A Model Predictive Control Approach for Catching Ball with Quadrotor

Siyuan Wu (5488362), Ranbao Deng(5460069),

***Abstract*—Catching a flying object with a quadrotor is a classic task in robotics. In this paper, a linear model predictive control (MPC) strategy is presented for this task. The feasibility of ball catching can be guaranteed due to the stability for the origin of the closed-loop linearized system. A 3D simulation in Python is presented to show that designed model predictive controller is able to achieve this task.**

I. Introduction

Quadrotors are getting more attentions because they can achieve many challenging tasks agilely. One of these interesting tasks is using quadrotor catch a throwing ball [1]. Since quadrotor is a nonlinear dynamic system, accurate control will be difficult. The typical approaches to control quadrotors is cascade PD controller[2], which requires delicate tuning under high accuracy motion capture systems. Furthermore, it is hard to handle operational constraints for cascade PD controller, which could limit its accuracy under agile flights. Therefore, catching a flying ball with quadrotor could be difficult.

In this paper, we present a model predictive control (MPC) approach to control a quadrotor for catching a flying ball. We assume the accurate position and velocity of the ball is fully observable and design our model predictive control model under this assumption. We will give the initial position and velocity theoretically based on stability theory.

A. Quadrotor dynamics

According to [3], the quadrotor is a nonlinear dynamic system. The full dynamics of quadrotor is shown as follows

$$\dot{\boldsymbol{p}}_{\text{quad}} = \boldsymbol{v}_{\text{quad}}, \tag{1a}$$
$$m\dot{\boldsymbol{v}}_{\text{quad}} = -mg\boldsymbol{e_3} + \mathbf{R}F\boldsymbol{e_3}, \tag{1b}$$
$$\dot{\boldsymbol{R}} = \boldsymbol{R}\hat{\omega}, \tag{1c}$$
$$\boldsymbol{I}\dot{\boldsymbol{\Omega}} = \tau - \boldsymbol{\Omega} \times \boldsymbol{I}\boldsymbol{\Omega}, \tag{1d}$$

Siyuan Wu and Ranbao Deng are master students at TU Delft, The Netherlands. E-mail addresses: { S.Wu-14, R.Deng}@student.tudelft.nl.

$\boldsymbol{p}_{\text{quad}}$ and $\boldsymbol{v}_{\text{quad}}$ are vectors, representing the position and velocity of quadrotor. $m \in \mathbb{R}_{\geq 0}$ is the mass of quadrotor. g is the gravitational acceleration. F is the thrust by the rotor. $\boldsymbol{e_3} = (0, 0, 1)^T$ is a unit vector. $\boldsymbol{I} \in \mathbb{R}^{3\times 3}_{\geq 0}$ is the moment of inertia, which is a 3-by-3 matrix and all the elements are equal to or greater than zero. The rotation matrix $\boldsymbol{R}$ from quadrotor's body frame to world frame is given by :

$$\mathbf{R} = \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi) \tag{2}$$

where ϕ, θ, ψ represents roll, pitch and yaw angles.

Since the quadrotor dynamics in equation 1b and 1c is given by rotation matrix $\boldsymbol{R}$, we can expand it with roll, pitch and yaw angles

$$\dot{v}_{\text{quad},x} = f(\sin\psi\cos\theta\sin\phi + \cos\psi\sin\theta) \tag{3a}$$
$$\dot{v}_{\text{quad},y} = f(-\cos\psi\cos\theta\sin\phi + \sin\psi\sin\theta) \tag{3b}$$
$$\dot{v}_{\text{quad},z} = -g + f\cos\theta\cos\psi \tag{3c}$$

where $f = \frac{F}{m}$ is the normalized thrust given by rotors.

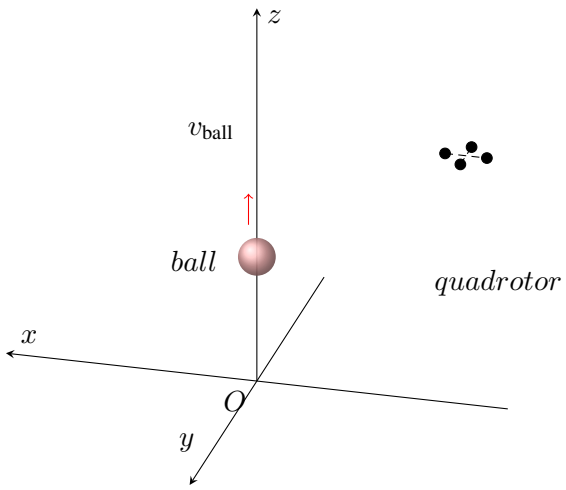

Fig. 1. Illustration of our task: a ball on the top is falling and the quadrotor on the right should approach and catch it

