

# Designing an MPC Controller for a Simplified Rocket Landing Problem

Sebastiano Zuddas

dept. Automatic Control & Systems Engineering

University of Sheffield

Sheffield, United Kingdom

sebastianozuddas1@gmail.com

**Abstract**—This document is a model and instructions for L<sup>A</sup>T<sub>E</sub>X. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. \*CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.

**Index Terms**—MPC, Rocket Landing, Model Predictive Control, Control Theory, Control Engineering

## I. INTRODUCTION

The objective of the assignment is to implement a model-predictive control (MPC) scheme to land a simplified model of a rocket. The problem is fundamentally one of 'controllability', ie  $x(0) = x_s \rightarrow x(t_f) = 0$ . The assignment gives an abstracted model, we start by rewriting the model as follows:

$$\begin{bmatrix} r_x(k+1) \\ r_y(k+1) \\ r_z(k+1) \\ v_x(k+1) \\ v_y(k+1) \\ v_z(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 & T_s & 0 \\ 0 & 0 & 1 & 0 & 0 & T_s \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_x(k) \\ r_y(k) \\ r_z(k) \\ v_x(k) \\ v_y(k) \\ v_z(k) \end{bmatrix} + \begin{bmatrix} \frac{T_s^2}{2m} & 0 & 0 \\ 0 & \frac{T_s^2}{2m} & 0 \\ 0 & 0 & \frac{T_s^2}{2m} \\ T_s & 0 & 0 \\ 0 & T_s & 0 \\ 0 & 0 & T_s \end{bmatrix} \begin{bmatrix} f_x(k) + w_x \\ f_y(k) + w_y \\ f_z(k) - mg \end{bmatrix} \quad (1)$$

Although we assume the system is reachable, it is important to check for controllability. Through MATLAB: `Rank = rank(ctrb(A, B))` we find the system is controllable, and we can proceed with designing a controller.

## II. DESIGN

### A. Controller Design

Three separate controllers are considered in this report. The first controller is an unconstrained, two stage MPC. The second controller builds on the first, and introduces both state and input constraints. The third controller further builds on the first two, introducing disturbance rejection. We start by looking at the unconstrained controller.

1) *Unconstrained*: The  $Q$  and  $R$  matrices are chosen as follows:

$$Q = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix}; R = I \cdot 0.1 \text{ for } I \in \mathbb{R}^{3 \times 3}$$

The  $Q$  matrix was chosen as to prioritise the vertical position and velocity. This represents the high level of importance given to landing at the correct altitude. Similarly, the prioritisation of velocity represents the importance of landing a velocity that prevents a hard landing. The lateral positions, ie  $r_x$  and  $r_y$ , are given a lower priority than the vertical position, but a higher priority than the lateral velocity,  $v_x$  and  $v_y$ . The reasoning behind this is that it is more important that the landing location is correct over the lateral velocities throughout the entire simulation.

The  $R$  matrix was chosen as to penalise the control inputs,  $f_x$ ,  $f_y$  and  $f_z$ , equally. The small value given to  $R$ , 0.1, represents the low cost associated with expending fuel to control the rocket, and encourages the rocket to use fuel to exercise control in all three directions equally. Essentially, control is cheap. The horizon length  $N$  was chosen to be 5 as it provided a spectral radius of 0.5975, which is well within the stability region of the system.

The poles of the closed loop system were placed through `K_2 = -place(A, B, [0.01 0.01 0 0.01 0 0])`. The most important poles, ie the poles representing  $r_z$ ,  $v_z$  were placed at zero. The limitations of the `place()` function meant that not every pole could be set to zero, hence the other poles were set to 0.01. The weighting matrix  $P$  was calculated using `P = dlyap((A+B*K_2), Q+K_2'*R*K_2)`. The prediction matrices  $F$  and  $G$  were generated using the `predict_mats()` function, and the corresponding cost matrices  $H$ ,  $L$  and  $M$  were generated using the `cost_mats()` function. We assume at this point that  $Q \succeq 0$  and  $R \succ 0$  which implies  $\bar{u}_N^*(k) = -H^{-1} \cdot Lx(k)$ . As such, the unique and optimal solution to the optimal control problem is obtained and is through quadratic programming at

each time step  $U_{opt} = \text{quadprog}(H, L \star x)$

2) *Constrained*: The constrained controller builds on the unconstrained controller by introducing both state and input constraints. The input constraints are chosen as follows: The linear inequality constraints are based on the brief, and are chosen as follows:

$$P_x = \begin{bmatrix} I_x \\ -I_x \end{bmatrix} P_u = \begin{bmatrix} I_u \\ -I_u \end{bmatrix} \text{ for } I_x \in \mathbb{R}^{6 \times 6} \text{ and } I_u \in \mathbb{R}^{3 \times 3}$$

