**MID TERM REPORT**

**Design and Implementation of Non-linear Model Predictive controller for a Non-linear Model of Quadrotor**

*A Midterm Project Report submitted in partial fulfilment*

*Of the requirement for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

**In**

**Instrumentation and Control Engineering**

*Submitted by*

**SRIRAM .T**

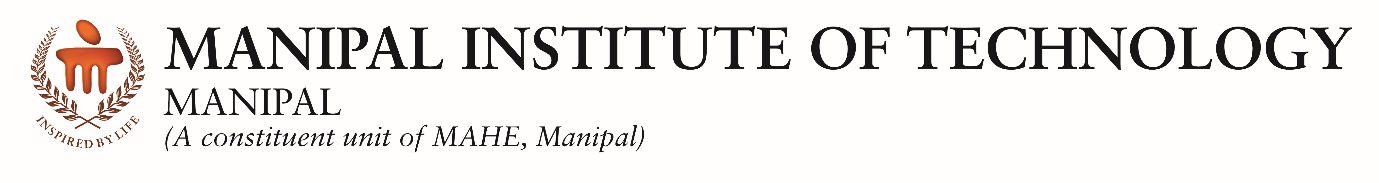
**Reg. No.:170921136**

*Under the guidance of*

**Dr.I Thirunnaukkarasu**

**Associate professor (Senior scale)**

**DEPARTMENT OF INSTRUMENTATION AND CONTROL ENGINEERING**

****

**MARCH/APRIL 2021**

**ABSTRACT**

One of the most important features of a quadrotor in order to properly work, generally in some sort of path tracking, is to have a suitable control. This thesis will approach the problem of controlling a quadrotor applying the control technique known as Nonlinear Model Predictive Control. The project work can motivate organisations/institutes to adopt unmanned aerial vehicles in their service. The research area in unmanned aerial vehicles in dynamic, but haven’t received much encouragement in the industry, therefore this work will promote the various applications of quadcopter along with major theory and practical results from the point of control theory.

The nonlinear model of the quadrotor is then used to implement a controller with input, output and state constraints. A quadratic cost function is used (discrete algebraic Riccati equation) to minimize the actuation and control signal. The quadratic cost function is minimized by using quadratic programming(QP). The controller algorithm is coded in matlab environment. The code is converted to embedded C code using matlab code generator and it is then implemented onto a flight control system for test flight.

The results obtained from the code depicts the important variables such as pitch, roll, yaw, velocity components in x, y and z axis and the four control inputs. These variables are used to control the system such that the error term in the feedback is minimized and an optimal control is achieved. The code also generates the trajectory tracking plot of the quadrotor with respect to reference trajectory, the results obtained were satisfactory. The trajectory tracking plot shows the performance of the control algorithm. First, this control should guarantee stability and feasibility, and then the control parameters are tuned to obtain the better possible performance. Proper simulations are performed by selecting different situations in terms of path tracking references, as well as in terms of the accuracy level of the control model with respect to the real system represented by a high-fidelity model.

The results obtained is used to implement onto a flight control system. The algorithm also lets other users to tweak changes to the code as per their requirements to test the logic for implementing into a quadrotor with different parameters.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Contents** | | | | | |
|  | | | | | Page No |
| Abstract | | | |  | 1 |
|  | | | | | |
| **Chapter 1** | | | **INTRODUCTION** | |  |
|  | **1.1** | Introduction | | |  |
|  | **1.2** | Motivation | | |  |
|  | **1.3** | Organization of Report | | |  |
|  | | | | | |
| **Chapter 2** | | | BACKGROUND THEORY LITERATURE REVIEW | |  |
|  | **2.1** | Literature Review | | |  |
|  | **2.2** | Mathematical Derivations | | |  |
|  | | | | | |
| **Chapter 3** | | | **METHODOLOGY** | |  |
|  | **3.1** | Non-Linear Model Predictive Control Formulation | | |  |
|  | **3.2** | Controller Design | | |  |
|  | | | | | |
| **Chapter 4** | | | **RESULT ANALYSIS** | |  |
|  | **4.1** | Introduction | | |  |
|  | **4.2** | Plots and conclusions | | |  |
|  | | | | | |
| **Chapter 5** | | | **CONCLUSION AND FUTURE SCOPE** | |  |
|  | **5.1** | Work Conclusion | | |  |
|  | **5.2** | Future Scope of Work | | |  |
|  | | | | | |
| **REFERENCES** | | | | |  |
| **ANNEXURES (OPTIONAL)** | | | | |  |
| **PROJECT DETAILS** | | | | |  |

**CHAPTER 1**

**INTRODUCTION**

**1.1 Introduction**

Unmanned aerial vehicles (UAVs) or drones are a rapidly growing ﬁeld of research. The applications of UAVs are growing day- by- day which can be categorized as scientiﬁc, commercial or military applications. Micro- UAVs or Micro Aerial Vehicles (MAVs) are classiﬁed as miniature UAVs of diﬀerent build conﬁgurations, which vary from tiny insect sized aircrafts to small quadrotors and ﬁxed wing aircrafts. MAVs due to their smaller size range are more useful in remote missions. MAVs show similarities to their UAV counterparts in various characteristics but diﬀer in terms of the forces experienced. MAVs are more susceptible to external forces due to their smaller size and lower inertia. Hence the problem of tracking and control of MAVs is much more challenging task to achieve.

The primary aim of this project is to model the MAV system mathematically and develop control algorithms for trajectory generation and tracking of an MAV. The control scheme intended to be used for the same is Non-linear Model Predictive Control (NMPC). Model Predictive Control (MPC) is a sophisticated process control method which has various applications in electro-mechanical systems which are commonly used in the chemical and petroleum industries. The MPC technique is a ﬁnite-boundary iterative optimization technique. It is very useful in-situ technique where the plant requirements vary with time. The MPC algorithm determines the control variables from the previously obtained values of the control variables. Linear MPCs are the most common form of control used in MPC applications with feedback mechanisms due to mismatch in the model and the process to be controlled. But there are several instances where linear MPCs can be inaccurate, leading way to modifying the algorithms to control the system in the non-linear scheme. The non-linear MPC or NMPC, which uses direct optimal control, is a model predictive control that use non-linear system model for the predictive function. Like MPC, NMPC also uses ﬁnite boundary conditions for the iterative process.

The goal of this work is to explore the use of Nonlinear Model Predictive Control (NMPC) applied to the control of a quadrotor in order to perform path tracking. The work will be centered on the simulation aspect, carried out with MATLAB, since the use of NMPC requires a high computational effort, therefore making it difficult to implement in a real-time application. Therefore, in this thesis it will be used a mathematical model for the quadrotor dynamics which will be based on the work of Bouabdallah in [2]. Using this mathematical model with some simplifications, a control model will be extracted to be used by the NMPC controller in the Optimal Control Problem (OCP). Afterwards, the controller will be designed. It will be showed the approach of using two control layers taking profit of the linear dynamics of the motors in order to reduce the computational burden.

