POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Control Techniques for Pinpoint Landing on Mars: Performance Enhancement through LQR and Model Predictive Control

Laurea Magistrale in Space Engineering - Ingegneria Spaziale

Author: Adriano Tessarollo

Advisor: Prof. Paolo Lunghi

Co-advisor: Ing. Paolo Martella

Academic year: 2024-2025

---

1. Introduction

Pinpoint landing on celestial bodies, such as the Moon and Mars, has emerged in recent years as a major challenge in space exploration. The increasing attention to this topic comes from the need for future missions to accurately reach predefined landing sites to support scientific, logistic, and potential human exploration objectives. The study of Mars in particular contributes to further broadening our knowledge of the Solar System and of planet formation and evolution.
In this scenario, the problem of pinpoint landing on Mars comes into play. Since the early 1960s, until recent years, different strategies have been adopted to land on Mars.
Today, the main critical aspect involving Mars landing is the transition from the so called ellipse landings to the precision landing. Ellipse landings are characterized by large landing areas, representing the uncertainty in the landing position on the ground.
These areas initially spanned hundreds of kilometers (Mars 3 from URSS, Viking 1 & 2 from NASA), to some kilometers in more recent missions (Mars Science Laboratory, Mars 2020 from NASA). Pinpoint landing aims to reduce the landing area from kilometers to some tens of meters from the target. The need of such precision is understandable considering the idea of building a base, having to move heavy materials; and the problem gets even worse taking into account humans landing kilometers away from the target. Among the many challenges posed by pinpoint landing, there is the delay in communications from Mars to Earth, which forces the system to be completely autonomous, requiring an advanced Guidance, Navigation, and Control subsystem (GNC).

1.1. Research objective

This study has been carried out in collaboration with Thales Alenia Space - Italy (TAS-I) in Turin, which provided the baseline GNC simulator in Matlab/Simulink.
The objective of this work, in the big picture of GNC, is to improve the performance of the simulator by implementing advanced techniques in the Control subsystem, to achieve pinpoint landing on Mars.
In particular, the baseline simulator is given with a Proportional-Integral-Derivative (PID) controller, and the steps are to implement a Linear Quadratic Regulator (LQR) controller and then a Model Predictive Control (MPC) logic.

2. Simulator, requirements and initial performances

The GNC simulator provided by TAS-I has been developed as design instrument in the frame of the studies for the ExoMars Rosalind Franklin Mission and has been later enhanced to support also the pinpoint landing. It is composed of the main modules representing the on-board software: Guidance, Navigation, and Control, and the Dynamics module, which models the real-world, therefore the environment, lander dynamics, kinematics, sensors and thrusters.

The Control block, which is the focus of this study, is composed of a translational controller, an attitude controller, a dispatch block to report the control actions in the thrusters space, and a gain manager suited for the PID controller.

The simulator has two sensors configurations: one with three sensors (inertial measurement unit, radar doppler altimeter, landing vision system) which will be referred to as RDA, and the other where the radar doppler is not provided and only the altimeter is available, which will be referred to as ALO (ALtimeter Only).

2.1. Landing requirements

The requirements at TouchDown (TD), in order to have a soft landing with stable attitude are:

- Vertical velocity $< 2.5\ m/s$
- Horizontal velocity $< 2\ m/s$
- Off vertical angular displacement $< 7\,°$
- Off vertical rate of displacement $< 3.2\,°/s$

The pinpoint landing requirements are:

- With RDA: Horizontal distance from the target < 60 m
- With ALO: Horizontal distance from the target < 100 m

The off vertical is the angle between the body vertical axis and the local vertical direction. Finally, there is a nice to have requirement to have propellant consumption lower than 300 kg.

2.2. Initial conditions

Performance analysis is conducted using a Monte Carlo analysis with a population of 100. Each Monte Carlo case has different initial conditions and, hence, different trajectory to ground, always with the aim of reaching the axes origin with a soft landing. The initial conditions for the Monte Carlo analysis are provided directly by TAS-I. The most relevant dispersions of the initial conditions are reported hereafter.