$$q_x = \begin{bmatrix} \frac{r_z}{\tan \phi} + 50 \\ \frac{r_z}{\tan \phi} + 50 \\ 500 \\ 20 \\ 20 \\ 15 \\ \frac{r_z}{\tan \phi} + 50 \\ \frac{r_z}{\tan \phi} + 50 \\ 0 \\ 20 \\ 20 \\ 15 \end{bmatrix} q_u = \begin{bmatrix} f_z \cdot \tan \theta \\ f_z \cdot \tan \theta \\ 12 \\ f_z \cdot \tan \theta \\ f_z \cdot \tan \theta \\ 0 \end{bmatrix}$$

The  $r_x$   $r_y$  constraints were given in the brief as  $|r_x|, |r_y| \leq \frac{r_z}{\tan \phi}$  where  $\phi$  represents the glide slope angle, and is chosen as a constant  $\phi = 30^\circ = 0.52$  rad. The input constraints given in the brief as  $|f_x|, |f_y| \leq f_z \tan \theta$  where  $\theta$  represents the maximum allowable angle of the rocket engines, and is chosen as  $\theta = 10^\circ = 0.17$  rad.

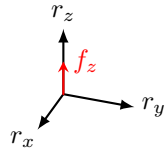


Fig. 1:  $f_z$  on the  $z$  axis.

Figure ?? provides a visualisation of how  $f_z$  acts on the point mass. Overall, this permits the implementation of Algorithm 6.1, pg. 102 from [?], since all the relevant optimal control and constraint matrices have been defined. The constraint matrices are obtained in code through `constraint_mats()`. Disturbances are included in the simulation despite no disturbance rejection implementation, mainly to be able to contrast the results and demonstrate the implementation of disturbance rejection.

3) *Disturbance Rejection*: Rank, reachability and observability tests were performed on  $B, E$  and the two pairs  $(A, B); (C, A)$  and it was found that disturbance rejection was possible. The disturbance rejection controller builds on both the unconstrained and constrained controllers. No reference tracking is included in the following controller. Only wind

disturbance is considered, and is modelled as a sinusoid as follows:

$$\vec{w}(k) = \begin{bmatrix} \sin \frac{50}{k} \\ \cos \frac{50}{k} \\ 0 \end{bmatrix}$$

The use of a sinusoid disturbance is intended to represent the changing windspeeds as the rocket descends through the atmosphere. The sinusoid functions were chosen to be out of phase as to yield visibly different results.

In terms of controller design, equation ?? is rewritten as follows:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + Ew(k) \\ y(k) = Cx(k) + Fd(k) \end{cases}$$

where  $E = B$  and  $F = 0$  since we ignore output disturbance, hence:

$$\begin{cases} x(k+1) = Ax(k) + B(u(k) + w(k)) \\ y(k) = Cx(k) \end{cases}$$

It is now possible to formulate the  $T$  matrix to find  $x_{ss}; u_{ss}$ .

$$T = \begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix}$$

In this case, we have  $C \in \mathbb{R}^{6 \times 6} \wedge B \in \mathbb{R}^{6 \times 3} \therefore p < m \iff 3 < 6$ . Similarly, we find that  $T$  is full row rank, confriming that for any pair  $(r, d) \exists$  a pair  $(x_{ss}, u_{ss})$ . As such,  $x_{ss}; u_{ss}$  are given by:

$$\begin{bmatrix} x_{ss}(k) \\ u_{ss}(k) \end{bmatrix} = T^{-1} \begin{bmatrix} B \cdot \vec{w}(k) \\ 0 \end{bmatrix}$$

From this result, the deviation variables become  $z := x - x_{ss}; v := u - u_{ss}$ . As a result,  $v^*(k|k)$  is calculated through `quadprog(H, L * z, qc + Sc * z)` at each iteration.

## B. Experiment Setup

The mass of the rocket is simplified and assumed to be constant for all experiments, and is chosen as  $m = 1$  kg. The starting parameters for each experiment are as follows:

$$\vec{x}(0) = \begin{bmatrix} 600 \\ 600 \\ 500 \\ 5 \\ 5 \\ -15 \end{bmatrix} \wedge \vec{u}(0) = \vec{0}$$

These starting conditions are the limits of what the assignment permits, i.e a maximum starting altitude of 500m, and a maximum lateral distance of 600m.

