CEAS-GNC-2024-043

EuroGNC Navigation Guidance Control CEAS

Bristol, UK June 11th-13th 2024

University of BRISTOL ROYAL AERONAUTICAL SOCIETY AIAA

Reinforcement Learning based adaptive control of landing manoeuvres in uncertain environments

Inês Zagalo GNC Engineer, Deimos Engenharia, 1070-061, Lisboa, Portugal. ines.zagalo@deimos-space.com

Paulo Rosa Head of the Avionics Business Unit, Deimos Engenharia, 1070-061, Lisboa, Portugal. paulo.rosa@deimos.com.pt

J. M. Lemos Professor, INESC-ID, Instituto Superior Técnico, University of Lisbon, 1000-029 Lisboa, Portugal jlml@inesc-id.pt

ABSTRACT

This work addresses the benefits and limitations of reinforcement learning (RL) - based adaptive control for the vertical landing of reusable launch vehicles. The RL algorithm selected is the so-called Q-learning, that allows to learn the optimal controller directly from plant data. The focus is on the development of an adaptive, model-free controller so that the launcher is able to follow a nominal predefined trajectory, even when subjected to disturbances and uncertainty. The main contribution of this work is the development of a cascaded control structure in which each loop is composed of a RL controller. The robustness of the proposed approach is tested in several scenarios and its performance is compared with that obtained with a Linear Quadratic Regulator (LQR). The main challenge in learning-based control approaches to the problem of vertical landing is the short trajectory duration, which limits the period for control adaptation. Another significant challenge consists in training the inner loop and outer loop controllers, that must be done with appropriate time-scales. By presenting the results obtained with the RL controller as well as its analysis, the study allows understanding, evaluating, and comparing its operating limits with the LQR control method in nonlinear simulations with parameter uncertainty.

Keywords: Reusable launch vehicles; Adaptive Control; Reinforcement Learning (RL); Q-learning; Linear Quadratic Regulator (LQR); Cascade control

Nomenclature

ψ, θ, ϕ	=	Euler angles (heading, pitch, roll)
u, v, w	=	velocity Cartesian components
p, q, r	=	angular velocity components
F	=	force
M	=	moment
ω	=	angular velocity vector
m	=	mass
thr	=	thrust force

1 Introduction

Reusability has been a major goal in the rocket industry. Reusable rockets are now the main driver in the new space economy since they allow companies to reduce cost, making space exploration more accessible and affordable, while reducing the amount of space debris generated by launches [1] [1].

Reinforcement learning is a class of machine learning methods that has been used in several applications. In general, the use of reinforcement learning techniques in optimal control [2, 3] has widened the horizon of the field and has overcome some of the limitations, such as the need for a full dynamic model in most of the traditional optimal control methods, providing model-free solutions to optimal control problems.

In this work, a reinforcement learning algorithm is adopted, driven by the fact that the reusability of a launcher implies landing trajectories that start at points of high uncertainty, accounting for uncertain launcher dynamics, significant wind disturbances, while, as a consequence of the ascent burn, the performance of the actuators may be degraded. Given the high uncertainty and the need to follow a reference trajectory accurately, classical control methods are challenged and adaptive control is thus selected. The main difficulty in its implementation consists in the short length of the trajectory, thereby reducing the period to train the controller.

The RL algorithm selected is Q learning, since the problem in question is a regulator problem, the system dynamics is approximately linear and the cost is quadratic, allowing the parameters of an approximation of the Q-function to be learn without knowing the system dynamics and developing an online solution for discrete-time systems with infinite horizon.

The goal of this study is to develop and test in simulation a control system for landing manoeuvres of reusable launch vehicles in highly uncertain environments, in order to compare adaptive control algorithms based on models with local controllers designed with robust controllers with reinforcement learning based adaptive controllers that adjust an initial robust controller, in the presence of uncertainties and sudden changes in dynamics ([4]).

2 Theoretical background

2.1 LQR controller

The objective of the linear controllers is to stabilize the system around the different equilibrium points. The reason for approximating the nonlinear system by a linear model is that, by so doing, one can apply rather simple and systematic linear control design techniques.

In order to design a linear controller, first of all, the system to be controlled has to be linear. The nomenclature considered for the discrete linear model is the following:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k). \end{aligned}$$

There are several types of linear controllers, however, in this work the LQR controller, a full state feedback controller, will be used due to the fact that it is an optimal controller that can provide not only a good stability, but also the stability margin of a system is guaranteed and it provides a better optimal energy compared to other linear controllers.

[1] R. Mike,The History Of Reusable Rockets, URL: `https://cosmospnw.com/history-of-reusable-rockets/`

Linear quadratic regulator (LQR) is a method that optimizes the linear feedback gains of state variables of a linear plant [5]. It is intended to minimize a cost function with the constraint imposed by the system dynamics. The cost function is given by

$$J = \frac{1}{2}\sum_{k=0}^{\infty}[\mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k) + \mathbf{R}\mathbf{u}^2(k)], \tag{1}$$

where **Q** (positive semi-definite) and **R** (positive definite) are state and input symmetric weighting matrices, respectively. Minimising the cost function provides optimal feedback

$$\mathbf{u}(k) = -\mathbf{K}\mathbf{x}(k), \tag{2}$$

which is a feedback of the state **x** with constant gain **K**.

The gain matrix, **K**, is calculated using the following expression:

$$\mathbf{K} = (\mathbf{B}^T\mathbf{S}\mathbf{B} + \mathbf{R})^{-1}\mathbf{B}^T\mathbf{S}\mathbf{A} \tag{3}$$

where the matrix **S** is the solution of the Riccati equation. The gain matrix **K** is calculated using the Matlab function: [K,S,E] = dlqr(A,B,Q,R).

It can be seen that as the value of **R** increases there are advantages and disadvantages. The advantages are that the amplitude of u decreases and the sensitivity to high frequency noise and modelling errors is reduced. The disadvantages are that the response of y to disturbances in the system increases and the system becomes slower. For a smaller value of **R**, the cost of the control parcel is lower allowing y to be smaller.

