A Model Predictive Control Approach for Trajectory Tracking of a Quadrotor

Weiming Chen (5015006), Vibhav Inna Kedege (4998596)

***Abstract*—In this paper, a Model predictive control (MPC) based controller has been designed for a quadrotor in order to enable it to follow a predefined trajectory. We discuss the modelling equations, stability, simulations and observations in order to show that the controller that has been designed works as per expectations in order to enable trajectory tracking functionality in a quadrotor.**

I. Introduction

In recent years, unmanned air vehicles (UAVS) have experienced a huge interest among academic researchers and industries for their exceptional agility and robustness. A quadrotor is a multirotor air vehicle with four attached propellers, one for each rotor. The quadrotor is a non-holonomic system due to the fact that only four actuators (rotors) are used to control all six degrees of freedom. However, using different combinations of these four rotor angular velocities, an equivalent force at each propeller is generated that translates and rotates the quadrotor in any possible direction.

Precise trajectory tracking is a demanding feature for UAVs to successfully perform required tasks, especially when operating in realistic environments where external disturbances may heavily affect the flight performance and when flying in the vicinity of structure.

MPC is a type of optimal control which predicts the system's future behaviour based on the model dynamic system. The key advantage is the ability to obtain the optimal control action to drive the output to the desired reference by solving an online optimization problem in which an objective function has to be minimized. This objective function is subject to a set of state and input constraints to control the system's behaviour in a fixed time horizon. Linear controllers have been proposed by several researchers in order to overcome dealing with the non-linear dynamics of the quadrotor during the trajectory tracking operation.

In this paper, a model predictive control strategy is implemented for trajectory tracking control, based on a linearization of the quadrotor model. The paper starts with an introduction to the quadrotor model, the states and its linearisation in section II.The approach of splitting the system into 4 independent systems and the control strategy for it has been descried in III. Section IV then elaborates on how the controller designed is asymptotically stable and finally section V explains in detail the graphical results of the quadrotor in following a trajectory and also compares its performance with a PID controller.

Weiming Chen and Vibhav Inna Kedege are master students at TU Delft, The Netherlands. E-mail addresses: {`W.Chen-20`, `V.InnaKedege`}`@student.tudelft.nl`.

II. System model

A. *Quadrotor Dynamics Model*

Euler angles describe the orientation of a rigid body in space, each angle represents the rotation of the quadrotor in a certain axis. It's used to construct the rotations matrices. These rotation matrices map the body frame onto the inertial frame. The final rotation matrix can be expressed as follows:

$$\begin{aligned} &R_{zyx}(\phi,\theta,\psi) = R_z(\psi)\cdot R_y(\theta)\cdot R_x(\phi) \\ &= \begin{bmatrix} c(\theta)c(\psi) & s(\phi)s(\theta)c(\psi)-c(\phi)s(\psi) & c(\phi)s(\theta)c(\psi)+s(\phi)s(\psi) \\ c(\theta)s(\psi) & s(\phi)s(\theta)s(\psi)+c(\phi)c(\psi) & c(\phi)s(\theta)s(\psi)-s(\phi)c(\psi) \\ -s(\theta) & s(\phi) & c(\theta) \end{bmatrix} \end{aligned} \quad (1)$$

where ϕ, θ and ψ are the quadrotor's roll, pitch, yaw angles respectively. While $s = sin$, $c = cos$ and $t = tan$ functions. T matrix is the transformation matrix which maps the angular velocities in the body frame onto the inertial frame:

$$T = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)} \end{bmatrix} \quad (2)$$

The nonlinear dynamic equations describing the quadrotor can be derived using Newton-Euler method as [1]:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} p + r[c(\phi)t(\theta)] + q[s(\phi)t(\theta)] \\ q[c(\phi)] - r[s(\phi)] \\ r\frac{c(\phi)}{c(\theta)} + q\frac{s(\phi)}{c(\theta)} \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} \dot{v_x} \\ \dot{v_y} \\ \dot{v_z} \end{bmatrix} = \begin{bmatrix} -\frac{f_t}{m}[s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta)] \\ -\frac{f_t}{m}[c(\phi)s(\psi)s(\theta) - c(\psi)s(\phi)] \\ g - \frac{f_t}{m}[c(\theta)c(\phi)] \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{Iy-Iz}{Ix}rq + \frac{\tau x}{Ix} \\ \frac{Iz-Ix}{Iy}pr + \frac{\tau y}{Iy} \\ \frac{Ix-Iy}{Iz}pq + \frac{\tau z}{Iz} \end{bmatrix} \quad (6)$$

where x, y and z are the linear positions and vx, vy and vz are the linear velocities, while ϕ, θ and ψ are the angular positions and p, q and r are the angular velocities.

