Performance of a Rocket Landing on a Moving Platform Problem with MPC and LQR and Optimal Tuning

George Makrygiorgos, Michael Wang, Alexis Ruiz, Fayyad Azhari
ME 231A - University of California, Berkeley

***Abstract*—In this paper, we discuss the design and simulation about landing of a rocket on a moving platform using Model Predictive Control (MPC) in 2D. For simulation and MPC, we will be using the non-linear rocket model with the MPC being the closed-loop trajectory of the landing. For control, we discretize in time, and the MPC problem will require as reference, the solution of the LQR, formulated as a QP, at each time step. Since the moving landing platform is randomly moving horizontal, the rocket will have side thrusters to contribute torque and enhanced control over the heading angle in order to provide extra controllability to the system to make sure the problem is feasible. We assume sensor information for horizontal displacement of the platform, which uses Kalman filter to give us velocity information. We analyze the performance of the control with the movement of the platform, disturbance and uncertainty. https://youtu.be/Mu4gLQCrZUE**

I. INTRODUCTION

The recent success by Space X in landing their Falcon 9 rocket on a swaying remote-moving sea platform is a great achievement for exploration since rocket reusing has been popular to reduce the cost of every mission. The rocket launches have more flexibility when a mobile landing platform is used. This mobile landing site could be place somewhere it is safe for example in the middle of the sea with undetermined waves. Model-based Predictive Control (MPC) can be proven to be effective for this rocket landing on moving platform scenario.

A. Model and assumptions

In this project, we will consider the following rocket system Fig 1 where we simulate a gimbaled thrust by adding left and right thruster. Our model simplifies the dynamics by ignoring l_n.

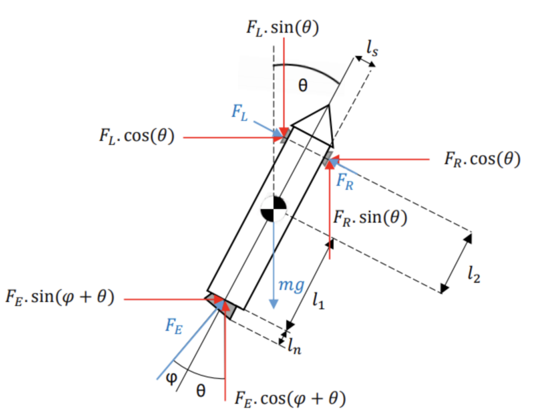


Fig. 1: Rocket and thrust [1]

The chosen model for the rocket is defined by the six equations which follow the following equations:

$$\begin{cases} \dot{\theta} = \omega \\ \dot{\omega} = \frac{-l}{2J}(F_E \sin(\delta) - F_S) \\ \dot{x} = V_x \\ \dot{V}_x = \frac{1}{m}(F_E \sin(\delta+\theta) + F_S \cos(\theta)) \\ \dot{z} = V_z \\ \dot{V}_z = \frac{1}{m}(F_E \cos(\delta+\theta) + F_S \sin(\theta) - mg) \end{cases}$$

Where θ, ω, x, z V_x, V_z - and their time derivatives - represent the relative angle and angular velocity, and the positions and velocities of the rocket in the x-horizontal and z-vertical directions.

The three inputs for our model are: F_E, δ and F_S where F_E is the main thruster, angle δ depicts the deviation between the rocket axis and the thrust F_E and F_S which is the differential thrust defined as the difference between F_L and F_R.

We also define the geometric parameters with the mass $m = 27648\ kg$, the moment of inertia $J = 8.47 \times 10^6\ kg.m^2$, the lengths $l_1 = l_2 = 35\ m$ and the maximal thrust of the rocket: $F_{max} = 1690\ kN$.

After a Jacobian linearization, we obtain a linear model based on the previous model. Zero-order hold discretization (sampling time $T_s = 0.5\ s$) is then applied to find the discrete dynamics:

$$x_{k+1} = Ax_k + Bu_k$$

with k the discrete-time instant, the seven-states vector $x_k = [\theta_k \;\; \omega_k \;\; x_k \;\; V_{x_k} \;\; z_k \;\; V_{z_k} \;\; g]^T$ and the input control vector $u_k = [F_{E_k} \;\; \delta_k \;\; F_{S_k}]^T$.

