ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

MODEL PREDICTIVE CONTROL PROJECT

DEVELOPING A MPC CONTROLLER FOR QUADCOPTER

Authors

DEVAKUMAR THAMMISETTY (SCIPER:285481)
ANIRVAN DUTTA (SCIPER: 293013)

12th January 2020

Introduction

In this report, we present the linear MPC implementation of linearized quadcopter as well as non-linear MPC implementation on nonlinear quadcopter dynamics. We present the implementation, results and discussion according to the Deliverables provided in the project document.

Deliverable 2.1

A quadcopter is an underactuated system with 6 DOF and 4 inputs. The quadcopter is defined by 12 dimentional state vector [$\omega\ \theta\ v\ p$] following the dynamical equations

$$\dot{\omega} = \mathcal{I}^{-1}(-\omega \times \mathcal{I}\omega \begin{bmatrix} M_\alpha \\ M_\beta \\ M_\gamma \end{bmatrix}) \tag{1a}$$

$$\dot{\theta} = \omega \tag{1b}$$

$$\dot{v} = (-mg + FR) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{1c}$$

$$\dot{p} = v \tag{1d}$$

where, ω is the angular velocity of the body co-ordinate frame w.r.t world co-ordinate frame, θ is the orientation v is the velocity and p is the position of the center of mass of the quadcopter. The 4 rotor inputs $u = [u_1, u_2, u_3, u_4]$ generate force and moment according to Equation 2, given below.

$$\begin{bmatrix} F \\ M_\alpha \\ M_\beta \\ M_\gamma \end{bmatrix} = \begin{bmatrix} k_F & k_F & k_F & k_F \\ 0 & k_M L & 0 & -k_M L \\ -k_M L & 0 & k_M L & 0 \\ k_M & -k_M & k_M & -k_M \end{bmatrix} u \tag{2}$$

Denoting the combined dynamics of Eq.1 and Eq.2 as $\dot{x} = f(x, u)$. The quadcopter dynamics is linearised around the steady state $\dot{x_s} = f(x_s, u_s) = 0$. Using Eq.1(b,d), results in zero values for ω and v. Also, equating Eq.1(c) to zero results in Eq.3

$$\begin{bmatrix} Fsin(pitch) \\ -Fcos(pitch)sin(roll) \\ -mg + Fcos(pitch)cos(roll) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{3}$$

This gives the values for θ=$roll, pitch$ to zero. Therefore, at steady state, the quadcopter could only have non zero values in $p = x, y, z$ and $\theta = yaw$. Further, as the rank of matrix in Eq.2 is 4, u can be used to independently control $F, M_\alpha, M_\beta, M_\gamma$, which can be utilised to control each of the non-zero steady states. Thus, on multiplying the inverse of matrix in Eq.2, results into decomposition of into 4 sub-systems.

Deliverable 3.1: Design MPC Regulators

Design Procedure for regulation

The MPC problem for regulation for each dimension x, y, z, yaw can be formulated as;

$$\min_{u} \quad \sum_{i=0}^{N-1} x_i^T Q x_i + u_i^T R u_i + x_N^T Q_f x_N \tag{4a}$$

$$\text{s.t} \quad x_{i+1} = A x_i + B u_i \tag{4b}$$

$$A_x x_i \leq b_x \tag{4c}$$

$$A_u u_i \leq b_u \tag{4d}$$

$$A_f x_N \leq b_f \tag{4e}$$

- Matrix A and B capture the linearized dynamics of each dimensions
- Matrix A_x and b_x are convex half-space state constraints provided as part of system, the values of which are provided in Table.1.
- Similarly, A_u and b_u are convex half-space input constraints whose values are mentioned in Table.1.
- To ensure recursive feasibility and stability, a terminal LQR controller was used in each dimension. The LQR controller uses discrete-time algebraic Riccati equation (DARE) as follows:

$$P = Q + A^T P A - A^T P B (R + B^T P B)^{-1} B^T P A \tag{5a}$$

$$K = -(R + B^T P B)^{-1} B^T P A \tag{5b}$$

$$Q_f = P \tag{5c}$$

- The terminal maximal invariant set using the LQR control law $u = Kx$ in the form of polyhedron A_f, b_f is computed. This ensures recursive feasibility of the MPC problem.
- To ensure stability in the MPC problem, terminal cost in form of $x_N^T Q_f x_N$ was added making the total cost a Lyapunov function.
- The parameters Q, R, N are the tuning parameter of the MPC. Horizon length, $N = 20$ was selected to ensure the MPC problem was feasible from set of initial states. It was observed that the ratio of Q and R was important, increasing the ratio improved faster response on expense of larger input. The terminal positive definite matrix was also analyzed for each domain to check which variable of each dimension to be weighted and scaled in Q. It was also observed that Q and R effected the size of the terminal set and if not correctly set, may render the MPC problem infeasbile. Q and R were iteratively tuned such that the performance criteria of settling time of eight seconds was achieved. Throughout the project $Q = I$ and $R = 5I$ was selected to be matrix of appropriate dimension for all sub systems.

The implementation of regulation is provided in Deliverable_3_1.zip

Controller	A_x	b_x	A_u	b_u
MPC_x	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0.035 \\ 0.035 \end{bmatrix}$	$\begin{bmatrix} 1 \\ -1 \end{bmatrix}$	$\begin{bmatrix} 0.3 \\ 0.3 \end{bmatrix}$
MPC_x	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0.035 \\ 0.035 \end{bmatrix}$	$\begin{bmatrix} 1 \\ -1 \end{bmatrix}$	$\begin{bmatrix} 0.3 \\ 0.3 \end{bmatrix}$
MPC_z	-	-	$\begin{bmatrix} 1 \\ -1 \end{bmatrix}$	$\begin{bmatrix} 0.3 \\ 0.2 \end{bmatrix}$
MPC_yaw	-	-	$\begin{bmatrix} 1 \\ -1 \end{bmatrix}$	$\begin{bmatrix} 0.2 \\ 0.2 \end{bmatrix}$

Table 1: Constraint parameters for the regulation formulation

Terminal Set plots

In this section we present the plots of the polytopic terminal set. As the x and y was higher than 2 dimension, projection of the plot are presented.