Here, we also consider the inputs that can be applied to the system in order to control the behavior of the quadrotor. Commonly, the control inputs that are considered are one for the vertical thrust and one for each of the angular motions. Here we consider the values of the input forces and torques proportional to the squared speeds of the rotors, their values are the following:

$$\begin{bmatrix} f_t \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ -bL & 0 & bL & 0 \\ 0 & -bL & 0 & bL \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} (\omega_1)^2 \\ (\omega_2)^2 \\ (\omega_3)^2 \\ (\omega_4)^2 \end{bmatrix} \tag{7}$$

where the thrust coefficient and the drag coefficient are denoted by the b and d respectively. The letter L represents the distance between the center line of the rotor and the quadrotor's center line.

B. Linearized System Dynamics

The motion equation that describes the quadrotor are complex, nonlinear and highly coupled and therefore is difficult to find the accurate behavior and also to perform the control in simulation. So the primary elements that are required while hovering and useful to move the quadrotor to the desired operational state are only considered. The remaining elements that become significant during high speeds are neglected. It is thus, possible to rewrite the equations of the quadrotor in the spate-space:

$$\dot{x} = Ax + Bu \tag{8}$$

The system states $\mathbf{x}$ can be expressed by the following vector:

$$\text{x}=\begin{bmatrix} \phi & \theta & \psi & p & q & r & v_x & v_y & v_z & x & y & z \end{bmatrix}^T \in \mathbb{R}^{12} \tag{9}$$

Set $\mathbf{u}$ the control vector:

$$u = \begin{bmatrix} f_t & \tau_x & \tau_y & \tau_z \end{bmatrix}^T \in \mathbb{R}^4 \tag{10}$$

A linearization procedure is then applied around an equilibrium point $\overline{x}$, which for fixed input $\overline{u}$ is the solution of the algebraic system. This solution is the value of state's vector, which on fixed constant input is the solution of algebraic system:

$$\dot{x} = f(\overline{x}, \overline{u}) = 0 \tag{11}$$

The linearization is done by approximating the sine function with its argument and the cosine function with unity. The approximation is valid if the argument is small. The equilibrium point obtained by doing this is,

$$\overline{x} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{x} & \bar{y} & \bar{z} \end{bmatrix}^T \in \mathbb{R}^{12} \tag{12}$$

From the equations, we find that the equilibrium point is obtained by the constant input value,

$$\overline{u} = \begin{bmatrix} mg & 0 & 0 & 0 \end{bmatrix}^T \in \mathbb{R}^4 \tag{13}$$

The resulting linearized system is described by the following equations:

$$\begin{bmatrix} \dot{v_x} \\ \dot{v_y} \\ \dot{v_z} \end{bmatrix} = \begin{bmatrix} -g\theta \\ g\phi \\ g - \frac{f_t}{m} \end{bmatrix} \tag{14}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{\tau x}{Ix} \\ \frac{\tau y}{Iy} \\ \frac{\tau z}{Iz} \end{bmatrix} \tag{15}$$

After determining the equilibrium point and the corresponding nominal input, we perform the actual linearisation and obtain the matrices associated with the linear system as,

$$\text{A}= \frac{\partial f(x,u)}{\partial x}\bigg|_{x=\overline{x}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{16}$$

$$B = \frac{\partial f(x,u)}{\partial u}\bigg|_{\substack{x=\overline{x} \\ u=\overline{u}}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 & \frac{1}{I_z} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{17}$$

III. Model predictive control design

Based on the dynamics model, the linear MPC scheme based on the linearized model for the quadrotor are designed to control the nonlinear quadrotor dynamic system. The general cost function used to solve the MPC problem at each iteration is defined as follows:

$$\begin{gathered} J(x_0, u) = \sum_{k=0}^{N-1} \ell(x(k), u(k)) + V_f(x(N)) \\ \text{s.t. } u \in \mathbb{U}, x \in \mathbb{X} \end{gathered} \tag{18}$$

To ensure the recursive constraint satisfaction with the horizon, the state and input should satisfy the constraints recursively when k increases from 0 to $N-1$. The control input and state constraints are given by:

$$\mathbb{U} = \left\{ u \in \mathbb{R}^4 | \begin{bmatrix} f_{t\min} \\ \tau_{x\min} \\ \tau_{y\min} \\ \tau_{z\min} \end{bmatrix} \leq u \leq \begin{bmatrix} f_{t\max} \\ \tau_{x\max} \\ \tau_{y\max} \\ \tau_{z\max} \end{bmatrix} \right\} \tag{19}$$

$$\mathbb{X} = \left\{ x \in \mathbb{R}^{12} | \begin{bmatrix} \phi_{\min} \\ \theta_{\min} \\ \psi_{\min} \\ p_{\min} \\ q_{\min} \\ r_{\min} \\ v_{x\min} \\ v_{y\min} \\ v_{z\min} \\ x_{\min} \\ y_{\min} \\ z_{\min} \end{bmatrix} \leq x \leq \begin{bmatrix} \phi_{\max} \\ \theta_{\max} \\ \psi_{\max} \\ p_{\max} \\ q_{\max} \\ r_{\max} \\ v_{x\max} \\ v_{y\max} \\ v_{z\max} \\ x_{\max} \\ y_{\max} \\ z_{\max} \end{bmatrix} \right\} \tag{20}$$