2.2 Reinforcement Learning Algorithms

Reinforcement learning is a machine learning algorithm in which the most important feature distinguishing it from other types of learning is that it refers to an actor or agent that interacts with its environment and modifies its actions, or control policies, based on stimuli received in response to its actions in order to maximize a reward or, equivalently, minimize a cost [6, 7]. In the optimal RL algorithms, the interest is no longer the system dynamics but a performance index that quantifies how close to optimality does the closed-loop control system operate. Thus, optimal behaviors are learned by observing the response from the environment to non-optimal control policies.

The type of RL algorithms to apply to a specific problem depends on its characteristics. The problems can have a finite or an infinite horizon. Although the construction of finite horizon problems is essentially realistic, it may be still impractical in large scale real problems, due to the curse of dimensionality. Thus, one simple solution to deal with these problems is to simply leave the terminal time unspecified and open. Futhermore, the state and action space can be continuous or discrete. In this study a continuous state space and action space will be considered, meaning that the states and possible actions are not confined to a previously defined set.

2.2.1 Problem Definitions and Important Concepts

It is considered the class of **systems** defined by

$$x_{k+1} = f(x_k) + g(x_k)u_k, \tag{4}$$

where $x_k \in R^n$ is the state and $u_k \in R^m$ the control input. A **control policy** is a function from the state space to the control space, that for every state defines a control action. In reinforcement learning, the control policy is learned in real time based on stimuli received from the **environment** [6]. The goal directed optimal behavior is captured defining the **performance measure (cost function)**

$$V_h(x_k) = \sum_{i=k}^{\infty} \gamma^{i-k} r(x_i, u_i), \tag{5}$$

with $0 < \gamma \leq 1$ the discount factor introduced such that the cost function remains bounded [6]. The performance index measures the cost-to-go of the state x from the current time k to the infinite horizon future with a defined feedback control policy. Here, the quadratic energy function

$$r(x_k, u_k) = x_k^T \mathbf{Q} x_k + u_k^T \mathbf{R} u_k, \tag{6}$$

where **Q** and **R** are positive definite matrices.

The **Bellman equation**, of which $V_h(x_k)$ is the unique solution, expresses a relationship between the value of a state and the values of its successor states and is given by

$$V_h(x_k) = r(x_k, h(x_k)) + \gamma V_h(x_{k+1}). \tag{7}$$

The Bellman's principle of optimality states: "An optimal policy has the property that, no matter what the previous controls have been, the remaining decisions must constitute an optimal policy with regard to the state resulting from those previous decisions" [6]. Accordingly,

$$V^*(x_k) = \min_{h(.)}(r(x_k, h(x_k)) + \gamma V^*(x_{k+1}), \tag{8}$$

which is the so-called **Bellman optimality equation**. The optimal policy is obtained from

$$h^*(x_k) = \arg\min_{h.}(r(x_k, h(x_k)) + \gamma V^*(x_{k+1}). \tag{9}$$

This equation defines a backwards procedure to determinate the value function.

The Bellman equation and the optimal Bellman equation are **fixed-point equations** used to develop forward-in-time methods. This means that, given an admissible policy $u_k = h(x_k)$, they have a unique fixed point, $V_h(x_k)$ and $V_h^*(x_k)$ respectively. Starting with any value of $V^0(x_k)$, it converges to $V_h(x_k)$ or $V^*(x_k)$ respectively. Using this property, there are two important algorithms: value iteration and policy iteration. Both algorithms have two main steps: the first step corresponds to calculation of the value function and the second step is responsible for the improvement. The main difference between both is in the way the evaluation step is considered. While in policy iteration the value of the current policy is determined using the Bellman equation, in the value iteration, the value of the states is updated using one iteration of the Bellman equation.

The main idea of **value function approximation** is to construct a relatively low dimensionally parameterised space for approximating the total cost functions. In this work, the value function is approximated using a polynomial basis functions.

2.2.2 Q Learning

Using the value function has the drawback that it does not depend explicitly on the control variable, implying that the plant model must be used. An alternative is Q learning, an algorithm that allows the learning of the value function where the model parameters are not known nor needed to be estimated and that explicitly depends on the control.

The Q (quality) function associated with the policy $u = h(x)$ is defined as

$$Q_h(x_k, u_k) = r(x_k, u_k) + \gamma V_h(x_{k+1}), \tag{10}$$

which is a function of both the state x_k and the control u_k at time k [6].

The optimal Q function is defined by

$$Q^*(x_k, u_k) = r(x_k, u_k) + \gamma V^*(x_{k+1}). \tag{11}$$

The Bellman Optimality equation is terms of Q^* is given by

$$V^*(x_k) = \min_u Q^*(x_k, u) \tag{12}$$

and the optimal control is given by

$$h^*(x_k) = \arg\min_u(Q^*(x_k, u)). \tag{13}$$

If there are no control constraints, the minimum value is obtained by

$$\frac{\partial}{\partial u} Q^*(x_k, u) = 0, \tag{14}$$

which can be calculated without knowing the system dynamics [6].

LQR case

According to (10), the Q function for the LQR case is

$$Q_K(x_k, u_k) = x_k^T \mathbf{Q} x_k + u_k^T \mathbf{R} u_k + x_{k+1}^T \mathbf{P} x_{k+1}, \tag{15}$$

where **P** is the solution to the Lyapunov equation for the gain (or policy) **K**.

Assuming the parametric approximator of Q of the form

$$Q_h(x, u) = \mathbf{W}^T \boldsymbol{\varphi}(x, u) = \mathbf{W}^T \boldsymbol{\varphi}(z) \tag{16}$$

with $\boldsymbol{\varphi}(x, u)$ a basis set of activation functions. In the LQR, the basis functions are all the combinations of products of two different entries of the state.

RLS or gradient-descent can be used to identify the Q function associated to a given policy **K**. These methods are used to estimate the parameters **W** using data from the system.