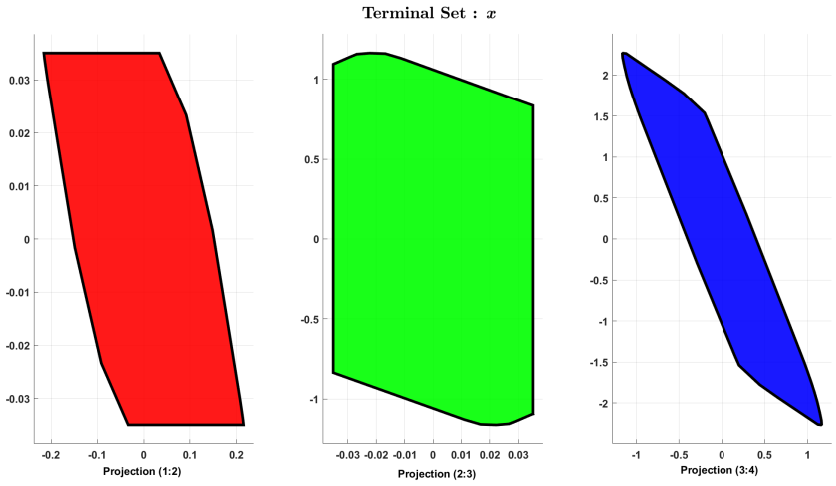

Figure 1: Terminal set of x

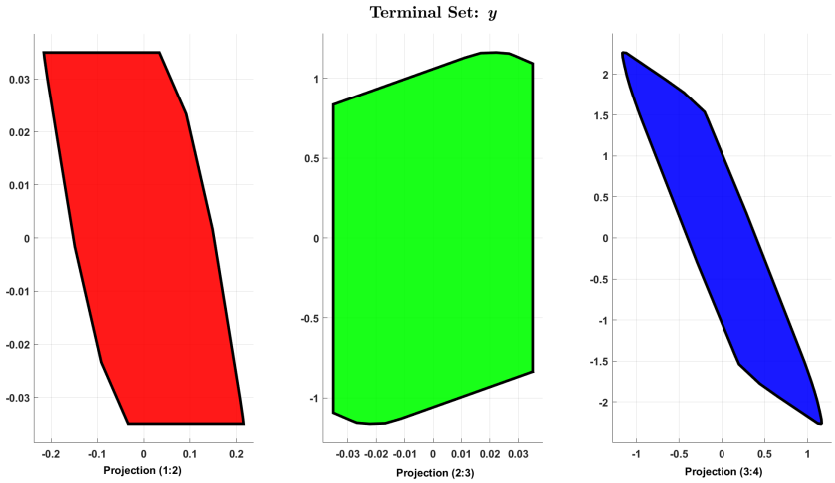

Figure 2: Terminal set of y

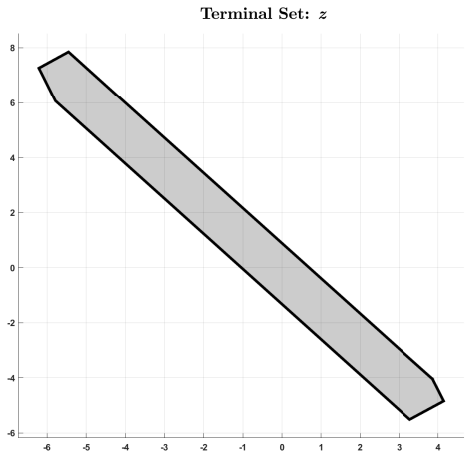

Figure 3: Terminal set of z

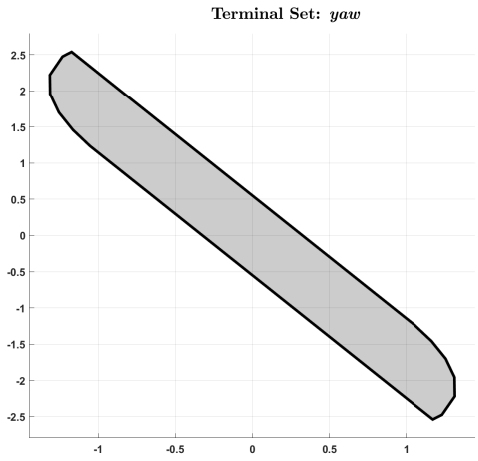

Figure 4: Terminal set of yaw

Regulation plots

In this section, we present the regulation result in each dimension. The initial value of x,y,z was set to -2 and yaw was set to $-\pi/4$. An settling time of 6 secs was observed for x and y while a settling time of 5 secs was observed for z and yaw, thus meeting the performance criteria. All the variables of each dimensions are presented to ensure that constraints are not violated.