**1.2 Motivation**

The quadrotor model depicts a non-linear behavior which is therefore difficult to control using traditional/linear control techniques, therefore this project work adopts a nonlinear optimal control technique known as non-linear model predictive control. This project work can stand an example to encourage the versatility of the NMPC control algorithm and wide range of applications relating to quadrotor.The benefits they provide in terms of maneuverability and simplicity of the mechanics make them an attractive option in order to explore new ideas and possibilities of application. For that reason, the quadrotor is recently having a high focus in terms of research platform, with the aim of providing new uses and/or improvements by developing or testing techniques ranging from computer vision and pattern recognition (with the aim of helping against forest fires) to several control techniques. The last one, in fact, has seen large focus in the research, since the control of the quadrotor is the basis of its performance, regardless of the application it is destined to. Nonlinear techniques, may make the controller able to work at larger operation ranges, and additionally to the advantage that nonlinear control provides in the operating range, nonlinear model predictive control (NMPC), like MPC in general, has the nice property of being able to handle constraints applied to the states and inputs during the computation of a suitable feedback law.

The Project Schedule is listed below along the timeline:

|  |  |
| --- | --- |
| *January 2020* | Understanding the quadrotor dynamics and MPC |
| *February 2020* | Literature review |
| *March 2020* | Coding of the model in Matlab/Simulink |
| *April 2020* | Implementation of the model in hardware |
| *May 2020* | Thesis preparation, Thesis presentation and Viva |

**1.3 Organization of Report**

The report has five chapters that entails introduction, background theory/ literature review, methodology, result analysis and conclusion.

Chapter 1: Introduction

Chapter 2: Background theory/ Literature review

In this chapter, the dynamics of the system are provided, as well as the corresponding simplifications in order to select the control-oriented model for the quadrotor rigid-body. Additionally, the dynamic model of the motors is selected, and the procedure to choose the suitable motor parameters is discussed.

Chapter 3: Methodology

In this chapter, the selection of a two-layer controller structure is stated, as well as the design procedure of the corresponding controllers. This chapter also presents the approach that is used to implement non-linear model predictive control and discusses about the definition of cost function and the method used to optimize it.

Chapter 4: Results analysis

This chapter presents the results of the performed simulations, in which the procedure of tuning the controller is showed, as well as results regarding the maximum possible linear velocity for the quadrotor, the controller’s performance in front of modelling errors and disturbances, and the necessary configuration in order to obtain real-time execution.

Chapter 5: Conclusions and Future work

This chapter will discuss the thoughts on the final state of the project, as well as the possibilities regarding additional features and improvements to add.

**CHAPTER 2**

**Background Review**

The literature survey consists of journal papers that includes the development of mathematical model and nonlinear control techniques for the orientation and position trajectory control and tracking of quadrotor-based MAVs and UAVs.

**2.1 Literature Review**

1. Kamesh Subbarao et.al.(2017) proposed a non-linear model predictive control strategy for unmanned aerial vehicles. The paper discusses the derivation and implementation of a nonlinear model predictive control law for tracking reference trajectories and constrained control of a quadrotor platform. The approach uses the state-dependent coefficient form to capture the system nonlinearities into a pseudo-linear system matrix. The same state-dependent coefficient form is exploited for obtaining a nonlinear equivalent of the model predictive control. The nonlinear model predictive control law is derived by first transforming the continuous system into a sampled-data form and then using a sequential quadratic programming solver while accounting for input, output and state constraints. The boundedness of the tracking errors using the sampled-data implementation was shown explicitly.
2. Nadia Miladi et. al. (2019) have proposed an explicit NMPC method based on a robust sliding mode observer for the trajectory tracking problem for quadrotor in real-time scenarios. The control algorithm is derived based on the mathematical model derived from Newton-Euler formulations. The authors have combined a high gain observer with a ﬁrst order sliding mode observer to deter-mine the states and the estimates of the disturbances using position and angular measurements to compensate for the environmental eﬀects and disturbances on the quadrotor. The estimations are then processed by the proposed predictive control algorithm to achieve trajectory tracking capability. The authors have established validity and convergence of the compound controller through numerical simulations and stability analysis.
3. M. Abdolhosseini et. al. (2013) have presented a model predictive control al-gorithm that uses fewer variables for prediction requiring lower computational capacity to control a unmanned aerial vehicle (UAV). The authors have reduced the system model using a model reduction technique to decrease the load of implementation of the MPC onto the on-board processing unit. The authors claimed to have reduced the calculation load compared to the general MPC algorithm. The simulation results have been presented by the authors to establish the performance characteristics and its advantages over traditional MPC algorithm.
4. Diganta Bhattacharjee et. al. (2020) has proposed a robust MPC for the trajectory tracking and control of a quadrotor, where attitude and position are controlled by a sliding mode control algorithm. The states in the inertial plane are tracked and controlled by the model predictive control to obtained virtual control inputs for the sliding mode controller. The authors have de-signed both control algorithms as on-board algorithm which does not require the pre-calculation of the reference inputs. The authors have included the sim-ulation results of the control strategy to illustrate the eﬃciency of the system. the authors have also validated the performance characteristics of the control scheme by comparison of the same with that of a non-linear control scheme and a backstepping control scheme
5. Tayfun Cimen (2008) has introduced the significance of, State-Dependent Riccati Equation (SDRE) strategies as general design methods which provide a systematic and effective means of designing nonlinear controllers, observers, and filters. The author points out that these methods overcome many of the difficulties and shortcomings of existing methodologies, and deliver computationally simple algorithms that have been highly effective in a variety of practical and meaningful applications. In a special session at the 17th IFAC Symposium on Automatic Control in Aerospace 2007, theoreticians and practitioners in this area of research were brought together to discuss and present SDRE-based design methodologies as well as review the supporting theory. It became evident that the number of successful simulation, experimental and practical real-world applications of SDRE control have outpaced the available theoretical results. The author reviews the theory developed to date on SDRE nonlinear regulation for solving nonlinear optimal control problems, and discusses issues that are still open for investigation. Existence of solutions as well as stability and optimality properties associated with SDRE controllers were the main contribution in this paper. The capabilities, design flexibility and art of systematically carrying out an effective SDRE design are also emphasized.
6. Sadegh Jalili et. al. (2016) have introduced an integral model predictive control strategy combined with fuzzy logic control scheme for a quadrotor UAV. The proposed control method is divided into two sub-parts: (1) a receding hori-zon design technique based MPC scheme integrated with a non-linear MPC algorithm, (2) a fuzzy logic control algorithm to maintain the error within the user-deﬁned system limits. The integration of a linear control scheme in the control system is said to reduce the resources used for the processing the system and the fuzzy logic control scheme is said to be a monitor for control gains of the non-linear model predictive control algorithm. The authors have presented simulation results to establish the eﬃciency of the proposed control technique on the quadrotor system.
7. Gang Cao et. al. (2017) have addressed the ﬂight path control problem of a quadrotor UAV with the help of a model predictive control strategy. The au-thors have used statistic Gaussian Process models from measured data to derive the UAV system model. The control structure is in the form a hierarchy with an inner loop to control the rotational subsystem and outer loop to control the translational subsystem, with controller designed separately for each subsystem. The authors have derived the GP model such that it enables the trajectory to become convex which are easier and more eﬃcient to solve. The authors have compared the performance of the proposed system with that of with existing benchmarked NMPC algorithms and an existing GP model based MPC algorithm. The proposed method is claimed to be more eﬀective and faster than the benchmark algorithms; this has been presented in the form of simulation results of the control algorithms.
8. Alexander Bogdanov et. al. (2015) have addressed an initial report on ﬂight experiments with a small, unmanned helicopter using a state dependent Riccati Equation (SDRE) controller for autonomous, agile maneuvering. The control design is based upon a full, 6-DoF, analytic nonlinear dynamic model, which is manipulated into a pseudo-linear form in which system matrices are given explicitly as a function of the current state. A standard Riccati equation is then solved numerically in each frame of a 50 Hz. control loop to design the state feedback control law on-line. Several ﬂights have been ﬂown with the helicopter to evaluate the accuracy of tracking under SDRE control in comparison with simulation results. The work was supported by the DARPA Software-Enabled Control program.

