Control of Aircraft Landing using MPC

Akshat Srivastava[1] and Shuhul Razdan[1]

***Abstract*— Aircraft jet landing has been studied extensively for performing optimized and accurate maneuvers. These conditions are of great value for air-coast and navy defence where the runway is short and thus landing within a restricted set of conditions is required. Some scenario also require a constant height or keeping a constant distance from refuelling planes. In this article, we attempt to pursue the problem of optimum control with MPC tools and attempt to highlight the differences obtained.**

I. INTRODUCTION

The project follows the guidelines presented in the literature regarding optimal control of aircraft landing, which pursues on designing different controllers synthesized by selecting different functional forms for the weighting factors appearing in the error index formulated from the performance requirements [1]. In this article, attempt is made to utilize the same linearized controller and implement finite and infinite horizon problem with various practical constraints. The latter would make use of LQR design succeeded by the MPC controller.

A. Dynamics of the Plant

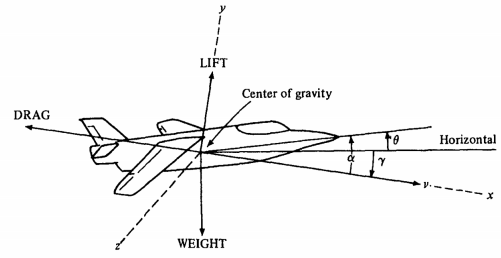

Fig. 1. Cordinate Frame of aircraft

We consider the vertical landing of the aircraft. The x-direction is along the flight path of the aircraft with α, θ and γ denoting the angle of attack, pitch angle and gliding path angle respectively. The model simplification is achieved by ignoring the influence of other axial (both y and z) dynamics and their co-dependence on the concerned axis, except for the angles involved, stated already. Random disturbances are not incorporated into the linearized model. Additionally, the gliding angle and velocity of the aircraft should be constraint to avoid the non-linearities of the system from corrupting the analysis. Additional information about the derivation of the plant could be found in the stated literature [1]. The resulting model dynamics are given by equation (1).

$$\begin{aligned}\dot{x}_1(t) &= x_2(t)\\ \dot{x}_2(t) &= a_{22}x_2(t) + a_{23}x_3(t)\\ \dot{x}_3(t) &= x_4(t)\\ \dot{x}_4(t) &= a_{42}x_2(t) + a_{43}x_3(t) + a_{44}x_4(t) + b_4u(t)\end{aligned} \tag{1}$$

Resulting in the following state space model.

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t) \quad \text{, where} \tag{2}$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & a_{22} & a_{23} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & a_{42} & a_{43} & a_{44} \end{bmatrix}; \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ b_4 \end{bmatrix}$$

$$x_1(t) = h(t), \quad x_2(t) = \dot{h}(t), \quad x_3(t) = \theta(t), \quad x_4(t) = \dot{\theta}(t)$$

The system states include the altitude above the flight deck h, altitude rate $\dot{h}$, pitch angle θ, and pitch rate $\dot{\theta}$. Each states could be measured by a suitable sensor for example, h can be measured by a radar altimeter, $\dot{h}$ by a barometric rate meter, θ and $\dot{\theta}$ by gyros. The longitudinal motion of the aircraft is controlled entirely by the elevator deflection angle $u(t)$. Each of the constants for matrix equation (2) are also known.

B. Desired Behaviour of the plant

The aircraft landing is practically feasible under certain intuitive conditions. Let's take the pitch angle for example, which should always be greater than 0° to perform a smooth landing. But this is much more important near the touchdown rather than the entire considered course of landing from a certain height. This constraint is important since to increase the landing velocity above the nominal falling off velocity, the aircraft would require to have a negative pitch angle, since for a plane with fixed thrusters, the pitch angle determines the direction of thrust.

Later on, throughout the course of implementation, complex coupling among different states are realized. This further restricts proper tracking of different states thus, our attention would be restricted to the vertical height, the landing velocity and the terminal pitch angle. It should also be emphasized that some constraints which cause the system to behave non-linearly are relaxed, because of the infeasibility produced by the optimizer/solver. For the landing problem, we would like our velocity to be of the profile shown in Figure 2, in which case, the altitude decrease would almost follow a constant slope, resulting in approximate runway of 1570ft, as shown. Furthermore, for reference tracking, we would like our jet aircraft to simulate

refuelling from a plane overhead. The desired profile for such a task is shown in Figure 3.

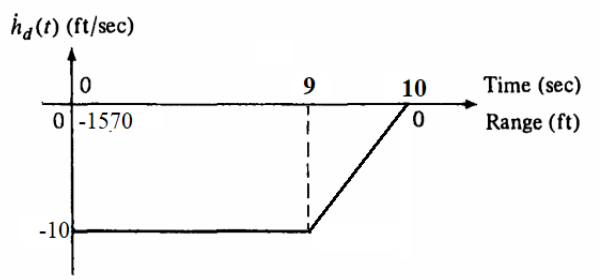

Fig. 2. Desired Landing Velocity profile

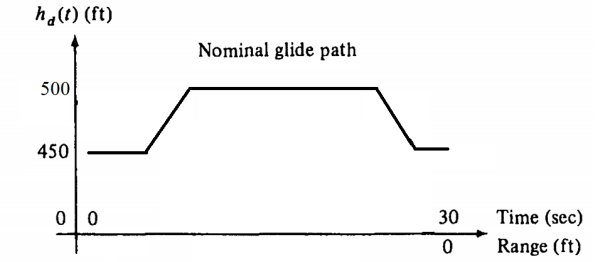

Fig. 3. Desired altitude profile for refuelling

II. MPC Formulation

The MPC is devised such as to perform the following maneuvers.

