MTP PROGRESS SEMINAR REPORT

on

MPC Controller Tuning for least energy quadrotor missions

Submitted by

Arya Mallick
(Roll No. 234102501)

Under the guidance of
Dr. Chayan Bhawal

DEPARTMENT OF ELECTRONICS AND ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

November 2024

Abstract

Unmanned Aerial Vehicles (UAVs), particularly drones, are increasingly becoming part of various sectors in society. In this work we focus on tuning a MATLAB-based controllers to enable a UAV to follow predetermined least energy trajectory while performing a mission. The solution of the optimal control problem(OCP) is generated by GPOPS-II. We then attempt to tune the Model Predictive Control (MPC) to track our desired optimal path.

List of Figures

Contents

Chapter 1

Introduction

Drone technology has found application in various sectors of industries. So for widescale usage energy efficieny and management in drones crucial.
Energy-efficient paths are crucial for drones as they extend flight duration, reduce battery consumption, and enable longer missions or greater payload capacity, which is essential for applications like delivery, surveillance, and search-and-rescue.

A controller ensures the drone accurately follows an energy-efficient path by regulating its motors and actuators to minimize energy usage while maintaining stability and trajectory accuracy. It optimally adjusts thrust, pitch, and yaw to reduce unnecessary maneuvers and ensure smooth transitions, aligning with the planned low-energy trajectory.

In this work we use Model Predictive Control (MPC) on a quadcopter model, which offers better predictive capabilities than convential PID and handles sharp turns with reduced tracking error.We try to tune the MPC controller paramters for best tracking performance.

Chapter 2

Literature Survey

In this chapter,we review the prior works done regarding MPC and energy optimal path planning. MPC has emerged as a promising control strategy for drones, due to its ability to handle multivariable systems and constraints effectively.

2.1 Trajectory Tracking and Path Planning

In [MBO+19], an MPC motion controller is developed based on a linearized model for waypoint tracking. Similarly, [LHC+19] employs MPC for flock control of quadrotors, incorporating collision avoidance strategies. In [IOI17], a linear MPC framework is implemented for quadcopter navigation, enhancing trajectory precision through real-time control refinements. Building on this [EDA+23] utilized a nonlinear MPC approach using CASADI to account for the UAV's complex dynamics, allowing for precise control over 3D paths. The study presented in [WPS+21] explores an efficient nonlinear model predictive control (NMPC) approach for trajectory tracking of a quadrotor unmanned aerial vehicle (UAV), achieved by integrating the desired trajectory into a reference dynamical system.

2.2 Energy Efficiency

With increase in flight durations, energy efficiency in UAV is becoming more vital.In this paper [JŻGB22], the energy efficiency of a quadrotor's dynamic model is analyzed. The study proposes a method for designing the dynamic model to estimate energy consumption accurately.
This paper[RTBP24] proposed an energy-aware MPC scheme designed to optimize power usage while maintaining accurate trajectory tracking. For energy optimal path-planning , the mission is considered for a fixed final time [MCL16], but the fixed final time may or may not be the optimal time for mission completition. So, the optimal time for mission completition need to be considered.

Chapter 3

Motivation and Problem Formulation

3.1 Motivation

While MPC has demonstrated significant effectiveness in various UAV applications, but achieving an optimal balance between computational efficiency and control accuracy remains a persistent challenge, particularly for nonlinear UAV systems.
In this work , we aim to address these gaps by developing a computationally efficient MPC strategy tailored to nonlinear UAV systems through an LPV (linear parameter-varying) MPC approach.

3.2 Problem Statement

An energy-optimal path between two points in 3D space is generated offline using the optimal control software GPOPS-II, which provides the advantage of handling complex trajectory optimization tasks. The primary objective is to develop a control strategy that enables the UAV to track this energy-efficient path with minimal time deviation while maintaining low energy consumption.

Chapter 4

Work Done So Far

4.1 Quadcopter Dynamics

Let us define the mathematical model of the quadcopter used. Refer [TF18].
The 4 control inputs of the drone are:

$$
\begin{aligned}
U_1 &= m\ddot{z} \\
U_2 &= I_x\ddot{\phi} \\
U_3 &= I_y\ddot{\theta} \\
U_4 &= I_z\ddot{\psi}
\end{aligned}
$$

Thrust U1 [N] (Throttle) MomentU2[Nm](Roll) MomentU3[Nm](Pitch) MomentU4[Nm](Yaw)

$$
\begin{aligned}
U_1 &= c_T \cdot \left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \\
U_2 &= c_T \cdot l \cdot \left(\Omega_4^2 - \Omega_2^2\right) \\
U_3 &= c_T \cdot l \cdot \left(\Omega_3^2 - \Omega_1^2\right) \\
U_4 &= c_Q \cdot \left(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2\right) \\
\Omega_{\text{total}} &= -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4
\end{aligned}
$$

The angular velocities of the rotors are denoted as Ω_1, Ω_2, Ω_3, and Ω_4 [rad/s] for the motors 1, 2, 3, and 4, respectively. cT [Ns2] and cQ [Nms2] are aerodynamic coefficients of thrust and drag.
A quadcopter has six degrees of freedom (DOF): three for position and three for attitude. The variables u, v, and w represent the velocities along the x, y, and z axes in the body frame (B-frame) [m/s], respectively. Similarly, p, q, and r denote the angular velocities about the roll (ϕ), pitch (θ), and yaw (ψ) axes in the B-frame [rad/s]. .Ix,Iy, Iz and Jtp [Nms2] are the moment of inertia around the x,y,z and propeller axis .The equations of motion in B-frame is:

$$\dot{u} = (vr - wq) + g\sin\theta$$

$$\dot{v} = (wp - ur) - g\cos\theta\sin\phi$$

$$\dot{w} = (uq - vp) - g\cos\theta\cos\phi + \frac{U_1}{m}$$

$$\dot{p} = qr\frac{I_y - I_z}{I_x} - \frac{J_{TP}}{I_x}q\Omega + \frac{U_2}{I_x}$$
$$\dot{q} = pr\frac{I_z - I_x}{I_y} + \frac{J_{TP}}{I_y}p\Omega + \frac{U_3}{I_y}$$
$$\dot{r} = pq\frac{I_x - I_y}{I_z} + \frac{U_4}{I_z}$$

let x,y and z be the translational position and ϕ, θ, and ψ be the angluar position in E-frame. Now we relate the translational and angular component in E-frame to B-frame.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

where the matrices R and T are:

$$R = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix}$$

$$T = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix}$$

T tends to the identity matrix when ϕ and θ are small, which holds true for non-acrobatic drones.