In the LQR case,

$$Q_K(x_k, u_k) = z_k^T \mathbf{H} z_k = \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T \begin{bmatrix} \mathbf{H}_{xx} & \mathbf{H}_{xu} \\ \mathbf{H}_{ux} & \mathbf{H}_{uu} \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix}. \tag{17}$$

Thus,

$$\mathbf{H}_{ux} x_k + \mathbf{H}_{uu} u_k = 0$$

$$u_k = -(\mathbf{H}_{uu})^{-1} \mathbf{H}_{ux} x_h \tag{18}$$

Since the quadratic kernel matrix **H** has been found using online reinforcement learning, the system dynamics is not needed for this step.

2.2.3 Q Learning Policy Iteration Algorithm

Initialization: Select any admissible (i.e., stabilizing) control policy $h_0(x_k)$.

Policy Evaluation Step: Determine the least-squares solution W_{j+1} to

$$\mathbf{W}_{j+1}^T(\boldsymbol{\varphi}(z_k) - \gamma\boldsymbol{\varphi}(z_{k+1})) = r(x_k, h_j(x_k))$$

Policy Improvement Step: Determine an improved policy using

$$h_{j+1}(x_k) = \arg\min_{h(.)}(\mathbf{W}_{j+1}^T\boldsymbol{\varphi}(x_k, u)) \tag{19}$$

Since $u_k = -\mathbf{K}x_k$, u_k depends on x_k therefore, the persistence of excitation needed on $(\boldsymbol{\varphi}(z_k) - \gamma\boldsymbol{\varphi}(z_{k+1}))$ doesn't exist. Thus, one must add noise, $u_k = -\mathbf{K}x_k + \eta_k$.

3 Methodology

For the development of the equations of motion of a vehicle in free flight, the following assumptions were made: the vehicle is a rigid body with fixed mass distribution; the position of the center of gravity is fixed and its position is the same as the one of the center of mass; the air is considered at rest relative to the Earth and the Earth surface can be approximated as flat due to the short duration of the landing.

The mathematical model describing the behaviour of a vehicle in free flight, based on Newton's and Euler's laws of a rigid-body motion, is represented by a continuous-time nonlinear state-space model [8].

The general structure is given by:

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)) & x(0) = x_0 \\ y(t) = g(x(t)) \end{cases} \tag{20}$$

where t is the time variable and the vectors $x(t)$, x_0 and $y(t)$ represent the state variables, their initial condition and the output variables, respectively. The system dynamics are governed by f and g, which are nonlinear functions of the state variables and model parameters.

First, the equations defining the 6DoF system were determined and then the system was linearised and discretized around the hover position. With the nominal trajectory defined, an LQR controller was designed as well as an RL controller. It is stressed that the linearised model is only required by LQR, the controller based on RL being model free. The nonlinear model was used for simulations of the controlled vehicle.

The aerodynamic coefficients were provided by [9]. For stable flight, the center of gravity must be above the center of pressure. In this work, its position is considered constant.

3.1 6DoF Landing Problem

For the development of the motion equations, which are needed to test the RL controller, the following reference frames have been considered:

- Recovery Pad frame: centred on the landing point, it is an Up-East-North reference frame, with the X axis pointing upwards and the Y (to east) and Z (to north) axes in the perpendicular plane. The Recovery pad frame is the inertial frame considered since the rotation of the Earth is ignored. This is a valid assumption because only the final phase of the landing is considered.

- Body frame: follows the body motion, centred on the centre of mass and with the x-axis coincident with the body symmetry axis with positive direction pointing downrange through the vehicle nose. Axes y and z in the perpendicular plane and with the directions of the aerodynamic fins. In this study, it can be assumed that Oxz and Oxy planes are planes of symmetry of the vehicle.

The figure 1 shows the angles in the axis system used.

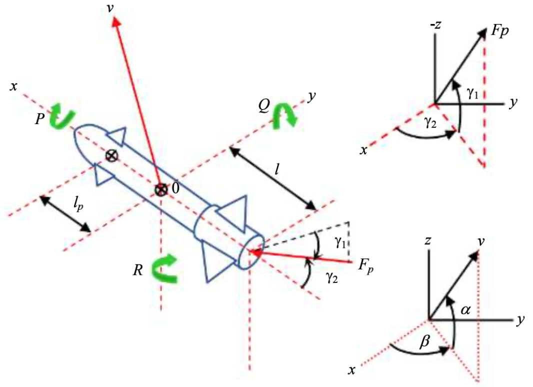


Fig. 1 Propulsion force from the nozzle of the rocket [10].

Translational motion The Newton's second law of motion can be written as

$$\mathbf{F_I} = \dot{m}\mathbf{V_I} + m\dot{\mathbf{V}}_\mathbf{I}, \tag{21}$$

where **F** are the forces applied on the body and $\mathbf{V_I}$ is its velocity in the inertial frame. The forces applied to the rocket are the thrust, the gravitational force, and the aerodynamic forces.

With regard to the change in mass, it follows that $m = m_0 - \frac{1}{I_{sp}} \int_0^t ||\mathbf{F}_{p_{ref}}||_2 \, dt$ where $\mathbf{F}_{p_{ref}}$ (N) is the reference thrust, I_{sp} (s) is the specific impulse of propellant, m_0 (kg) is the rocket mass at time zero and t (s) is the simulated time.

Rotational motion For a rocket, the rotational motion can be described as

$$\dot{\omega} = I^{-1}[\mathbf{M} - \omega \times \mathbf{I} \cdot \omega], \tag{22}$$

where **I** is the inertia tensor (which actually varies along the path because the mass decreases), $\boldsymbol{\omega}$ is the angular velocity, **M** is the sum of the torques and the operator × is the outer product. The tensor of inertia is a matrix 3×3.