- Executing the landing with and without disturbance.
- Executing the refuelling maneuver.
- Maintaining constant trajectory with Disturbance rejection and state observer

The cost function used throughout the analysis is given as follows

$$J(x_0, u) = \sum_{k=0}^{N-1} \ell(x(k), u(k)) + V_f(x(N)) \quad \text{s.t. } u \in \mathbb{U}, x \in \mathbb{X},\ x(N) \in \mathbb{X}_f. \tag{3}$$

where $\mathbb{U}$ and $\mathbb{X}$ represent the admissible control input and state set defined by (8) and (9) respectively, x_0 the initial state, $\ell(\cdot)$ the stage cost and V_f the terminal cost defined as follows.

$$\ell(x(k), u(k)) = x(k)^T Q x(k) + u(k)^T R u(k)$$
$$V_f(x(N)) = x(N)^T P x(N) \tag{4}$$

where $Q, R \succ 0$, opted suitably for desired performance. To solve for the constraints discussed in the following subsection, equation (4) is compactly rewritten as matrix form in terms of only the control input using the dynamic recursion (10).

$$\boldsymbol{x}_{N+1} = T x_0 + S \boldsymbol{u}_N \tag{5}$$

where $\boldsymbol{x}_{N+1} \in \mathbb{R}^{Nn \times 1}$, T is the matrix containing ascending power of state transition matrix A, and S is the repeated controllability matrix upto the power $N-1$ as shown below.

$$T = \begin{bmatrix} A^0 \\ A \\ A^2 \\ . \\ . \\ . \\ A^N \end{bmatrix} \qquad S = \begin{bmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ B & 0 & 0 & 0 & \dots & 0 \\ AB & B & 0 & 0 & \dots & 0 \\ A^2B & AB & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & & \vdots \\ A^{N-1}B & A^{N-2}B & A^{N-3}B & \dots & & B \end{bmatrix} \tag{6}$$

By substituting the equation (5) into (4), ,taking care of dimensions, for e.g. reforming Q and R into block diagonal matrices $\boldsymbol{Q}_{N-1}$ and $\boldsymbol{R}_{N-1}$ respectively, and ignoring all the terms independent of $\boldsymbol{u}_N$, we attain a the standard objective function for quadratic optimization.

$$V_N(\boldsymbol{u}_N) = \frac{1}{2} \boldsymbol{u}_N^\top H \boldsymbol{u}_N + h^\top \boldsymbol{u}_N$$
$$\text{where} \quad H = (\boldsymbol{R}_{N-1} + S^T \bar{Q} S),$$
$$h = S^T \bar{Q} T x_0,$$
$$\bar{Q} = \begin{bmatrix} \boldsymbol{Q}_{N-1} & 0 \\ 0 & P \end{bmatrix} \tag{7}$$

$\bar{Q}$ is formed such as to combine the terminal state and preceding states into single vector and therefore produce a cumulative matrix equation in terms of only control input $\boldsymbol{u}_N$.

A. Regulation MPC

We begin by analysing a system with the assumption that all states are measured and available, as described in section I. The system has been further dicretized using a sampling time of $0.1s$ with first-order hold approximation. The costs-in a preliminary case-where bounded constraints are impressed upon the control input, is taken as $Q = 10^4 \mathbf{I}_4$. This sort of high value is required for the feasibility of the optimizer. Choosing $R = 1$ delivers suitable results as well. The terminal cost is chosen as the optimal cost of the unconstrained infinite horizon LQR problem. The motivation for the latter is elaborated in section III. More discussion on the results and different weights can be found in section IV. The control horizon is taken as $1.5s$ (N=15) and a total duration of $30s$. Such settings enables low computational complexity and well as allow more space for experimentation with different terminal constraints.

The control input, i.e. elevator angle is used as the sole input, keeping thrust and other parameters constant. This is often so the case in aircraft landing, where other flaps including aileron, spoiler and winglet are actuated using hydraulic pistons, visible clearly from passenger window near the wings. The actuation of elevator is restricted by mechanical structure and the piston capabilities. In our investigation, we assume the angle of elevator to be bounded by $35°$. The input constraints $u \in \mathbb{U}$ in equation (3) are

therefore defined as

$$|u(i)| \leq T_{\max} = 35^\circ \quad \forall i \in [0, .., N-1] \tag{8}$$

It should be noted that the algorithm utilizes radians and thus necessary conversion is carried out implicitly. As stated previously in the section I, certain assumptions have been made to produce a linear estimate of the system, (2). To retain linearity, we impose state constraints on those states which are most likely to produce significant divergence to non-linear behaviour. The altitude rate is bounded by $20ft/s$ to avoid complex aerodynamic effects. An obvious minimum horizontal height constraint is also placed on the altitude. In order to preserve the feasibility of the solver, the nose tilt angle is bounded within 180° and it's respective angular velocity is left unbounded. The collective constraints are given as follows

$$\begin{aligned} & x_1(k) \geq 0 \text{ ft} \\ & |x_2(k)| \leq 20 \text{ ft/s} \quad , k \in [1, \ldots, N-1] \\ & |x_3(k)| \leq 180^\circ \end{aligned} \tag{9}$$