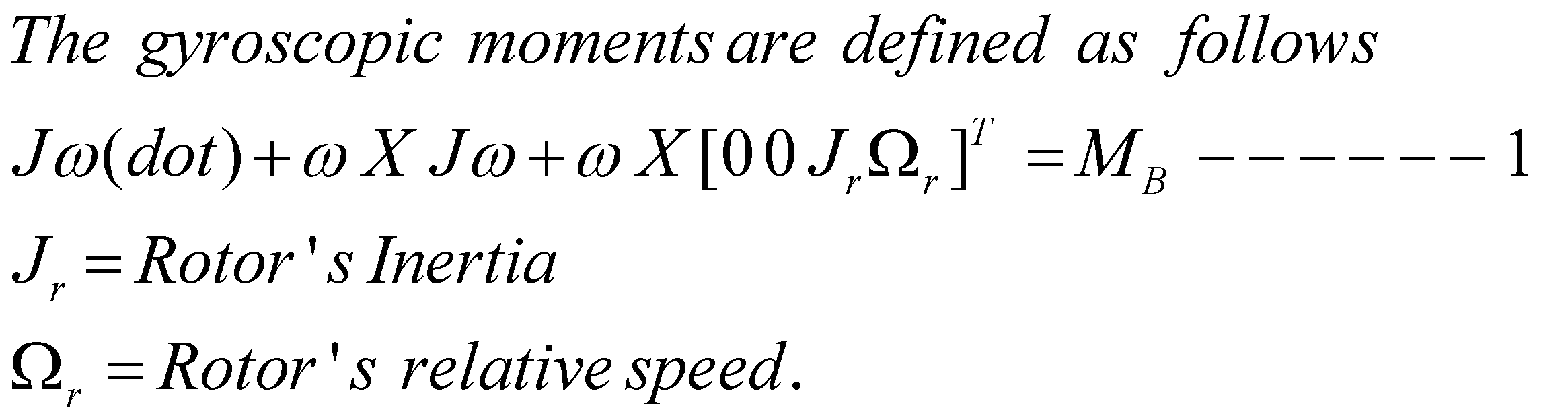
Objectives:

* To design a Non linear Model predictive controller to perform trajectory tracking.
* To develop a Matlab code for the Non-linear model predictive control.
* To convert the Matlab code to embedded C code.
* To interface the hardware and software relating to the quadcopter and test the code on the hardware.

**2.2 Mathematical Derivations**

* The motion of the quadrotor can be divided into two subsystems; rotational subsystem (roll, pitch and yaw) and translational subsystem (altitude and x and y position).
* The rotational subsystem is fully actuated while the translational subsystem is underactuated.





Inertia Matrix

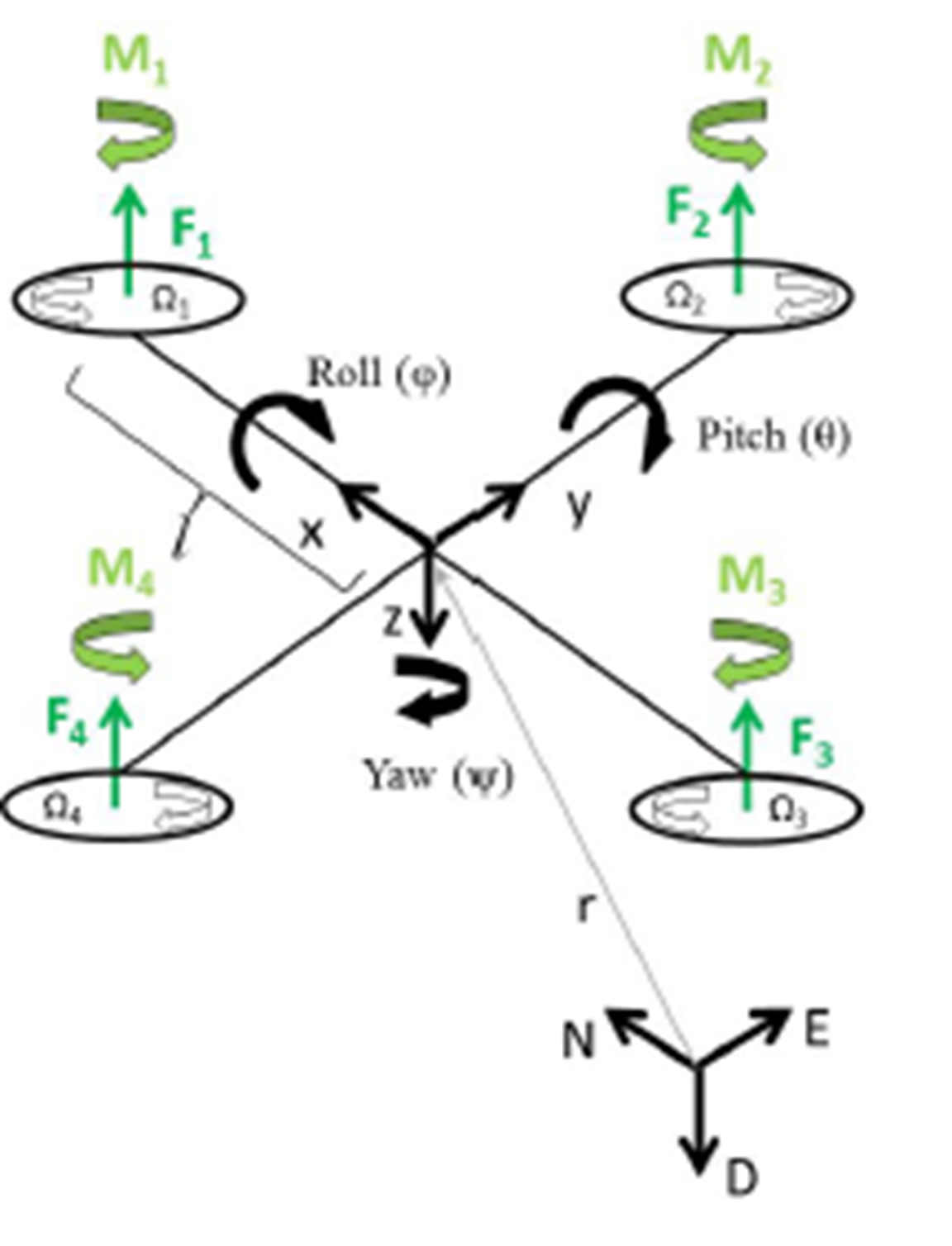
The inertia matrix for the quadrotor is a diagonal matrix, the oﬀ-diagonal elements, which are the product of inertia, are zero due to the symmetry of the quadrotor**.**

****

Where Ixx, Iyy and Izz are the area moments of inertia about the principle axes in the body frame.