The parameters of the AscTec Hummingbird drone used are:

Parameter	**Value**
Mass (m)	$0.698\,\mathrm{kg}$
Moments of inertia	$I_x = I_y = 0.0034\,\mathrm{kg \cdot m^2}$, $I_z = 0.006\,\mathrm{kg \cdot m^2}$
propeller moment of inertia (J_{tp})	$1.302 \times 10^{-6}\,\mathrm{N \cdot s^2}$
Thrust coefficient (C_t)	$7.6184 \times 10^{-8} \times \left(\frac{60}{2\pi}\right)^2\,\mathrm{N \cdot s^2}$
Torque coefficient (C_q)	$2.6839 \times 10^{-9} \times \left(\frac{60}{2\pi}\right)^2\,\mathrm{N \cdot m^2}$
Arm length (l)	$0.171\,\mathrm{m}$

4.2 Optimal Control Problem

The electrical model of a brushless DC motor is shown:

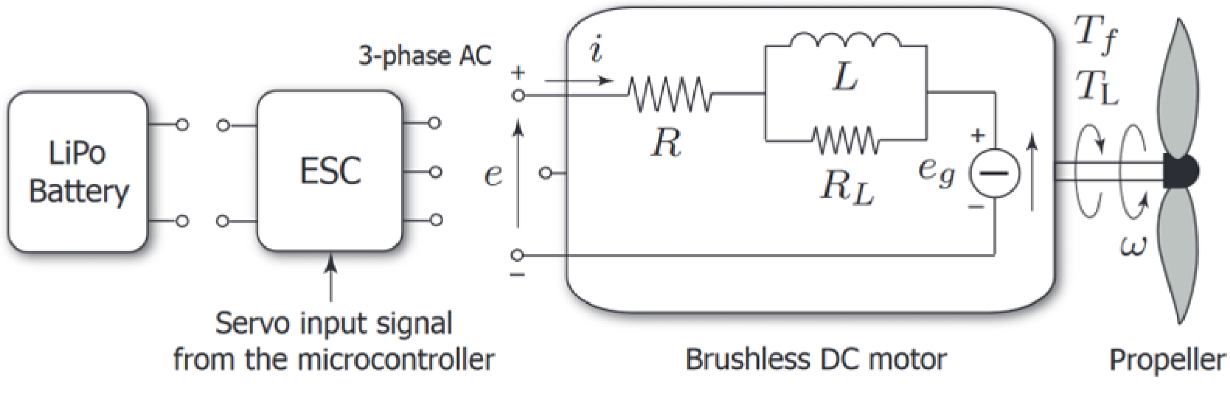

Figure 4.1: BLDC Motor equivalent model

we write the electric propulsion dynamics of the motors (actuators). In time [t0,tf], the total energy consu

$$E_M = \int_{t_0}^{t_f} \sum_{k=1}^{4} e_k(t) i_k(t) dt \tag{4.1}$$

where $e_k(t)$ and $i_k(t)$ are the instantaneous voltage across and current drawn by each motor respectively.

$$e(t) = Ri(t) + K_E \Omega(t) + L \frac{di(t)}{dt} \tag{4.2}$$

$$i(t) = \frac{1}{K_T} \left[T_f + T_L(\Omega(t)) + D_f \Omega(t) + J \frac{d\Omega(t)}{dt} \right] \tag{4.3}$$

where:

$$\begin{aligned}
R &: \text{phase winding resistance } [\Omega] \\
L &: \text{phase winding inductance } [H] \\
K_E &: \text{motor voltage constant } [Vs/\text{rad}] \\
\Omega(t) &: \text{angular velocity of the motor shaft [rad/s]} \\
K_T &: \text{torque constant of the motor [Nm/A]} \quad (\text{Note: } K_T = K_E) \\
K_V &: \text{velocity constant of the motor [rpm/V]} \\
T_f &: \text{motor friction torque [Nm]} \\
T_L(\Omega(t)) &: \text{load friction torque [Nm]} \quad (\text{assumed } T_L(\Omega(t)) = k_{t0} \Omega(t)^2) \\
D_f &: \text{viscous damping coefficient of the motor [Nms/rad]} \\
J &: \text{moment of inertia [kgm}^2] \quad (J = J_m + J_p) \\
J_m &: \text{moment of inertia of the motor} \\
J_p &: \text{moment of inertia of the propeller}
\end{aligned}$$

Assumptions: *1)The quadrotor mission starts from a hovering state and ends in a hovering state, i.e.,* $\Omega(t_0) = \Omega(t_f)$
2)The efficiency of the motor is 1 (ideal).
3)The onboard devices ,controller, ESC and gearbox uses negligible energy.