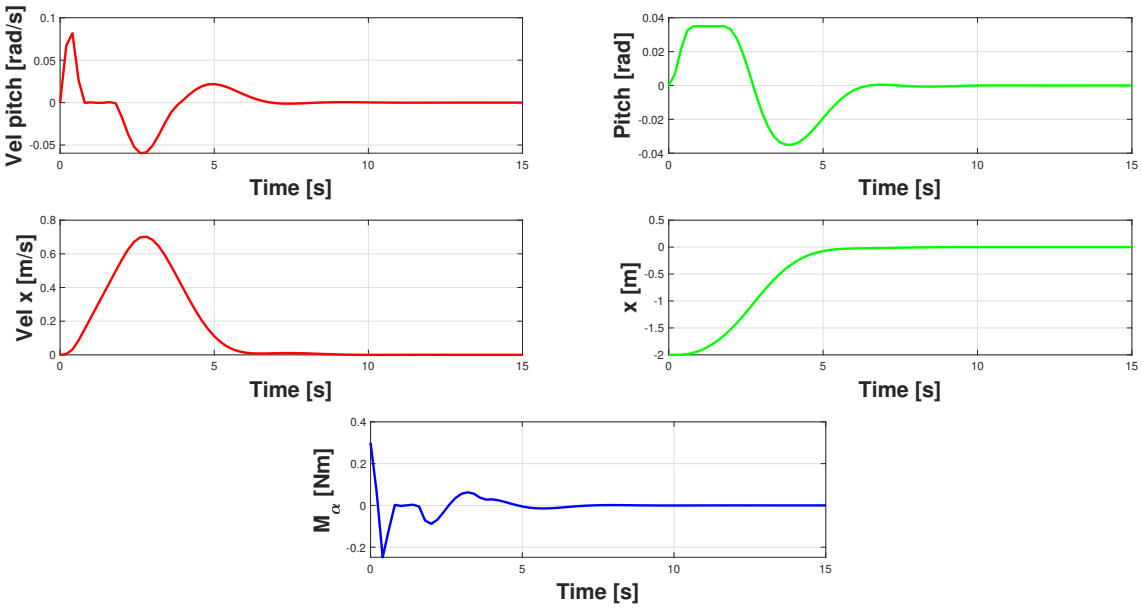

Figure 5: Regulation response of x dimension, starting from -2

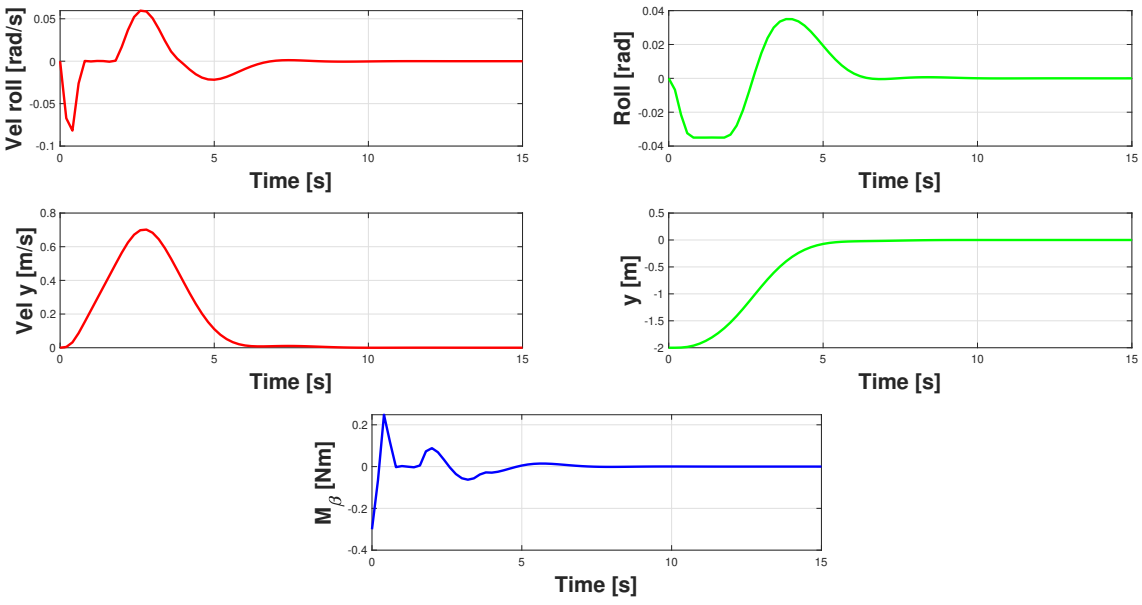

Figure 6: Regulation response of y dimension, starting from -2

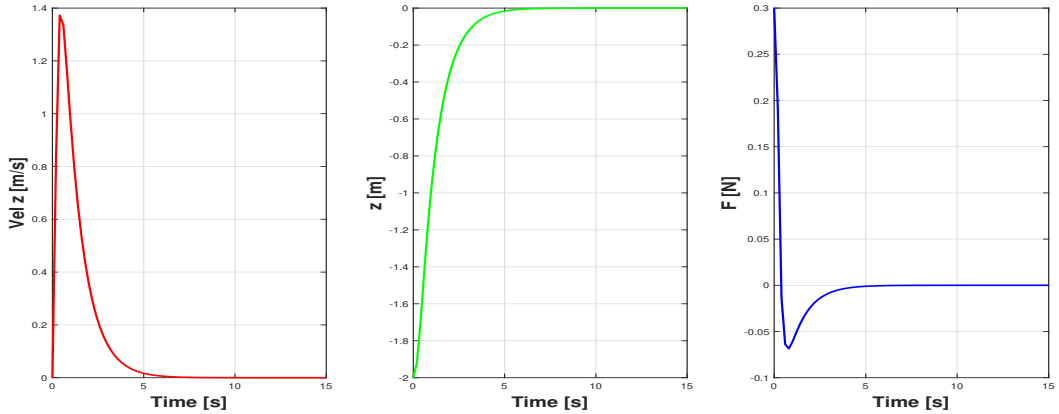

Figure 7: Regulation response of z dimension, starting from -2

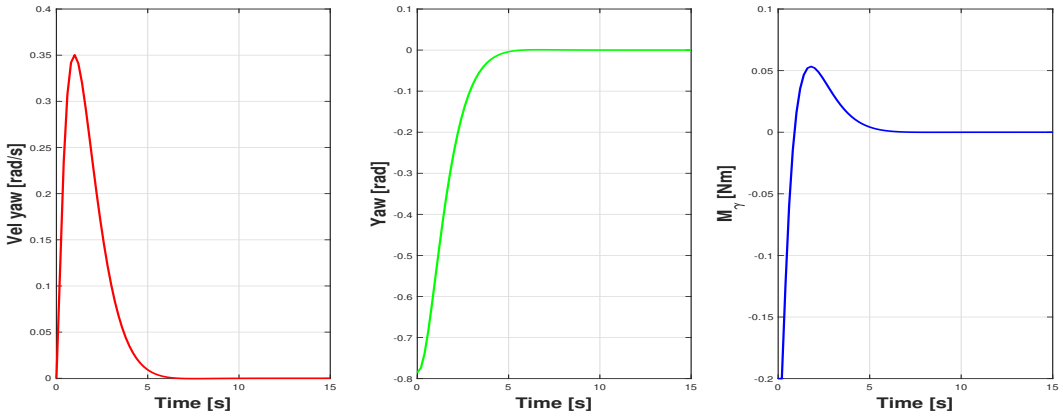

Figure 8: Regulation response of yaw dimension, starting from -$\frac{\pi}{4}$

Deliverable 3.2 : Design MPC tracking controllers

Design Procedure for tracking

Optimization problem is formulated for tracking the reference states (*ref*). The solution for Eq.6 is obtained using YALMIP for each of the controllers (x, y, z, yaw), implementation is provided with the Deliverable_3_2.zip.

$$\min_{u_s, x_s} \quad u_s^T R_s u_s \tag{6a}$$

$$\text{s.t} \quad x_s = Ax_s + Bu_s \tag{6b}$$

$$\texttt{ref} = Cx_s \tag{6c}$$

$$A_x x_s \leq b_x \tag{6d}$$

$$A_u u_s \leq b_u \tag{6e}$$

In the formulation given in Eqs.6,

- Parameters A, B, C correspond to respective discrete time state-space model matrices.
- x_s, u_s are steady state values for the state and control input respectively.
- Further, The tuning parameter in objective function (R_s) is set to identity matrix, without loss of generality.
- `ref` - Value of the parameter begin tracked.