Kinematic equations The attitude of a vehicle in flight is defined as the angular orientation of the body with respect to Earth-fixed axes. Since in this case nearly rectilinear trajectories are dealt with, it is reasonable to express the kinematic equations using Euler angles [8]. In 6DoF simulations, the rocket attitude is computed directly by integrating the set of equations that define Euler angle rates.

$$\begin{cases} \dot{\phi} = p + (q \sin\phi + r\cos\phi)\tan\theta \\ \dot{\theta} = q\cos\phi - r\sin\phi \\ \dot{\psi} = \frac{(q\sin\phi + r\cos\phi)}{\cos\theta} \end{cases} . \tag{23}$$

3.2 Forces and moments acting on the body

Since the earth is assumed to be flat, the gravitational force always has the direction $\mathbf{e}_{x_I}$ and a negative sign. Near the surface of the earth, $\mathbf{F}_{\text{grav}} = mg\mathbf{e}_X$ with $g(x) = g_0(\frac{R_E}{R_E+x})^2$, where g_0 is the gravitational acceleration, $R_E = 6371\text{km}$ is the earth's radius and x is the rockets altitude. Since the altitudes considered in the problem are relatively low, this simplified model can be used.

Considering the thrust model, $\mathbf{F}_{\text{thr}}$ is the thrust force vector, γ_2 is the angle from x_b-axis projecting the thrust vector $\mathbf{F}_{\text{thr}}$ on $x_b y_b$-plane and γ_1 is the angle projecting thrust vector $\mathbf{F}_{\text{thr}}$ on $x_b y_b$-plane to the thrust vector $\mathbf{F}_{\text{thr}}$. When the thrust force is not aligned with the x-axis of the body ($\gamma_1 \neq 0$ or $\gamma_2 \neq 0$), it produces momentum and the arm is given by the distance between the point of application of the force (motor) and the centre of mass: l_{Thr}.

The components of the propulsion force on the body frame are given by:

$$\begin{cases} F_{\text{thr}_{x_b}} = F_{\text{thr}} \cos\gamma_1 \cos\gamma_2 \\ F_{\text{thr}_{y_b}} = F_{\text{thr}} \cos\gamma_1 \sin\gamma_2 \\ F_{\text{thr}_{z_b}} = -F_{\text{thr}} \sin\gamma_1 \end{cases} \Leftrightarrow \begin{cases} \gamma_1 = \arctan \frac{F_{\text{thr}_{z_b}}}{\sqrt{F^2_{\text{thr}_{x_b}} + F^2_{\text{thr}_{y_b}}}} \\ \gamma_2 = \arctan \frac{F_{\text{thr}_{y_b}}}{F_{\text{thr}_{x_b}}} \end{cases} \tag{24}$$

Since $\mathbf{r}_{\text{thr}} - \mathbf{r}_{\text{cm}} = (r_{\text{thr}} - r_{\text{cm}})\mathbf{e}_{x_b} = l_{\text{thr}}\mathbf{e}_{x_b}$,

$$\mathbf{M}_{\text{Thr}} = l_{\text{thr}}\mathbf{e}_{x_b} \times F_{\text{thr}} \begin{bmatrix} \cos\gamma_1 \cos\gamma_2 & \cos\gamma_1 \sin\gamma_2 & -\sin\gamma_1 \end{bmatrix}^T$$

3.3 Guidance architecture

Although for landing there are several phases to be considered, in this study, only the powered descent and landing phases are considered, where the engine is ignited, the thrust force vector is used to control the translational motion of the vehicle – attitude pitch and yaw motion is controlled by the TVC system and the RCS provide roll control. Thus, the trajectory is divided in two phases: an initial curve in which the rocket starts with a non-zero attitude (yaw or pitch) and the TVC system is used to reach zero attitude, and the vertical trajectory which starts with zero attitude and the goal is to allow to touch down on the targeted landing area with accuracy and a sufficiently low velocity that can be absorbed by the landing system.

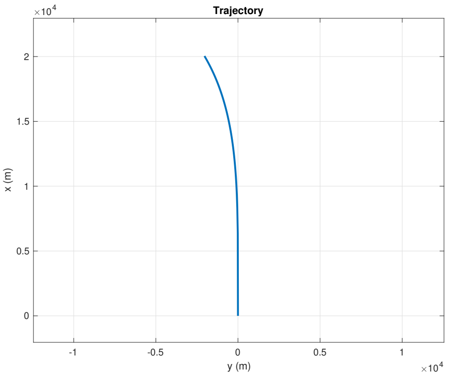


Fig. 2 Nominal trajectory.

The nominal trajectory is defined, without loss of generality, in the *Oxy* plane, since it facilitates the process and represents the simplest possible trajectory. Using a rotation of reference frames, it allows any type of 3D trajectory to be obtained. The nominal trajectory is represented in figure 2, which is not optimal and lasts approximately for 180 seconds.

Thus, in the design of the linear controller, the following models are considered:

- Motion in the xy-plane, with the controller on the yaw angle through the deflection of the TVC γ_2 angle. $x_{\text{yaw}} = \begin{bmatrix} m & x & y & u & v & r & \psi \end{bmatrix}^T$
- Motion in xz-plane with the controller on the pitch angle through the deflection of the TVC γ_1 angle. $x_{\text{pitch}} = \begin{bmatrix} m & x & z & u & w & q & \theta \end{bmatrix}^T$