Another approach to generating discrete matrices A and B is numerical differentiation. Here A is generated by $\frac{S_+ - S_-}{2\varepsilon}$, where $S_+ = x + \varepsilon I$ and $S_- = x - \varepsilon I$. Similarly, B is found by shifting on inputs. The plant model, using the full nonlinear dynamics and a high-fidelity ODE solver, was used to find A and B. Our MPC controller performed significantly better using this approach.

The state constraints, only applied on the first two states, are defined such as: $|x(k)| \leq x_{max}$ with $x_{max} = \begin{bmatrix} \frac{20\pi}{180} \\ \frac{5\pi}{180} \end{bmatrix}$

The input constraints can be written such as $H_u u \leq K_u$ where

$$H_u = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix}, \text{ and } K_u = \begin{bmatrix} F_{max} \\ -F_{max}/6 \\ \frac{5\pi}{180} \\ \frac{5\pi}{180} \\ F_{max}/200 \\ F_{max}/200 \end{bmatrix}$$

B. Reference Trajectory

From [1], the target trajectory of the rocket can be defined by the following equation:

$$X_{\text{target trajectory}} = \begin{bmatrix} \theta_0 * (0.3 + e^{-\beta t}) \\ \gamma\theta_0 \\ x_0 + \frac{x_f - x_0}{1+e^{-t}} \\ x_0 - x_f \\ z_t[t_0 \ldots T_h] \\ z_0 * e^{\text{linspace}(0,2,T_h)} \end{bmatrix}$$

Where θ_0, x_0, t_0, z_0, is the current state of each components, z_t is the target vertical position, γ and β are constants that need to be optimized. This formulation, while computationally fast, resulted in largely infeasible trajectories. Thus, we look toward optimization to generate reference trajectories.

II. LQR and MPC formulation

We use the simulation time $M = 40\ s$, the MPC horizon length is $N_P = 4$ and the LQR horizon length is $N = 80$. Each time step is 0.5s. For both MPC and LQR, we tune matrices P, Q, and R which will be chosen positive semi-definite and optimized thanks to Bayesian techniques.

A. Linear-Quadratic Regulator formulation

The LQR with terminal cost can be expressed as below:

$$\min_{x_0,\ldots,x_{N-1},u_0,\ldots,u_{N-1}} x_n' P x_n + \sum_{k=0}^{N-1} [x_k' Q x_k + u_k' R u_k] \tag{1}$$

s.t.:

$$\begin{cases} x_{k+1} = Ax_k + Bu_k \ \forall k \in [\![0\ ;\ N-1]\!] \\ x_{min} \leq x(k) \leq x_{max} \ \forall k \in [\![0\ ;\ N-1]\!] \\ u_{min} \leq u(k) \leq u_{max} \ \forall k \in [\![0\ ;\ N-1]\!] \\ x_0 = x(0) \\ x_N \in \mathscr{X} \end{cases}$$

B. Model Predictive Control formulation

The MPC formulation, with slack variables, used is the following:

$$\begin{aligned} \min_{x_0,\ldots,x_{N_P-1},u_0,\ldots,u_{N_P-1}} \sum_{k=0}^{N_P-1} & [(x_k - \widetilde{x}_{ref})' Q (x_k - \widetilde{x}_{ref}) \\ & + (u_k - \widetilde{u}_{ref})' R (u_k - \widetilde{u}_{ref})] \\ & + (x_N - \widetilde{x}_N)' P (x_N - \widetilde{x}_N) \\ & + \nu' \varepsilon + ||\varepsilon||^2 \end{aligned}$$

s.t.:

$$\begin{cases} x_{k+1} = Ax_k + Bu_k \ \forall k \in [\![0\ ;\ N_P-1]\!] \\ x_{min} \leq x(k) \leq x_{max} \ \forall k \in [\![0\ ;\ N_P-1]\!] \\ u_{min} \leq u(k) \leq u_{max} \ \forall k \in [\![0\ ;\ N_P-1]\!] \\ x_0 = x(t) \\ \varepsilon \geq 0 \end{cases}$$