It is quite apparent that constraints discussed above require to be expressed in terms of control input $u_i(k)$ since optimization is with respect to u only. To do so, we utilize the dynamic recursion obtained through the discrete representation of (2).

$$x(k) = A^k x_0 + \sum_{j=0}^{k-1} A^{k-j-1} B u(j) \tag{10}$$

This leaves us with terms containing only $x_0 = [450 \text{ ft}, 0, 0, 0]$ and $u(j)$. To produce the standard form of constraint, i.e. $Au_N \leq b$ for the quadratic objective function, the above expression is re-written as follows.

$$\begin{aligned} & |Px_0 + Au_N| \leq x_{\text{limit}} \\ \Longrightarrow & \begin{cases} Au_N \leq -Px_0 + x_{\text{limit}} \\ Au_N \geq -Px_0 - x_{\text{limit}} \end{cases} \end{aligned} \tag{11}$$

with $b = -Px_0 - x_{\text{limit}}$ and x_{limit} chosen suitably to represent (9).

B. Reference Tracking using MPC

What we aim to achieve is a trajectory similar to the one shown in Figure 3 with its respective interpretation specified in section I. This is achieved by obtaining the optimum reference states evaluated for each iterating horizon. This is not necessarily the same as the reference trajectory, and to avoid confusion, is termed as optimum target (i.e. x_{ref} and u_{ref}), with the algorithm used to evaluate the latter called optimum target selection (OTS).

$$(x_{\text{ref}}, u_{\text{ref}})\,(\hat{d}, y_{\text{ref}}) \in \begin{cases} \arg\min_{x_r, u_r} J(x_r, u_r) \\ \text{s.t.} \begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_r \\ u_r \end{bmatrix} = \begin{bmatrix} 0 \\ y_{\text{ref}} \end{bmatrix} \\ (x_r, u_r) \in \mathbb{Z} \end{cases} \tag{12}$$

After obtaining the optimal trajectories, the stage cost and the terminal cost has to be reformulated such that, we now have to penalize the difference between the current state and our desired state, i.e. x_{ref} and u_{ref}.

$$\begin{aligned} \ell(x(k), u(k)) &= (x(k) - x_{\text{ref}})^T Q (x(k) - x_{\text{ref}}) \\ &\quad + (u(k) - u_{\text{ref}})^T R (u(k) - u_{\text{ref}}) \\ V_f(x(N)) &= (x(N) - x_{\text{ref}})^T P (x(N) - x_{\text{ref}}) \end{aligned} \tag{13}$$

The above equation is formulated into the matrix form, similar to (14).

$$\begin{aligned} V_N(\boldsymbol{u}_N) &= \frac{1}{2} \boldsymbol{u}_N^\top H \boldsymbol{u}_N + \boldsymbol{u}_N^\top [h_{ur} \;\; h_{xr} \;\; h_{x0}] \\ \text{where} \quad H &= (R + S^T \bar{Q} S), \\ h_{x0} &= S^T \bar{Q} T, \\ h_{xr} &= S^T \bar{Q} \boldsymbol{X}_{ref}, \\ h_{ur} &= -R \boldsymbol{U}_{ref} \end{aligned} \tag{14}$$

Few subtleties have to be taken care of. For example, x_{ref} and u_{ref} must be resized into a lengthy vector $\boldsymbol{X}_{ref}$ and $\boldsymbol{U}_{ref}$ respectively such as to match the dimensions. Following a similar procedure as described for the Regulation MPC design, the cost weighting matrices were opted as.

$$Q_{\text{Land}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 10^8 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad Q_{\text{Refuel}} = \begin{bmatrix} 10^7 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

with $R = 1$. Finally, the prediction horizon N was selected to be 15 with a sampling time of 0.1 seconds.

C. Disturbance Rejection MPC

In real systems, the assumption of having full state information at our disposal is uncommon. Such situations also arise in case of faulty sensor in which, a virtual sensor (observe/estimator) is utilized. To replicate this scenario in our aircraft model, we modify our output equation in (2) such that only altitude information and nose tilt angle is available for analysis. To make it more practical, we can also implement a constant disturbance in the process and the output sensor measurement equation (2) resulting in the following augmented discretized state space model.

$$\begin{aligned} \begin{bmatrix} x(k+1) \\ d(k+1) \end{bmatrix} &= \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} x(k) \\ d(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u \\ y &= \begin{bmatrix} C & C_d \end{bmatrix} \begin{bmatrix} x(k) \\ d(k) \end{bmatrix} \end{aligned} \tag{16}$$

where the matrices B_d and C_d are used to represent the affect of a constant disturbance on the x and y states. In order to ensure the augmented system is observable, it is required that the original (A, C) system is observable as well as the following matrix is detectable.

$$\left[\begin{pmatrix} A & B_d \\ 0 & I \end{pmatrix}, \begin{pmatrix} C & C_d \end{pmatrix} \right] \tag{17}$$

With the above assumptions, an observer could be implemented by opting a suitable Luenberger gain matrix

such that the estimations asymptotically converge to the actual states. It is empirically discovered that the convergence should be as smooth as possible, which in turn required experimenting with different poles to obtain the most suitable one.