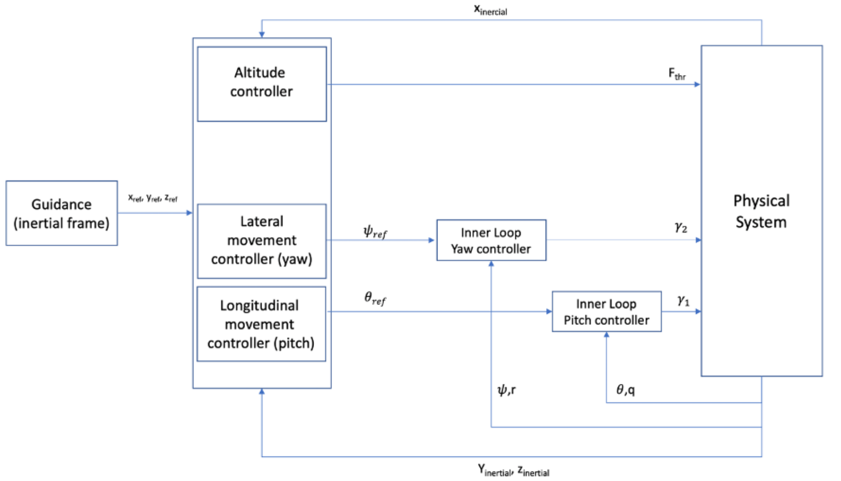
The mass is present in both models and is common to them, however, it is not a variable that can be controlled. Besides this, the roll motion is not considered because it is considered that the control of this variable is made separately (with aerodynamic fins for example) so that it is almost zero, by having a roll controller that maintains a zero roll rate despite disturbances, the pitch and yaw motion can be separately controlled.

3.4 Control architecture

The objective of the linear controllers is to stabilize the system around the different equilibrium points. When designing the controller, the nominal system will be considered, and thus no disturbances, such as aerodynamic forces, are considered.

Given that the coupling effects between the lateral and longitudinal dynamics are negligible, the two modes can be decoupled.

First, the rocket is symmetrical around the x_b axis (vertical) therefore, the inertia matrix is diagonal and the coupling effects between the movements in the planes xy and xz are minimal. Furthermore, the fact that the control surfaces used are conventional, the aerodynamic fins are mainly responsible for the roll control, the magnitude of thrust force and γ_2 are responsible for the movement in the xy plane and the magnitude of thrust force and γ_1 are responsible for the movement in the xz plane. The last reason is that the roll angle is practically zero since it is assumed that the roll control system is faster than the others.



$x_{inercial}$
Altitude controller
F_{thr}
Guidance (inertial frame)
x_{ref}, y_{ref}, z_{ref}
Lateral movement controller (yaw)
ψ_{ref}
Inner Loop Yaw controller
γ_2
Physical System
Longitudinal movement controller (pitch)
θ_{ref}
Inner Loop Pitch controller
γ_1
ψ,r
θ,q
$y_{inertial}$, $z_{inertial}$

Fig. 3 Control scheme.

Thus, in the design of the linear controller, the following models are considered: motion in the xy-plane, with the controller on the yaw angle through the deflection of the TVC γ_2 angle and motion in xz-plane with the controller on the pitch angle through the deflection of the TVC γ_1 angle.

After decoupling the yaw, pitch and roll motion, the linear states must be separated from the angular states, since the attitude controller must be faster than the position controller. Therefore, a cascaded controller will be used, where the inner loop (attitude controller) bandwidth must be wider than the outer loop (position controller) bandwidth. By using the cascade controller, the disturbances of the inner loop do not propagate to the outer loop. The outer loop is used to provide the desired attitude angle, and the inner loop is used to track these angles to obtain the desired position and speed.

Due to pitch and yaw symmetry in geometry, mass distribution, and the identical actuating systems for the gimbal angles, γ_1, γ_2 respectively, the feedback gains for both pitch and yaw loops can be the same. Thus, the linear control architecture for yaw is studied in detail and the one for pitch is similar.

The nomenclature considered for the continuous-time linear model is the following:

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

Yaw: linear state representation

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{u}\\ \dot{v}\\ \dot{r}\\ \dot{\psi}\end{bmatrix}=\begin{bmatrix}0 & 0 & c\psi_0 & -s\psi_0 & 0 & -u_0 s\psi_0 - v_0 c\psi_0\\ 0 & 0 & s\psi_0 & c\psi_0 & 0 & u_0 c\psi_0 + v_0 s\psi_0\\ 0 & 0 & 0 & r_0 & v_0 & g s\psi_0\\ 0 & 0 & -r_0 & 0 & -u_0 & g c\psi_0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x\\ y\\ u\\ v\\ r\\ \psi\end{bmatrix}+\begin{bmatrix}0 & 0\\ 0 & 0\\ \frac{c\gamma_{20}}{m} & \frac{-F_{\mathrm{thr}_0} s\gamma_{20}}{m}\\ \frac{s\gamma_{20}}{m} & \frac{F_{\mathrm{thr}_0} c\gamma_{20}}{m}\\ \frac{l_{\mathrm{thr}} s\gamma_{20}}{I_{zz}} & \frac{F_{\mathrm{thr}_0} l_{\mathrm{thr}} c\gamma_{20}}{I_{zz}}\\ 0 & 0\end{bmatrix}\begin{bmatrix}F_{\mathrm{thr}}\\ \gamma_2\end{bmatrix} \tag{25}$$

Linearizing around $v_0, r_0, \psi_0 = 0$, $u_0 = 0$ m/s, $\gamma_{20} = 0$ rad and the equilibrium value for F_{thr} (hover state) with the initial mass value:

$$A=\begin{bmatrix}0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & g\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0\end{bmatrix} B=\begin{bmatrix}0 & 0\\ 0 & 0\\ \frac{1}{m} & 0\\ 0 & \frac{F_{\mathrm{thr0}}}{m}\\ 0 & \frac{F_{\mathrm{thr}_0} l_{\mathrm{thr}}}{I_{zz}}\\ 0 & 0\end{bmatrix} \tag{26}$$

It can be seen that the vertical motion and the lateral motion are decoupled. The thrust force acts only on the vertical motion while the gimbal angle is the actuator used for the lateral motion.