III. Optimal Controller Tuning

The selection of the tuning parameters plays a crucial role. In particular, several key elements that affect the solution are the the prediction horizon N_p, the input cost matrix R as well as the state cost matrix Q that appear in the objective function. Nevertheless, tuning can be performed in any other relevant parameter. Tuning those properly overall affects the solution quality. We denote as θ the tuning parameters considered here. Given that the objective is to optimize the performance of a system denoted by $\mathscr{M}$, we pose the following minimization problem

$$\min_{\theta} f(\theta|\mathscr{M}) \tag{2}$$

$$s.t. \quad \theta \in \mathbb{R}^{n_\theta}, \tag{3}$$

where f defines a general performance metric. This metric can include several sub-metrics related to the objective over the entire simulation horizon, the number of constraint violations (especially in the case of soft-constraints) and other relevant terms [3], [4]. In general, the mapping between θ and $f(\theta)$ is unknown *a priori* so the optimization problem of (2) is typically treated as a black-box, data-driven optimization problem. Moreover, evaluation of f can be expensive. For this reason, a popular solution strategy for (2) is based on Bayesian Optimization (BO) [2]. BO iteratively samples the tuning parameter space and the data are used to construct and refine a statistical model/surrogate for the real function f. This is commonly done using a Gaussian Process (GP). Thus, we assume that that the objective function is given by $f(\theta) \sim \mathscr{GP}(m(\theta), k(\theta, \theta'))$, where $m(\theta)$ is the prior mean and $k(\theta, \theta')$ is the covariance function between points in the tuning parameter space. The explored samples are picked by optimizing some acquisition function.

IV. Simulation and Results

In this section we present our case study results. We implemented the previous model and solved an LQR and an MPC problem to compare the performance. For the first part of the results, we rely on "trial-and-error" for obtaining good tuning parameters for our problem. A systematic way of tuning is presented next on sample case studies

A. Simulation and plots

Using open-loop LQR, we generated the optimal trajectory and inputs of the thrusts and angle using the linear, discrete matrices from numerical differentiation. Trivially, running these inputs directly on the high-fidelity plant will result in unstable behavior. Instead, these references will be used when solving the reference-tracking problem with MPC. In our MPC simulation, the loop is closed with the high-fidelity, nonlinear plant model mentioned earlier, and we also update A and B with numerical differentiation at each time step.

The objective was to land the rocket with zero angle orientation and angular speed. After solving the MPC, we plot the closed-loop simulation of the controller with the reference trajectory to evaluate the performance. We generate new reference trajectories every 7.5 seconds in order to track the landing platform and reduce infeasibility. The updated reference is then tracked when solving the MPC. In Fig. 2, one can observe that the rocket is following the reference and going to the desired state, tracking a moving platform that starts at 0 and moves right at a rate of 0.5m/s.

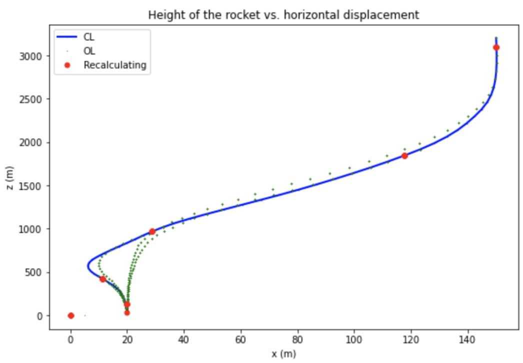
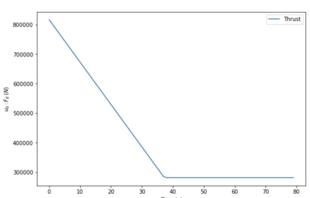
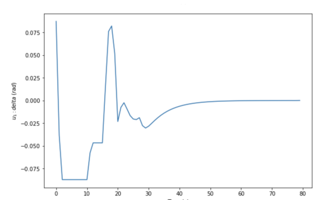


Fig. 2: Reference trajectories and closed-loop simulation

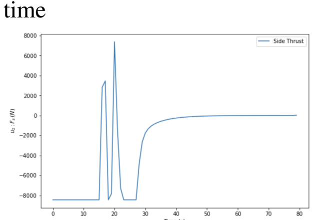
In the following figures, we have the optimal inputs generated when solving the open-loop LQR. We have the thrust F_E, the angle δ and the differential thrust F_S versus time. The angle δ and the differential force F_S converge to zero as expected because we want to land the rocket without angle orientation and angular speed.