Moments Acting on the Quadrotor (MB):

* For the last term of equation (1), there is a need to deﬁne two physical eﬀects which are the aerodynamic forces and moments produced by a rotor.
* As an eﬀect of rotation, there is a generated force called the aerodynamic force or the lift force and there is a generated moment called the aerodynamic moment.



A simplified equation for moment and force produced by ith can be given as:



Translational Equation of motion

The translational equation of motion for the quadrotor are based on newton’s second law and they are derived in earth’s inertial frame



Where,

m = mass of the quadrotor

r = quadrotor distance from inertial frame

Fb = non gravitational forces acting on the quadrotor

g = gravitational acceleration



State space model:

****

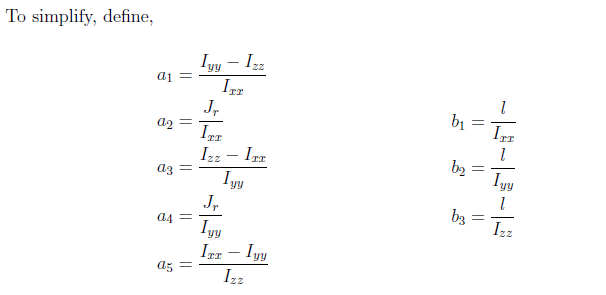
****

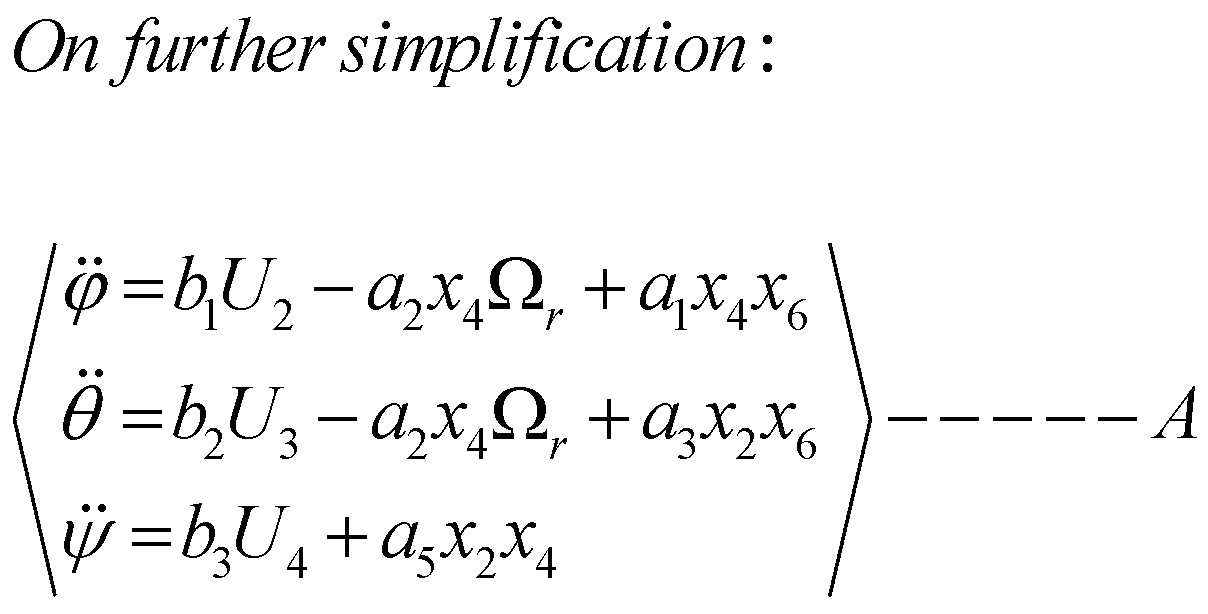
* U1 is the resulting upwards force of the four rotors which is responsible for the altitude of the quadrotor and its rate of change (z, z˙).
* U2 is the diﬀerence in thrust between rotors 2 and 4 which is responsible for the roll rotation and its rate of change (φ, φ˙).
* U3 on the other hand represents the diﬀerence in thrust between rotors 1 and 3 thus generating the pitch rotation and its rate of change (θ, θ˙).
* U4 is the diﬀerence in torque between the two clockwise turning rotors and the two counter-clockwise turning rotors generating the yaw rotation.

If the rotor velocities are needed to be calculated from the control inputs, an inverse relationship between the control inputs and the rotors’ velocities is needed, which can be acquired by inverting the matrix in the above equation to give:

****







Translational equation of motion:



It is clear here that the translational subsystem is under-actuated as it dependant on both the translational state variables and the rotational ones.

State space representation:

Using the equations of the rotational angular acceleration. Equations (A), and those of translation, Equations (B), the complete mathematical model of the quadrotor can be written in a state space representation as follows:



**CHAPTER 3**

**Methodology**

The work is based on results by Pengkai Ru and Kamesh Subbarao, the mathematical model derived is based on the formulation given in chapter 2. The detailed non linear model predictive control system has been derived below.

* The nonlinear model of the quadrotor is then used to implement a controller with input, output and state constraints.
* A cost function is used (discrete algebraic Riccati equation) to minimize the actuation and control signal.
* The quadratic cost function is minimized by using quadratic programming(QP).
* Coding of the model is done using Matlab software.
* The matlab code is converted to C code using targetlink/ Matlab code generator and it is then implemented onto a flight controller (Phikhawk) for test flight.

3.1 Non-Linear Control Formulation:

Constraints on angle:

Given the nature of quadrotor, the range of operation of the vehicle is explained below:



Input and state constraint handling:

Now attention is given to constraint handling capabilities of the NMPC problem formulation after the cost function is described:

It is necessary to constraint both:

* Total **thrust force** on each rotor
* **Restrict the magnitude of the angles** in order to stay **within the limits** allowed by Euler angles formulation discussed before

Input constraints:





Output and state constraints:

* It is also of high importance **to limit the angles so that no kinematic singularities** are encountered due to the limitations in the model description.
* In case of the quadrotor, the angles need to be within the bounds specified earlier.





For example term like does not exist for tehta equal to zero. To prevent this from

happening, the first three terms from Taylor series expansion is used.







Similarly, the terminal state and output is given by:



Cost Function:



The cost function is chosen in such a way that predicted outputs derived from the predicition horizon,are **driven to a desired state** and at the same time it should **minimize the controller effort required**.