Using Eq. (4.2), Eq. (4.3), and Assumption 1, we rewrite Eq. (4.1) as

$$E_M = \int_{t_0}^{t_f} \sum_{i=1}^{4} \left(c_1 + c_2 \Omega_i(t) + c_3 \Omega_i^2(t) + c_4 \Omega_i^3(t) + c_5 \Omega_i^4(t) + c_6 \dot{\Omega}_i^2(t) \right) dt \tag{4.4}$$

where,

$$c_1 = \frac{R T_f^2}{K_T^2}, c_2 = \frac{T_f}{K_T} \left(\frac{2 R D_f}{K_T} + K_E \right), c_3 = \frac{D_f}{K_T} \left(\frac{R D_f}{K_T} + K_E \right) + \frac{2 R T_f k_{t0}}{K_T^2},$$
$$c_4 = \frac{k_{t0}}{K_T} \left(\frac{2 R D_f}{K_T} + K_E \right), c_5 = \frac{R k_{t0}^2}{K_T^2}, c_6 = \frac{R J^2}{K_T^2},$$

$c_1, c_2, \ldots, c_6$ are constants characteristic of the motor-propeller pair.
For a given mission $\mathcal{M}(x_i, y_i, z_k)$, E_M denotes the total energy consumed by the quadrotor to

traverse a trajectory $\mathcal{T}(x_i(t), y_j(t), z_k(t))$.The OCP being solved here is of the form:
Objective:

$$E_M^* = \min_{u(t),t_f} E_M = \min_{u(t),t_f} \int_{t_0}^{t_f} \left[\sum_{i=13}^{16} \left(c_1 + c_2 x_i(t) + c_3 x_i^2(t) + c_4 x_i^3(t) + c_5 x_i^4(t) \right) + c_6 \sum_{k=1}^{4} \alpha_k^2(t) \right] dt \tag{4.5}$$

System Constraints:

$$\dot{x}(t) = f(x(t)) + Bu(t) \tag{4.6}$$

State Constraints:

$$x_{i,\min} < x_i < x_{i,\max} \tag{4.7}$$

Input Constraints:

$$\alpha_{k,\min} < \alpha_k < \alpha_{k,\max} \tag{4.8}$$

Boundary Conditions:

$$x_i(t_0) = x_{i,t_0}, \quad x_i(t_f) = x_{i,t_f} \tag{4.9}$$

$i \in \{1, 16\}$, $k \in \{1, 4\}$. where

$$\mathbf{X}_{\text{vector}}^\top = \begin{bmatrix} x & \dot{x} & y & \dot{y} & z & \dot{z} & \phi & \dot{\phi} & \theta & \dot{\theta} & \psi & \dot{\psi} & \Omega_1 & \Omega_2 & \Omega_3 & \Omega_4 \end{bmatrix}$$

$$\mathbf{u}_{\text{vector}}^\top = \begin{bmatrix} \dot{\Omega}_1, & \dot{\Omega}_2, & \dot{\Omega}_3, & \dot{\Omega}_4 \end{bmatrix} = \begin{bmatrix} \alpha_1, & \alpha_2, & \alpha_3, & \alpha_4 \end{bmatrix}$$

$f(x) \in \mathbb{R}^{16\times1}$ is the drone dynamics,$B \in \mathbb{R}^{16\times4}$ is the control input matrix

$$f(x) = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ \frac{k_{lh}}{m}(c_{x_7}c_{x_9}s_{x_8} + s_{x_7}s_{x_9}) \sum_{i=13}^{i=16} x_i^2 \\ \frac{k_{lh}}{m}(c_{x_7}s_{x_8}s_{x_9} - c_{x_9}s_{x_7}) \sum_{i=13}^{i=16} x_i^2 \\ -g + \frac{k_{lh}}{m}(c_{x_8}c_{x_7}) \sum_{i=13}^{i=16} x_i^2 \\ x_{10} \\ x_{11} \\ x_{12} \\ \frac{I_y - I_z}{I_x} x_{11} x_{12} - \frac{J}{I_x} x_{11} \Omega_5 + \frac{l\kappa_b}{I_x}(x_{14}^2 - x_{16}^2) \\ \frac{I_z - I_x}{I_y} x_{10} x_{12} + \frac{J}{I_y} x_{10} \Omega_5 + \frac{l\kappa_b}{I_y}(x_{15}^2 - x_{13}^2) \\ \frac{I_x - I_y}{I_z} x_{10} x_{11} + \frac{\kappa_t 0}{I_z}(x_{13}^2 + x_{15}^2 - x_{14}^2 - x_{16}^2) \end{bmatrix} \qquad B = \begin{bmatrix} \mathbf{0}_{12\times4} \\ I_{4\times4} \end{bmatrix}$$

Solving it we get $u^*(t)$ and t_f^* to reach $x_{i,f}$ with the least energy E_M^*. Using $u^*(t)$ and t_f^*, we compute the optimal state $x^*(t)$ and the trajectory $\mathcal{T}(x_i^*(t), y_j^*(t), z_k^*(t))$ for the mission $\mathcal{M}(x_i, y_j, z_k)$.

4.3 Controller Model

The control architecture consists of two interconnected controllers: an outer position controller and an inner attitude controller. The position controller is responsible for managing the position variables x, y, and z. Using state feedback linearization, it computes the thrust input U_1 and the desired roll (ϕ) and pitch (θ) angles. These outputs ensure that the UAV can track the target position (x, y, z). The computed thrust U_1 is directly applied to the open-loop system, while the desired angles ϕ and θ are provided as reference inputs to the attitude controller. This inner controller, based on an LPV-MPC (Linear Parameter Varying Model Predictive Control) framework, calculates the remaining control inputs: U_2, U_3, and U_4. These inputs control the UAV's roll, pitch, and yaw dynamics, enabling precise and stable trajectory tracking.