Further, the constraint parameters (A_x, b_x, A_u, b_u) for each of the controllers are defined in Table 2 according to the given problem description.

Additionally, the MPC formulation defined in regulation is modified to include the steady states for tracking.

$$\min_{u} \quad \sum_{i=0}^{N-1} (x_i - x_s)^T Q (x_i - xs) + (u_i - u_s)^T R (u_i - us) + (x_N - xs)^T Q_f (x_N - x_s) \tag{7a}$$

$$\text{s.t} \quad x_{i+1} = Ax_i + Bu_i \tag{7b}$$

$$A_x x_i \leq b_x \tag{7c}$$

$$A_u u_i \leq b_u \tag{7d}$$

$$A_f (x_N - x_s) \leq b_f \tag{7e}$$

In the formulation given in equations (7),

- Parameters A, B correspond to respective discrete time state-space model matrices.
- x_s, u_s are steady state values for the state and control input respectively.
- Further, The tuning parameter in objective function (R) is set to be 5 times higher compared to matrix (Q) which is set to be identity matrix. The optimal weights are arrived at by trial and error to get best settling time response and terminal feasibility.
- Terminal constraint set (A_f, b_f) is obtained using LQR following the same methodology given in Equations 4.

Controller	A_x	b_x	A_u	b_u
MPC_x	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0.035 \\ 0.035 \end{bmatrix}$	$\begin{bmatrix} 1 \\ -1 \end{bmatrix}$	$\begin{bmatrix} 0.3 \\ 0.3 \end{bmatrix}$
MPC_y	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0.035 \\ 0.035 \end{bmatrix}$	$\begin{bmatrix} 1 \\ -1 \end{bmatrix}$	$\begin{bmatrix} 0.3 \\ 0.3 \end{bmatrix}$
MPC_z	-	-	$\begin{bmatrix} 1 \\ -1 \end{bmatrix}$	$\begin{bmatrix} 0.3 \\ 0.2 \end{bmatrix}$
MPC_yaw	-	-	$\begin{bmatrix} 1 \\ -1 \end{bmatrix}$	$\begin{bmatrix} 0.2 \\ 0.2 \end{bmatrix}$

Table 2: Constraint parameters for the tracking formulation

Tracking response plots

In this section we present the tracking result of each sub-system. To ensure ease of plotting and maintain consistency, the plotting function of quad was used with slight modification of passing the reference command while simulating.