- Maximum initial distance from the target equal to 3000 m.
- Initial altitude with minimum around 2400m and maximum around 2700 m.
- Max. horizontal velocity around 7 m/s.
- Initial vertical velocity between 80 m/s and 100 m/s.

2.3. PID initial performances

In the following table, the initial performance of the simulator with PID in RDA configuration is reported. The results are presented in terms of percentiles, reporting the most relevant ones.

	50%	**99%**	**Max**	**Req.**
Hor. Dist. [m]	29.147	56.059	61.023	60
|Vert Vel.| [m/s]	1.494	2.432	2.458	2.5
Hor. Vel. [m/s]	0.427	1.059	1.090	2
Off Vert. Ang. [°]	0.617	1.643	1.733	7
Ang. @Spin Ax. [°]	0.027	0.678	0.707	
Trans. Ang. Rate [°/s]	1.349	3.489	3.653	3.2
Rate @Spin Ax. [°/s]	0.004	0.528	0.531	
Prop. Mass [kg]	230.731	291.043	292.035	

Table 1: PID touchdown values, RDA config.

As can be seen from the table, the PID controller almost satisfies all the requirements, slightly exceeding the horizontal distance one for pinpoint landing, and failing in the maximum value of the transversal angular rate (i.e. off vertical rate of displacement) at touchdown.

The requirements on velocities are met, even if with a maximum (absolute) vertical velocity of 2.458 m/s the requirement is only marginally satisfied. The same table is reported for the ALO configuration in Tab. 2. The performance is comparable to that of the previous configuration. Again, the PID struggles to satisfy the horizontal distance requirement, the transversal angular rate requirement is not met, and the vertical velocity maximum value lies close to the requirement limit.

	50%	**99%**	**Max**	**Req.**
Hor. Dist. [m]	33.349	93.353	101.168	100
|Vert Vel.| [m/s]	1.569	2.439	2.474	2.5
Hor. Vel. [m/s]	0.672	1.902	1.910	2
Off Vert. Ang. [°]	0.779	1.752	1.925	7
Ang. @Spin Ax. [°]	0.002	0.697	0.720	
Trans. Ang. Rate [°/s]	1.358	3.283	3.445	3.2
Rate @Spin Ax. [°/s]	0.007	0.546	0.564	
Prop. Mass [kg]	227.854	287.383	287.685	

Table 2: PID touchdown values, ALO config.

3. LQR control

The first control strategy investigated is an LQR controller, which is the most popular optimal control problem. Initially, it is applied only on translation, and subsequently an integrated controller in both translation and rotation is implemented. The mathematical formulation is taken from [1], where the goal is to find a control that minimizes the cost functional J. The scenario analyzed is a tracking problem; therefore, state and control vector are expressed as error variables.
Given the following state space representation:

$$\delta\dot{\mathbf{x}} = \mathbf{A}\,\delta\mathbf{x} + \mathbf{B}\,\delta\mathbf{u} \tag{1}$$

Where $\mathbf{A}$ is the $n \times n$ dynamics matrix and $\mathbf{B}$ is the $n \times m$ control matrix, after linearization, the aim is to minimize the cost function:

$$J(\mathbf{x}, t) = \int_t^\infty (\delta\mathbf{x}^\mathbf{T}\mathbf{Q}\,\delta\mathbf{x} + \delta\mathbf{u}^\mathbf{T}\mathbf{R}\,\delta\mathbf{u})\,d\tau \tag{2}$$

With $\mathbf{Q} = \mathbf{Q}^T \geq 0$ state weights (cost) matrix, and $\mathbf{R} = \mathbf{R}^T > 0$ control weights (cost) matrix. By solving the Riccati equation, the optimal control is computed as follows:

$$\mathbf{u}^* = -\mathbf{K}\mathbf{x} \tag{3}$$

Here, matrix $\mathbf{K}$ is the optimal state-feedback gain matrix.