With the references given at each iteration, a reference tracking MPC problem is aimed at being solved. The stage and terminal costs are defined as follows in order to account for the reference signals:

$$\begin{aligned} \ell(x(k), u(k)) &= (x(k)-x_{\text{ref}})^T Q\,(x(k)-x_{\text{ref}}) \\ &\quad + (u(k)-u_{\text{ref}})^T R\,(u(k)-u_{\text{ref}}) \\ V_f(x(N)) &= (x(N)-x_{\text{ref}})^T P\,(x(N)-x_{\text{ref}}) \end{aligned} \tag{21}$$

where Q is the penalty on the state error, R is the penalty on control input error and P is the terminal state error penalty. These matrices are all positive definite and relevant to the cost computation. The choices of Q and R were obtained following an iterative tuning procedure which would ensure a balance between control action and settling time. P is obtained to be the solution to the discrete algebraic Riccati equation (DARE), which is given by:

$$P = A^T PA - A^T PB\left(R + B^T PB\right)^{-1} {}^T PA + Q \tag{22}$$

In our MPC design, we split the whole system into 4 independent control systems:

- System-X. Pitch moment input τ_x to position x. The system has four states: ϕ, p, v_x, x.
- System-Y. Roll moment input τ_y to position y. The system has four states: θ, q, v_y, y.
- System-Z. Total thrust input $_t$ to position z. The system has two states: v_z, z.
- System-Yaw. Yaw moment input τ_z to yaw angle ψ. The system has two states: ψ, r.

The four independent sub-system design also follow the constraints corresponding to the input and states. To ensure the recursive constraint satisfaction with the horizon, the implementation of the state and input constraints recursively are defined as:

$$x(k+1) = Ax(k) + Bu(k) \tag{23}$$

$$F_x \cdot x(k) \leq f_x \tag{24}$$

$$F_{u,x} \cdot u(k) \leq f_{u,x} \tag{25}$$

For example, in the yaw sub-system, we have:

$$F_x = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}, f_x = \begin{bmatrix} \psi_{\min} \\ \psi_{\max} \\ r_{\min} \\ r_{\max} \end{bmatrix} \tag{26}$$

$$F_{u,x} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, f_{u,x} = \begin{bmatrix} \tau_{z\min} \\ \tau_{z\max} \end{bmatrix} \tag{27}$$

x_{ref} and u_{ref} are respectively the target steady state vector and target control input at time k. Similarly, the expected values of x_{ref} and u_{ref} according to the given reference for tracking also follows the same constraint implementation,

$$F_x \cdot x_{ref} \leq f_x \tag{28}$$

$$F_{u,x} \cdot u_{ref} \leq f_{u,x} \tag{29}$$

Also, the value of x_{ref} and u_{ref} should satisfy the equation:

$$x_{ref} = Ax_{ref} + Bu_{ref} \tag{30}$$

$$r = Cx_{ref} + Du_{ref} \tag{31}$$

where r is the tracking reference.

Then an objective function $V(x_{ref}, u_{ref})$ is defined with the aim to minimize input:

$$V\left(x_{ref}, u_{ref}\right) = u_{ref}^{\mathrm{T}} u_{ref} \tag{32}$$

It should be noted that if the reference cannot be reached, the objective function $V(x_{ref}, u_{ref})$ should try to minimize the difference between output and reference, which is:

$$V(x_{ref}, u_{ref}) = (r - Cx_{ref} - Du_{ref})^{\mathrm{T}}(r - Cx_{ref} - Du_{ref}) \tag{33}$$

The tuning procedure is illustrated in Section V. Note that only the first control input is applied to the system, and the process is repeated the next time step in a receding horizon fashion.

IV. Asymptotic stability

To prove stability of the MPC controller, we consider the LQR regulator for the MPC and prove theorem 2.24 as given in [2], which mentions there is a Lyapunov function in $\mathbb{X}_N$ that drives the given closed loop system $x^+ = Ax$ to the origin, given that the assumptions 2.2, 2.3 and 2.23 needs to have been satisfied. This is commonly referred to as the asymptotic stability with stage cost. In this section, we consider the linear system of equation 8 whose A and B matrices are given by 16 and 17 and as we consider the case of an LQR regulator, we consider the equations mentioned in 21 to be represented as,

$$\begin{aligned} \ell(x(k), u(k)) &= x^T Qx + u^T Ru \\ V_f(x(N)) &= x^T Px \end{aligned} \tag{34}$$

For asymptotic stability, we move to prove the above assumptions and hence the theorem.