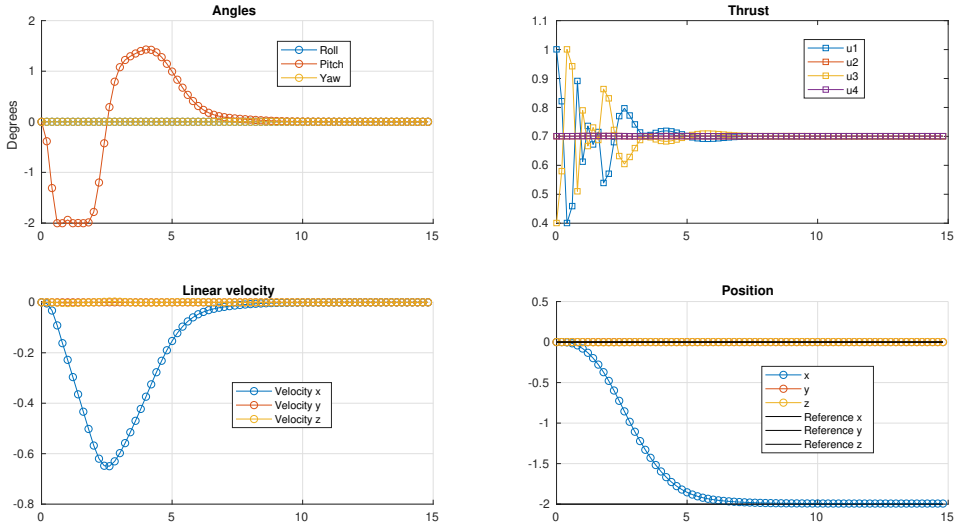

Figure 9: Tracking response of x dimension = -2, starting from origin

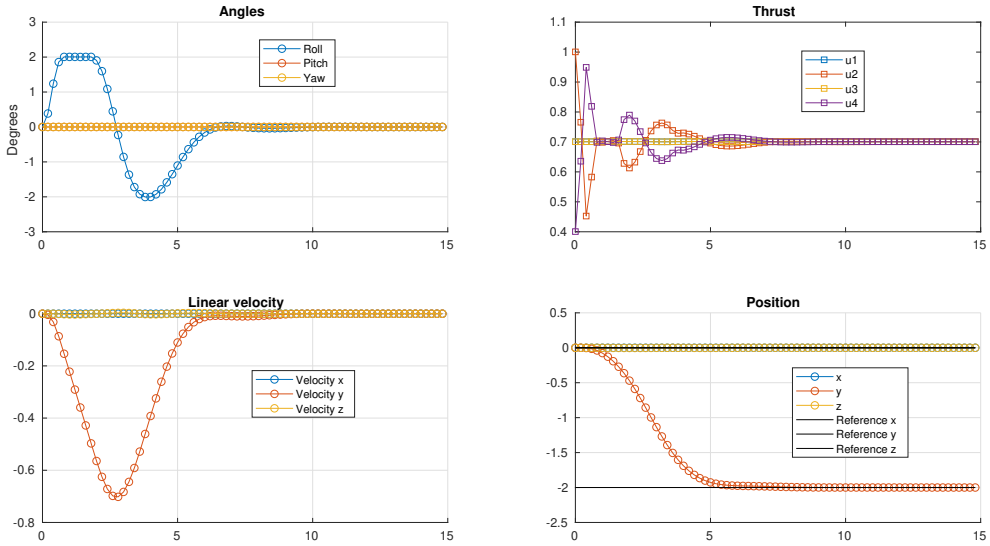

Figure 10: Tracking response of y dimension = -2, starting from origin

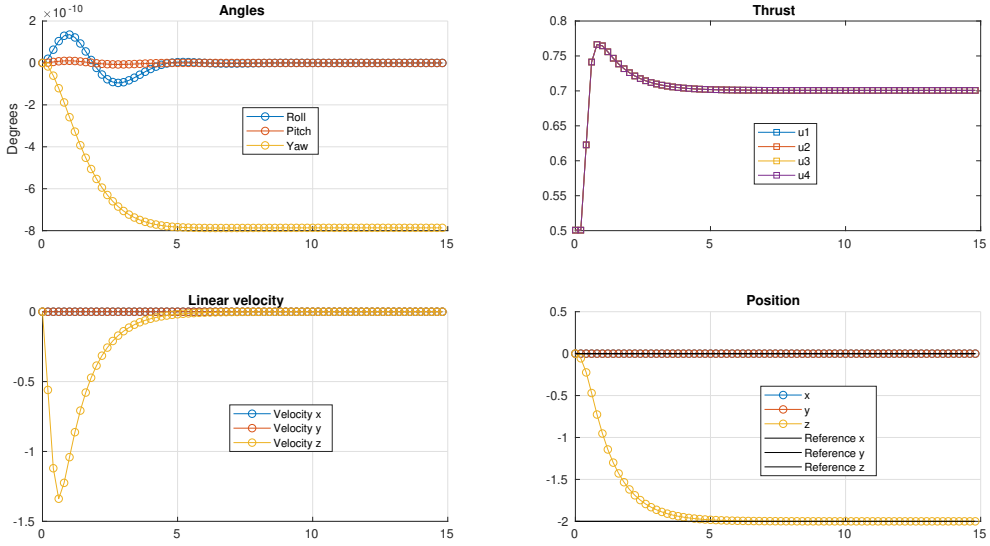

Figure 11: Tracking response of z dimension = -2, starting from origin

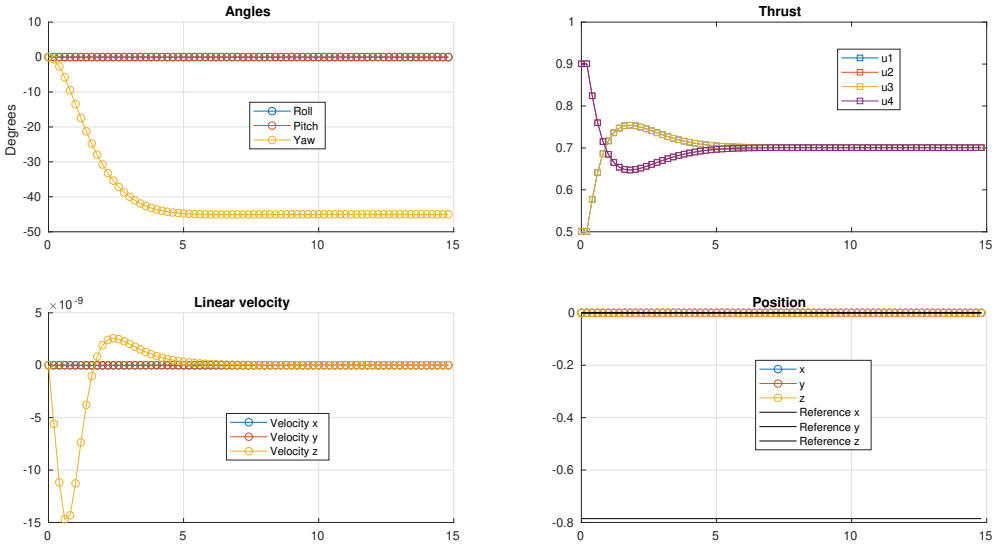

Figure 12: Tracking response of yaw angle = -$\frac{\pi}{4}$ rad, starting from origin

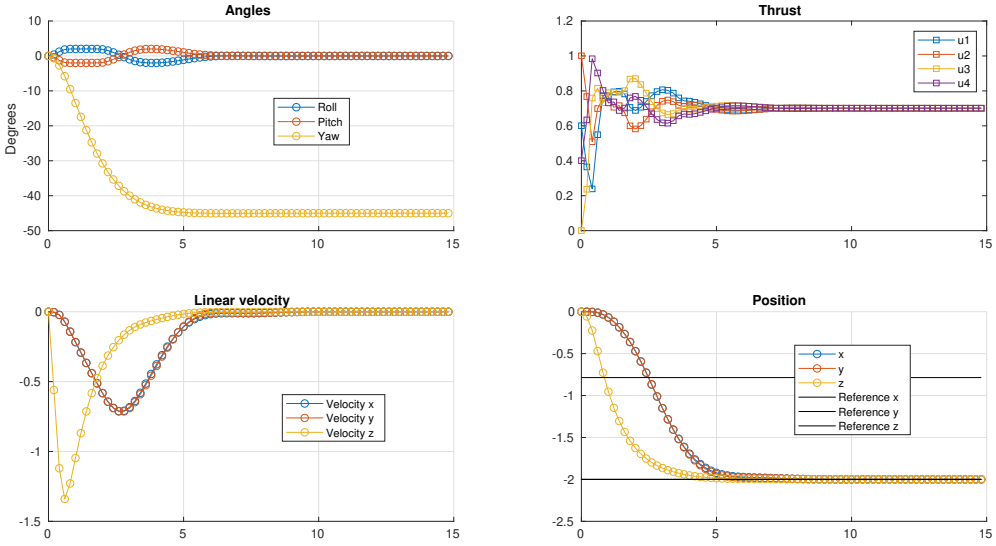

Figure 13: Tracking response of ref = (-2, -2, -2, -$\frac{\pi}{4}$), starting from origin

m-codes for the controllers are attached in Deliverable_3_2.zip. Main script for producing the plots is titled Deliverable_3_2.m

Deliverable 4.1

In this section, we present the plots of all the controllers simultaneously working to tracking a reference trajectory with $T_f = 40$. The m-code is provided in Deliverable_4_1.zip.

