Rocket Landing Control With MPC

Akinola Alexander Dada[1]

***Abstract*— Recent developments in optimal control have made the landing of rockets possible. This investigation develops a model predictive control scheme capable of safely bringing to rest from a variety of starting conditions a linearised model representing a rocket. A comparison is then made between the devised controller's performance and that of a linear quadratic regulator implementation.**

I. Introduction

The SpaceX company achieved the first successful propulsive vertical landing of an orbital-class rocket stage in December 2015. The rocket in question, Falcon 9, is equipped with Merlin 1D rocket engines, capable of vectored thrust, and grid fins which deploy from the stage-1 fuselage following separation; these actuators allow sufficient controllability of the rocket to permit a safe vertical landing. From a technical point of view, the successful landings were also enabled by theoretical advances in how the kind of nonlinear optimal control problem associated with safe rocket landing can be modelled and solved.
The rocket landing problem can be simplified to generate a mock problem to explore the application of linear model predictive control.This simplification will treat the rocket as a point mass and will only look at position velocity control by assuming that attitude stabilization is handled separately.

II. Problem Statement

The rocket is simplified to produce a model of form:

$$\begin{bmatrix} r(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} I & TI \\ 0 & I \end{bmatrix} \begin{bmatrix} r(k+1) \\ v(k+1) \end{bmatrix} + \frac{1}{m} \begin{bmatrix} 0.5T^2I \\ TI \end{bmatrix} f \qquad (1)$$

With States:

$$r = \begin{bmatrix} r_x & r_y & r_z \end{bmatrix}^T$$

$$v = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T$$

and inputs:

$$\frac{f}{m} = \begin{bmatrix} \frac{f_x}{m} & \frac{f_y}{m} & (\frac{f_z}{m} - g) \end{bmatrix}^T$$

With, T being discrete sampling time, I and 0 being a 3 by 3 identity and zero matrix respectively.
The task is to design, implement and tune an MPC controller on the simplified rocket model that is capable of achieving safe landing from initial altitudes up to 500 m and initial lateral distances up to 600 m from the target. while satisfying input constraints:

$$0 \leq \frac{f_z}{m} \leq 12Nkg^{-1}$$

$$|f_x| \leq f_z \tan\theta, |f_y| \leq f_z \tan\theta$$

and state constraints:

$$|r_x| \leq \frac{r_z}{\tan\phi}, |r_y| \leq \frac{r_z}{\tan\phi}$$

$$|V_x| \leq 20, |v_y| \leq 20, |v_z| \leq 15$$

With ϕ being the glide slope and θ being a max thrust vectoring angle.

The controller should be feasible for a range of different initial positions and also non-zero initial velocities, this is to represent a potential real situation where the rocket is initially moving with downward and lateral speed at the commencement of landing control.
The controller should aim to minimise of control effort while also minimising the time taken till landing

III. Design

The objective of the control system is to regulate all system states to 0, within some finite time, f where:

$$r_f = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$$

$$v_f = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$$

and we define the system states X, where:

$$X = \begin{bmatrix} r & v \end{bmatrix}^T$$

In order to accomplish this and design a control system capable of producing guaranteed results, the system's uncontrolled characteristics must be understood for model.

We define state matrices:

$$Adt = \begin{bmatrix} I & TI \\ 0 & I \end{bmatrix}$$

$$Bdt = \begin{bmatrix} 0.5T^2I \\ TI \end{bmatrix}$$

[2]Akinola Alexander Dada, 160140802, is with the Department of Automatic Control and Systems Engineering, The University of Sheffield, UK

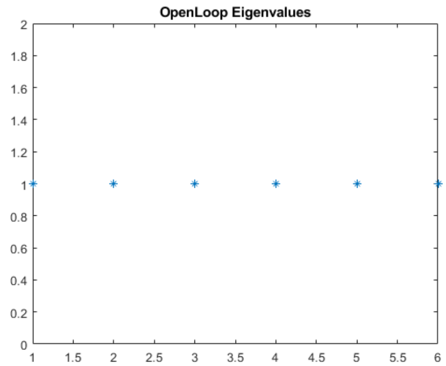

Fig. 1: Open Loop Eigenvalues

A. Stability

For linear state-space systems, stability can be assertained from the locations of the eigenvalues, λ, of the A_{dt} matrix, This process is know as Lyapunov's indirect method where:

$$det|A_{dt} - \lambda I| = 0 \tag{2}$$

For discrete-time systems, stability can be said to be asymptotic if all eigenvalues are strictly within the unit circle, Lyapunov if all eigenvalues are within or on the unit circle, and unstable otherwise. Figure 1 reveals the presence 6 eigenvalues with a value less than 1. Therefore the linear systems model can be said to be stable in the sense of Lyapunov.

B. Reachability

A system is said to reachable if for any initial set of state values X_0, and any final set of state value, X_f, there is a control signal, f, that can take the system from X_0 at k = 0, to X_f with some finite time.
A system is said to be reachable if the reachability matrix:

$$W_r = \begin{bmatrix} B_{dt} & A_{dt}B_{dt} & \dots & A_{dt}^{n-1}B_{dt} \end{bmatrix} \tag{3}$$

is full rank equalling the number of states. Thus, as this is the case, the system can be said to be reachable.