3.1. LQR on translation

The mathematical model is implemented in the simulator that operates in the discrete-world with a sampling time for the controller of $\Delta t = 0.1\,s$.
The chosen working principle is to have an altitude-varying behavior of the weights in the matrix $\mathbf{Q}$, this choice enables a more refined tuning and a more accurate control of the lander. The weights selection is done exploiting the Bryson's rule; therefore, the diagonal terms of the weighting matrices are chosen as the inverse of the square of the maximum acceptable value of the corresponding state or control variable. These maximum acceptable values are derived by looking at the evolution of the errors as the altitude decreases during the simulation with the baseline simulator with the PID controller.
The same procedure is used, by looking at the maximum acceptable accelerations, to define the weights in the $\mathbf{R}$ matrix, which are instead kept constant.
To have altitude-varying gains, 1-D LookUp Tables on Simulink are exploited. The LookUp Table takes in input the estimated altitude given by the Navigation module, and gives as output the desired gain. The altitude is partitioned into arbitrarily defined breakpoints, building in this manner an altitude breakpoints vector Z_{bp}. At each breakpoint corresponds a chosen value of the gain.
For each breakpoint, the set of maximum acceptable values (for the translation in this case) has to be defined. This set of parameters is called σ, and they define a different $\mathbf{Q}$ matrix for each altitude breakpoint.
The gains are retrieved, after choosing the forms of $\mathbf{Q}$ and $\mathbf{R}$ matrices, following the algorithm reported hereafter.

Algorithm 1 Gains set computation

1: Empty gains vector definition, one for each non-zero term of the gain matrix $\mathbf{K}$.
2: **for** i from 1 to $length(Z_{bp})$ **do**
3: Define $\mathbf{Q}(i)$ with the $i-th$ set of σ
4: Compute $\mathbf{K}(i)$ matrix with `K = dlqr(A,B,Q,R)`
5: Allocate the non-zero terms of $\mathbf{K}$ to the previously created vectors.
6: **end for**
7: Gains vector computed, to be passed to the associated LUT together with the altitude breakpoint vector.

The output of this controller are the accelerations needed to control the lander. The accelerations are then multiplied by the mass, obtaining the force, which is then transformed into the space of the thrusters using the dispatch block.

3.1.1 Results

In Tab. 3 and Tab. 4 the results for the RDA and the ALO configurations are reported, respectively.
As can be seen, the performance is enhanced for both configurations in terms of translational control compared to the PID controller, with a lower distance from the target at TD and safer margins on the landing velocities.

	50%	99%	Max	Req.
Hor. Dist. [m]	26.404	50.537	50.658	60
|Vert Vel.| [m/s]	1.132	1.786	1.807	2.5
Hor. Vel. [m/s]	0.484	1.160	1.174	2
Off Vert. Ang. [°]	2.526	5.314	5.450	7
Ang. @Spin Ax. [°]	0.017	0.630	0.632	
Trans. Ang. Rate [°/s]	1.197	3.837	4.317	3.2
Rate @Spin Ax. [°/s]	0.007	0.639	0.663	
Prop. Mass [kg]	215.905	274.875	275.360	

Table 3: LQR on trans. TD values, RDA config.

	50%	99%	Max	Req.
Hor. Dist. [m]	30.521	85.324	92.553	100
|Vert Vel.| [m/s]	0.888	1.243	1.262	2.5
Hor. Vel. [m/s]	0.643	1.738	1.765	2
Off Vert. Ang. [°]	0.823	1.753	1.878	7
Ang. @Spin Ax. [°]	0.054	0.678	0.730	
Trans. Ang. Rate [°/s]	1.195	5.400	4.613	3.2
Rate @Spin Ax. [°/s]	0.006	0.609	0.616	
Prop. Mass [kg]	224.799	291.249	293.202	

Table 4: LQR on trans. TD values, ALO config.

Moreover, the propellant consumption in the RDA configuration is reduced by around 15 kg for each percentile. The rotational performance has not improved instead, as predictable having two different controllers for translation and attitude, and the performance in transversal angular rate is slightly larger than the requirement even if not critical.

3.2. Integrated LQR

Now, an integrated LQR controller in both rotation and translation is implemented. The working principle for the attitude controller is the same as that of translation.
The only implementation difference is that the control output is not the angular acceleration but directly the torque needed to control the lander. As before, LookUp Tables are used to have an altitude-varying behavior of the gains, exploiting the Algorithm 1.