**F ,H and C** are prediction matrices relating to system ,input and output respectively **. Q and R** are weight matrices

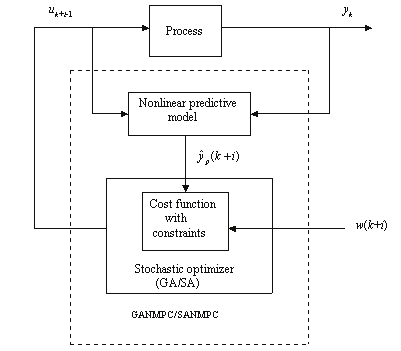
.



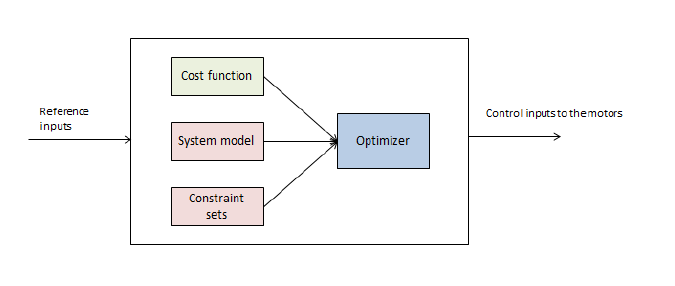


3.2 Controller Design:

A general closed-loop system with no disturbances or noise can be represented in a block diagram such as in figure below



* In our case, being the controller based in the NMPC method, the full state set of the quadrotor is required, which would make necessary the use of an state estimator. However, it will be assumed that all states of the quadrotor can be accessed and therefore no observer will be designed.
* The controller of the quadrotor is designed by using Nonlinear Model Predictive Control. Therefore, the structure of the controller can be depicted in Figure below





NMPC Definition:

Once selected the mathematical model for the quadrotor rigid body, having defined the structure of the controller the NMPC problem has to be defined. The structure of a general MPC problem consists in the computation of control inputs that minimize a cost function, subject to a certain set of constraints that includes the control oriented model of the system, over a certain prediction horizon N . This translates into the following algorithm for a basic NMPC:

* Measure the state x(n) of the system (or real model, in our case).
* Set x0 = x(n), solve the Optimal Control Problem



**CHAPTER 4**

**Result Analysis**

4.1 Introduction

As explained above, the simulation is done in MATLAB software platform. The results obtained are represented in the below ﬁgures. The quadrotor system subjected to a rectangular reference trajectory as shown in the ﬁgures below. The plots represent the system response along with the reference trajectory, to establish the system performance and its eﬀectiveness.

4.2 Plots and conclusions

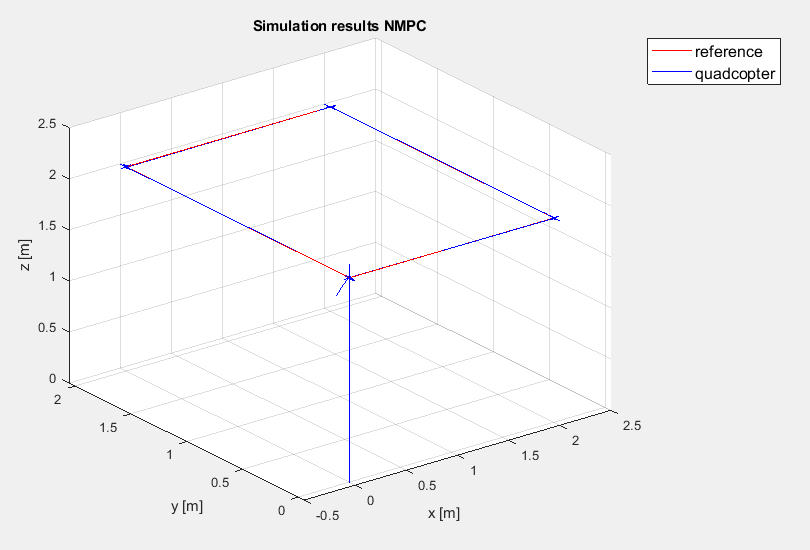


Figure 1: System position response plot in the Y-axis for the NMPC controlled system with rectangular reference trajectory

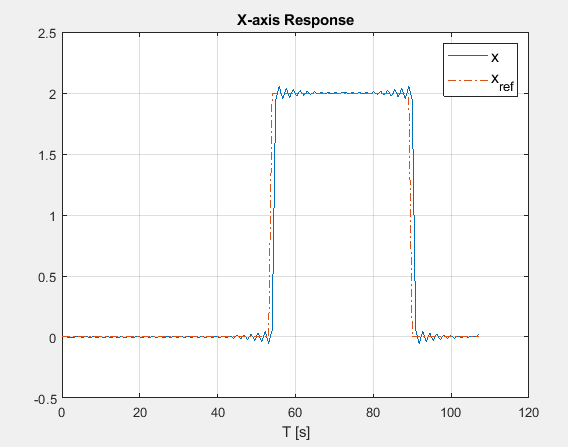


Figure 2: System position response plot in the X-axis for the NMPC controlled system with rectangular reference trajectory

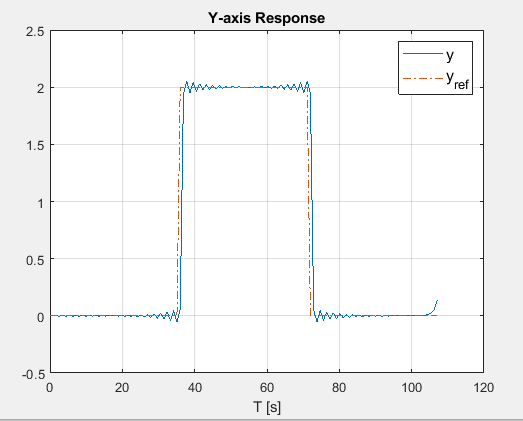


Figure 3: System position response plot in the Y-axis for the NMPC controlled system with rectangular reference trajectory

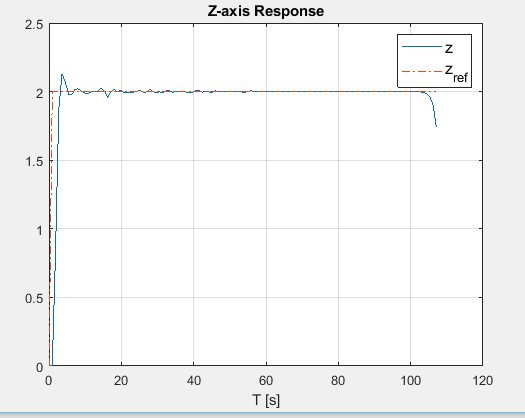


Figure 4: System position response plot in the Z-axis for the NMPC controlled system with rectangular reference trajectory



Figure 5: Quadrotor hardware completely assembled

The results obtained from the code depicts the important variables such as pitch, roll, yaw, velocity components in x, y and z axis and the four control inputs. These variables are used to control the system such that the error term in the feedback is minimized and an optimal control is achieved. The code also generates the trajectory tracking plot of the quadrotor with respect to reference trajectory, the results obtained were satisfactory. The trajectory tracking plot shows the performance of the control algorithm.