The OTS formulation for creating offset free MPC controller is shown in (18)

$$(x_{\text{ref}}\,, u_{\text{ref}}\,)\,(\hat{d}, y_{\text{ref}}\,) \in \begin{cases} \arg\min_{x_r,u_r} J\,(x_r, u_r) \\ \text{s.t.} \begin{bmatrix} I-A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_r \\ u_r \end{bmatrix} = \begin{bmatrix} B_d\hat{d} \\ y_{\text{ref}} - C_d\hat{d} \end{bmatrix} \\ (x_r, u_r) \in \mathbb{Z} \\ Cx_r + d \in \mathbb{Y} \end{cases} \tag{18}$$

Although we follow most of the stated formulations strictly in our algorithm, some modifications had to be made for proper execution of the optimization problem. For example, while implementing OTS, a different measurement dynamic matrix (C in (2)) had to be utilized, such that (12) is not over-constrained. If we were to use full state information, without disturbance to execute reference tracking (and that too with small enough horizon), our OTS problem becomes infeasible because many of the arguments can be independently determined, simply by using the equality constraints.

We came across two ways to deal with the stated problem. One is, as stated above, to modify our C matrix. This translates to focusing on reference tracking only for particular states, rather than attempting to track all of them. Another way is to increase our horizon, such that as few as possible constraints are imparted on the arguments to be determined in the optimization problem. Since the latter introduces heavy computational burden on our algorithm run, we pursue the former. This is the reason why in section I, we have only attempted to track either the altitude rate, or the altitude itself. Terminal constraints are implemented as inequality constraints which make the final trajectory lie within the set $\mathbb{X}_f$.

III. Stability Analysis

The stability analysis is more in form of a proof about convergence of MPC controller. In principle, we utilize closed lyapunov function's (CLF) decrease to establish the stated. Our CLF is the optimal cost function, $V_N^0(\cdot)$. This is closely linked with the LQR controller primarily due to terminal cost of MPC and that of infinite horizon LQR being the same, i.e. $\frac{1}{2}x^TPx$, where P is the solution to DARE [3, Pg 19]. Within the set $\mathbb{X}_f$, the MPC constrained problem solution coincides with the infinite horizon unconstrained LQR input. (19).

$$u(t) = Kx(t) \tag{19}$$

We know that our system at hand is controllable which implies that origin is GAS for $x^+ = (A+BK)x$. But in the tracking problem formulation, we require that the stability is defined with respect to x_{ref}, obtained from the OTS. By performing a shift of coordinates [4, Pg 107,54], one could transform $x_{ref} \to 0$, but this will consequently change the state and input constraints as well and it becomes necessary to also transform $u_{ref} \to 0$. With this, the problem would boil down to regulation to the origin. Let us now gain some insight about each assumption, see how they are satisfied and use them for establishing standard properties of CLF (2.14), (2.15) and (2.16) in the reference text [3].

The motivation for the subsequent theorems stems from the fact that not every constraint optimization problem has a solution [3, Pg 97]. To be able to ensure that our chosen CLF exist (note that our CLF is the solution of optimization problem), we need that our cost to be continuous.

Assumption 2.2 Since we have an hurwitz LTI system, $f(0,0) = 0$ and continuous. It is also intuitive to see that with $\ell(\cdot)$ defined in (4) and respective weights being non-zero, $\ell \in \mathbb{R}_+, \ell(0,0) = 0$. The terminal cost, $V_f = x^TPx$ with $P \succ 0$ satisfies $V_f(0) = 0, V_f(0) \in \mathbb{R}_+$.

Also, we would also like to ensure that our constraint set is compact. Another way to look at this assumption is to provide BIBO stability, i.e. for a constraint set of input $\mathbb{U}$, we would obtain a compact (or closed) set of states $\mathbb{X}$ and hence the CLF does not blow up, or equivalently, our optimum cost is finite.

Assumption 2.3 The constraints defined in (8) and (9) are compact, closed and contain the origin. Furthermore, the set $\mathbb{X}_f$ obtained from the algorithm [2] is compact as well. This is quite intuitive since the resulting region is itself given in the form $Ax \leq b$ with b being finite.

Another theorem which follows deserves particular attention because it is used to induce lower bounds to our CLF. Let us first state and then discuss how it achieves it.