B. Thrown ball dynamics

Since catching a flying ball is consider in our project, we need to analysis the dynamic of thrown ball, and

model its states in our MPC design. Therefore, we assume the position and velocity of thrown ball is known in our project. To simplify the problem, we consider the motion of thrown ball as a vertical up throw, and we do not consider its horizontal displacement in x and y axis. The initial position of dropping box is $\boldsymbol{p}_{\text{ball}}^0 = [0, 0, h_0]^\intercal$ and position at time t is $\boldsymbol{p}_{\text{ball}} = [0, 0, h(t)]^\intercal$. We assume that the initial vertical velocity is $\boldsymbol{v}_{\text{ball}}^0 = [0, 0, v_h^0]^\intercal$. The dynamic of dropping box is described as follows.

$$\dot{\boldsymbol{p}}_{\text{ball}} = \boldsymbol{v}_{\text{ball}} \tag{4a}$$
$$\dot{\boldsymbol{v}}_{\text{ball}} = -g\boldsymbol{e}_3 \tag{4b}$$

where $\boldsymbol{e}_3$ carries the same meaning in equation 1d.

C. Linearization and discretization of quadrotor-ball system

To linearize quadrotor system, we need to assume the quadrotor is controlled near hovering state therefore roll, pitch angle of quadrotor is small enough to be linearized. Next, due to the assumption that the motion of the ball is thrown vertically upwards, we do not need to consider the direction of the quadrotor at the time of catching. Therefore, we restrict quadrotor from yawing, that is $\psi = 0$. The advantage of this simplification is equation 1d can be largely simplified:

$$\dot{v_x} = f \sin\theta \tag{5a}$$
$$\dot{v_y} = -f \sin\phi \tag{5b}$$
$$\dot{v_z} = -g + f\cos\theta \tag{5c}$$

Then, we can linearize the nonlinear dyanmics of quadrotor by applying 1st-order Taylor-expansion around the hovering state assuming the roll and pitch angle is near zero. The linearized dynamics is given by:

$$\dot{v}_{\text{quad},x} = g\theta \tag{6a}$$
$$\dot{v}_{\text{quad},y} = -g\phi \tag{6b}$$
$$\dot{v}_{\text{quad},z} = -g + f \tag{6c}$$

Consider quadrotor and ball as a single system in the catching ball task. We do not care the absolute position and orientation of quadrotor and the ball, however, the relative position and velocity is vital in this task. We subtract $\boldsymbol{p}_{\text{ball}}$ and $\boldsymbol{v}_{\text{ball}}$ from $\boldsymbol{p}_{\text{quad}}$ and $\boldsymbol{v}_{\text{quad}}$ separately, given by $\boldsymbol{p} = \boldsymbol{p}_{\text{quad}} - \boldsymbol{p}_{\text{ball}}$ and $\boldsymbol{v} = \boldsymbol{v}_{\text{quad}} - \boldsymbol{v}_{\text{ball}}$ as the quadrotor-ball system. Next, we discretize the quadrotor-ball system using Euler methods. The discrete quadrotor-ball dynamics system is given by:

$$p_x(k+1) = p_x(k) + v_x(k) \tag{7a}$$
$$p_y(k+1) = p_y(k) + v_y(k) \tag{7b}$$
$$p_z(k+1) = p_z(k) + v_z(k) \tag{7c}$$
$$v_x(k+1) = v_x(k) + g\theta(k) \tag{7d}$$
$$v_y(k+1) = v_y(k) - g\phi(k) \tag{7e}$$
$$v_z(k+1) = v_z(k) + f \tag{7f}$$
$$\theta(k+1) = \theta(k) + \omega_\theta \tag{7g}$$
$$\phi(k+1) = \phi(k) + \omega_\phi \tag{7h}$$

where p_x, p_y, p_z are relative displacement from quadrotor to the ball at x, y, z axis; ω_θ and ω_ϕ represent pitch and roll rate.

D. System dynamics

Although the dynamics of combined system is nonlinear due to the property of quadrotor dynamics, we can linearize it near hovering state as derived in Sec. I-C therefore the system we will analysis is a linear system,

$$x^+ = f(x, u) = \mathbf{A}x + \mathbf{B}u, \tag{8}$$

where the states are

$$x = [p_x, p_y, p_z, v_x, v_y, v_z, \theta, \phi]^\intercal . \tag{9}$$

The control input of the system is given by:

$$u = [f, \omega_\theta, \omega_\phi]^\intercal . \tag{10}$$

We will design a linear model predictive controller based on the discretized system dynamic above.

II. Model predictive control design

In this section, we will describe the design and formulation of our model predictive control strategy. We will clarify the state and input constraints of this MPC problem.

