
$$

Where h is the prediction horizon in seconds, f is the sampling frequency, and T_{max} is the maximum allowable thrust. Preliminary results shows that MPC is able to stabilize the system even in the presence of noise, highlighting the robustness.



(a) Trajectory

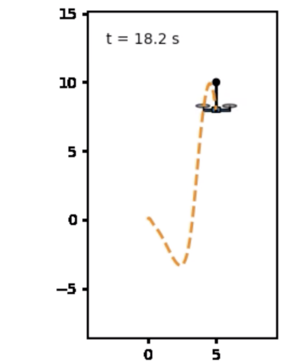
(b) Position

(c) Angles

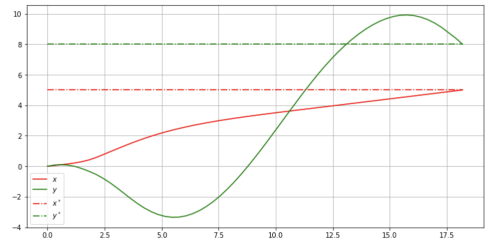
Figure 5: MPC method

This method however suffers from slower convergence and higher control effort, so other objective functions based on direct methods were also considered. A objective aimed at minimizing control inputs for a fix final time boundary conditions was developed as follows, with significantly lower control input required than the previous MPC formulation.

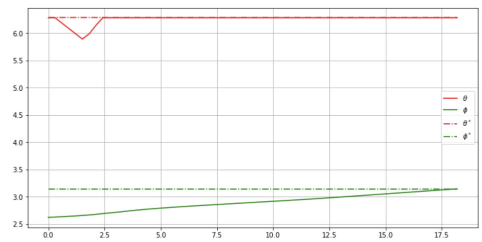
$$
\min \int_0^{t_f} T_1(t)^2 + T_2(t)^2 dt
$$



(a) Trajectory



(b) Position

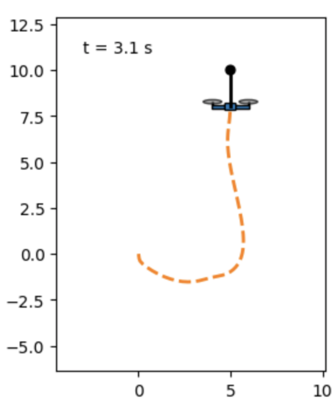


(c) Angles

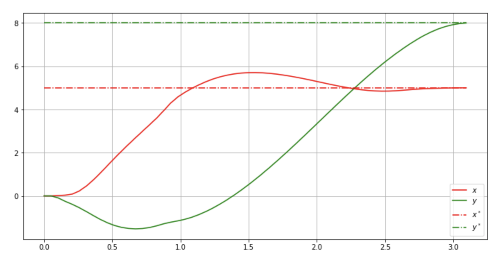
Figure 6: Direct Method with control inputs ≥ 0

Another objective function was developed that aimed to minimize control inputs and final time, with a free final time boundary condition. From Figure 7, we can see that the drone is doing very aggressive moves in order to balance pendulum in a short time. It has to move upside down to balance pendulum quickly.

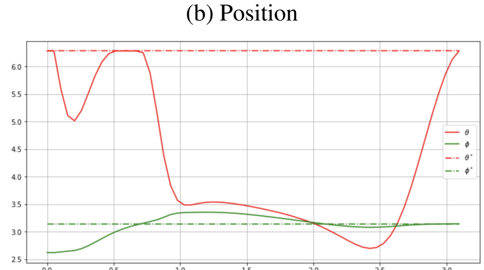
$$\min \int T_1(t)^2 + T_2(t)^2 dt + \frac{1}{2} t_f$$



(a) Trajectory



(b) Position



(c) Angles

Figure 7: Direct Method for free final time scenario without control input limitation

7 Next Steps

Our next step will focus on making the drone be able to swing up an inverted pendulum to the top of quadrotor using MPC controls and develop our RL controller using methods such as Proximal Policy Optimization or Advantage Actor Critic methods to compare with the baseline controller. [6] [7].

References

[1] Christoph Hintz, Shakeeb Ahmad, Joseph Kloeppel, and Rafael Fierro. Robust hybrid control for swinging-up and balancing an inverted pendulum attached to a uav. In *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pages 1550–1555, 2017.

[2] Moses Bangura and Robert Mahony. Real-time model predictive control for quadrotors. *IFAC Proceedings Volumes*, 47(3):11773–11780, 2014.

[3] Andrzej Jezierski, Jakub Mozaryn, and Damian Suski. A comparison of lqr and mpc control algorithms of an inverted pendulum. 07 2017.

[4] Rafael Figueroa, Aleksandra Faust, Patricio Cruz, Lydia Tapia, and Rafael Fierro. Reinforcement learning for balancing a flying inverted pendulum. In *Proceeding of the 11th World Congress on Intelligent Control and Automation*, pages 1787–1793. IEEE, 2014.

[5] Dylan Bates. A hybrid approach for reinforcement learning using virtual policy gradient for balancing an inverted pendulum. *arXiv preprint arXiv:2102.08362*, 2021.

[6] Alexandre El Assad, Elise Fournier-Bidoz, Pierre Lachevre, and Javier Sagastuy. Inverted pendulum on a quadcopter: A reinforcement learning approach. In *2017 CS229 Final Project*, 2017.

[7] Fang-I Hsiao, Cheng-Min Chiang, and Alvin Hou. Reinforcement learning based quadcopter controller. In *2019 CS238 Final Project*, 2019.