Assumption 2.14(a)The algorithm used to obtain the set $\mathbb{X}_f$ ensures that the set is invariant. This invariance in-turn provides that $\exists u$ such that if $(x,u) \in \mathbb{X}_f \cup \mathbb{U}$, then $f(x,u) \in \mathbb{X}_f$. Next we ought to show that $V_f(\cdot)$ is decreasing. This we do using the infinite horizon LQR control law (19). One intuitive reason behind it is that since we are in $\mathbb{X}_f$, we cannot get out of it, and thus the constraint of MPC could be dropped. Then the resulting problem resembles the infinite horizon LQR problem. The elaborate proof is stated in the reference [5, slide 7] with following conclusion.

$$V_f(x^+) \leq V_f(x) - \ell(x,u), \quad \forall x \in \mathbb{X}_f \tag{20}$$

This conclusion is also confirmed through numerical simulation, the result of which is displayed in Figure 4.

Assumption 2.14(b) $\ell(x,u)$ is a function of PSD matrix Q and R, which are both taken as factors of identity (i.e. $\alpha\mathbf{I}_4$ and $\beta\mathbf{I}_1$ respectively). Thus, a suitable $\mathcal{K}_\infty$ function to lower bound $\ell(\cdot)$ could be $\frac{1}{2}\alpha||x||$. To upper bound $V_f(\cdot)$, we utilize a known fact for PSD matrices. Their respective quadratic expressions are always bounded by the multiples of their smallest and the largest eigenvalues, i.e.

$$\lambda_{min}||x||_P \leq ||x||_P \leq \lambda_{max}||x||_P$$

thus the suitable $\mathcal{K}_\infty$ function to upper bound the terminal cost is $\lambda_{max}||x||_P$.

With all the assumptions satisfied, we now show that our CLF satisfies property (2.14), (2.15) and (2.16) in brief.

- Through DP recursion, we can state that

  $$V_N^0(x) = \min_{u\in\mathbb{U}}\{\ell(x,u) + V_{N-1}^0(f(x,u))\} \quad \forall x \in \mathcal{X}_{N-1}$$

  which implies that,

  $$V_N^0(x) \geq \ell(x, \kappa_N(x))$$

  where $\kappa_N(x)$ is the optimal control sequence. From above, we understand that to lower bound $V_N^0(\cdot)$, we need to lower bound $\ell(\cdot)$ which is accomplished through Assumption 2.14(b).
- Given the monotonicity property of optimal value function holds, we would have

  $$V_{j+1}(\cdot) \leq V_j(\cdot) \quad \forall x \in \mathcal{X}_j$$

  This property states something quite obvious, i.e. for a larger horizon of concern - $(j+1)$ as compared to j - the optimal cost would be smaller for the former. Think in terms of covering a 7ft by foot. More energy would be expended in covering it under 4 steps, than covering it under 10. Energy is analogous to optimal cost. This could be easily extended (and thus forming **Proposition 2.18)** to,

  $$V_N^0(\cdot) \leq V_f(\cdot)$$

  In which case, an upper bound on $V_f(\cdot)$ would imply an upper bound on our CLF, which is also achieved using Assumption 2.14(b). Furthermore Assumption 2.3 also indicates a finite value of our CLF.
- Lyapunov decrease is shown explicitly in the referenced text [3, Pg 116].

The above remarks could also be traced as conclusions of **Theorem 2.19**, which additionally states that the origin is asymptotically stable in $\mathcal{X}_N$. This is also an important conclusion since, for a lyapunov function, we also require that

$$V_N^0(x^+) - V_N^0(x) + \ell(x, \kappa_N(x)) \leq 0 \quad \forall x \in \mathcal{X}_N \backslash \{x_e\}$$

From the aforementioned theorem, we know that stable origin is equilibrium and thus $x_e = \{0\}$, which fits perfectly! What remins to complete the proof is to evaluate the maximum invariant set, $\mathcal{X}_N$ for which the MPC exist and the state could be driven to $\mathbb{X}_f$.

A. Terminal Set, $\mathbb{X}_f$

We determine here the invariant set $\mathbb{X}_f$ within which, the MPC coincides with the LQR problem having the control input (19), where K is the optimal LQR gain. The algorithm discussed in [2] is utilized to obtain a set of 96 linear inequalities describing the region $\mathbb{X}_f$. The basic principle behind this algorithm is empirically generating different set for which the control law is able to drive the state to origin and then taking the union of all the set obtained. Since we are dealing with $\mathbb{R}^4$, it is not possible to show exactly how the regions formed look like.

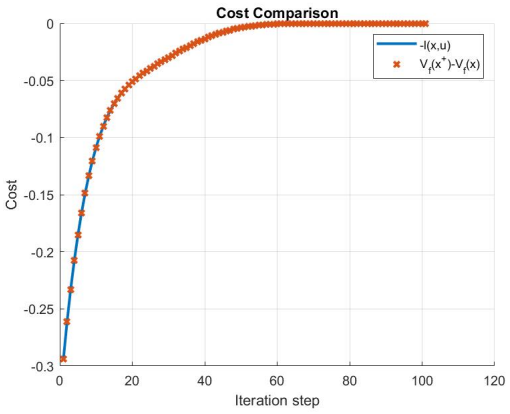

Fig. 4. Cost Decrease for initial point on $\mathbb{X}_f$

B. Maximum Invariant Set, $\mathcal{X}_N$

Our system is controllable, as stated previously. Hypothetically then, if the control input constraints were not applicable, the whole $\mathbb{R}^n$ would be our $\mathcal{X}_N$ since controllability allows us to drive our state anywhere we desire within N steps. But due to the constraint (8), the region described by $\mathcal{X}_N$ would take a more conservative shape altogether. With a similar algorithm described by [2] and instruction from exercise 4 of the assisted lecture in MPC course, a large set of linear inequalities are obtained, describing the region $\mathcal{X}_N$.