The goal of our task is to catch a flying ball with a quadrotor, in another words, the relative displacement $\boldsymbol{p}$ should be 0. To avoid damage while catching, the relative speed between the quadrotor and the ball should as small as possible, that is $\boldsymbol{v} = 0$. We consider a receding horizon MPC strategy with control horizon $N = 8$. The MPC optimization problem for this goal can be formulated as:

$$\begin{aligned} J(x_0, u) &= \sum_{k=0}^{N-1} \ell(x(k), u(k)) + V_f(x(N)) \\ &\text{s.t. } u \in \mathbb{U}, x \in \mathbb{X} \end{aligned} \tag{11}$$

The stage cost $\ell(x,u)$ and terminal cost $V_f(x(N))$ are defined as quadratic forms as:

$$\begin{aligned}\ell(x,u) &= \frac{1}{2}(x^\intercal \mathbf{Q} x + u^\intercal \mathbf{R} u)\\ V_f(x(N)) &= \frac{1}{2}(x(N)^\intercal \mathbf{P} x(N))\end{aligned} \tag{12}$$

where $\mathbf{Q}, \mathbf{R}$ are positive definite diagonal matrix given by:

$$\begin{aligned}\mathbf{Q} &= \mathrm{diag}([100, 100, 100, 1, 1, 1, 1, 1])\\ \mathbf{R} &= \mathrm{diag}([0.1\ 0.1\ 0.1])\end{aligned} \tag{13}$$

and $\mathbf{P}$ is given by Discrete Algebraic Riccati Equation (DARE):

$$\mathbf{P} = \mathbf{A}_K^\intercal \mathbf{P} \mathbf{A}_K + \mathbf{Q}_K \tag{14a}$$

$$\mathbf{Q}_K = \mathbf{Q} + \mathbf{K}^\intercal \mathbf{R} \mathbf{K} \tag{14b}$$

$$\mathbf{K} = -(\mathbf{B}^\intercal \mathbf{P} \mathbf{B} + \mathbf{R})^{-1} \mathbf{B}^\intercal \mathbf{P} \mathbf{A}^\intercal \tag{14c}$$

where $\mathbf{K}$ is the optimal gain given by unconstrained infinite-horizon LQR problem. In our project, we use a python function `scipy.linalg.solve_discrete_are` to solve DARE numerically.

The state and input constraints are obvious. Since the linearization is based on the assumption that quadrotor is near the hovering states, the roll and pitch angle should have a upper bound. We also assume the relative horizontal velocity are limited. So in general the state constraints $x \in \mathbb{X}$ are:

$$v_x \le v_{\max} \tag{15a}$$

$$v_y \le v_{\max} \tag{15b}$$

$$-\theta_{\max} \le \theta \le \theta_{\max} \tag{15c}$$

$$-\phi_{\max} \le \phi \le \phi_{\max} \tag{15d}$$

Due to the limited thrust and torque, the input normalized thrust f and absolute roll, pitch rate should have a upper bound. The input constraints $u \in \mathbb{U}$ can be formulated as:

$$f \le f_{\max} \tag{16a}$$

$$|\omega_\theta| \le \omega_{\max} \tag{16b}$$

$$|\omega_\phi| \le \omega_{\max} \tag{16c}$$

The terminal region $\mathbb{X}_f$ is given by the invariant constraint admissible set for the unconstrained optimal control dynamics $x^+ = \mathbf{A}_K x$. A typical solution of this equation is $x = 0$, whose physical meaning is the quadrotor is catching the ball and falling with the ball. This is a control invariant of $\mathbb{X}_f$ for two reasons. First, roll and pitch are zero so that the system will keep this state. Second, $u = \mathbf{K}x = 0$ so the unconstrained optimal control of this particular state is zero.

III. Asymptotic stability

In this section, we show that the designed linear MPC controller asymptotically stabilized the closed-loop system.

A. Linearized system analysis

Consider our linearized system

$$x^+ = \mathbf{A}x + \mathbf{B}u \tag{17}$$

where $\mathbf{A}$ is a 8-by-8 matrix and $\mathbf{B}$ is a 8-by-3 matrix. They represent the approximated system dynamics in Sec. I-D. We can verify this linearized system $(\mathbf{A}, \mathbf{B})$ is controllable with the help of `sympy`, which is an open-source Python library for symbolic mathematics:

$$\mathrm{rank}(\mathbf{W}_c) = \mathrm{rank}(\begin{bmatrix}\mathbf{B}\ \mathbf{AB}\ \mathbf{A}^2\mathbf{B}\cdots\ \mathbf{A}^7\mathbf{B}\end{bmatrix}) = 8. \tag{18}$$

So the controllability matrix is full rank and this linearized system is controllable. The stage cost of this MPC problem is positive definite.

The terminal cost $V_f(x)$ is given by

$$V_f(x) = \frac{1}{2} x^\intercal \mathbf{P} x \tag{19}$$

where $\mathbf{P}$ is the solution of Discrete Algebraic Riccati Equation (DARE).