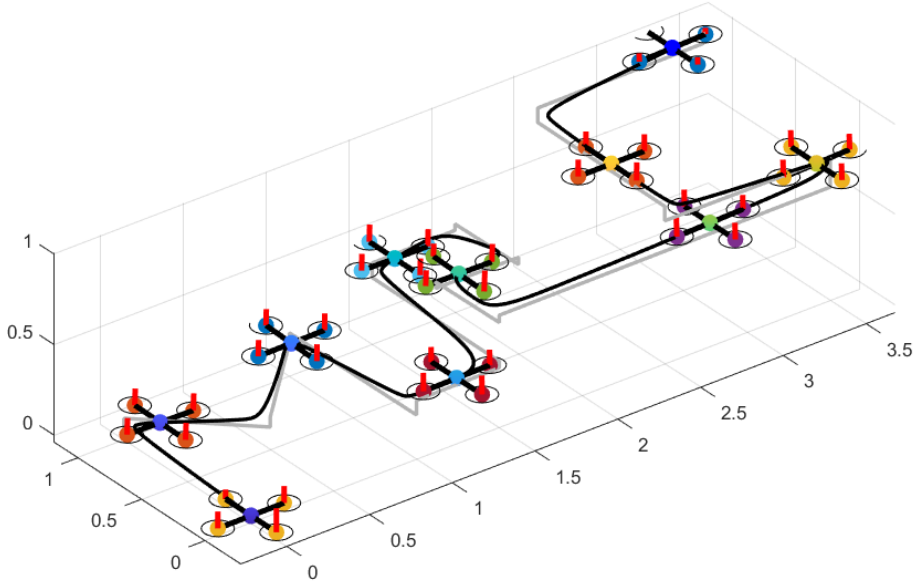

Figure 14: Orthographic view of the quadcopter tracking the *MPC* reference trajectory

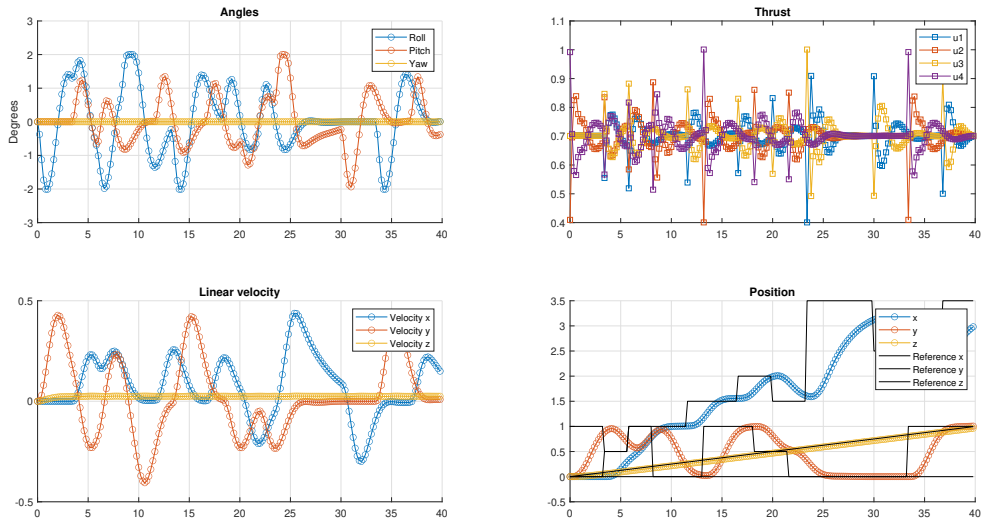

Figure 15: Response of all the sub-systems

Deliverable 5.1

Design Procedure for offset-free tracking

Controllers for x, y, yaw implemented in the section with Deliverable 3.2 are used as such in this section. However, z-controller problem formulation is updated to include the unknown disturbance. In both cases terminal set constraints are used. Further, the matrices Q and R are retained same as that of Deliverable 3.2. Updated formulation for the z-controller is given in equations (8, 9).

Z-controller formulation with offset

$$\min_{u} \quad \sum_{i=0}^{N-1}(x_i - x_s)^T Q(x_i - x_s) + (u_i - u_s)^T R(u_i - u_s) + (x_N - x_s)^T Q_f(x_N - x_s) \tag{8a}$$

$$\text{s.t} \quad x_{i+1} = Ax_i + Bu_i + B\mathbf{d}_i \tag{8b}$$

$$d_{i+1} = d_i \tag{8c}$$

$$A_u u_i \leq b_u \tag{8d}$$

$$A_f(x_f - x_s) \leq b_x \tag{8e}$$

$$\tag{8f}$$

In the formulation given in equations (8),

- Parameters A, B correspond to respective discrete time state-space model matrices.
- x_s, u_s are steady state values for the state and control input respectively. Further, d_s is the constant unknown disturbance.
- Further, The tuning parameter in objective function (R) is set to be 5 times higher compared to matrix (Q) which is set to be identity matrix. The optimal weights are arrived at by trial and error to get best settling time response and terminal feasibility.
- Terminal constraint set (A_f, b_f) is obtained using LQR following the same methodology given in Equations 4.

Z-steady state target formulation with offset

$$\min_{u_s, x_s} \quad u_s^T R u_s \tag{9a}$$

$$\text{s.t} \quad x_s \quad = Ax_s + Bu_s + B\mathbf{d}_s \tag{9b}$$

$$\texttt{ref} = Cx_s \tag{9c}$$

$$A_x x_s \leq b_x \tag{9d}$$

$$A_u u_s \leq b_u \tag{9e}$$

In the formulation given in equations (9),

- Parameters A, B, C correspond to respective discrete time state-space model matrices.
- x_s, u_s are steady state values for the state and control input respectively. Further, d_s is the constant unknown disturbance.

- Further, The tuning parameter in objective function (R) is set to identity matrix, without loss of generality.
- `ref` - Value of the parameter begin tracked.