4.3.1 General constrained MPC:

$$J = \min_{e_t,u_t} \left\{ \frac{1}{2} \sum_{k=0}^{N-1} \left(e_t^T Q e_{t+k} + u_t^T R u_{t+k} \right) + \frac{1}{2} e_t^T Q e_{t+N} \right\}, \tag{4.10}$$

where $e_k = r_k - y_k = r_k - Cx_k and u_k = u_{k-1} + \Delta u_k$.
We augment the system by including the previous input from one sample of past as an additional state.

$$x_{\tilde{k+1}} = \begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix} + \begin{bmatrix} B \\ I \end{bmatrix} \Delta u_k = \tilde{A}\tilde{x}_k + \tilde{B}\Delta u_k$$

$$y_k = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix} = \tilde{C} \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix}$$

Here for our drone A and B are in LPV(linear parameter varying) format where A and B are goverened from the following state space equations:

$$\begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Omega\frac{J_r P}{I_x} & 0 & \frac{I_y - I_z}{I_x} \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \Omega\frac{J_r P}{I_y} & 0 & 0 & 0 & \frac{I_z - I_x}{I_y} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & \frac{\dot{\theta}}{2}\frac{I_x - I_y}{I_z} & 0 & \frac{\dot{\phi}}{2}\frac{I_x - I_y}{I_z} & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{I_z} \end{bmatrix} \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} \tag{4.11}$$

We calculate at each time step prediction for the entire horizon period(N). It can be compactly represented as:

$$\tilde{x} = \begin{bmatrix} \tilde{B} & & & & \\ \tilde{A}\tilde{B} & \tilde{B} & & & \\ \tilde{A}^2\tilde{B} & \tilde{A}\tilde{B} & \tilde{B} & & \\ \vdots & \vdots & \ddots & \ddots & \\ \tilde{A}^{N-1}\tilde{B} & \tilde{A}^{N-2}\tilde{B} & \cdots & \tilde{A}\tilde{B} & \tilde{B} \end{bmatrix} \Delta u + \begin{bmatrix} \tilde{A} \\ \tilde{A}^2 \\ \vdots \\ \tilde{A}^N \end{bmatrix} \tilde{x}_t = \overline{\overline{C}}\Delta u + \hat{A}\tilde{x}_t. \tag{4.12}$$

$$J = \frac{1}{2}\left((\overline{\overline{C}}\Delta u + \hat{A}x_t)^T \overline{\overline{Q}} (\overline{\overline{C}}\Delta u + \hat{A}x_t) + \frac{1}{2}\Delta u^T \overline{\overline{R}} \Delta u - r^T \overline{\overline{T}} (\overline{\overline{C}}\Delta u + \hat{A}x_t) \right), \text{ where} \tag{4.13}$$

$$\overline{\overline{Q}} = \begin{bmatrix} \tilde{C}^T Q\tilde{C} & 0 & \cdots & 0 \\ 0 & \tilde{C}^T Q\tilde{C} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \tilde{C}^T S\tilde{C} \end{bmatrix}, \overline{\overline{T}} = \begin{bmatrix} Q\tilde{C} & 0 & \cdots & 0 \\ 0 & Q\tilde{C} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & S\tilde{C} \end{bmatrix}, \quad \overline{\overline{R}} = \begin{bmatrix} R & 0 & \cdots & 0 \\ 0 & R & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R \end{bmatrix}$$

Finally, J becomes

$$J = \frac{1}{2}\Delta u^T \overline{\overline{H}} \Delta u + f^T \Delta u, \text{ where } \overline{\overline{H}} = \left(\overline{\overline{C}}^T \overline{\overline{QC}} + \overline{\overline{R}}\right),$$

$$f^T = \begin{bmatrix} \tilde{x}_t^T & r^T \end{bmatrix} \begin{bmatrix} \hat{A}^T \overline{\overline{QC}} \\ -\overline{\overline{TC}} \end{bmatrix} \tag{4.14}$$

We then solve the optimization problem within interval $T_s = 0.1$ sec using MATLAB's QUADPROG. Also, we subject the UAV to some constraints on $\Omega_{\max} = 110 \cdot \frac{\pi}{3}$ and $\Omega_{\min} = 0$.
By substituting them, we get constraints on rotor forces and torques as $\mathtt{min}, \mathtt{max}(U_1, U_2, U_3, U_4)$.

4.3.2 Feedback linearization as position controller:

$$\ddot{x} = (\cos\theta \sin\phi + \sin\theta \sin\psi)\frac{U_1}{m} \tag{4.15}$$

$$\ddot{y} = (\cos\theta \sin\phi \sin\psi + \cos\theta \cos\psi)\frac{U_1}{m} \tag{4.16}$$

$$\ddot{z} = -g + \cos\theta \cos\phi \frac{U_1}{m} \tag{4.17}$$

We define:

$$\begin{aligned} &e_x = xR_t - x_t, \dot{e_x} = \dot{x}R_t - \dot{x}_t, \ddot{e_x} = -\ddot{x}_t = v_x, e_y = yR_t - y_t \\ &\dot{e_y} = \dot{y}R_t - \dot{y}_t, \ddot{e_y} = -\ddot{y}_t = v_y, e_z = zR_t - z_t \\ &\dot{e_z} = \dot{z}R_t - \dot{z}_t \ddot{e_z} = -\ddot{z}_t = v_z \end{aligned}$$

The variables vx, vy and vz are selected as control inputs for the linearized state feedback control strategy and are defined as:

$$\begin{aligned} v_x &= -k_{1_x} e_x - k_{2_x} e_x \\ v_y &= -k_{1_y} e_y - k_{2_y} e_y \\ v_z &= -k_{1_z} e_z - k_{2_z} e_z \end{aligned}$$

We choose real and negative poles, such that the constants in these equations can be computed as the error tends to zero as time tends to infinity. Thus, by setting $v_x = 0$, $v_y = 0$, and $v_z = 0$ and solving the three equations, we obtain the references θ, ϕ, and U_1.

$$\begin{aligned} \theta &= \tan^{-1}(ac + bd) \\ \phi &= \tan^{-1}((ad - bc)/\sqrt{1 + (ac + bd)^2}) \\ U_1 &= \frac{(v_z + g)m}{\cos(\phi)\cos(\theta)} \\ a &= v_x/(v_z + g), \quad b = v_y/(v_z + g), \quad c = \cos(\psi), \quad d = \sin(\psi) \end{aligned}$$

The feedback linearization controller calculates the thrust input U_1, which is directly applied to the nonlinear UAV model to control altitude and overall propulsion. Simultaneously, the desired roll (ϕ) and pitch (θ) angles are generated as reference signals and passed to the MPC-based attitude controller. This approach ensures coordination between the outer and inner control loops for precise trajectory tracking.