3.2.1 Results

In Tab. 5 the performance with the RDA configuration in presented. The translational controller provides approximately the same results, while the rotational controller performance is improved.
In particular, the transversal angular rate requirement is now largely met, still keeping a good margin in the velocities touchdown values and still having a lower propellant consumption compared to the PID controller.

	50%	99%	Max	Req.
Hor. Dist. [m]	35.295	52.626	52.768	60
|Vert Vel.| [m/s]	1.184	1.778	1.833	2.5
Hor. Vel. [m/s]	0.401	1.185	1.201	2
Off Vert. Ang. [°]	3.407	5.242	5.277	7
Ang. @Spin Ax. [°]	0.053	0.677	0.719	
Trans. Ang. Rate [°/s]	0.377	0.821	0.828	3.2
Rate @Spin Ax. [°/s]	0.005	0.057	0.059	
Prop. Mass [kg]	214.121	274.682	275.030	

Table 5: LQR TD values, RDA config.

The results of the ALO configuration simulation are reported in Tab. 6. The most valuable improvement is given by the values of the horizontal landing distance from the target, which show a significant enhancement. The maximum value now is around 65 m, which means a gain of around 35 meters in accuracy compared to the PID, and approximately 27 meters compared to the previous simulation. Moreover, also here, the requirement on transversal angular rate is largely met, keeping a safe margin on the velocities TD values.

	50%	99%	Max	Req.
Hor. Dist. [m]	21.328	58.875	65.423	100
|Vert Vel.| [m/s]	0.994	1.986	2.116	2.5
Hor. Vel. [m/s]	0.725	1.970	1.976	2
Off Vert. Ang. [°]	0.952	2.055	2.084	7
Ang. @Spin Ax. [°]	0.048	0.678	0.716	
Trans. Ang. Rate [°/s]	0.202	0.530	0.579	3.2
Rate @Spin Ax. [°/s]	0.000	0.227	0.233	
Prop. Mass [kg]	205.325	283.683	289.531	

Table 6: LQR TD values, ALO config.

4. Model Predictive Control

The Model Predictive Control (MPC) is part, as the LQR, of the big family of optimal controllers. In short, MPC is a constrained receding-horizon optimization framework in which control actions are determined by solving the optimal control problem iteratively at each time step.

4.1. Problem formalization

The formalization hereafter presented follows the approach illustrated in [2].
The three fundamental parameters of MPC are:

1. **Sampling time.** $\Delta t = 0.1\,s$. In the MPC context, it defines the time span between two successive optimizations.
2. **Prediction Horizon.** It is the period of time covered by the optimization, computed as $p_H = N_s \Delta t$. N_s is the number of time steps covered, which is set to 12 in this work. Therefore, $p_H = 1.2\,s$.
3. **Control Horizon.** It defines how many of

the control actions computed by the MPC are applied. The chosen solution is to have a control horizon $N_m = 1$, recomputing the control actions each time step.

As MPC is an optimal control problem, the aim is to solve, at each time step k of the simulation:

$$min_{\mathbf{u}} J(\mathbf{x_i}, \mathbf{u_i}), \quad i = 1, ..., N_s \tag{4}$$

Where i is the $i-th$ step, called *stage*, in the prediction horizon length. Eq. 4 is subject to:

$$\mathbf{x}_{i+1} = f(\mathbf{x}_i, \mathbf{u}_i), \quad \mathbf{x}_1 = \mathbf{x}(k) \tag{5}$$

$$\mathbf{x}_i \in \mathbf{X}, \ \ \mathbf{u}_i \in \mathbf{U} \tag{6}$$

Where the initial conditions in Eq. 5 are given by the $k-th$ state, and in Eq. 6 the sets $\mathbf{X}$ and $\mathbf{U}$ are the state and control constraints. These represent one of the biggest advantages of MPC over LQR, allowing an explicit handle of physical limitations and actuator saturation. The general form of the cost function is as follows:

$$J(\mathbf{x}_k, \mathbf{u}_k) = \sum_{i=0}^{N_s-1} \left(\mathbf{x}_{k+i}^T \mathbf{Q} \mathbf{x}_{k+i} + \mathbf{u}_{k+i}^T \mathbf{R} \mathbf{u}_{k+i} \right) + \mathbf{x}_{k+N_s}^T \mathbf{S} \mathbf{x}_{k+N_s} \tag{7}$$