C. Observbility

As the full-state is available, the system is fully observable.

D. LQR

Full-state feedback is a control technique which makes use of X to modify the system's eigenvalue so as to change its characteristic responses by determining some set of gains, K, that applied to the system states and fed back into the system adjust the system dynamics. The Linear Quadratic Regulator (LQR) is a technique for finding the set of gains K, in an optimal manner by minimising a quadratic cost function over an infinite horizon. This produces a closed loop system of form:

$$X(k+1) = (A_{dt} - B_{dt}K)X(k) \tag{4}$$

With cost function of form:

$$min_f \sum_{k=0}^{k=\infty} X^T(k)QX(k) + f^T(k)Rf(k) \tag{5}$$

s.t

$$X(k+1) = A_{dt}X(k) + B_{dt}f(k), k = 1,2,3\dots$$

Where $Q^{n\times n}$, and $R^{m\times m}$, are 2-norm weighing matrices on the states and inputs respectively. This controller adjusted the eigenvalues in closed-loop as shown in Figure 2, where all state are within the unit circle of value and thus the system is now stable asymptotically.

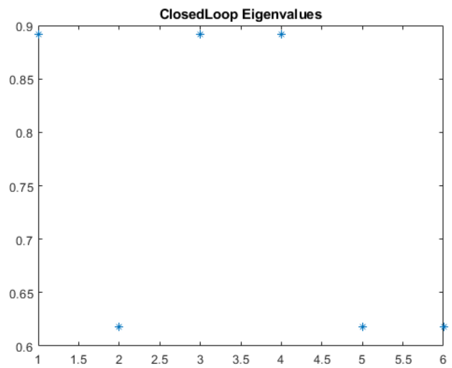

Fig. 2: Closed Loop Eigenvalues

E. MPC

MPC refers to an approach to tackling control problems, containing certain key concepts and ideas implemented in various ways. There are 3 main components in an MPC algorithm:

- Prediction
- Optimisation
- The Receding Horizon Principle

To implement an MPC algorithm, we define a prediction stage where the model is recursively implemented unto itself to produce a higher order prediction models over specified horizon lengths. Given a state space model, a state prediction model for a horizon of length N, can be defined as:

$$\bar{X}(k) = FX(k) + G\bar{f}(k) \tag{6}$$

where:

$$\bar{X}(k) = \begin{bmatrix} X(k+1|k) \\ X(k+2|k) \\ \vdots \\ X(k+N|k) \end{bmatrix}, \bar{f}(k) = \begin{bmatrix} f(k|k) \\ f(k+1|k) \\ \vdots \\ f(k+N-1|k) \end{bmatrix} \tag{7}$$

$$F = \begin{bmatrix} A_{dt} \\ A_{dt}^2 \\ \vdots \\ A_{dt}^N \end{bmatrix}, G = \begin{bmatrix} B_{dt} & 0 & \dots & 0 \\ A_{dt}B_{dt} & B_{dt} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_{dt}^{N-1}B_{dt} & A_{dt}^{N-2}B_{dt} & \dots & B_{dt} \end{bmatrix} \tag{8}$$

The Optimisation stage formulates and solve an optimisation problem to obtain the minimising control sequence over the whole horizon, $\bar{f}$, evaluated at each time step k, where:

$$min_f = \sum_{i=0}^{N-1}[X^T(k+i|k)QX(k+i|k)+f^T(k+i|k)Rf(k+i|k)]+X^T(k+N|k)PX(k+N|k) \quad (9)$$

$s.t$

$$X(k|k) = X(k)$$

$$X(k+1+i|k) = A_{dt}X(k+i|k)+B_{dt}\mu(k+i|k), i = 1,2,3\ldots,N-1$$

$$\begin{bmatrix}(-f_z+g)\tan\theta \\ (-f_z+g)\tan\theta \\ 12-g\end{bmatrix} \leq f \leq \begin{bmatrix}(f_z+g)\tan\theta \\ (f_z+g)\tan\theta \\ g\end{bmatrix}$$

$$\begin{bmatrix}-\frac{r_z}{\tan\phi} \\ -\frac{r_z}{\tan\phi} \\ 0 \\ -20 \\ -20 \\ -15\end{bmatrix} \leq X \leq \begin{bmatrix}\frac{r_z}{\tan\phi} \\ \frac{r_z}{\tan\phi} \\ inf \\ 20 \\ 20 \\ 15\end{bmatrix}$$

Where Q, and R, are the same 2-norm weighing matrices as with the LQR on the states and inputs respectively.

The value of P is derived through satisfying the application of the Lyapunov equation:

$$(A_{dt}+BK)^T P(A_{dt}+BK)-P = S \quad (10)$$

Where S is the infinite horizon cost and K is the LQR gain. Therefore, with the addition of P, if K is stabilising then, the derived MPC algorithm produces a series of control sequences that would be guaranteed stabilising in the unconstrained case if the system is

- reachable, but at least stabilizable
- Observable
- Q is at least Positive Semi Definite and R is Positive Definite.