### III. RESULTS

#### A. Unconstrained

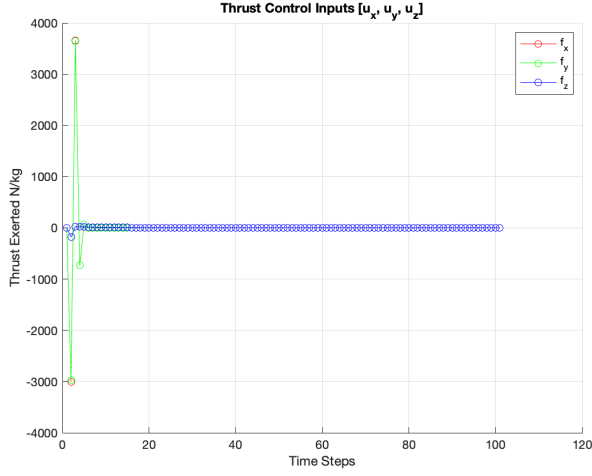


Fig. 2: Unconstrained Thrust Control Input

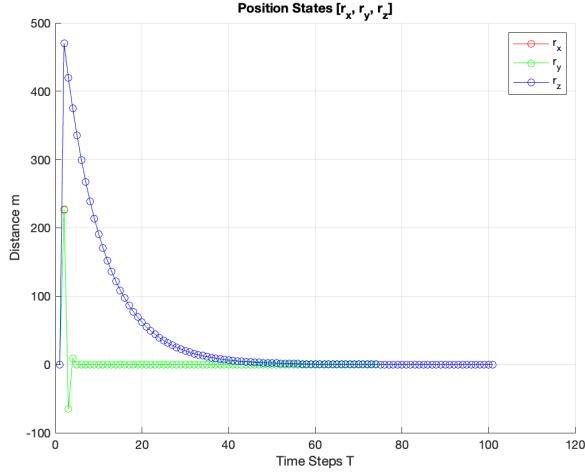


Fig. 3: Unconstrained Position State Plot

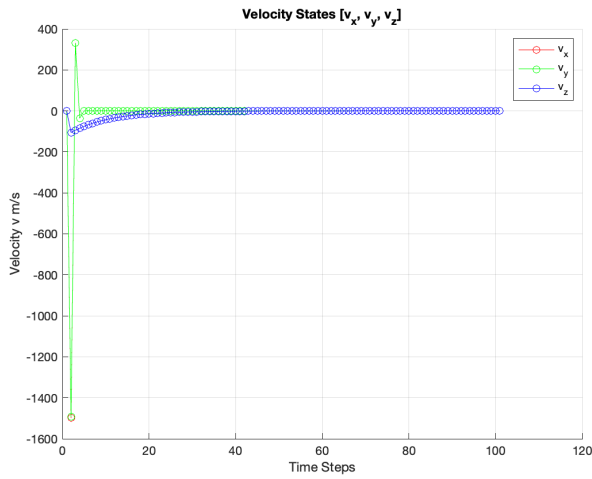


Fig. 4: Unconstrained Velocity State Plot

#### B. Constrained

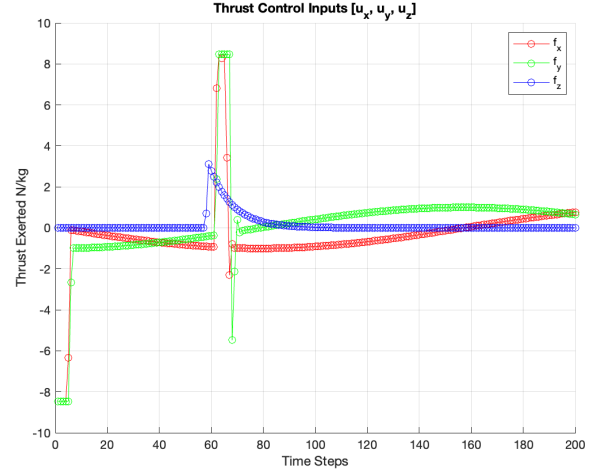


Fig. 5: Constrained Thrust Control Input

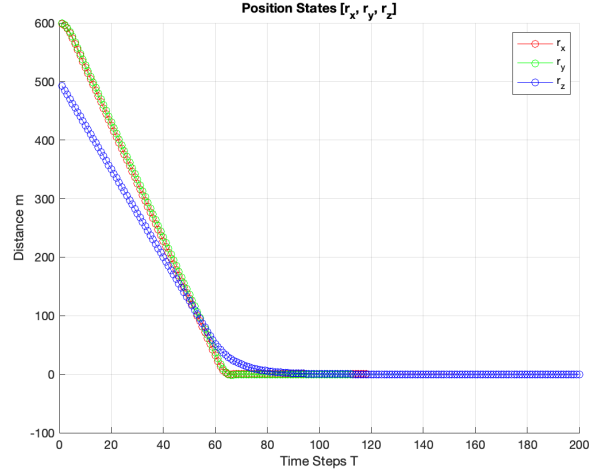


Fig. 6: Constrained Position State Plot

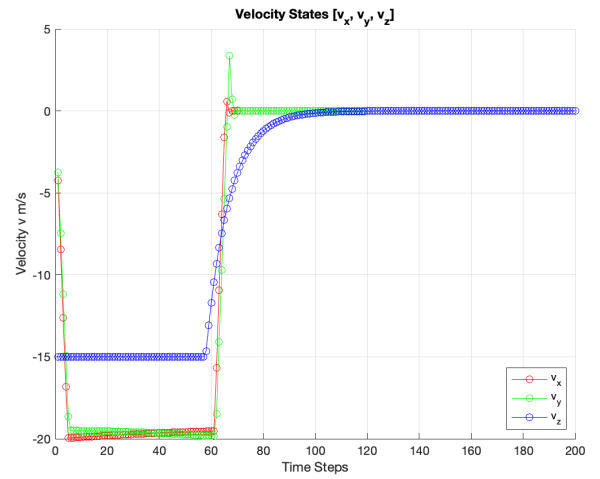


Fig. 7: Constrained Velocity State Plot

### C. Disturbance Rejection

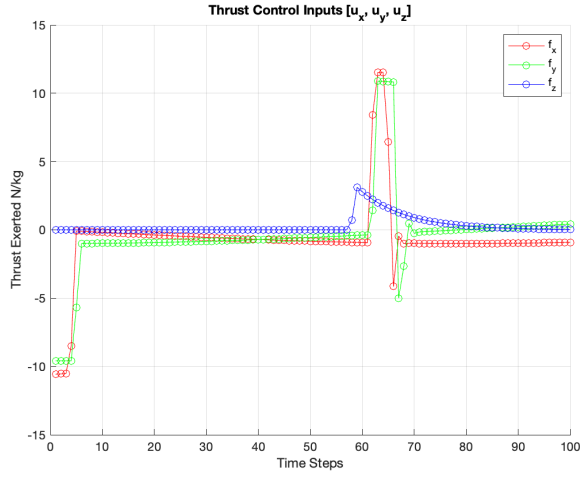


Fig. 8: Disturbance Rejection Thrust Control Input

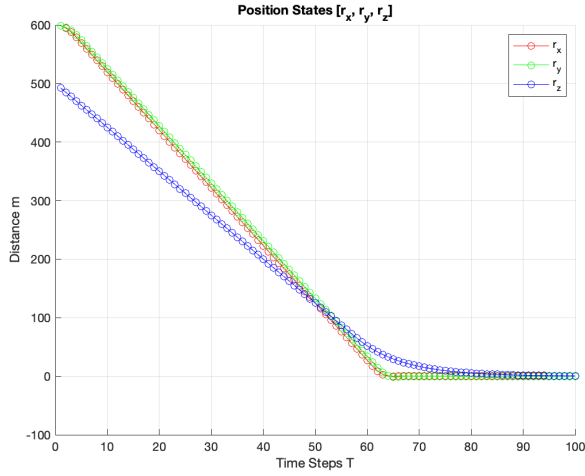