IV. Numerical Simulation

In this section, we show some of the results obtained by varying parameters in the algorithm such that the target trajectories are as close to the ones discussed in section I. Some deviations are encountered and discussed upon as well.

A. Prediction Horizon, N

The Figure 5 shows the response of altitude for different horizon. For small enough horizon, e.g. $N = 3$, the system is unstable. For larger horizon, the system seems to stabilize suitably. This indicates the how horizon effects the span of set $\mathcal{X}_N$, which in turn dictates the stability of the system itself.

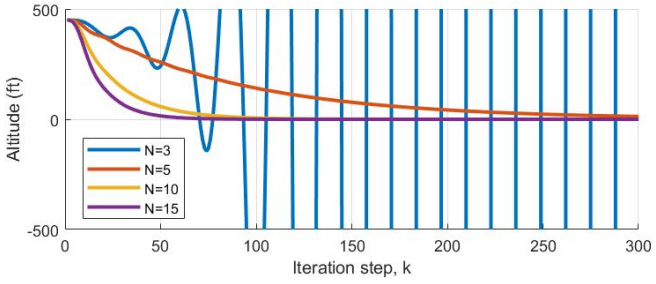

Fig. 5. Altitude for different prediction horizon,$Q = 0.1\mathbf{I}_4$

B. LQR and MPC

In this section, we explore the performance of LQR and MPC for simple regulation case. In the first case, we initialize our system outside the set $\mathbb{X}_f$. Both the problems are constrained by the input only, i.e. (8) holds. The input cost weight, R of the LQR problems was adjusted such as to accommodate the stated constraint.

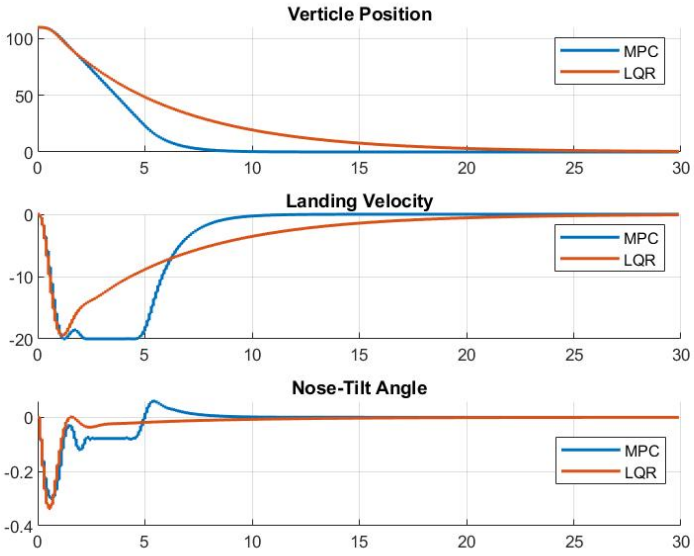

Fig. 6. $h(t)$, $\dot{h}(t)$ and $\theta(t)$ for LQR and MPC controller outside $\mathbb{X}_f$

Figure 6 shows that MPC outperforms LQR for the stated case. The angular velocity was ignored since we deemed it unimportant for this analysis.

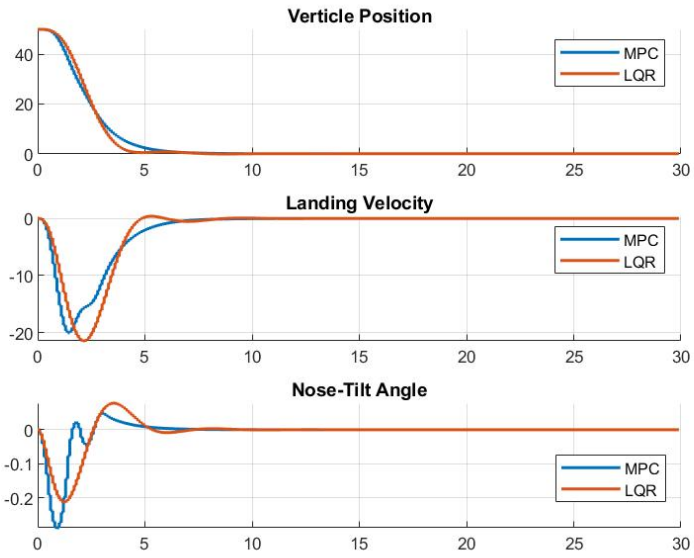

Fig. 7. $h(t)$, $\dot{h}(t)$ and $\theta(t)$ for LQR and MPC controller inside $\mathbb{X}_f$

When the initial state is within, $\mathbb{X}_f$, we expect the MPC and LQR to behave similar to each other. This is what simulation also shows, as in Figure 7. Faster convergence is observed since the initial altitude is low.