**CHAPTER 5**

**CONCLUSION AND FUTURE SCOPE OF WORK**

The work consisted on designing a non-linear model predictive control for a quadrotor from a mathematical model discussed in chapter 2. The control logic is coded in matlab environment

by taking into considerations of the parameters relating to real system setup. The control logic takes into account of both soft and hard constraints on the system dynamics to bring a best possible control signal and minimize the error. The NPMC is a versatile control technique that also solves an optimal control problem by minimizing the cost function in this work, I have considered a quadratic cost function which is solved using quadratic programming. The weight matrices (Q and R) are tuned in such a way that the desired output is achieved. The input reference trajectory is a rectangular plot in a finite state space. The trajectory tracking plot shows the performance of the control algorithm. The control guaranteed stability and feasibility, and then the control parameters are tuned to obtain the better possible performance. Proper simulations are performed by selecting different situations in terms of path tracking references, as well as in terms of the accuracy level of the control model with respect to the real system represented by a high-fidelity model.

Conclusions:

The results obtained were satisfactory in terms of trajectory tracking and time domain

performance objectives. The simulation results are discussed in detail in chapter 4. In this project, it has been shown that the use of NMPC to control a quadrotor is able to guarantee stability and feasibility and provide a considerably good performance. This have been the case for simple reference inputs as well as for complete path tracking, and considering an ideal model for the quadrotor, as well as considering a real model with added effects with respect to the control model.

However, the major drawback of this approach remains after the conclusion of this project, which is the inability to obtain real-time control. Even after several attempts to minimize

the time ratio, the best obtained value, considering the unmodelled dynamics and wind disturbances. This is relatively far from a reasonable value, which should be low enough (lower than one) such that it would guarantee real-time control even after considering additional time delays due to communication, taking data from the sensors and processing it, etc.

Therefore, any future work should focus on obtaining real-time control. This could be approached by considering the use of a commercial and more powerful nonlinear optimization solver, such as Knitro from Artelys and/or the use of a PC with a more powerful microprocessor.

**CHAPTER 6**

**ENVIRONMENTAL, SOCIAL AND ECONOMIC IMPACT**

Nowadays, there is an extensive and constantly increasing use of quadrotors, or, more generally, UAVs. These vehicles provide a wide set of possibilities in terms of applications, with the corresponding consequences in terms of environmental, social and economic impact. Therefore, being a proper path control the most basic feature of any application, then the impact these applications may have are, by extension, affected by the control of the UAV. These vehicles, themselves, have low environmental impact in terms of contamination (theoretically zero, since it is an electric vehicle, but the energy sources would have to be taken into account). A proper control of the quadrotors may help reduce its power consumption. However, the main benefit it could provide comes at the hand of its possible applications. For instance, in the Introduction it was cited a research case consisting in the use of quadrotors for fire prevention. This example helps to understand the potential beneficial impact that UAVs can have for the environment.

With respect to the social impact, UAVs, in particular quadrotors, still have the main share of civilian use related to entertainment. Of course, the use of a quadrotor with a controller able to guarantee stability, as well as to provide robustness and good performance leaves a better experience for the user. Additionally, and on a more serious note, there are other uses of UAVs that can have a potentially big social impact, such as the case, stated in the Introduction, of the German consortium that successfully implemented a service to provide medicines to remote and low-accessible areas with the use of quadrotors.

Finally, regarding the economic impact, this is probably the aspect that provides a higher development rate for UAVs (along with military applications). The potential benefits that the use of UAVs can have for a company is mainly due to the reduction in costs that, for a similar application using a manned vehicle, would be considerably higher. Particularly, distribution companies have the opportunity of offering a service to their clients that can be faster than the traditional ground transportation, while at the same time cutting in personnel cost.

**REFERENCES**

1. Pengkai Ru and Kamesh Subbarao , ‘Nonlinear Model Predictive Control for Unmanned Aerial Vehicles’, Flight Mechanics Conference, AIAA Aviation Forum, Dallas, TX, USA, 22–25 June 2015; Kamesh Subbarao, Carlos Tule and Pengkai Ru, Nonlinear Model Predictive Control Applied to Trajectory Tracking for Unmanned Aerial Vehicles, No. AIAA 2015-2857.
2. Diganta Bhattacharjee and Kamesh Subbarao,’ Robust Control Strategy for Quadcopters using Sliding Mode Control and Model Predictive Control’, AIAA Scitech 2020 Forum.
3. Tayfun Çimen,’ State-Dependent Riccati Equation (SDRE) Control: A Survey’, Proceedings of the 17th World Congress, The International Federation of Automatic Control Seoul, Korea, July 6-11, 2008.
4. Subbarao. K., Tule. C, Pengkai Ru. Nonlinear model predictive control applied to trajectory tracking for unmanned aerial vehicles. In Proceedings of the AIAA Atmospheric Flight Mechanics Conference, AIAA Aviation Forum, Dallas, TX, USA, 22–26 June 2015; Number AIAA 2015-2857.
5. Limon D, Alamo T, Raimondo D, De La Peña D. M, Bravo J, Ferramosca A, and Camacho E, ’Input-to-state stability: a unifying framework for robust model predictive control’ Nonlinear model predictive control, Springer, 2009, pp. 1–26.
6. Camacho E.F, Bordons C, ‘Model Predictive Control’ Springer: London, UK, 1999; pp. 18–20.

**ANNEXURES**

Codes / algorithms :

function [X\_vector,U\_vector]=nmpc2(refs)

% Data

m = 0.8; % Mass of Quadrotor

L = 0.165; % Moment arm length

k\_t = 3 \* 10^(-6); % Thrust constant

b = 1 \* 10^(-7); % Moment constant

g = 9.81; % Acceleration due to gravity

k\_d = 0.25; % Drag constant

Ixx = 5 \* 10^(-3); % Momentds of inertia

Iyy = 5 \* 10^(-3);

Izz = 1 \* 10^(-2);

c\_m = 1 \* 10^(4); % Moment constant

Tmax = 50; % Maximum Time duration

% System Sizes

nx = 12; % Nb of states

nu = 4; % Nb of inputs

ny = 6; % Nb of outputs

x0\_quadcopter = zeros(nx,1);

% Inputs at equilbrium point at all states=0

u\_eq = g\*m/(4\*k\_t\*c\_m);

N = 15; % prediction horizon

umax = 100-u\_eq; umin = u\_eq; % input limit

zmax = inf; zmin = inf; % room limits

xymax = inf;

%%

% Construction of the linear state space model

% symbolic variables

syms x y z v\_x v\_y v\_z phi theta psi1 p q r u1 u2 u3 u4

% state vector

state = [x; y; z; v\_x; v\_y; v\_z; phi; theta; psi1; p; q; r];

% input vector

input = [u1; u2; u3; u4];