A. controllability

In order to prove that the system is controllable, we compute the controllability/reachability matrix given by $W_r = \begin{bmatrix} B & AB & A^2B.. & A^{11}B \end{bmatrix}$ and its associated rank. If this gives us a full rank, then the system is controllable. In this case the rank was computed as 12 and therefore we conclude it is full rank and hence controllable.

B. Assumption 2.2

This assumption is referred to as the continuity of system and cost. This states that the system dynamics (f), the stage cost (ℓ) and the terminal cost (V_f) are continuous and $f(0,0) = 0$, $\ell(0,0) = 0$ and $V_f = 0$.

It can be seen from the above equations, that the linear function f is 0 when the state and input are 0 that is, $f(0,0) = 0$. It can also be seen that the stage cost $\ell(0,0) = 0$ and $V_f(0) = 0$. Further the functions f, ℓ and V_f are also continuous in nature. Due to this, assumption 2.2 is satisfied.

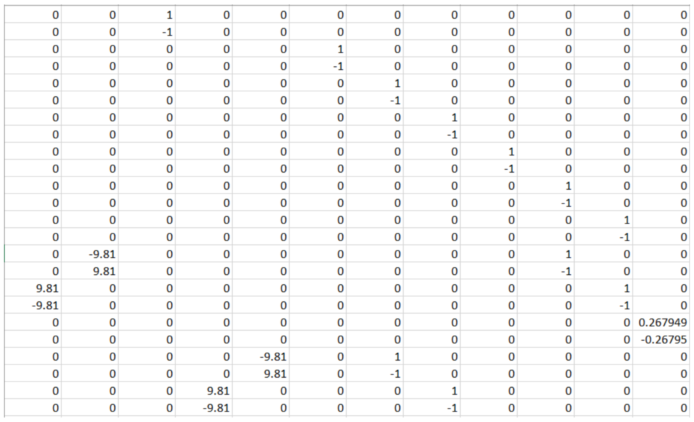
C. Assumption 2.3

This assumption is referred to as the property of constraint sets and requires that the set $\mathbb{Z}$ consisting of $\mathbb{U}$ and $\mathbb{X}$ needs to be closed and consist of the origin. Further, the set $\mathbb{U}$ needs to also be bounded and the terminal set $\mathbb{X}_f \subset \mathbb{X}$.

From table I we see that each of the states and control inputs are closed, bounded and consists of the origin in the interior. Due to this, assumption 2.3 is satisfied. Further the terminal set is also computed to lie within the set of state constraints. This will be proved in the next section.

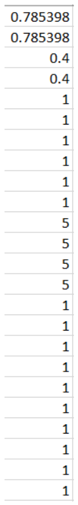
D. Invariant Terminal Set $\mathbb{X}_f$

In order to compute the invariant terminal set $\mathbb{X}_f$, we implement algorithm 1 from question 2 of the MPC course exercise set 4 [3].In this, an LQR optimal gain K is computed, which is then used to form a closed loop system matrix $A_k = A + BK$. From this, the invariance is taken care of as the set is computed such that $x^+ = A_{k+1}x$ also lies in the set. It is important to note that this set is represented in an $A_f x \leq b_f$ form and the results of the matrices are as given in figure 1 and figure 2.



0	0	1	0	0	0	0	0	0	0	0	0
0	0	-1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	-1	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	-1	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	-1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	-1	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	-1	0	0
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	-1	0
0	-9.81	0	0	0	0	0	0	0	1	0	0
0	9.81	0	0	0	0	0	0	0	-1	0	0
9.81	0	0	0	0	0	0	0	0	0	1	0
-9.81	0	0	0	0	0	0	0	0	0	-1	0
0	0	0	0	0	0	0	0	0	0	0	0.267949
0	0	0	0	0	0	0	0	0	0	0	-0.26795
0	0	0	0	-9.81	0	1	0	0	0	0	0
0	0	0	0	9.81	0	-1	0	0	0	0	0
0	0	0	9.81	0	0	0	1	0	0	0	0
0	0	0	-9.81	0	0	0	-1	0	0	0	0

Fig. 1. A_f



0.785398
0.785398
0.4
0.4
1
1
1
1
1
1
5
5
5
5
1
1
1
1
1
1
1
1
1
1

Fig. 2. b_f

Further, algorithm 1 and 2 in exercise set 3 of the MPC course were used in order to compute the region of attraction (χ_N). However the algorithm did not work well for high dimensional states and input. Nevertheless, in the numerical simulations, we would compute the initial state x_0 by ensuring that it lies within $\mathbb{X}_f$ by checking the condition that $A_f x_0 \leq b_f$. On doing, this we found that the MPC controller was able to track the trajectory satisfactorily and hence was able to drive the states to the origin.