Fig. 9: Disturbance Rejection Position State Plot

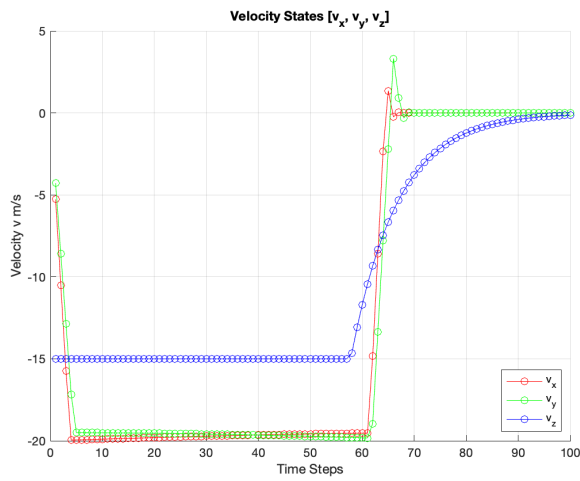


Fig. 10: Disturbance Rejection Velocity State Plot

### IV. ANALYSIS & DISCUSSION

Due to the limitations to the number of pages, the analysis will mainly discuss the effectiveness of the controllers using constraints and disturbance rejection.

### V. CONCLUSION

## APPENDIX

**Temporary page!**

L<sup>A</sup>T<sub>E</sub>X was unable to guess the total number of pages correctly. As there was some unprocessed data that should have been added to the final page this extra page has been added to receive it.

If you rerun the document (without altering it) this surplus page will go away, because L<sup>A</sup>T<sub>E</sub>X now knows how many pages to expect for this document.