B. Estimating terminal region set $\mathbb{X}_f$

Algorithm 1: Algorithm for finding $\mathbb{X}_f$

Input: linearized system dynamics: $\mathbf{A}, \mathbf{B}$, state penalty matrix $\mathbf{Q}$, input penalty matrix $\mathbf{R}$

$k \leftarrow 0$;
$\mathcal{S} = \{1, 2, 3, \cdots, s\}$;
while do
  foreach $i \in S$ **do**
    $x_i^\star \leftarrow$
    $$\begin{cases} \underset{x}{\mathrm{argmax}} f_i(\mathbf{A}_K^{k+1} x) \\ \text{s.t.} \quad f_j(\mathbf{A}_K^t x) \le 0, & \forall j \in \mathcal{S} \\ & \forall t \in \{1, 2, 3, \cdots, k\} \end{cases}$$
  end
  if $f_i(\mathbf{A}_K^{k+1} x_i^\star) \le 0 \quad \forall i \in \{1, 2, \ldots, s\}$ **then**
    $\mathbb{X}_f \leftarrow \{x \in \mathbb{R}^8 \mid f_i(\mathbf{A}_K^t x) \le 0, \forall i \in \mathcal{S}, \forall t \in \{0, 1, \ldots, k\}\}$;
    Break
  else
    $k \leftarrow k + 1$
  end
end
Result: $\mathbb{X}_f$

$\mathbb{X}_f$ is called terminal region in [4], which is an invariant constraint admissible set for $x^+ = \mathbf{A}_K x$ where $\mathbf{A}_K = \mathbf{A} + \mathbf{BK}$ and $\mathbf{K}$ is the optimal control gain defined by the unconstrained infinite-horizon Linear Quadratic Regulator (LQR) problem in equation 14c. The stability in the terminal region X_f is guaranteed by the LQR control law.

The numerical computation algorithm to estimate $\mathbb{X}_f$ is shown in Algorithm 1 according to [5].

Figure 2 illustrates the numerical results given by Algorithm 1 projected into p_x and p_z plane, indicating the initial setup of p_x and p_z inside the blue region is stable guaranteed by uncontrained LQR law. The $\mathbb{X}_f$ is symmetric in x-direction but asymmetric in z-direction, indicating the height of the ball should be a little bit higher than the quadrotor for successful catching.

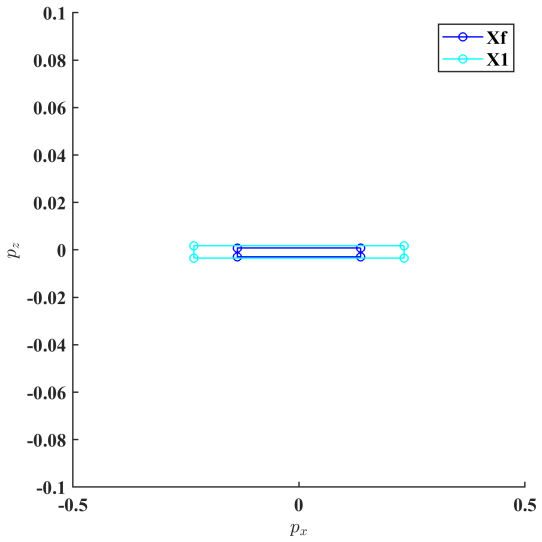

Fig. 2. Terminal region $\mathbb{X}_f$ (blue) and 1 stage feasible set $\mathcal{X}_1$ (cyan) projected to p_x and p_z plane. p_z and p_x are relative position of the quadrotor to the ball in z and x axis

C. Stability proof

According to **Theorem 2.41** of [4], to prove the origin is asymptotically stable for the closed-loop system, we need to check if the closed-loop system satisfy **Assumption 2.2**, **Assumption 2.3** in [4].

1) ***Assumption 2.2*** *(Continuity of system and cost):* It's obvious that the function f, the stage cost $\ell(x,u)$ and terminal cost $V_f(X(N))$ is continuous. It's easy to verify that $f(0,0) = 0$, $\ell(0,0) = 0$ and $V_f(0) = 0$ because dynamics functions f is linear, stage and terminal cost have quadratic form. Therefore this assumption is satisfied.

2) ***Assumption 2.3*** *(Properties of constraint sets):* According to equation 15 and 16 both state constraints

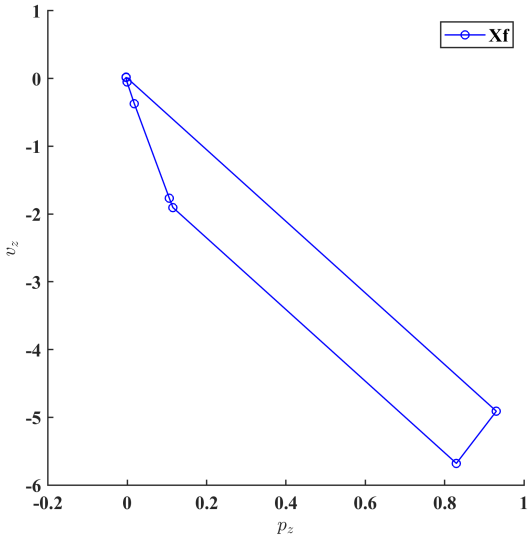

Fig. 3. Terminal region $\mathbb{X}_f$ (blue) projected to p_z and v_z plane, v_z is the relative velocity of the quadrotor to the ball in z axis.