Estimator for the states and disturbance

Based on the system dynamics given in equations (8), we design an estimator for the states and disturbance as follows.

$$\begin{bmatrix} \hat{x}_{k+1} \\ \hat{d}_{k+1} \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} + \begin{bmatrix} \hat{x}_k \\ \hat{d}_k \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k + \begin{bmatrix} L_x \\ L_d \end{bmatrix} (C\hat{x}_k - y_k) \tag{10a}$$

where the observer gain $L = [L_x; L_d]$ in the above formulation is selected to have stable estimator by placing the poles of the error dynamics within unit circle. Further, we place poles at $[0.4, 0.5, 0.6]$ to have stable fully observable estimator dynamics.

The formulation described in Equations (8, 9, 10) is implemented in the controller template m-code. Simulation results for a given $bias = -0.1$ in z-dynamics are shown in Figures (16, 17, 18), which confirm that the off-set free tracking is achieved and meets the specifications.

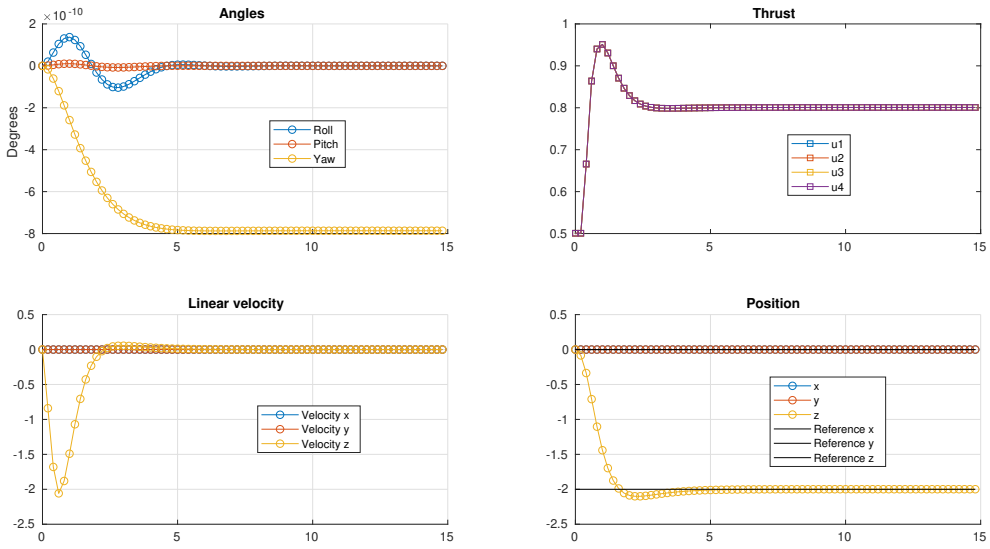

Figure 16: Tracking z=-2 with a z-disturbance of -0.1

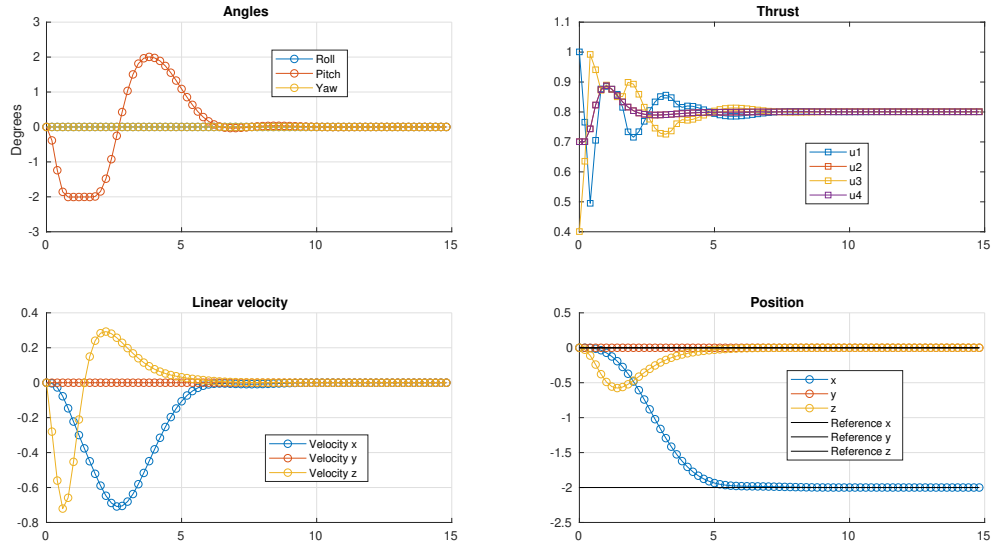

Figure 17: Tracking x=-2 with a z-disturbance of -0.1 in z-model

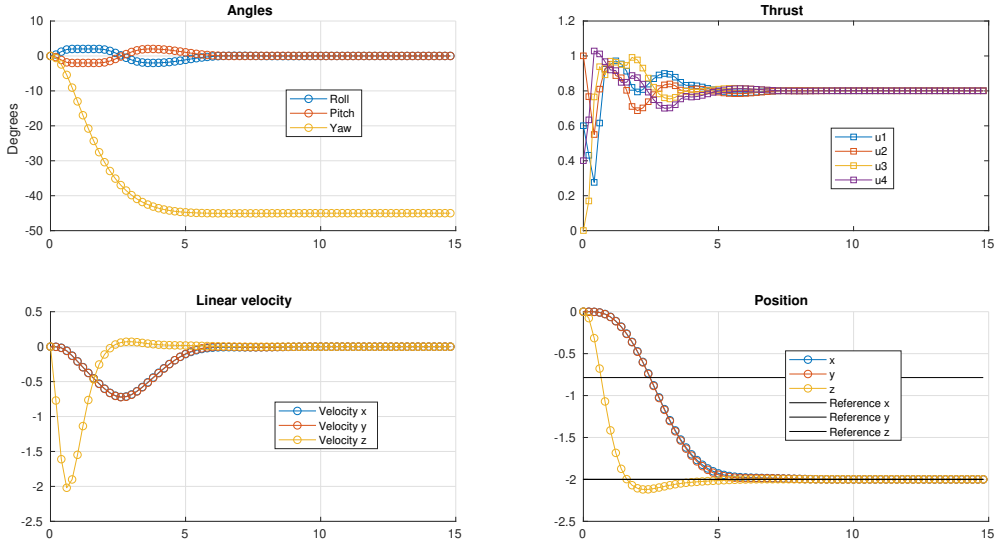

Figure 18: Tracking (x, y, z, yaw)=(-2, -2, -2, -π/4) with a z-disturbance of -0.1