C. Tracking Manuever Profiles

As stated in section I, we aim to follow the desired trajectory for certain maneuvers. We first consider the flight with the stated altitude rate profile as shown in Figure 2. To achieve this, we had to adjust our cost function weights to (15) such that, any deviation from the reference value of velocity is penalized significantly. Figure 8 is obtained as a result.

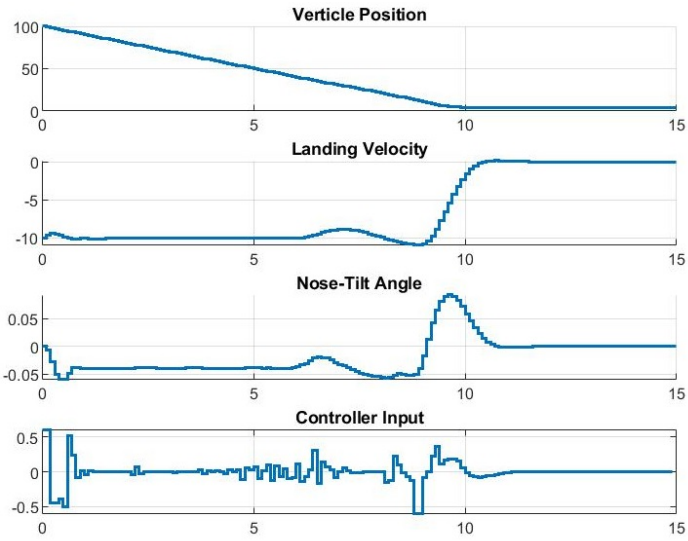

Fig. 8. Landing Dynamics for desired constant velocity

The velocity profile is not exactly what we aimed for. This is because, other than velocity reference, other reference, even though not explicitly mentioned, are aimed at regulation (i.e. 0). Thus the resulting control sequence does produce some deviation off from the desired velocity profile such as to satisfy the problem as whole. This deviation is much noticeable in the case when a sudden disturbance in horizontal velocity, x_2 is induced along $\frac{1}{3}$ of the path, as shown in Figure 9. No matter what penalty is imparted upon the concerned state, the profiles remains invariant. It should be noted that this simulation was performed under the assumptions of full state information.

Now we explore the scenario of refuelling, within which, the aircraft is to keep a constant height. Using the same procedure and weights discussed in (15), we obtained the results shown in Figure 9. As compared to the velocity tracking, no deviations from the desired profile are observed. This is because (as stated earlier), when no reference is given, simple regulation is performed (i.e. tracking zero). In the refuelling case, it so happens that while at a constant altitude, the aircraft ideally has no vertical velocity and nose tilt angle is zero, thus all the states are tracked without contradicting

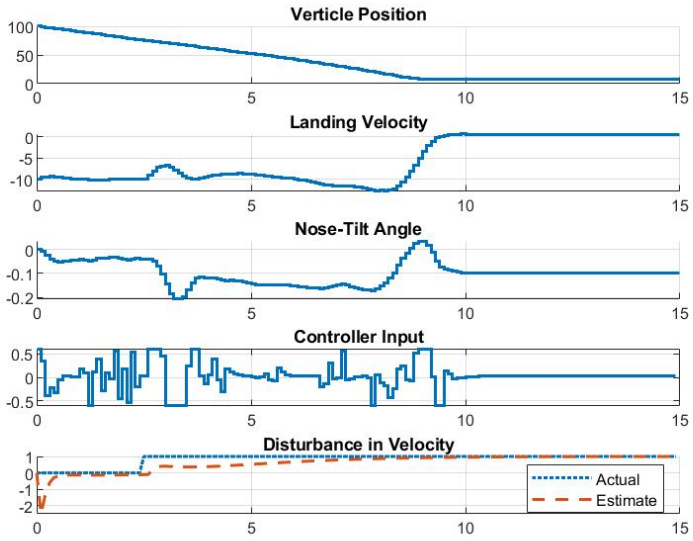

Fig. 9. Landing Dynamics for desired constant velocity

each other. This is not the case in constant velocity, where tracking one state contradicts the tracking of other.

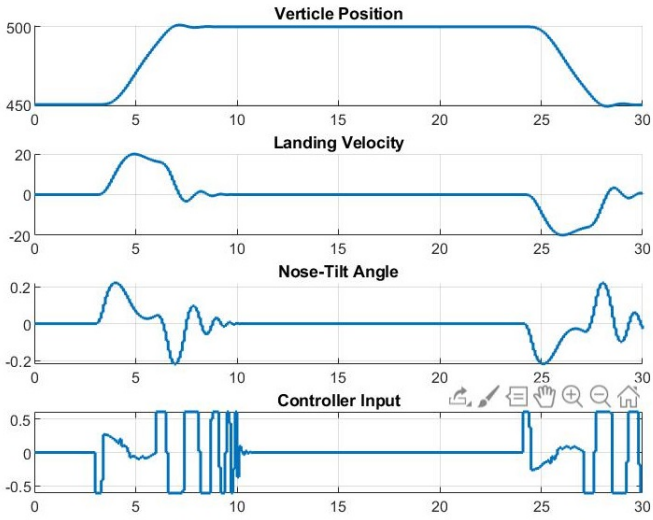

Fig. 10. Dynamics for Refuelling of Jet

Furthermore, let us consider the case of partial state information and disturbance rejection combined. As stated previously, we had reduced our output system equation such as to satisfy the feasibility of the solver, but this did not influence the observability of the system. Additionally, the disturbance and the guessed initial condition of the system were empirically determined to be within severely restricted range (i.e. $x_0 = [445,\ 0,\ 0,\ 0]^T$ and $d(t) = \pm 1$). This limitation is due to the state and terminal constraints. Figure 11 displays the result obtained for such simulation, in which the controller maintains a constant trajectory to hover at an altitude of 500 ft.