$x \in \mathbb{X}$ and input constraints $u \in \mathbb{U}$ are closed, compact. The terminal region $\mathbb{X}_f \subseteq \mathbb{X}$ is also compact. And $0 \in \text{int}(\mathbb{X}_f)$ is satisfied. Therefore this assumption is satisfied.

3) ***Assumption 2.23*** *(Modified basic stability assumption) :* According to [4], to prove the asymptotic stability of the origin, $V_f(\cdot)$, $\mathbb{X}_f$ and $\ell(\cdot,\cdot)$ must satisfy following conditions:

- For all $x \in \mathbb{X}_f$, there exists a u such that $u \in \mathbb{U}$ satisfying
$$V_f(f(x,u)) - V_f(x) \leq -\ell(x,u), \quad f(x,u) \in \mathbb{X}_f \tag{20}$$
- For all $x \in \mathbb{X}$ and $u \in \mathbb{U}$ satisfying
$$\ell(x,u) \geq \alpha_1(|(x,u)|) \tag{21}$$
- For all $x \in \mathbb{X}_f$ satisfying
$$V_f(x) \leq \alpha_f(|x|), \quad x \in \mathbb{X}_f \tag{22}$$

Condition 21 and 22 are satisfied because the stage cost and terminal cost has quadratic form:

$$\ell(x,u) = \frac{1}{2}(x^\intercal \mathbf{Q} x + u^\intercal \mathbf{R} u) \geq \frac{1}{2}(x^\intercal \mathbf{Q} x) \geq \frac{1}{2}\lambda(\mathbf{Q})|x|^2, \tag{23a}$$

$$V_f(x) = \frac{1}{2}x^\intercal \mathbf{P} x \leq \frac{1}{2}\lambda_{\max}(\mathbf{P})(|x|^2) \tag{23b}$$

Given the terminal invariant set $\mathbb{X}_f$ in Fig. 2, equation 20 can be verified numerically. Fig. 4 provides a verification of Lyapunov decrease given the inital state $p_x = -0.01, p_y = 0.01, p_z = 0.01, v_z = -0.1$.

4) ***Theorem 2.41****:* According to Theorem 2.41, because Assumption 2.2, Assumption 2.3 and Assumption 2.23 are satisfied, the origin is exponentially stable for the closed-loop system.

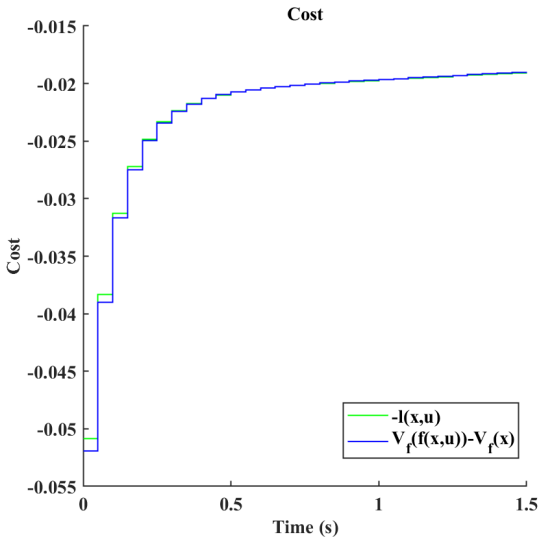

Fig. 4. A verification of Assumption 2.23 equation 20

D. Estimating feasible set $\mathcal{X}_N$

Since $\mathbb{X}_f$ is calculated numerically, we can deductive backward and estimate $\mathcal{X}_N$. The idea is simple, we will check possible discrete initial states x_0 if the terminal stage $x(N)$ satisfy the terminal constraints $\mathbb{X}_f$. If $x(N) \in \mathbb{X}_f$, then x_0 is feasible so that should be included in the feasible set $\mathcal{X}_N$.

Algorithm 2: Algorithm for finding $\mathcal{X}_N$

Input: the MPC problem $\mathbb{P}_N(x_0, \boldsymbol{u})$, initial state $x(0)$, terminal invariant set $\mathbb{X}_f$

$\mathcal{X}_N \leftarrow \emptyset$;
foreach $x_0 \in \mathcal{X}_{IC}$ **do**
  if $x(N) \in \mathbb{X}_f, u \in \mathbb{U}, x \in \mathbb{X}$ **then**
    $\mathcal{X}_N \leftarrow x_0 \subseteq \mathcal{X}_N$;
  end
end
Result: $\mathcal{X}_N$

In Figure 2, the outer polygon $\mathcal{X}_1$ in cyan represents a 2D projection of the 1-stage feasible set $\mathcal{X}_N$ when receding horizon length $N = 1$. According to our computation $\mathcal{X}_1$ contains 299 linear inequalities in total. And number of linear inequalities increases exponentially when N increases. Due to limited computation resources, we cannot provide X_N with longer horizon length.