% The functions f:

f1 = v\_x;

f2 = v\_y;

f3 = v\_z;

f4 = -k\_d/m \* v\_x + k\_t\*c\_m/m \*(sin(psi1)\*sin(phi)+cos(psi1)\*cos(phi)\*sin(theta))\*(u1+u2+u3+u4);

f5 = -k\_d/m \* v\_y + k\_t\*c\_m/m \*(cos(phi)\*sin(psi1)\*sin(theta)-cos(psi1)\*sin(phi))\*(u1+u2+u3+u4);

f6 = -k\_d/m \* v\_z - g + k\_t\*c\_m/m \*(cos(theta)\*cos(phi))\*(u1+u2+u3+u4);

f7 = p + q\*(sin(phi)\*tan(theta)) + r\*(cos(phi)\*tan(theta));

f8 = q\*cos(phi) - r\*sin(phi);

f9 = sin(phi)/cos(theta) \*q + cos(phi)/cos(theta) \* r;

f10 = L\*k\_t\*c\_m/Ixx \* (u1-u3) - (Iyy-Izz)/Ixx \* q\*r;

f11 = L\*k\_t\*c\_m/Iyy \* (u2-u4) - (Izz-Ixx)/Iyy \* p\*r;

f12 = b\*c\_m/Izz \* (u1-u2+u3-u4) - (Ixx-Iyy)/Izz \* q\*p;

F = [f1; f2; f3; f4; f5; f6; f7; f8; f9; f10; f11; f12];

%%

% Deriving the functions in the state variables (Jacobian)

J = jacobian(F, state);

% Evaluating the jacobian in the equilibrium values: the result is A

A = subs(J,[state; input],[zeros(nx,1); u\_eq\*ones(nu,1)]);

A = double(A);

% Deriving the functions in the input variables

J = jacobian(F, input);

% Evaluating the derivatives in the equilibrium values: the result is B

B = subs(J, [state; input],[zeros(nx,1); u\_eq\*ones(nu,1) ]);

B = double(B);

% The output consists of states 1 to 3 and 7 to 9, so C selects these and D

% is zero

C = [eye(3), zeros(3,9);

zeros(3,6), eye(3), zeros(3,3)];

D = zeros(ny,nu);

% Creating the continuous time system

c\_sys = ss(A,B,C,D);

%Checking the stability

disp('Poles(continous system):')

disp(eig(A))

%%

%Discretization

T\_s = 0.9; % Sampling time

M = inv(eye(nx) - A\*T\_s/2)

[A\_d,B\_d,C\_d,D\_d]=c2dm(A,B,C,D,T\_s,'zoh'); %Discretization using zero order hold

% Creating the discrete time system

sys = ss(A\_d,B\_d,C\_d,D\_d,T\_s);

%%

%Checking the stability

disp('Poles(discretized system):')

disp(eig (A\_d))

disp ('Controllability matrix');

CO = ctrb(A\_d,B\_d);

disp('Rank of the controllability matrix:');

rank(CO)

disp ('Observability matrix');

OB = obsv(A\_d,C\_d);

disp('Rank of the observability matrix:');

rank(OB)

%% Setpoints

[M,~] = size(refs);

y\_ref\_vector = reshape(refs',[M\*ny,1]); % Built optimization problem that finds u such that (y-y\_ref) minimized

F = zeros(M\*ny,nx); % Establish O

F(1:ny,:) = C\_d;

for i=1:M-1

F(i\*ny+1:(i+1)\*ny,:) = F((i-1)\*ny+1:i\*ny,:)\*A\_d;

end

H\_o = zeros(M\*ny,nu); % Establish H: H\_0 = D, H\_k = CA^{k-1}B

H\_o(1:ny,:)=D\_d;

temp = B\_d;

for k=1:M-1

H\_o(k\*ny+1:(k+1)\*ny,:)=C\_d\*temp;

temp = A\_d\*temp;

end

H = zeros(M\*ny,M\*nu);

for i=0:M-1

H(:,i\*nu+1:(i+1)\*nu) = H\_o;

H\_o = [zeros(ny, nu);

H\_o(1:end-ny,:)];

end

Q = diag(repmat([0.2,0.2,0.2,0,0,0],1,M));

R = 1.5e-4\*eye(M\*nu);

u\_ref\_vector = quadprog(R+H'\*Q\*H,H'\*Q'\*(y\_ref\_vector-F\*x0\_quadcopter));

u\_ref\_vector=-u\_ref\_vector;

u\_ref = reshape(u\_ref\_vector,[nu,M])'; % Reshape the inputs as rowvectors for each time t

[Y, T, x\_ref] = lsim(sys,u\_ref); % Simulation to get actual outputs and reference states

x\_ref\_vector = reshape(x\_ref',[M\*nx,1]);

close all

%%

for i=(M+1):(M+N) % Reference vector with N repititions of the last input/state

x\_ref\_vector((i-1)\*nx+1:i\*nx) = x\_ref\_vector(i-nx+1:i);

u\_ref\_vector((i-1)\*nu+1:i\*nu) = u\_ref\_vector(i-nu+1:i);

end

qdiag = ones(1,nx);

rdiag = 1e-7\*ones(1,nu);

bigQ = repmat(qdiag,1,N);

bigR = repmat(rdiag,1,N);

H = 2\*diag([bigQ,bigR]);

A\_eq = [eye(nx\*N), zeros(nx\*N,nu\*N)];

for i=1:N

if(i<N)

A\_eq(i\*nx+1:(i+1)\*nx,(i-1)\*nx+1:i\*nx) = -A\_d;

end

A\_eq((i-1)\*nx+1:i\*nx, nx\*N + (i-1)\*nu+1:nx\*N + i\*nu) = -B\_d;

end

A\_limit = [eye(nx\*N), zeros(nx\*N,nu\*N); -eye(nx\*N), zeros(nx\*N,nu\*N); % Limiting the space of the quadcopter and the input size

zeros(nu\*N,nx\*N), eye(nu\*N); zeros(nu\*N,nx\*N), -eye(nu\*N)];

b\_limit\_u\_max = repmat([umax],nu,1); % Inputs between 0 and umax

b\_limit\_u\_min = repmat([umin],nu,1);

b\_limit\_x\_max = [xymax; xymax; zmax; inf\*ones(nx-3,1)]; % XY coordinates between + and - xymax, z coordinate between 0 and zmax

b\_limit\_x\_min = [xymax; xymax; zmin; inf\*ones(nx-3,1)];

b\_limit = [repmat(b\_limit\_x\_max,N,1); repmat(b\_limit\_x\_min,N,1) ;repmat(b\_limit\_u\_max,N,1); repmat(b\_limit\_u\_min,N,1)];

x = x0\_quadcopter;

for k=1:M

x\_ref = x\_ref\_vector((k-1)\*nx+1:(k+N-1)\*nx); % Extract time horizon

u\_ref = u\_ref\_vector((k-1)\*nu+1:(k+N-1)\*nu);

f = -H\*[x\_ref; u\_ref]; % Complete optimization matrices

b\_eq = zeros(N\*nx,1);

b\_eq(1:nx) = A\_d\*x;

xu = quadprog(H,f,A\_limit,b\_limit,A\_eq,b\_eq); % Solve optimization problem

%xu = quadprog(H,f,[],[],A\_eq,b\_eq);

u = xu(nx\*N+1:nx\*N+nu); % Applying first input

y = C\_d\*x + D\_d\*u; % Update state

x = A\_d\*x + B\_d\*u;