4.3.3 Combined Controller Structure

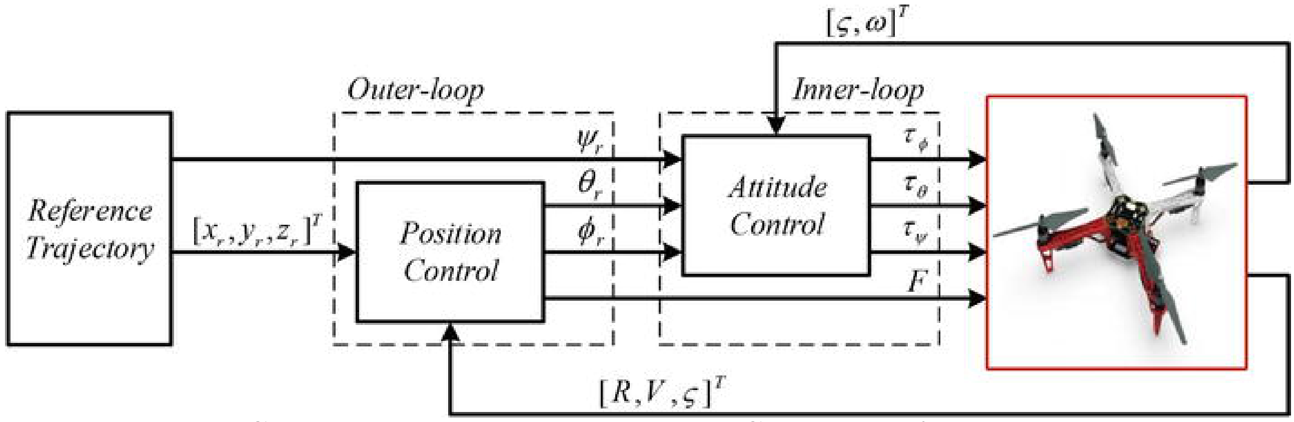

Figure 4.2: Cascade control with inner loop-MPC and outer-feedback linearization

Chapter 5

Results And Discussion

These are the best possible performance results achieved by the controller for reaching the point (X,Y,Z)=(10,25,15) starting from(0,0,0) hile respecting the quadcopter's actuator constraints.

The minimum time required for mission completion is 7.2 seconds. In comparison, the energy-optimal path generated by GPOPS-II can be traversed in approximately 5.42 seconds.

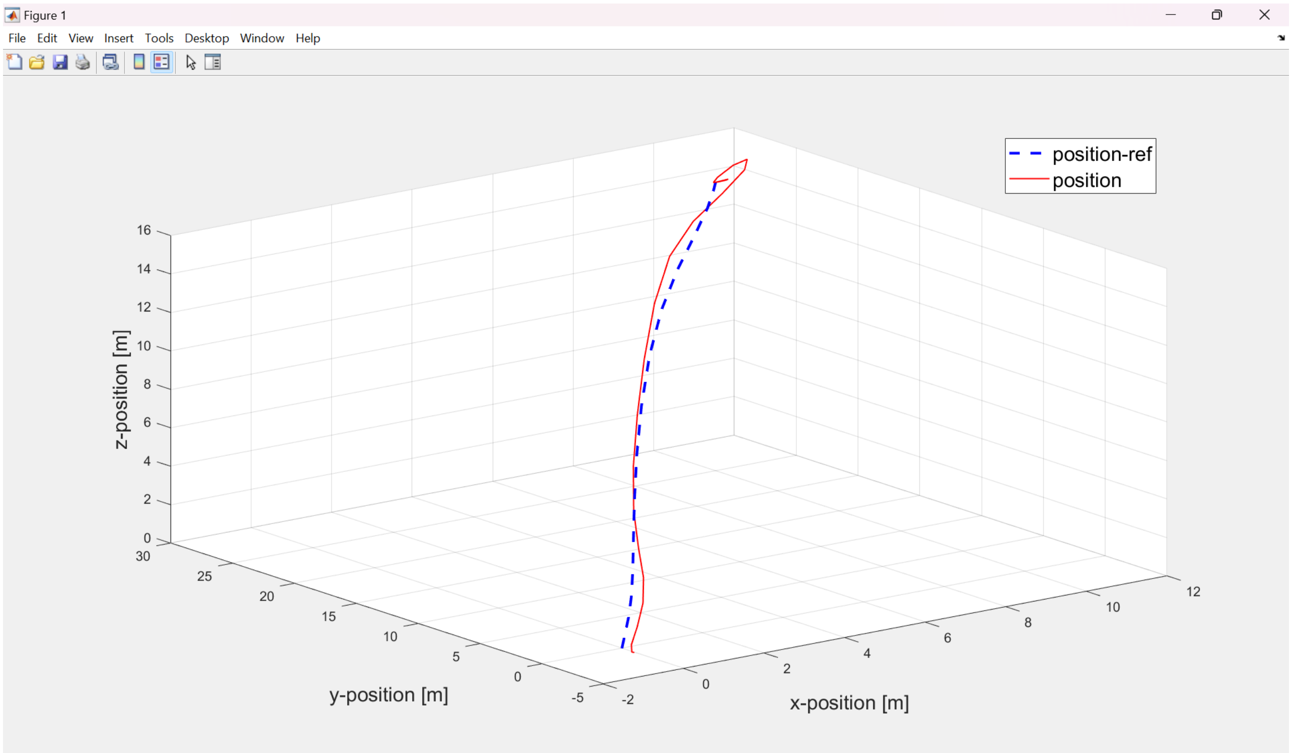

Figure 5.1: Mission completion in minimum of 7.2 seconds

There is some initial difference at starting,and also some before it hovers at final position.