IV. Numerical Simulations

In this section, we run several numerical simulations, as shown in Figure 5, where we campare our MPC with and LQR controller and test the performance of our MPC controller to see the effect of different parameters on our model predictive controller. The figures shown below are only showing the relative distance in z and x direction for the sake of brevity, since y-axis behaves similar to x-axis. The environment being tested is $(p_{z0} = -0.5m, p_{x0} = 0.3m, v_{bz} = 3m/s)$, which means the quadrotor has $-0.5m$ offest in z direction and $0.3m$ in x direction. The ball was given a initial velocity of $3m/s$ upwards.

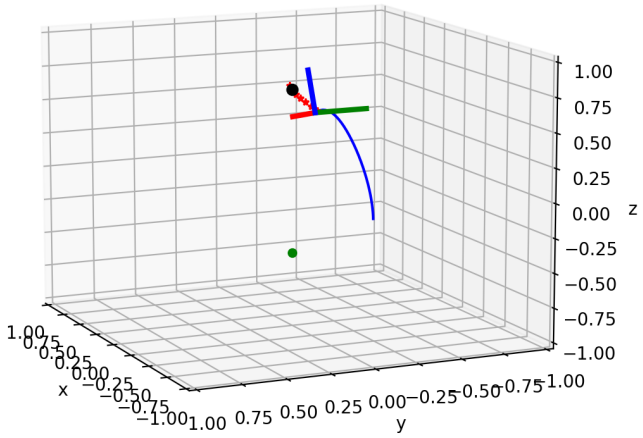

Fig. 5. 3D simulation of our MPC.

A. MPC and LQR comparison N

Given the initial condition within the terminal region, the LQR and MPC controller behaves similar as can be shown in Figure 6. This is as intended since the MPC was designed with DARE terminal cost.

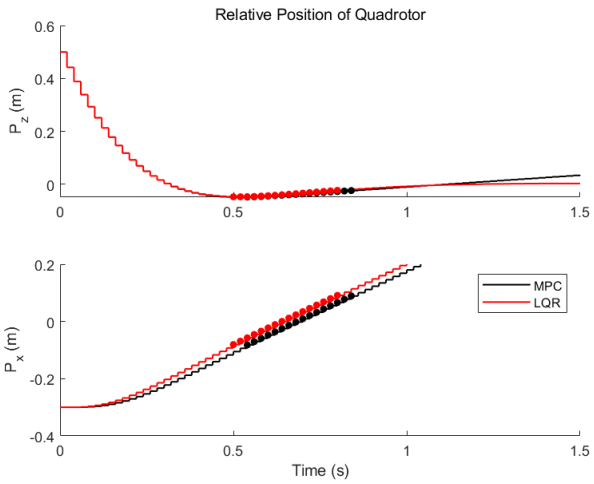

Fig. 6. Comparison between MPC and LQR controllers. Initial states in terminal region. States where the quadrotor successfully caught the ball is marked with dots.

However, when given a initial condition outside the terminal region, the LQR would fail to control the quadrotor. The MPC outperforms LQR greatly under such scenarios as can be shown in Figure 7.

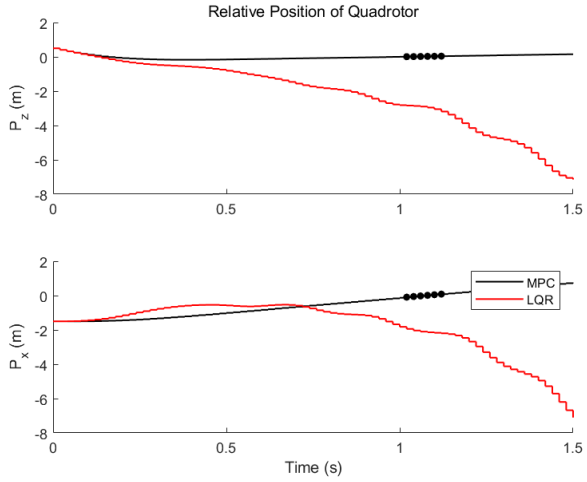

Fig. 7. Comparison between MPC and LQR controllers. Initial state outside terminal region. States where the quadrotor successfully caught the ball is marked with dots.

B. Different Prediction Horizon N

To see how different prediction horizons would effect our MPC's performance, tests with diffrent N are conducted on the same environment as mentioned above. Figure 8 shows the states trajectories with different prediction Horizons. It can be seen that when N is larger, the MPC would succeed in catching the ball faster. However, when N is too high, the quadrotor would "try too hard" to catch the ball, results in higher velocity and less smooth trajectories. Thus N was kept as 8 in further tests.

C. Different State Stage Cost $\mathbf{Q}$

This part illustrates how different choices of cost function weights change the MPC performance, and specifically the state stage cost $\mathbf{Q}$. The default $\mathbf{Q}$ was set as $\mathrm{diag}([100, 100, 100, 1, 1, 1, 1, 1])$ based on experience and trials. As can be seen from Figure 8, the effect of $\mathbf{Q}$ begins to saturate around $\mathbf{Q} = I$. It might be due to the fact that with a larger $\mathbf{Q}$, larger inputs are needed to lower the errors generated, However, the constraints on inputs make this impossible and results in a saturation.