(a) F_E vs. time



(b) δ vs. time



(c) F_S vs. time

Fig. 3: Optimal inputs obtained from solving the LQR problem

In our next case, we have another MPC simulation where the landing zone is moving at $V_{platform} = 1\, m \cdot s^{-1}$, but it stops halfway. From our filter, the velocity of the platform is tracked, and given the total time horizon for landing, we predict that the platform will be at 40m. As shown in Figure 4, the rocket first tracks this point, but is able to update its trajectory and successfully lands on the platform when it stops, while the original reference trajectory does not reach the right landing zone.

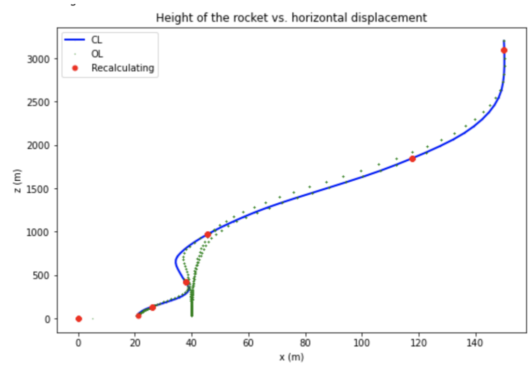
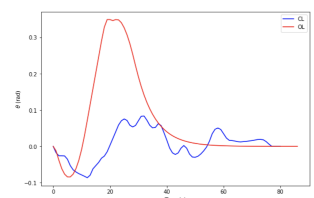
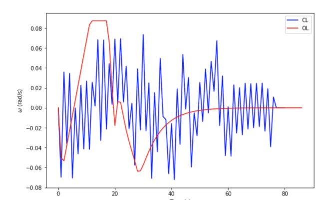


Fig. 4: Reference trajectories and closed-loop simulation with moving platform

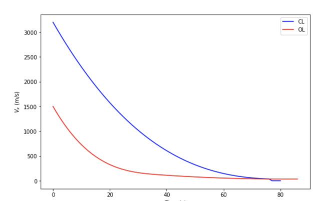
The evolution of the other states $(\theta, \omega, V_x, V_z)$ in comparison to their reference are given in the four plots in Fig 5. We can see the same dynamic but, there is a variation between the optimal open-loop reference and the closed-loop states on the plant from MPC control.



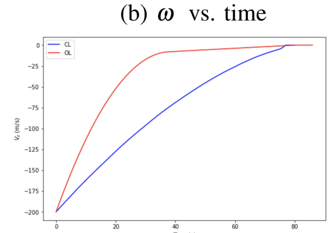
(a) θ vs. time



(b) ω vs. time



(c) V_x vs. time



(d) V_z vs. time

Fig. 5: Comparison of the states

B. Optimal Tuning

To demonstrate the effect of optimal parameter tuning, we first apply this methodology to the open-loop LQR optimization problem of the rocket landing. We assume that the cost matrices have the following form ; $Q = [\widetilde{q}I_6, 0I_1]$, where I_n is $n \times n$ identity matrix. The matrix R is given by $R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{(\frac{\alpha\pi}{180})^2} & 0 \\ 0 & 0 & \frac{\beta}{F_{max}^2} \end{bmatrix}$. Finally, we define $P = \gamma Q$. We consider now that $\theta = \{\widetilde{q}, \alpha, \beta, \gamma\}$ are the tuning parameters and perform BO by solving

$$\min_{\theta} J_{opt}(\theta|\mathscr{M}) \quad (4)$$
$$s.t. \quad \theta \in \Theta, \quad (5)$$

where the Θ is the bounded set for the tuning parameters and J_{opt} is the minimum cost that is obtained given a set of parameters θ. We perform 50 BO iterations and the samples obtained are shown in the following figure.