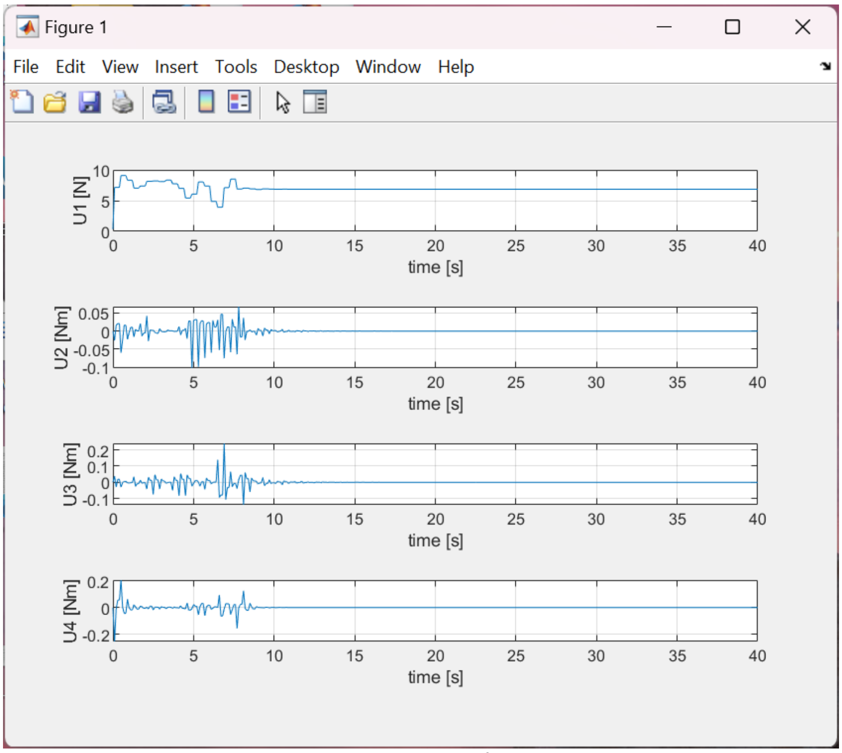

Figure 5.2: Thrust, roll, pitch forces and yaw torque

We observe rarely any input (force and torques) saturation occurs. All are within feasible bounds.

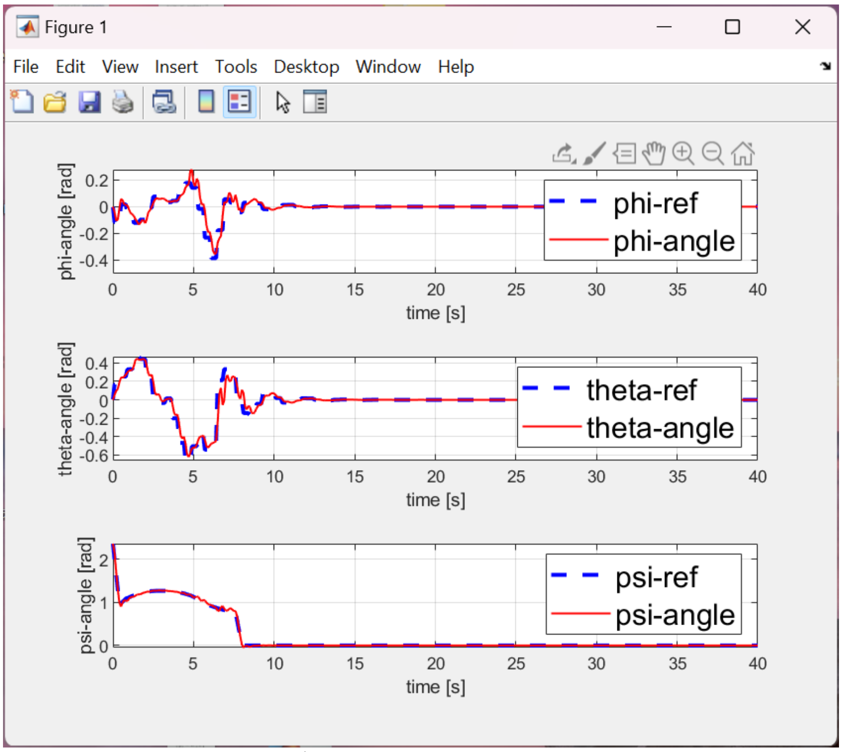

Figure 5.3: Attitude tracking performance

We can say the attitude controller(MPC) gives satisfactory accuracy in tracking.