Deliverable 6.1

Design Procedure for Nonlinear MPC

The nonlinear MPC can be formulated as follows

$$\min_{u} \quad \sum_{i=0}^{N-1} (x_i - ref)^T Q (x_i - ref) + u_i^T R u_i \tag{11a}$$

$$\text{s.t} \quad x_{i+1} = f_{discrete}(x_i, u_i) \tag{11b}$$

$$A_x x_i \leq b_x \tag{11c}$$

$$A_u u_i \leq b_u \tag{11d}$$

The discretization of the nonlinear quad dynamics $\dot{x} = f(x, u)$ was done using Runga Kutta-4 with h=0.2. This was wrapped inside the $f_diiscrete$.

$$\dot{x} = f(x, u) \tag{12a}$$

$$x_{k+1} = x_k + h(\frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6}) \tag{12b}$$

$$k_1 = f(x_k, u_k) \tag{12c}$$

$$k_2 = f(x_k + \frac{h}{2} k_1, u_k) \tag{12d}$$

$$k_3 = f(x_k + \frac{h}{2} k_2, u_k) \tag{12e}$$

$$k_4 = f(x_k + h k_3, u_k) \tag{12f}$$

CASADI was utilised to implement the above formulation using the template provided with the project description. The tuning parameters were taken as $Q = 5I$, $R = I$, $N = 50$.

Performance of Nonlinear MPC

We can observe that the performance of non-linear controller in Fig.19 is faster, the inputs are smooth function which go to steady value much earlier compared to MPC controller in Deliverable 3.2 shown in Fig.9. This may be due to the fact that the optimization problem has better information of the gradients and cost projections in future as the dynamics is not linearized. Additionally, one can also say that in when the system was decomposed into 4 parts, the authority of the input constraint was equally distributed in 4 sub domains, making the optimization problem much restricted. Whereas in nonlinear MPC, the optimization has much larger input set to utilise.

Plots showing the performance of nonlinear controller

Non-linear formulation given in Deliverable 6.1 is programmed in MATLAB, the MPC problem is solved for tracking a given input in x =-2 so that we can compare the results with MPC

controller obtained in Deliverable 3.2. Figure 19 give the tracking response of the non-linear MPC controller. Since the terminal constraints are dropped in the present MPC formulation, the MPC horizon $N = 50$ is selected to ensure the feasibility of the problem.

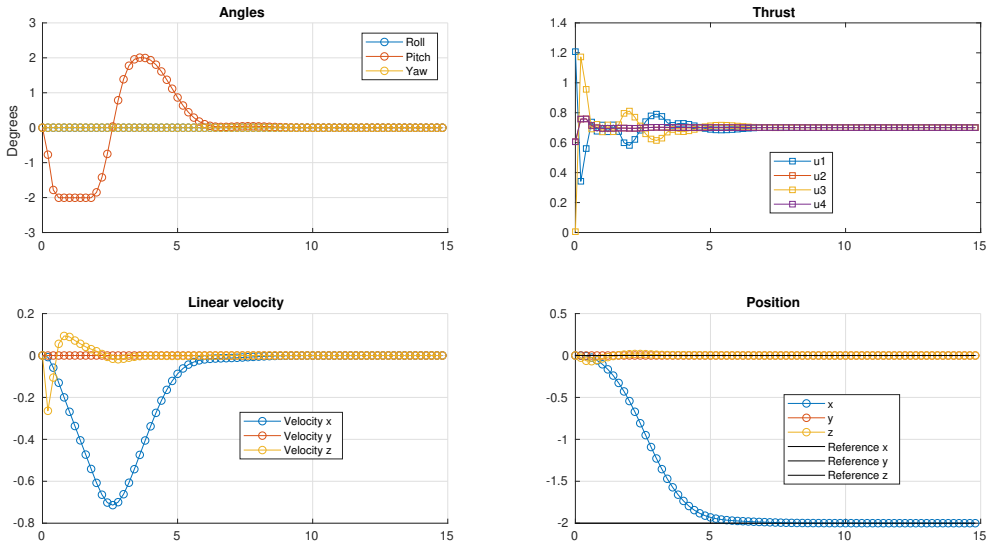

Figure 19: Tracking (x, y, z, yaw)=(-2, 0, 0, 0) using non-linear MPC

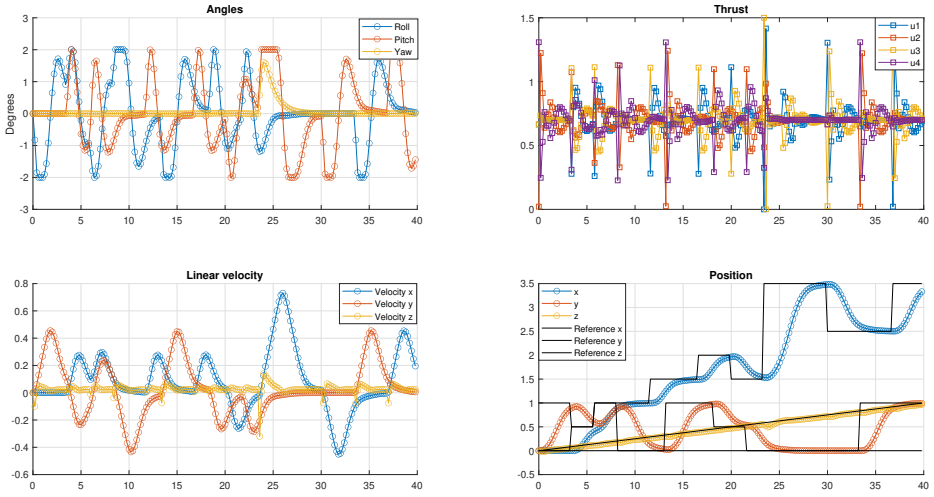

Figure 20: Response of nonlinear controller in *MPC* tracking

The code for generating the NMPC controller and plots is given in Deliverable_6_1.zip which can be executed using Deliverable_6_1.m script. For Fig.19, one has to provide user defined value in the `ref` variable. If kept empty, it will follow the inbuilt 'MPC' trajectory.