The yaw state space in discrete time is the following (sampling frequency 100Hz):

$$\mathbf{x}(k+1)=\begin{bmatrix}x(k+1)\\ y(k+1)\\ u(k+1)\\ v(k+1)\\ r(k+1)\\ \psi(k+1)\end{bmatrix}=\begin{bmatrix}1 & 0 & 0.01 & 0 & 0 & 0\\ 0 & 1 & 0 & 0.01 & 1.6350\times10^{-6} & 4.9050\times10^{-4}\\ 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 4.9050\times10^{-4} & 0.0981\\ 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 0.01 & 1\end{bmatrix}\begin{bmatrix}x(k)\\ y(k)\\ u(k)\\ v(k)\\ r(k)\\ \psi(k)\end{bmatrix}+ \\ \begin{bmatrix}4.7511\times10^{-9} & 0\\ 0 & 4.9048\times10^{-4}\\ 9.5022\times10^{-7} & 0\\ 0 & 0.0981\\ 0 & -0.0442\\ 0 & -2.2077\times10^{-4}\end{bmatrix}\begin{bmatrix}\Delta F_{\mathrm{thr}}\\ \Delta\gamma_2\end{bmatrix} \tag{27}$$

Pitch: linear state representation

$$\begin{bmatrix} \dot{x} \\ \dot{z} \\ \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & c\theta_0 & s\theta_0 & 0 & -u_0 s\theta_0 + w_0 c\theta_0 \\ 0 & 0 & -s\theta_0 & c\theta_0 & 0 & -u_0 c\theta_0 - w_0 s\theta_0 \\ 0 & 0 & 0 & -q_0 & -w_0 & g s\theta_0 \\ 0 & 0 & q_0 & 0 & u_0 & -g c\theta_0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ z \\ u \\ w \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{c\gamma_{10}}{m} & \frac{-F_{\text{thr}_0} s\gamma_{10}}{m} \\ \frac{s\gamma_{10}}{m} & \frac{F_{\text{thr}_0} c\gamma_{10}}{m} \\ \frac{l_{\text{thr}} s\gamma_{10}}{I_{yy}} & \frac{F_{\text{thr}_0} l_{\text{thr}} c\gamma_{10}}{I_{yy}} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} F_{\text{thr}} \\ \gamma_1 \end{bmatrix} \quad (28)$$

Analysis of poles and zeros of the discrete linear system The outer loop control is hard to design. The poles and zeros of the lateral movement (considering y, v, ψ and r and the inner loop control) are: stable poles and the zero in $z = -1$ from the inner loop controller; two poles in $z = 1$; one zero inside the unit circle (from the discretization process) and one non minimal phase zero (outside the unit circle). Because of the two poles in $z = 1$ and the zero outside the unit circle, the outer loop controller is more complex than expected and a single feedback gain of y does not stabilize the system. Thus, all the state variables need to have a feedback gain.

4 Results and Discussion

4.1 RL controller

The RL control algorithm was added to the control architecture. Therefore, a learning time (TLearn) is defined for each controller. For time instants up to this value, the LQR controls the system, while RL just learns the optimal gains. From that moment on, the gains used to control the system are those calculated by RL.

The addition of dither is fundamental for parameter estimation. Thus, dither is added to the actuators' input (γ and thrust force) as well as to the reference angle (ψ or θ) for the estimation of the parameters of the external loop controller. The dither has two parameters to adjust: sampling time and noise power. It is important to emphasise that the adjustment that is made to the dither parameters depends on the operating/perturbation range that the controller is intended to have.

While in the case of the LQ controller it is possible to define an observer, this is not the case with the RL controller. The main advantage of using the RL controller is that no knowledge about the system dynamics is required. For the observer design, the system dynamics have to be known. It is thus assumed that the state is accessible. However, when the state is not accessible, the output and its derivatives can be used, typically only the first, which corresponds to a PD or PID controller if integral action is used.

The performance of the RL controller - learning time and closed loop system response - depends on the values given to the parameters that can be adjusted, namely, the LQR parameters, the discount factor, the estimation algorithms initialization parameters, the learning time of each controller and the dither. When defining the dither sequence one must keep in mind that there is a compromise: the higher the dither, the better and faster the estimation of the parameters is. However, the actuators have limitations and the higher the dither, the more oscillations the system presents. Another possibility is to consider a dither with variable power throughout the simulation. Simulations were made but the controller performance does not change much. A disadvantage of this method consists in obtaining a dither sequence that only presents good results for the seed considered in its definition.

The designed controller has the following learning times: 20 s (attitude control), 40 s (lateral position controller) and 50 s (vertical movement controller) which represent a small part of the trajectory.

4.2 Performance metrics

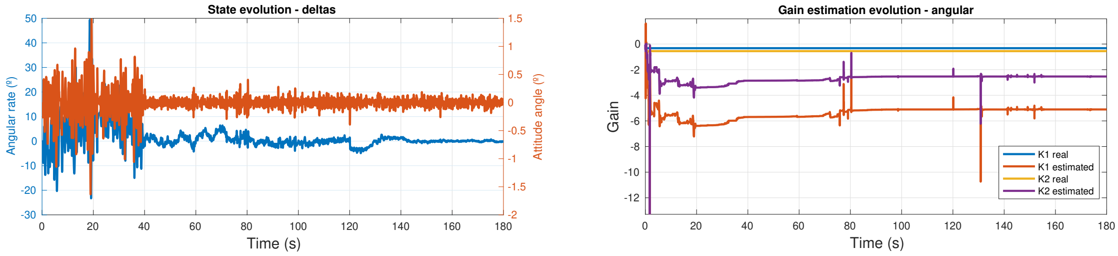
The performance metrics of the controlled system considered are:

- On landing (since a precise landing is important): lateral deviation of position and linear velocity as well as angular deviation and rate of variation.
- Along the trajectory: average value of control signals; measurement of fuel consumed; mean position error along the trajectory.

4.3 Bandwidth analysis - cascade controller

For a cascaded controller to work properly, the bandwidth of the outer loop controller must be lower than that of the inner loop, which has to be faster.