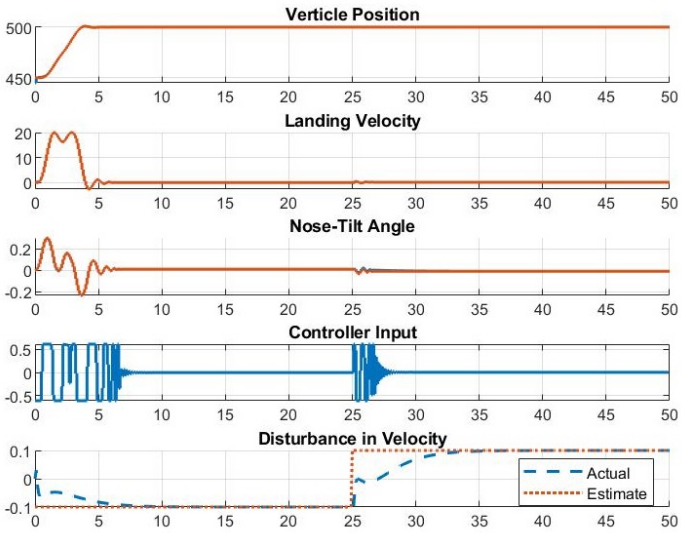

Fig. 11. Disturbance Tracking

In an attempt to explore the possibility of accommodating an extended range of initial state of the observer and disturbance magnitude, the constraint on velocity and altitude were relaxed and some extreme values of initial observer state and disturbance were opted. The results is illustrated in Figure 12..

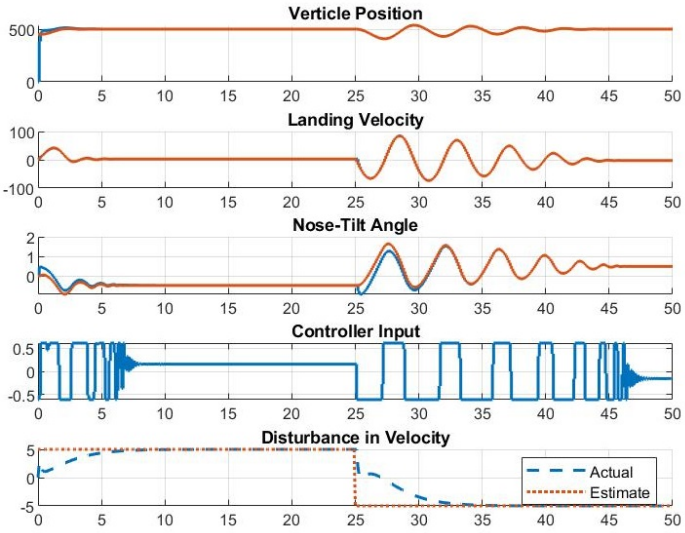

Fig. 12. Disturbance Tracking Extended

V. Conclusion

In addition to the above work, we attempted to pursue output cost based MPC as well, though not documented. The thought behind it was to handle a system arising out of subspace identification. It often happens that states for a model identified using subspace does not posses any physical meaning. Only its output does. Thus, it is more sensible to penalize the output, rather than the states. Furthermore, the constraints would also be applicable only to the output. This

in turn would require our equation 5 to be expressed in terms of matrices C, D. Additionally, the stability analysis for such an output based mpc would require us to manipulate the constraints and utilize the pseudo-inverse of matrix C for obtaining the limits on states. A strong theoretical justification/base was missing (except the trivial derivations) and thus the pursuit was further avoided. The simulation files for the attempt could be found in the folder attached herewith. The reference article has explored many other problems but with the perspective of optimal control. We understand that these problems have a room for being explored through MPC as well. Interested reader could also find more information in the referenced book [3].

REFERENCES

[1] J. Rawlings and D. Mayne, "Model Predictive Control: Theory and Design" 5th ed. Madison: Nob Hill Publishing, 2015.

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

[3] Donald E.Kirk, "Optimal Control Theory: An introduction". Printice-Hall Electrical Engineering Series.

[4] K.J. Astrom, R.M.Murray, "Feedback System: An Introduction to scientist and engineers". Printice University Press (2008)

[5] S. Grammatico, "Model Predictive Control", Lecture 4 slides

NOTES FOR SIMULATION FILES

- All the files require YALMIP directory to be indicated at the top of each.
- Figures indicating the LQR/MPC comparison utilize the files from Regulation folder. Use the `main.m` file.
- The Figures under the section *Tracking Maneuver Profiles* could be replicated using the files in Tracking folder. All the files with name `main_*.m` respectively produce the Figure for stated names.
- Output cost based MPC is contained in the folder with same name.