D. Different Input Stage Cost $\mathbf{R}$

Similar tests are conducted on input stage cost weight $\mathbf{R}$ to see its effect on our MPC. Figure 10 shows that

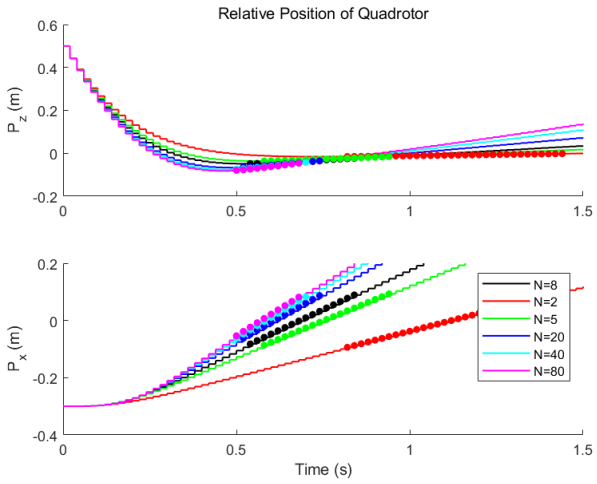

Fig. 8. State trajectories under different Horizons N. States where the quadrotor successfully caught the ball is marked with dots.

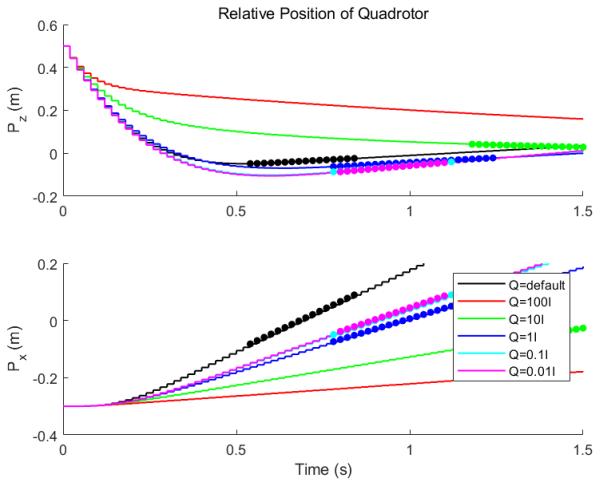

Fig. 9. State trajectories with different stage cost weight $\mathbf{Q}$. States where the quadrotor successfully caught the ball is marked with dots.

lower $\mathbf{R}$ tends to saturate and would stop have much influence till round $\mathbf{R} = 0.1I$. Figure 11 shows that with $\mathbf{R}$ lower than $0.1I$ the input are more responsive and results in better chance of catching the ball. It seems that with extreme high $\mathbf{R}$, the cost would be too high to perform any meaningful action.

E. Different Terminal Cost $\mathbf{P}$

To ensure stability, the terminal cost weight is calculated via DARE as mentioned above. However, to see how the terminal cost would affect the MPC, we tested several different $\mathbf{P}$ as shown in Figure 12.

F. Different Environment Settings

To see how our MPC performs on different environment and to prove that we have the right feasible and

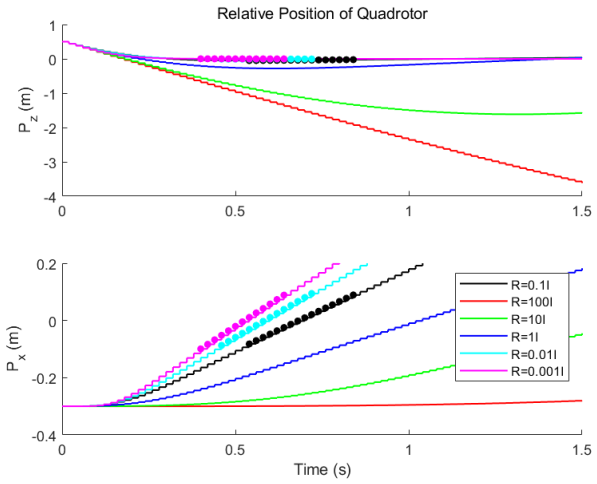

Fig. 10. State trajectories with different input stage cost weight $\mathbf{R}$. States where the quadrotor successfully caught the ball is marked with dots.