The landing maneuver comprises a divert phase to recover entry horizontal error and a final vertical descent; as a design choice, MPC is used during the divert and LQR during the final descent. This is done because, as the vehicle approaches the ground, as explained in [3], a standard receding-horizon MPC may continue to "pushing" the terminal event forward: the predicted landing time is repeatedly postponed, so the optimizer continues to generate commands that achieve landing in the future rather than committing to touchdown. This behavior can be solved with a decreasing-horizon MPC. Switching to LQR in the terminal phase plays an equivalent role, to practically reduce the horizon.

4.2. Modeling and implementation

The system of equations representing the state function is highly non-linear, therefore the MPC problem is extended to a **Non-Linear Model Predictive Control (NMPC)** problem.

In the cost function, it has been chosen not to include the terminal cost given by $\mathbf{S}$ matrix. This choice is made since deriving matrix $\mathbf{S}$ (solution of a Riccati-like equation) at each time step would significantly increase the computational burden, which is already relevant. The final form of the cost function adopted is:

$$J_k = \sum_{i=1}^{N_s} \left(\mathbf{e}_i^T \mathbf{Q} \mathbf{e}_i \, + \, \mathbf{u}_i^T \, \mathbf{R} \, \mathbf{u}_i \right) \tag{8}$$

The implementation principle is depicted in the block scheme in Fig. 1.

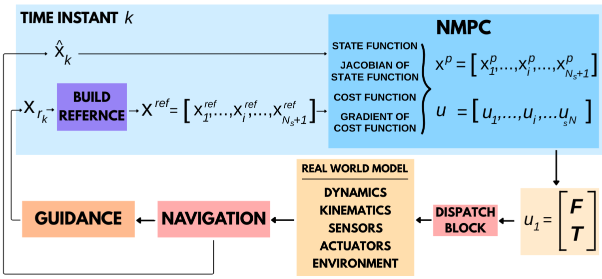

Figure 1: NMPC working principle.

The NMPC is implemented using the *Nonlinear Model Predictive Control Multiple Stages* block on Simulink. It takes in input the estimated state $\mathbf{\hat{x}_k}$ and the set of reference states $\mathbf{x^{ref}}$ along the prediction horizon, retrieved by propagating the state $\mathbf{x_{r_k}}$ provided by the Guidance using a dedicated reference builder function. Inside the NMPC block the *lander* object is defined, to which the following functions are associated: state function and its jacobian, cost function and its gradient. The NMPC block solves the optimization problem computing a set of N_s control actions, and therefore predicting a total of $N_s{+}1$ states. Then, only the first of the control actions is applied and the process is repeated at the next time instant. The constraints imposed are:

$$\begin{cases} z \geq 0 \\ 0 \leq m \leq 1500 \end{cases} \tag{9}$$

$$\begin{cases} -1500 \leq F_x \leq 1500 \\ -1500 \leq F_y \leq 1500 \\ 0 \leq F_z \leq 15000 \\ -2500 \leq \tau_x \leq 2500 \\ -2500 \leq \tau_y \leq 2500 \\ -1000 \leq \tau_z \leq 1000 \end{cases} \tag{10}$$

With z altitude, m mass, F forces, τ torques. The control constraints are chosen by looking at the trends of forces and torques in the simulations with LQR.

4.3. Results

The results for the RDA configuration are reported hereafter with the usual table.

	50%	99%	Max	Req.
Hor. Dist. [m]	24.852	38.940	40.385	60
|Vert Vel.| [m/s]	0.952	1.179	1.189	2.5
Hor. Vel. [m/s]	1.038	1.700	1.722	2
Off Vert. Ang. [°]	0.929	1.960	1.995	7
Ang. @Spin Ax. [°]	0.471	7.299	7.596	
Trans. Ang. Rate [°/s]	0.167	0.618	0.761	3.2
Rate @Spin Ax. [°/s]	0.002	1.886	1.914	
Prop. Mass [kg]	239.750	302.878	303.151	

Table 7: MPC TD values, RDA config..