When setting the parameters of the LQR controllers, in the nominal situation, this limitation was met, however, it was found that the RL controller (starting at 40 s) of the inner loop learned different gains from the theoretical ones, giving rise to a better system response (figure 4). The fact that the RL controller has learnt different gains from the theoretical ones, does not mean that they are wrong, and this situation exemplifies one of the great advantages of the RL controller: when considering the aerodynamic forces, the actuators model and the dither, the dynamics of the system to be controlled is different from the one considered when designing the LQR, so it is normal that the optimal gains are different. In figure 4, where K_1 and K_2 are the feedback gains, $\gamma_2 = -K_1 r - K_2(\psi - \psi_{\text{ref}})$.



(a) State perturbations evolution.

(b) Angular controller gains.

Fig. 4 Inner loop controller - RL learning

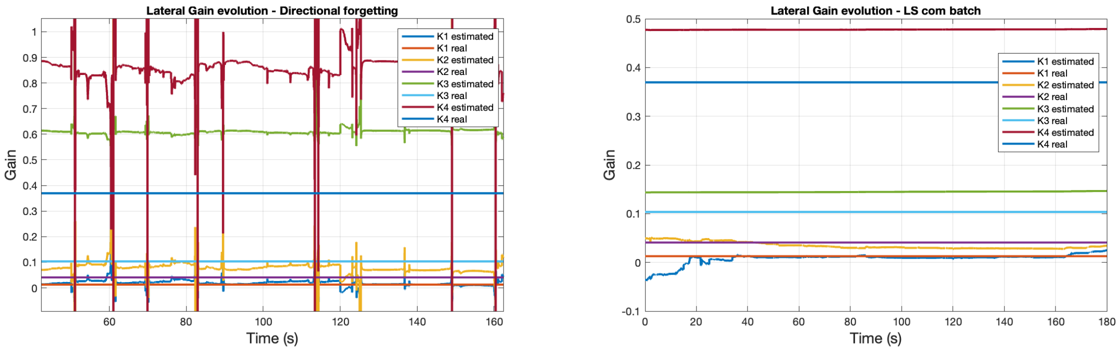
Thus, the RL converged to gains corresponding to a bandwidth (0.5118 rad/s) smaller than the bandwidth of the outer loop controller (0.7052 rad/s), so that it became impossible to stabilise the system. As a consequence, the outer loop controller had to be changed in order to have an even smaller bandwidth.

4.4 Outer Loop controller

With the simulations performed, it was verified that in the outer loop, directional forgetting [2] could not estimate correctly the parameters related to the state variables r and ψ. To estimate these parameters, the LS batch was tested [2], however, the estimation problem remained. Therefore, it was decided to fix these parameters values.

The real and the estimated parameters using both estimation algorithms are represented in figure 5, where, $\psi_{\text{ref}} = -\mathbf{K}\begin{bmatrix} y \\ v \\ \hat{r} \\ \hat{\psi} \end{bmatrix} = -\begin{bmatrix} K_1 & K_2 & K_3 & K_4 \end{bmatrix}\begin{bmatrix} y \\ v \\ \hat{r} \\ \hat{\psi} \end{bmatrix}$ ("ˆ" indicates the output of the outer loop).

The parameters have a smooth evolution, unlike what happens with directional forgetting where they have lots of peaks.



(a) RLS Directional forgetting.

(b) LS Batch.

Fig. 5 Outer loop controller: Comparison of different estimation algorithms.

4.5 Controller sampling period analysis

Since the designed controllers are discrete, the sampling period is an important parameter to define. However, given the complexity of this work, an analysis of the controller performance as a function of the sampling period was not performed. Varying the sampling time implies changes in the discrete linear model as well as in the LQR matrices R and Q and consequently in the RL controller parameters.

It can be said that the chosen sampling period was 0.01 seconds which is a reasonable value and it is used in the literature. For a larger sampling period, the RL controller had more difficulty in learning the parameters by having access to less data (verified in simulation for a sampling frequency of 30 Hz) and the disturbance reaction time would be longer, for a larger sampling period, the estimation of the gains is expected to improve but, with the actuators constraints, a big advantage in increasing the sampling period is not expected - unless buffers are used to store intermediate data for example. Also, sufficiently low sampling period leads to difficulties associated with non minimal phase effects.

4.6 Comparison of RL and LQR performances

In order to compare the performance of the LQR controller during the whole simulation with the LQR controller at the beginning and the RL from the learning time on, the performance measures defined in section 4.2 were used. Each performance value is obtained by performing several simulations with distinct dither seeds.

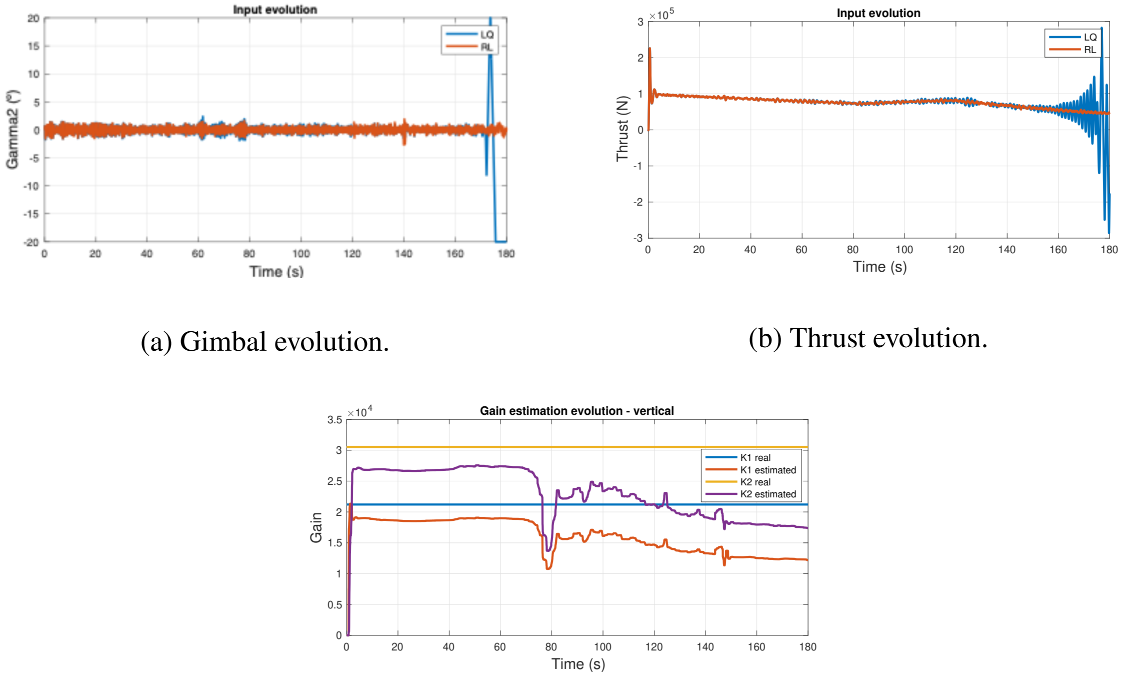
To make the model as realistic as possible, the existence of wind in the atmosphere was considered. In order to guarantee the functioning of the controller for different seeds values, the maximum wind intensity at 6 m altitude would have to be 2 m/s.