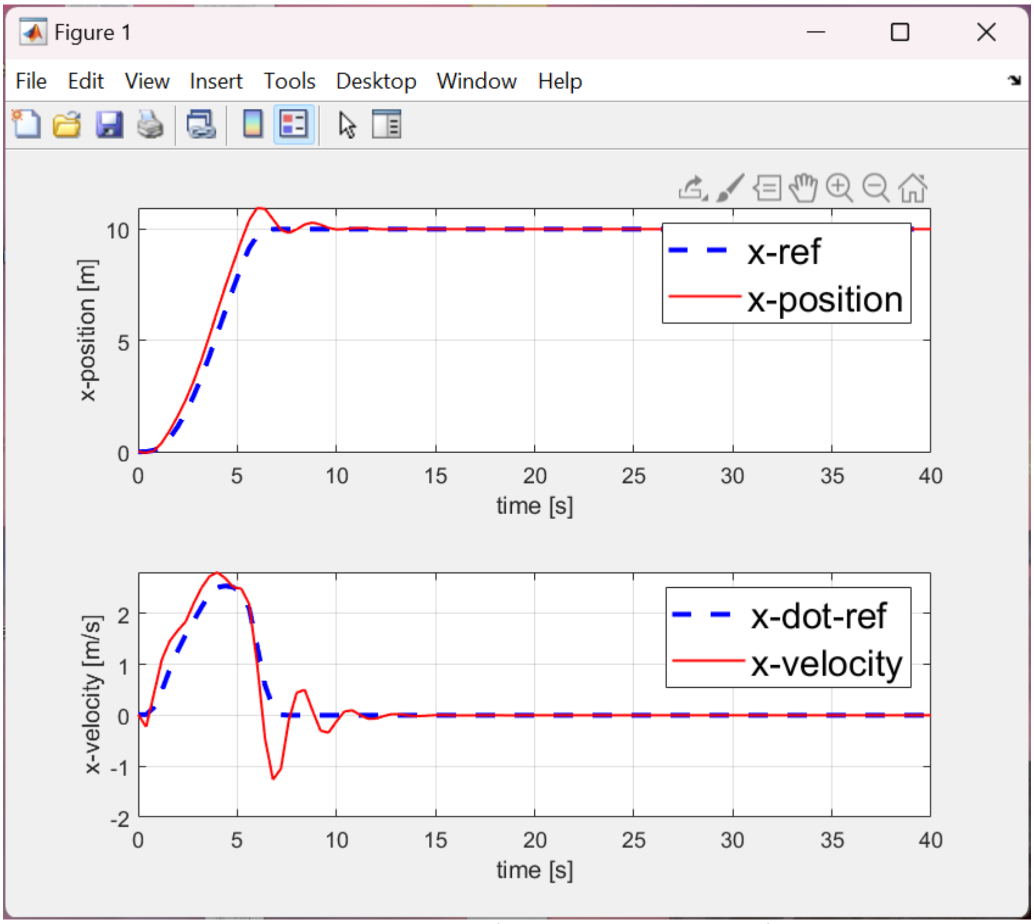

Figure 5.4: Inertial X-Reference tracking performance

There is much oscillation in tracking the x direction psotion and velocity.

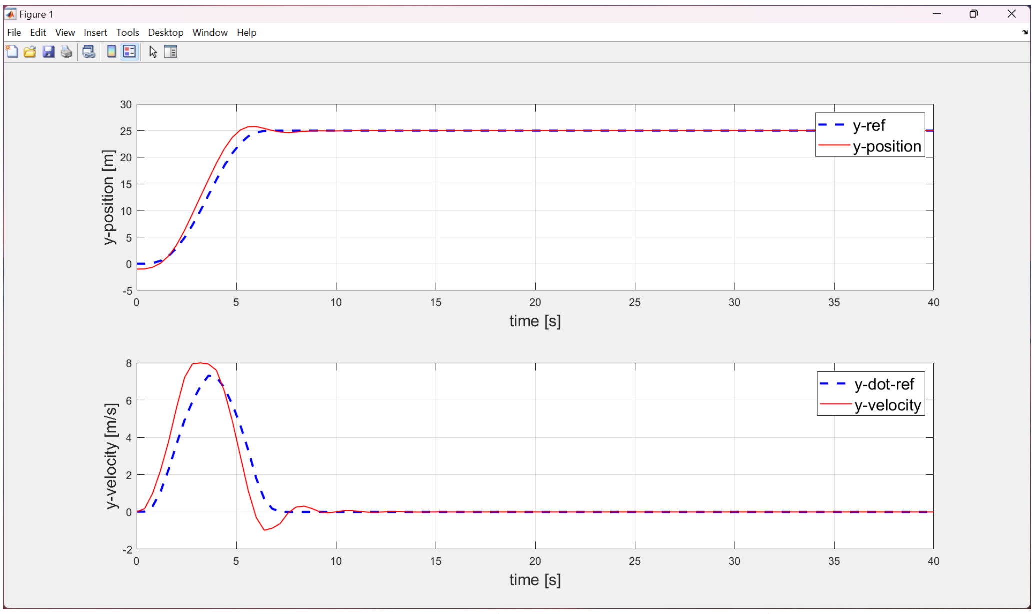

Figure 5.5: Inertial Y-Reference tracking performance

There occurs some lag in tracking both the x direction psotion and velocity.