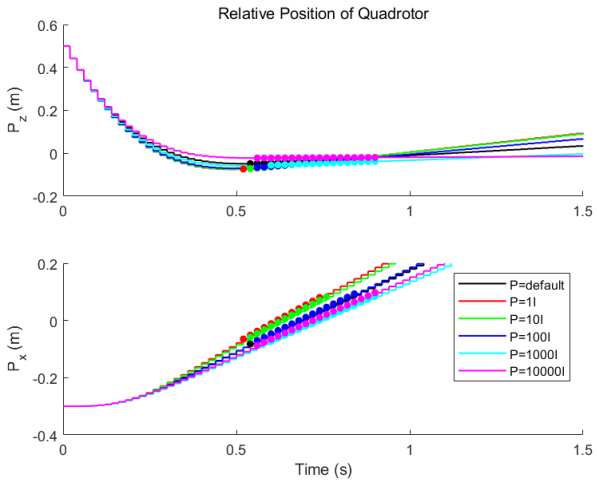

Fig. 12. State trajectories with different terminal cost weight $\mathbf{P}$. States where the quadrotor successfully caught the ball is marked with dots.

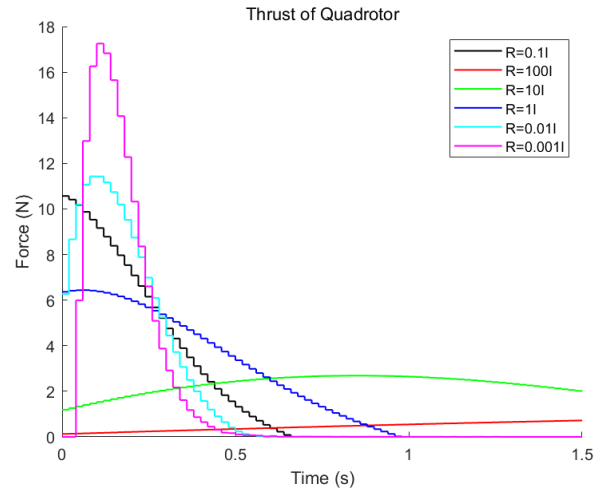

Fig. 11. Input thrust under different input stage cost $\mathbf{R}$.

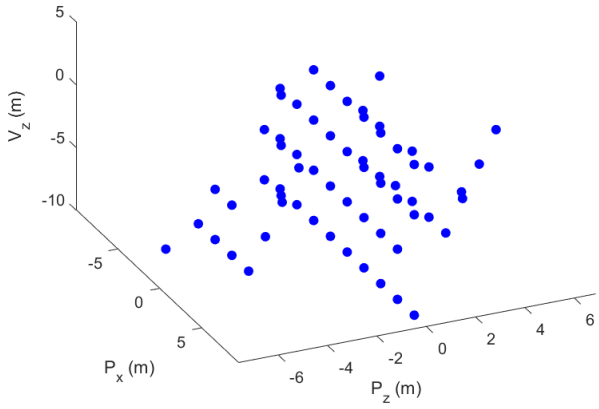

Fig. 13. Feasible environment where the ball could be caught by MPC.

terminals set, we did the catching test with different p_{z0}, p_{x0}, v_{bz}. The catching was considered successful if the quadrotor was close enough (0.1m) to the ball. The results could be seen in 13. It can be shown that if the ball is higher than the quadrotor, it would be easier to catch. And when an initial velocity upward (negative) is given, the horizontal distance could be allowed to be very large. This also aligns with the intuitions. Compared to what we derived, the terminal region in Fig. 2 and Fig. 3, in previous part, this feasible region is somewhat similar in both shape and size.

V. Conclusion

The ball-catching quest for a quadrotor via MPC is an interesting and challenging problem. The dynamics of the quadrotor is highly non-linear. To simplify the problem, the quadrotor and the ball has integrated dynamics by using the relative distance and velocity in z direction, and the ball was limited to move only up and down. Furthermore, the dynamics are linearized and thus the stability and feasibility could be satisfied. However, due to physical constraints of the system (e.g. no negative thrust for quadrotor), the quadrotor could not always keep the trajectory with the ball in practise. That's not a fatal problem after all, as long as the ball is caught, the quadrotor is free for a new task. Keeping track with the ball before catching the ball is also a good feature to prevent the quadrotor from being hardly hitted by the dropping ball.

The comparisons on LQR, different parameters and

environments give more insights on the principles of MPC, which would help us in future, to make better MPC designs.

REFERENCES

[1] K. Su and S. Shen, "Catching a flying ball with a vision-based quadrotor," in *International Symposium on Experimental Robotics*. Springer, 2016, pp. 550–562.

[2] "Controller Diagrams — PX4 User Guide." [Online]. Available: https://docs.px4.io/v1.12/en/flight_stack/controller_diagrams.html

[3] P. Wang, Z. Man, Z. Cao, J. Zheng, and Y. Zhao, "Dynamics modelling and linear control of quadcopter," in *2016 International Conference on Advanced Mechatronic Systems (ICAMechS)*, 2016, pp. 498–503.

[4] J. Rawlings and D. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2008.

[5] E. Gilbert and K. Tan, "Linear systems with state and control constraints: the theory and application of maximal output admissible sets," *IEEE Transactions on Automatic Control*, vol. 36, no. 9, pp. 1008–1020, 1991.