E. Assumption 2.23(a)

In [2], this is referred to as the modified basic stability assumption. In this assumption, there exists a lyapunov function for the terminal cost V_f such that, $V_f(f(x,u) - V_f(x) \leq -\ell(x,u)$ for all, $f(x,u) \in \mathbb{X}_f$.

This can be proved by considering $V_f = \frac{1}{2}x^T Px$, where P is the the solution to the DARE equation. As shown in the previous subsection, the constraints of $\mathbb{X}_f \subseteq \mathbb{X}$ is satisfied by considering the closed loop optimal LQR to be applied that gives rise to to using the modified matrix $A_k = A + BK$. Following this the DARE equation solution becomes $P = A_K^\top P A_K + Q_K$. Thus for the system, $x^+ = A_K x$, we have the following,

$$\begin{aligned} V_f(x^+) = V_\mathrm{f}(A_K x) &= \frac{1}{2}\left(A_K x\right)^\top P\left(A_K x\right) \\ &= \frac{1}{2}x^\top Px - \frac{1}{2}x^\top Q_K x \\ &= V_\mathrm{f}(x) - \ell(x,u) \end{aligned}$$

Thus, from the above it can be seen that $V_f(x^+) - V_\mathrm{f}(x) = -\ell(x,u)$ and therefore the assumption is satisfied.

F. Assumption 2.23(b)

This assumption states that there exists a lower and upper bound on the stage and terminal costs. This is given by,

$$\ell(x,u) \geq \alpha_1(x,u) \quad \forall x \in \mathcal{X}_N, \forall u \in \mathbb{U}$$

$$V_f(x) \leq \alpha_2(|x|) \quad \forall x \in \mathbb{X}_f$$

where, α_1 and α_2 are κ_{inf} functions.

In this case, an LQR regulator has been chosen which is as defined above. If we first consider the stage cost, then

$$l\ell(x,u) = \frac{1}{2}\left(x^\top Qx + u^\top Ru\right) \geq \frac{1}{2}x^\top Qx \geq \frac{1}{2}\lambda_{\min}(Q)|x|^2 \tag{35}$$

In the above equations, $\lambda_{\min}$ is the minimum eigenvalue. Also the function $|x|^2$ is κ_{inf} as it is equal to 0 when $x = 0$ and as $|x|^2 \to \inf$ as $x \to \inf$. Further, the terminal cost is given by the following equation,

$$V_\mathrm{f}(x) = \frac{1}{2}x^\top Px \leq \frac{1}{2}\lambda_{\max}(P)|x|^2 \tag{36}$$

Therefore, from the above 2 inequalities we see that assumption 2.23(b) is satisfied.

Thus, with the above assumptions verified, we can say that Theorem 2.24 holds true for the closed loop system and the origin is asymptotically stable in χ_N.

V. Numerical simulations

In this section, simulations are carried out using the YALMIP optimisation problem solver in MATLAB 2019b along with interfacing done to a model developed on Simulink and another open-source tool called CASADI.

First, we will research the influence of the prediction horizon and the cost function weightings Q, R and P. Second, we will simulate the linearized six degrees of freedom model along with the MPC algorithm, which optimizes the actuator effort based on the constraints. The control inputs are saved and applied to the nonlinear dynamic model. For all of the simulation results, the initial equilibrium condition was that of a hover. Three different reference scenarios are generated to evaluate the effectiveness of the control laws: (a) hover stability, (b) circular trajectory tracking and (c) helical trajectory tracking. As a final step, a PID controller is implemented on the nonlinear dynamic model, to be compared with the MPC control scheme. The values of the state and input constraints have been given in Table I.

Term	Range
$[\phi_{\min}, \phi_{\max}]$	$[-\pi/4, \pi/4]$
$[\theta_{\min}, \theta_{\max}]$	$[-\pi/4, \pi/4]$
$[\psi_{\min}, \psi_{\max}]$	$[-\pi/4, \pi/4]$
$[p_{\min}, p_{\max}]$	[-0.4, 0.4]
$[q_{\min}, q_{\max}]$	[-0.4, 0.4]
$[r_{\min}, r_{\max}]$	[-0.4, 0.4]
$[v_{x\min}, v_{x\max}]$	[-1, 1]
$[v_{y\min}, v_{y\max}]$	[-1, 1]
$[v_{z\min}, v_{z\max}]$	[-1, 1]
$[x_{\min}, x_{\max}]$	[-5, 5]
$[y_{\min}, y_{\max}]$	[-5, 5]
$[z_{\min}, z_{\max}]$	[-5, 5]
$[f_{t\min}, f_{t\max}]$	[-10, 10]
$[\tau_{x\min}, \tau_{x\max}]$	[-1.625, 1.625]
$[\tau_{y\min}, \tau_{y\max}]$	[-1.625, 1.625]
$[\tau_{z\max}, \tau_{z\max}]$	[-1.625, 1.625]

TABLE I
Input and State Cconstraints

A. Prediction Horizon N

Prediction horizon N will influence the stability, feasibility and invariance. With small N, the solution may be infeasible since the constraints on the states and input could no longer be satisfied. By increasing N, we could get a feasible solution but the settling time will exceed the expectation. A comparison was done for different N values and the positional RMS error is shown in table II.