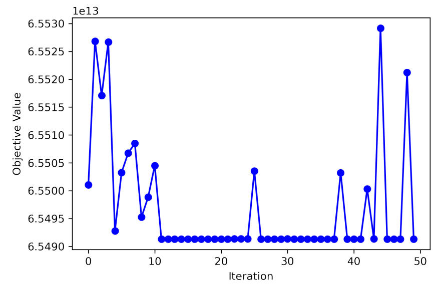


Fig. 6: Objective function evaluated during the BO iterations for the LQR case study

The optimal solution in this case converged to the lower and upper bounds of the parameters involved, which is $\theta = 400, 10, 100, 0.8$. This is expected as those values minimize the the elements of the cost matrices, leading to an overall decreased cost. In other words, for any optimal input and state profile, the cost is minimized when the cost matrices are also element-wise minimized.

Next, we have performed BO on the MPC case study whose results are given on Fig. 2. The objective here is different, as using the objective from LQR would lead to a similar behavior. This time, we aim to tune the controller so that the side thrust follows as close as possibly some reference input profile. Apparently, all solutions aim to track the reference, but the BO loop ensures that we discover a solution where particular emphasis is given to side thrust tracking.

$$\min_{\theta} \|u_{ref}^{st} - u_{MPC}^{st}\|_2 \quad (6)$$
$$s.t. \quad \theta \in \Theta, \quad (7)$$

In this case, all elements of our cost matrices Q and R are a function of 4 parameters, $\theta = [\alpha_c, \beta_c, \gamma_c, \delta_c]$. We performed BO for 30 iterations in order to find the parameters that minimize the objective of (6) and the results are given in Fig.7

The inset of the Fig.7 shows that the desired trajectory tracking is achieved, as expected, in the minimum. The ranges and the optimal value that is found are given in Table 1.

It should be mentioned that this is not necessarily the true global optimum, but the best discovered value given the computational budget available.

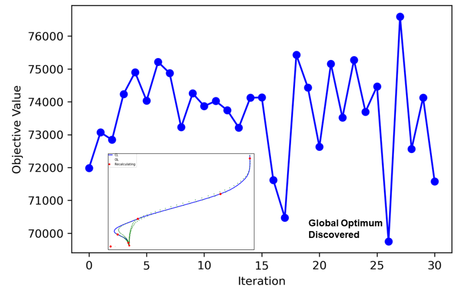


Fig. 7: Objective function evaluated during the BO iterations for the MPC case study

V. Conclusions

Using MPC, to solve the reference tracking problem of the open-loop LQR, we managed to land the rocket with zero angle orientation and angular speed, for the two scenarios that we simulated. Therefore, the MPC is proven to be an effective way to control the rocket landing on a moving platform. We also demonstrated how Bayesian Optimization techniques can help in discovering better tuning parameters by optimizing for arbitrary metrics related to the performance of the controlled system. Further work on the model can include cost on the fuel consumption, consideration of change in mass, aerodynamics, analysis of feasible sets, and reactions to wind disturbances.

References

[1] R. Ferrante, A Robust Control Approach for Rocket Landing. University of Edinburgh, 2017

[2] J. Snoek, H. Larochelle, R.P. Adams, Practical bayesian optimization of machine learning algorithms, Advances in neural information processing systems, 2012

[3] Q. Lu, R. Kumar, V.M. Zavala, MPC Controller Tuning using Bayesian Optimization Techniques, arXiv, 2020

[4] D. Piga, M. Forgione, S. Formentin, A. Bemporad, Performance-Oriented Model Learning for Data-Driven MPC Design, IEEE Control Systems Letters, 2019

TABLE I: Optimal tuning parameters for the MPC problem

Parameter	Range	Optimal Value
α_c	$[0.05, 0.125]$	0.088
β_c	$[0.3, 0.6]$	0.37
γ_c	$[4, 6]$	44.68
δ_c	$[1.5, 2.5]$	1.78