This constrained finite horizon optimisation problem is then put into the standard form for a quadratic program:

$$min_f 0.5f^T(k)Hf(k)+X^T(k)L^T f(k)+X^T(k)MXk \quad (11)$$

$s.t$

$$Pcf(k) \leq qc+ScX(k)$$

by substituting equation 6 into equation 9 and putting the constraints in matrix form. This problem is solved using the matlab quadprog function.

A receding horizon is implemented by the selection and implementation of only the first set of control actions produced over the N length horizon, and discarding the rest of the control sequence, before repeating this process at thee next time step. This is what adds feedback authority to the controller as new state information is taken into account at each time interval.

Control Parameters	
Character	Value
T	0.5 seconds
ϕ	30 degrees
θ	10 degrees

TABLE I: Table of Control Parameters

IV. Results, Analysis and Discussion

The weights Q and R on both the LQR and MPC implementation are selected via trial and error while observing the systems responses,where:

$$Q = diag(0.05,0.05,0.05,1,1,1) \quad (12)$$

$$R = diag(1,1,1); \quad (13)$$

For the MPC implementation a horizon length of 5 was selected.

Shown in Figures 3 and 4 is are the inputs and state responses of the LQR implementation. The LQR implementation's utilisation of an infinite horizon in its optimisation problem, which would ordinarily be intractable, is only solved through the adoption of dynamic programming. This however makes the controller unable to be solved with constraints placed upon it, thus the controller is unconstrained and violates both input and state constraints, despite successfully regulating all states to 0 within a finite time, settling within 25 seconds from a variety of initial conditions up too and including r = [600,600,500] and non 0 starting velocities.

Shown in Figures 5 and 6 are results of the MPC control implementation, where the system respects all input and velocity state constraints where the system is able to regulate all states to 0 within a finite time, settling in less than 60 seconds.

With the implementation of constraints on the position glide slopes however, the solutions becomes increasingly unfeasible at certain points. The phenomenon arises where due to the constraints, the system stability is no longer decoupled from the horizon length N, as with the unconstrained case, in the presence of a stabilising terminal cost. Beyond N = 10, the system becomes less feasible which leads to a performance dropping resulting in increased optimisation cost.

This can be rectified through the implementation of terminal state constraint where X(k+N) = 0.

V. Conclusion

This investigation was a able to:

- Design and implement an LQR controller as a benchmark
- Design and implement a constrained MPC controller
- Determine metrics and specifications for analysis of the MPC controller.
- Compare the perfomance of the MPC controller against the LQR and discuss results

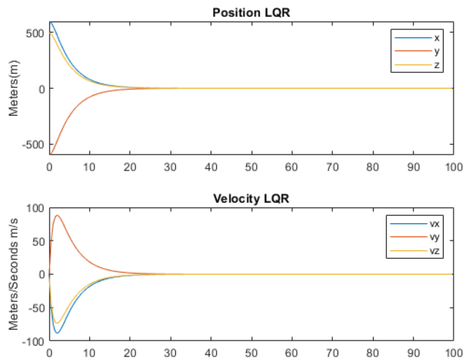

Fig. 3: LQR States

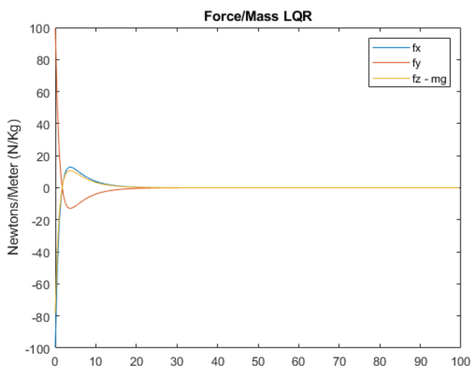

Fig. 4: LQR Inputs

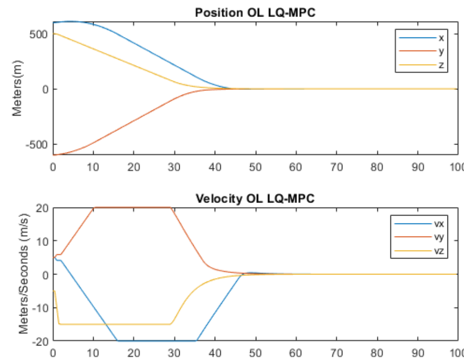

Fig. 5: MPC States

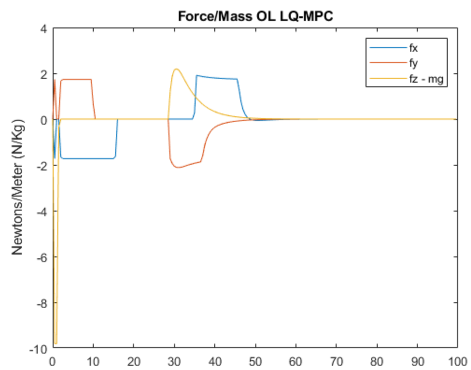

Fig. 6: MPC Inputs