U\_vector(k,:) = u';

X\_vector(k,:) = x';

Y\_vector(k,:) = y';

end

close all

%%

figure % Plotting

plot(T,Y\_vector(:,1));

hold on

plot(T,refs(:,1),'-.');

xlabel('T [s]')

legend({'x','x\_{ref}'},'FontSize',12);

title('X-axis Response');

grid on

figure

plot(T,Y\_vector(:,2));

hold on

plot(T,refs(:,2),'-.');

xlabel('T [s]')

legend({'y','y\_{ref}'},'FontSize',12);

title('Y-axis Response');

grid on

figure

plot(T,Y\_vector(:,3));

hold on

plot(T,refs(:,3),'-.');

xlabel('T [s]')

legend({'z','z\_{ref}'},'FontSize',12);

title('Z-axis Response');

grid on

figure

plot3(refs(:,1),refs(:,2),refs(:,3),'r');

hold on

plot3(Y\_vector(:,1),Y\_vector(:,2),Y\_vector(:,3),'b');

legend({'reference','quadcopter'},'FontSize',12);

xlabel('x [m]')

ylabel('y [m]')

zlabel('z [m]')

title('Simulation results NMPC')

grid on

% figure

% plot(T,X\_vector);

% legend({'x','y','z','v\_x','v\_y','v\_z','\phi','\tau','\psi','\omega\_x','\omega\_y','\omega\_z'},'FontSize',12);

% title('States');

% xlabel('T [s]')

% figure

% plot(T,X\_vector(:,1));

% legend('x','FontSize',12);

% title('States');

% xlabel('T [s]')

% figure

% plot(T,X\_vector(:,2));

% legend('y','FontSize',12);

% title('States');

% xlabel('T [s]')

% figure

% plot(T,X\_vector(:,3));

% legend('z','FontSize',12);

% title('States');

% xlabel('T [s]')

figure

plot(T,X\_vector(:,4));

legend('v\_x','FontSize',12);

title('States');

xlabel('T [s]')

grid on

figure

plot(T,X\_vector(:,5));

legend('v\_y','FontSize',12);

title('States');

xlabel('T [s]')

grid on

figure

plot(T,X\_vector(:,6));

legend('v\_z','FontSize',12);

title('States');

xlabel('T [s]')

grid on

figure

plot(T,X\_vector(:,7));

legend('\phi','FontSize',12);

title('States');

xlabel('T [s]')

grid on

figure

plot(T,X\_vector(:,8));

legend('\theta','FontSize',12);

title('States');

xlabel('T [s]')

grid on

figure

plot(T,X\_vector(:,9));

legend('\psi','FontSize',12);

title('States');

xlabel('T [s]')

grid on

% figure

% plot(T,X\_vector(:,10));

% legend('\omega\_x','FontSize',12);

% title('States');

% xlabel('T [s]')

% figure

% plot(T,X\_vector(:,11));

% legend('\omega\_y','FontSize',12);

% title('States');

% xlabel('T [s]')

% figure

% plot(T,X\_vector(:,12));

% legend('\omega\_z','FontSize',12);

% title('States');

% xlabel('T [s]')

figure

plot(T,U\_vector(:,1));

title('Control input U1');

xlabel('T [s]')

grid on

figure

plot(T,U\_vector(:,2));

title('Control input U2');

xlabel('T [s]')

grid on

figure

plot(T,U\_vector(:,3));

title('Control input U3');

xlabel('T [s]')

grid on

figure

plot(T,U\_vector(:,4));

title('Control input U4');

xlabel('T [s]')

grid on

end

%%

PROJECT DETAILS

|  |  |  |  |
| --- | --- | --- | --- |
| *Student Details* | | | |
| **Student Name** | Sriram.T | | |
| Register Number | 170921136 | Section / Roll No | B / 31 |
| Email Address | sriram29598@gmail.com | Phone No (M) | 9741400597 |
| **Student Name** |  | | |
| Register Number |  | Section / Roll No |  |
| Email Address |  | Phone No (M) |  |
|  | | | |
| *Project Details* | | | |
| **Project Title** | Design and Implementation of Non-linear Model Predictive controller for a Non-linear Model of Quadrotor | | |
| Project Duration | 5 months | Date of reporting | 11.1.21 |
| Expected date of completion of project | 31.5.21 |  |  |
|  |  | | |
| *Organization Details* | | | |
| **Organization Name** | **NA** | | |
| Full postal address with pin code | NA | | |
| Website address | NA | | |
|  |  | | |
| *Supervisor Details* | | | |
| **Supervisor Name** | **NA** | | |
| Designation | NA | | |
| Full contact address with pin code | NA | | |
| Email address | NA | Phone No (M) | NA |
|  |  | | |
| *Internal Guide Details* | | | |
| **Faculty Name** |  | | |
| Full contact address with pin code | Dept. of Instrumentation & Control Engineering, Manipal Institute of Technology, Manipal – 576 104 (Karnataka State), INDIA | | |
| Email address | itarasu1881@gmail.com | | |
| *Co- Guide Details (if any)* | | | |
| **Faculty Name** | **NA** | | |
| Full contact address with pin code | Dept. of Instrumentation & Control Engineering, Manipal Institute of Technology, Manipal – 576 104 (Karnataka State), INDIA | | |
| Email address | NA | | |

**General Guidelines (Delete this page when making the report submission)**

* Project Report to be minimum 18 pages.
* Project report to be maximum 20 - 25 pages (preferred)
* Page margin; Left = Right = Top = Bottom Margins = 1”
* Page Numbering Position: Bottom with right justified and continuous numbering from the Introduction Chapter
* Use Times New Roman Font with Normal Style, paragraph justified and 1.25 line spacing
* Paragraph Heading: Times New Roman Font, Bold, Font Size 14; Paragraph Matter: Times New Roman Font, Normal, Font Size 12;
* Sub-paragraphs be appropriately numbered as in 1.1, 1.2, 1.3 etc; Sub-paragraph Heading: Times New Roman Font, Italics, Font Size 12; Sub-paragraph Matter: Times New Roman Font, Normal, Font Size 12;
* Figure captions below Figure with chapter wise numbering
* Tables captions above Table with chapter wise numbering
* All references must be quoted in ascending order (follow IEEE format for referencing)
* Project Details page must be the last page in the project report
* **Arrangement of contents**

[1] Cover page (same as inner page)

[2] Abstract

[3] Table of contents

[4] Chapters 1, 2, 3, 4, 5

[5] References (follow IEEE format)

[6] Annexures

[7] Project Details (Last page of the report)

* The above guidelines should be used only as a help guide and is more or less a standard way of report writing
* Report formatting should not be disturbed in any form.
* **Project students are requested to discuss with their department guides regarding the contents of the project report**