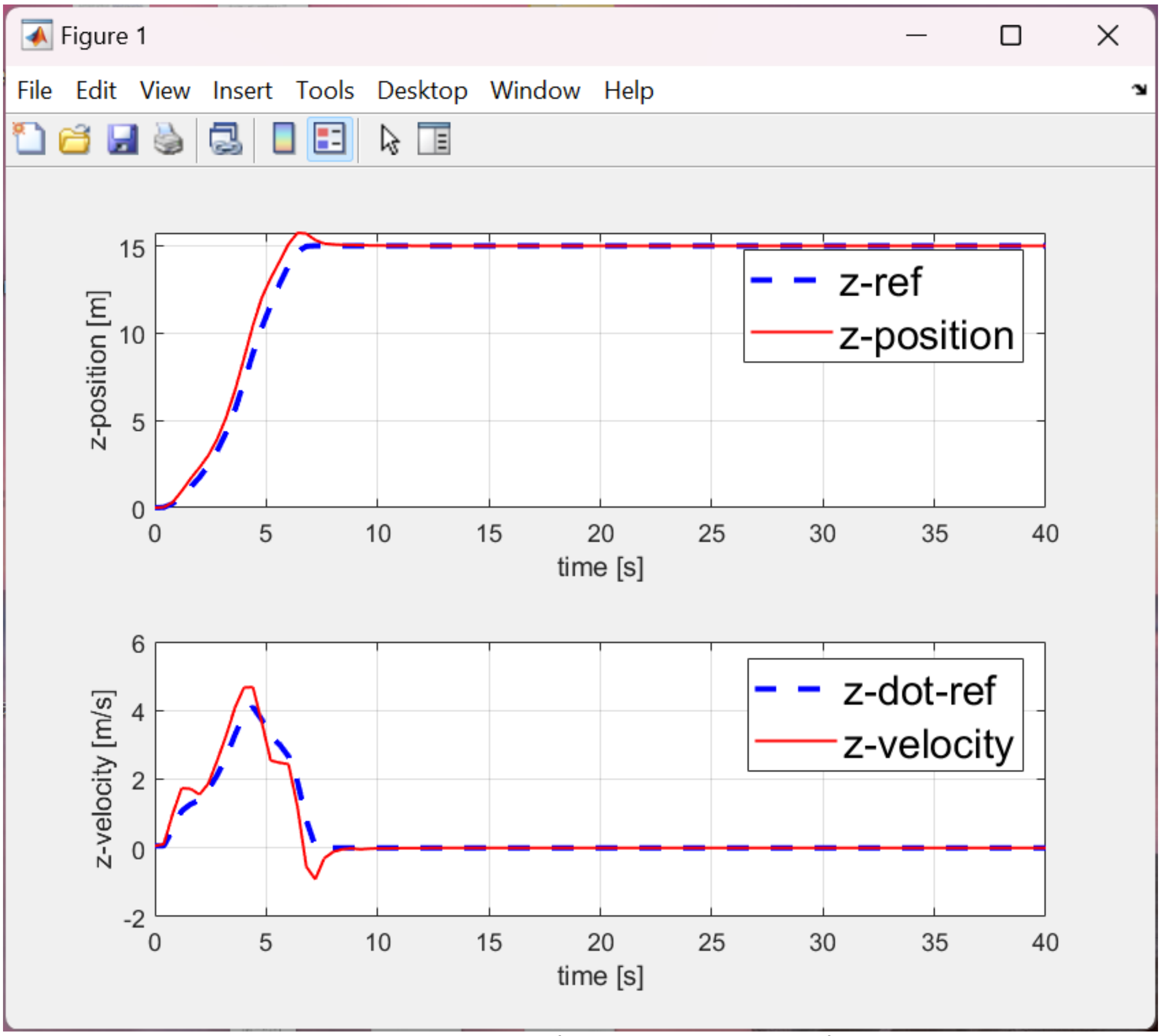

Figure 5.6: Inertial Z-Reference tracking performance

Similarly there is some deviation in tracking the z position and velocity. So we can conclude the position controller(feedback linearization controller) poles need to be fine tuned or the structure has to be modified for better performance.

Remarks

1. In our MPC controller Q,R and S are kept identity. The prediction horizon is kept 4.No control horizon is used.Also the frequency of the inner loop is kept 4 times that of the outer loop. The closed loop poles of the feedback linearization controller is chosen as -2 and -1 for convergence. These are the hyper-parameters that can be further fine tuned.
2. Solving the big optimization within .1 seconds might be challenging due to limited computational capacity of the computer(Acer Aspire 5 14 with CPU as iNTEL CORE i5 and 8 GB RAM) used for the purpose of optimization in mpc.

Chapter 6

Work To Be Done

The work presented in this report are having significant room for further exploration in this area.
This work does not address unexpected disturbances, which could introduce uncertainty into the model's predictions. Additionally, sensor noise—affecting measured states and outputs—was not accounted for, adding another layer of potential uncertainty. Future research could address these challenges by enhancing the LPV-MPC and the position controller to better handle disturbances and noise. For example, integrating a filter such as a Kalman filter into the control loop could improve tracking robustness.
Lastly, with these modifications in place, testing the entire control strategy on a real UAV would provide valuable insights into its practical performance. MATLAB code could be converted to C/C++ and implemented on an actual drone to evaluate the control system's trajectory-tracking capabilities under real-world conditions.

Bibliography

[EDA+23] Mohamed Elhesasy, Tarek N Dief, Mohammed Atallah, Mohamed Okasha, Mohamed M Kamra, Shigeo Yoshida, and Mostafa A Rushdi. Non-linear model predictive control using casadi package for trajectory tracking of quadrotor. *Energies*, 16(5):2143, 2023.

[IOI17] Maidul Islam, Mohamed Okasha, and MM Idres. Dynamics and control of quadcopter using linear model predictive control approach. In *IOP conference series: materials science and engineering*, volume 270, page 012007. IOP Publishing, 2017.

[JŻGB22] Mariusz Jacewicz, Marcin Żugaj, Robert Głębocki, and Przemysław Bibik. Quadrotor model for energy consumption analysis. *Energies*, 15(19):7136, 2022.

[LHC+19] Yang Lyu, Jinwen Hu, Ben M Chen, Chunhui Zhao, and Quan Pan. Multivehicle flocking with collision avoidance via distributed model predictive control. *IEEE transactions on cybernetics*, 51(5):2651–2662, 2019.

[MBO+19] Nathan Michel, Sylvain Bertrand, Sorin Olaru, Giorgio Valmorbida, and Didier Dumur. Design and flight experiments of a tube-based model predictive controller for the ar. drone 2.0 quadrotor. *IFAC-PapersOnLine*, 52(22):112–117, 2019.

[MCL16] Fabio Morbidi, Roel Cano, and David Lara. Minimum-energy path generation for a quadrotor uav. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1492–1498. IEEE, 2016.

[RTBP24] Joshua A Robbins, Andrew F Thompson, Sean Brennan, and Herschel C Pangborn. Energy-aware predictive motion planning for autonomous vehicles using a hybrid zonotope constraint representation. *arXiv preprint arXiv:2411.03189*, 2024.

[TF18] Carlos Trapiello Fernández. Control of an uav using lpv techniques. Master's thesis, Universitat Politècnica de Catalunya, 2018.

[WPS+21] Dong Wang, Quan Pan, Yang Shi, Jinwen Hu, and Chunhui Zhao. Efficient nonlinear model predictive control for quadrotor trajectory tracking: Algorithms and experiment. *IEEE Transactions on Cybernetics*, 51(10):5057–5068, 2021.