As can be seen, all the requirements are met, and the main improvement given by the MPC is in the horizontal distance at TD, which is improved by more than 10 meters compared to the LQR and more than 20 meters compared to the PID. The main issue concerns the propellant consumption, which is increased by around 10 kg w.r.t. PID and by around 28 kg w.r.t. LQR.
Regarding the ALO configuration, the MPC does not provide completely satisfactory results. Different tuning strategies have been used, but none of them yielded a Monte Carlo analysis with 100/100 successful cases. Initially, after relaxing the weights on horizontal position, a solution with 94/100 successful cases is obtained. All of these 6 cases exceed the limit of horizontal velocity at TD of around 0.5 m/s. Although these velocities could be accepted, considering that they exceed the limit by a small margin, the main problem is in the landing accuracy and propellant consumption: significantly higher than with the other controllers, with distances from target over 100 meters, and consumption again almost around 300 kg.
A proposed solution is to reduce the prediction horizon, as [3] suggests. Therefore, the horizon is reduced from $p_H = 12$ to $p_H = 5$, but still 4 cases present a too high horizontal velocity at TD. The reason why this is happening is because, not having direct measurements on the velocities provided by the doppler, the performances of the MPC during the divert maneuver are degraded. This is understandable since the MPC have to perform a propagation forward in the future, and not having a precise estimation of the velocities does not allow an accurate predicted state, hence, the predicted control actions are not suitable in order to follow the reference profile.

5. Conclusions

To conclude, a final comparison table for both configurations is reported. In particular, in Tab. 8 for the RDA, every scenario analyzed is indicated, while in Tab. 9 the MPC is omitted, since the performance was not satisfactory.

Max Parameter	**PID**	**LQR Tr.**	**LQR**	**MPC**
Hor. Dist. [m]	61.023	50.658	52.768	40.385
|Vert. Vel.| [m/s]	2.458	1.807	1.833	1.189
Hor. Vel. [m/s]	1.090	1.174	1.201	1.722
Off-vert. Ang. [°]	1.733	5.450	5.277	1.995
Trans. Ang. Rat. [°/s]	3.653	4.317	0.828	0.761
Prop. Cons. [kg]	292.035	275.360	275.030	303.151

Table 8: RDA config.: performance comparison.

Max Parameter	**PID**	**LQR Tr.**	**LQR**
Hor. Dist. [m]	101.168	92.553	65.423
|Vert. Vel.| [m/s]	2.474	1.262	2.116
Hor. Vel. [m/s]	1.910	1.765	1.976
Off-vert. Ang. [°]	1.925	1.878	2.084
Trans. Ang. Rat. [°/s]	3.445	4.613	0.579
Prop. Cons. [kg]	287.685	293.202	289.531

Table 9: ALO config.: performance comparison.

In conclusion, the LQR manages to perform the pinpoint landing with good performances for both configurations, and the results are improved compared to those provided by the PID controller. In particular, the results with the LQR controller improved when an integrated LQR was implemented in both translation and rotation. The MPC succeeded in providing reliable and acceptable results with the RDA configuration, while with the ALO configuration it struggled to obtain a comfortable solution, due to its inherent looking-forward nature and the basic navigation sensor set, which highlights a limit in the command definition when the navigation is affected by larger uncertainties.

References

[1] Feng Lin. Robust control design – an optimal control approach, Publicated in 2007.

[2] V. Pesce and A. Colagrossi et al. In *Modern Spacecraft Guidance, Navigation, and Control – From System Modeling to AI and Innovative Applications*, chapter 10, pages 609–612. 2023. Edited by F. Cavenago et al.

[3] G. Z. Prous et al. A decreasing horizon model predictive control for landing reusable launch vehicles, January 2025. URL `https://doi.org/10.3390/aerospace12020111`.