As shown in Table II, both $N = 3$ and $N = 5$ show a relatively large error. For $N = 10$ and $N = 20$, there are minimal differences in the performance, but since larger N also ensures a larger region of attraction, we finalized N to 20.

B. Cost Function Weightings

Q and R matrices are relevant to the cost computation. To choose these values, we test five different settings to find the

		N=3	N=5	N=10	N=20
Hovering	X	5.89e-02	6.25e-04	1.65e-07	1.64e-07
	Y	4.90e-05	6.88e-07	2.00e-09	2.04e-09
	Z	1.2873	0.9831	0.7857	0.7936
Circular tracking	X	0.7693	0.4197	0.2492	0.2583
	Y	0.6983	0.3823	0.1757	0.1692
	Z	6.87e-02	2.47e-03	6.70e-04	6.92e-04
Helical tracking	X	0.7453	0.4962	0.3397	0.3571
	Y	0.8742	0.5639	0.3854	0.3702
	Z	0.8236	0.5365	0.3716	0.3635

TABLE II
RMS error of position tracking errors

optimum value. For testing stage state cost Q, we tried $Q1 = 0.001 \times I$, $Q2 = 0.01 \times I$, $Q3 = 0.1 \times I$ $Q4 = I$ and $Q5 = 10 \times I$. The R matrix was kept at constant I to have a fair comparison. The results can be observed in Fig. 3. The states are the quadrotor position x, position z and pitch angle ψ.

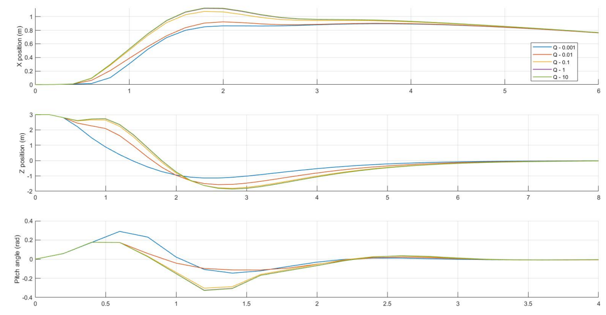


Fig. 3. State trajectories for different state cost matrices Q

It can be learned from Fig. 3 that state-weighting matrix Q has a relatively significant influence on the result. Small state costs tend to cause longer settling times and more overshoot as opposed to larger costs.

Similarly, we also tested $R1 = 0.001 \times I$, $R2 = 0.01 \times I$, $R3 = 0.1 \times I$ $R4 = I$ and $R5 = 10 \times I$ and kept the Q matrix equal to constant value I. The results can be observed in Fig. 4.

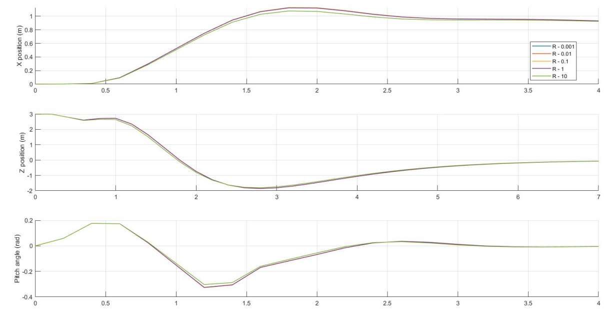


Fig. 4. State trajectories for different input cost matrices R

It can be seen from Fig. 4 that the input-weighting matrix R only has a small influence on the results. Fig. 5 also indicates

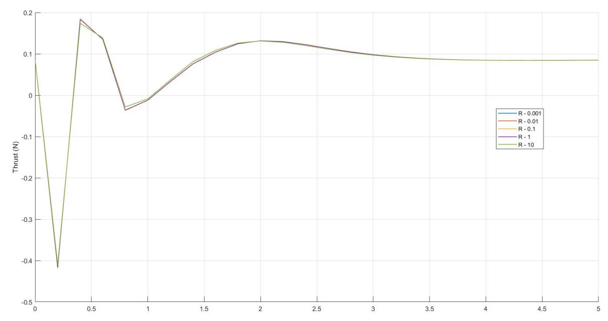


Fig. 5. Input sequences for one of the rotors for different input cost R matrices.

that the input sequences are nearly constant for different values of R. This could be explained by the fact that the constraints we set for the input is more than sufficient for the quadrotor movement. Finally, with the results from the figures, we take $Q4$ and $R4$ as our final value in all our application testings.

C. Hover Stability

In this problem, we give an unstable initial state and different reference locations for the quadrotor and check its capability to maintain the hover state. In the simulation, the initial roll angle is given as $40\,\mathrm{deg}$ while the reference in hover is $0\,\mathrm{deg}$. Also, the reference z axis position is changed to 4 meters while the initial position is 0 meter. Fig. 6 shows the quadrotor states position x, z and roll angle ϕ stabilizing over time.

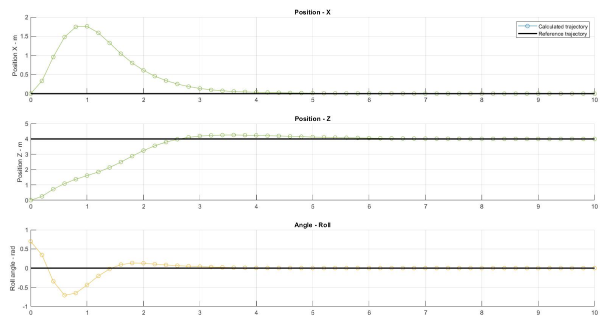


Fig. 6. States for hover stability

D. Trajectory Tracking

First, a circular trajectory was generated which was given as the reference for the controller. Fig. 7 shows the quadrotor states changing over the given circular reference. Fig. 8 shows the circular trajectory tracking.

We also simulate the controller performance when conducting helical trajectory tracking. Fig. 9 shows the quadrotor states changing over the given helical reference. Fig. 10 shows the helical trajectory tracking.

It can be seen that the controller achieves good tracking performance in both scenarios, despite a minor delaying in direction x and y.

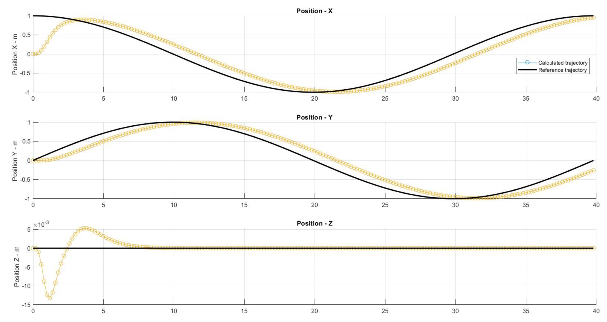


Fig. 7. States for circular trajectory tracking

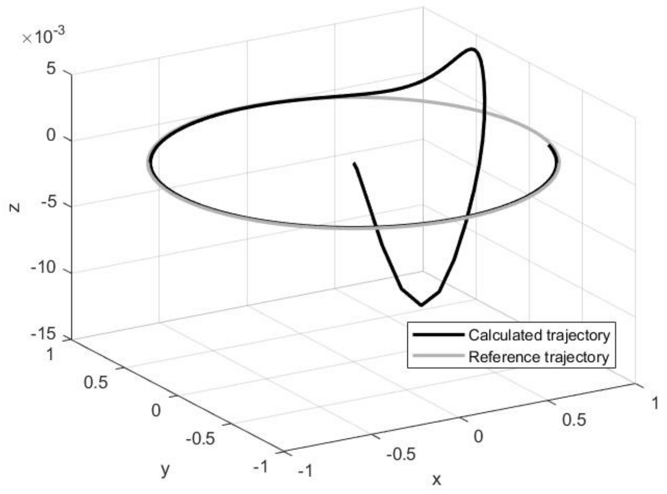


Fig. 8. Circular trajectory tracking

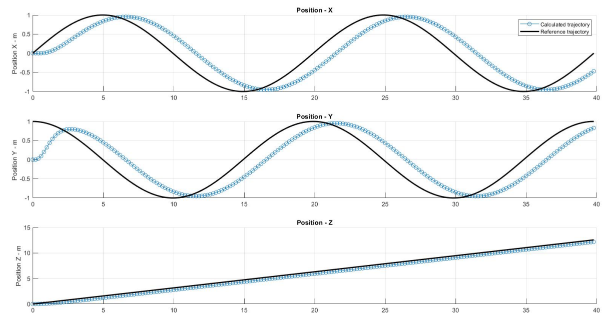


Fig. 9. States for helical trajectory tracking

E. Comparison with PID controller

The same trajectory tracking is performed using a PID controller. The designed control scheme is based on [4]. Here we use three PIDs: position, angle and motor. The equation of the PID control input is shown as follow:

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau)d\tau + K_D \frac{de(t)}{dt} \tag{37}$$

where e is the tracking error. K_P , K_I and K_D are proportional(P), integral (I), and derivative (D) gains respectively.

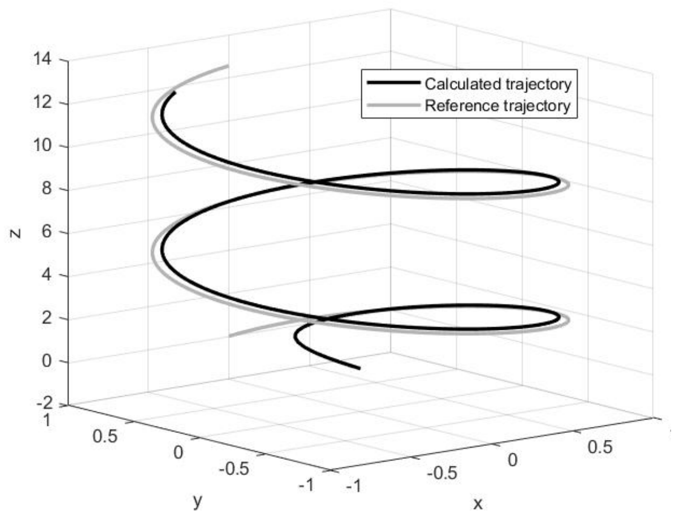


Fig. 10. Helical trajectory tracking

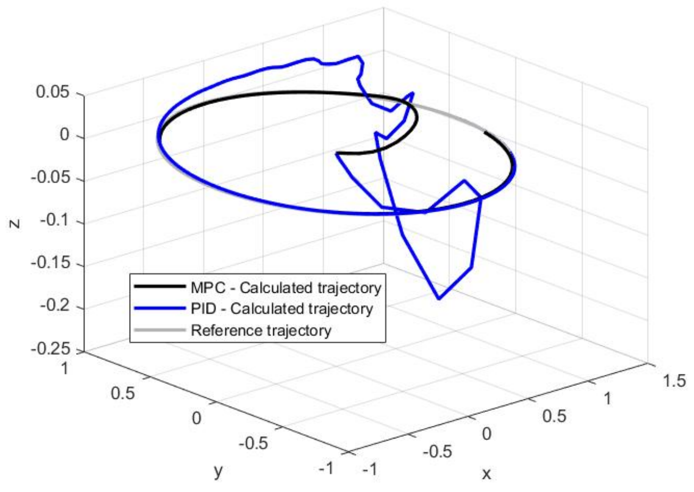


Fig. 12. Trajectory tracking comparison

Controller	RMSE$_x$	RMSE$_y$	RMSE$_z$
PID	0.7089	0.7053	0.0357
MPC	0.2465	0.1757	0.0022

TABLE IV
RMSE VALUE COMPARISON

The simulation is done by manually tuning of PIDs. The values of control gain obtained for the control loop are given in Table III. The comparison results is shown in Fig. 11 and Fig. 12.

	K_P	K_I	K_D
Position	15	3	3
Angle	5	0	10
Motor	1	10	0

TABLE III
PID CONTROLLER GAINS

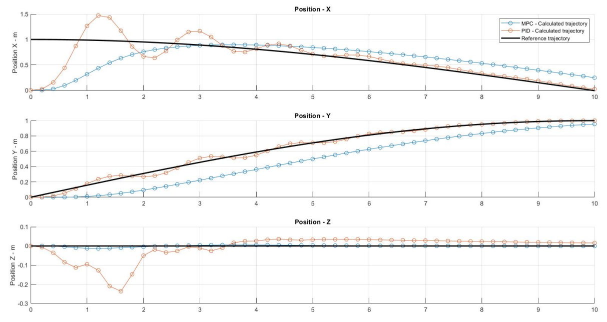


Fig. 11. Comparison results

It can be seen from the figures that the performance of linear MPC and PID is compared. By comparison we can see that MPC has better control characteristics despite a longer settling time. The PID has a sustained oscillation at the beginning of the simulation, and a constant deviation in z direction. Furthermore, to guarantee a feasible and stable result, the PID gain has to be small. If K_P is increased, the oscillations will become more apparent as a result of the time delay and the system will become unstable if K_I and K_D are increased. A quantitative comparison can also be made by observing that the Root mean square error values in table IV are much lower in the case of an MPC controller as compared to the PID controller. Thus, due to the above observations, a linear MPC maybe preferred over a PID controller.

VI. CONCLUSION

From the simulations, we can conclude that an asymptotically stable MPC controller was successfully designed that was able to result in hover stabilisation, circular trajectory and helical trajectory tracking of a quadrotor. We also found that the MPC controller has better trajectory tracking when compared to a PID controller.

REFERENCES

[1] F. Sabatino, “Quadrotor control: modeling, nonlinearcontrol design, and simulation,” 2015.
[2] J. Rawlings and D. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2019.
[3] “Exercise set 4: Mpc, (sc42125).”
[4] T. Bresciani, “Modelling, identification and control of a quadrotor helicopter,” *MSc Theses*, 2008.