4.6.1 *Disturbances*

Besides the base wind, random wind gusts were considered. It was concluded from the analysis performed that the RL algorithm guarantees the stability of the system for gusts with magnitude up to 2 m/s on each axis, and which can occur at any instant of the trajectory. However, it is observed that the closer to the ground they occur the better is the response of the RL controlled system. In general, the system controlled with LQR is more robust to gusts with higher amplitude.

Difference specific impulse of the propellant The different specific impulse of the propellant is a measure of its efficiency. Decreasing I_{sp}, the efficiency of the propellant is decreased - meaning that for the same thrust force, the mass flow has to increase. The value for which the controller was designed is 282 s. The range of values considered in this study is between 235 and 282 s.

In landing, both the position error and the lateral speed error are always smaller with the RL controller. For an Isp of 235 s, the LQ controller started to show signs of instability as can be seen in figure 6, where it can be seen that the gains of the RL vertical controller tend towards different values than the theoretical ones. The smaller the Isp value is, the greater the difference in vertical speed on landing with the two controllers considered, with the RL controller presenting better results. It should be noted that for the value used for the controller design, the LQR presents a better result, although the difference is not significant. Regarding the angular metrics, the response of the two controllers is similar. The mean position error along the trajectory is always higher for the LQR.



(a) Gimbal evolution.

(b) Thrust evolution.

(c) Gain evolution: Vertical controller, $F_{\text{thr}} = -\begin{bmatrix} K_1 & K_2 \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$.

Fig. 6 Control system: system response for an ISP of 235 s.

Step in Thrust force Step-shaped disturbances of 5×10^2 N were applied to the actuator input at different time instants (30 s, 50 s, 70 s, 90 s, 110 s, 130 s, 150 s and 170 s). It can be concluded that the RL controlled system performs better at the moment of landing in lateral speed and position and angular position as seen in figure 7. In general, the average position error is about 2m lower with the RL controller and that with the exception of the disturbance applied at 30 seconds, the error remains practically constant for the remaining instants of application of the disturbance. The thrust level has lower values for the system controlled with RL and that the difference between RL and LQR decreases the later the disturbance is applied. The vertical speed at the moment of landing remains lower with the LQR independently of the instant of application of the disturbance.

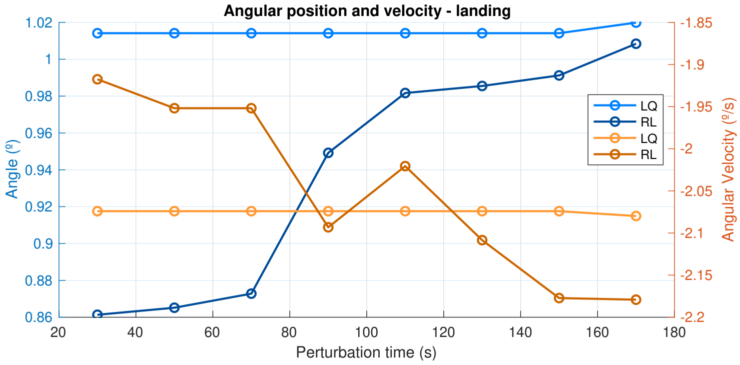


Fig. 7 Angular position and velocity at landing - step disturbance in the thrust force.

Step in Gimbal angle In order to analyse the effect of a perturbation at the gimbal angle input, a step of 3 ° that can start at different simulation instances was considered (the same as before). The maximum magnitude of the step disturbance that can be guaranteed to work for the various seeds and at various time instants is 3 °. Additionally, no such perturbations should be applied to the gimbal angle input before 30s as it affects the learning process of the inner loop gains and the system may become unstable. Furthermore, perturbations of higher intensity can be applied at specific time instants, but without guarantee of stabilisation.

Regardless of the instant at which the perturbation is applied, the RL controller performs better with respect to the lateral speed at landing, the mean position error along the trajectory and the thrust power, while the opposite is true for the vertical speed at landing (figure 8), the mean thrust value and the mean gimbal angle value.

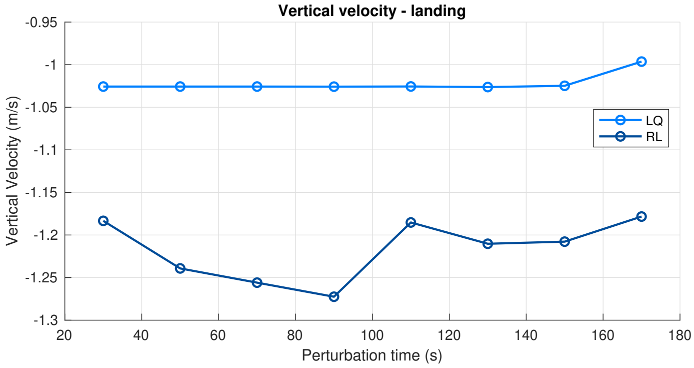
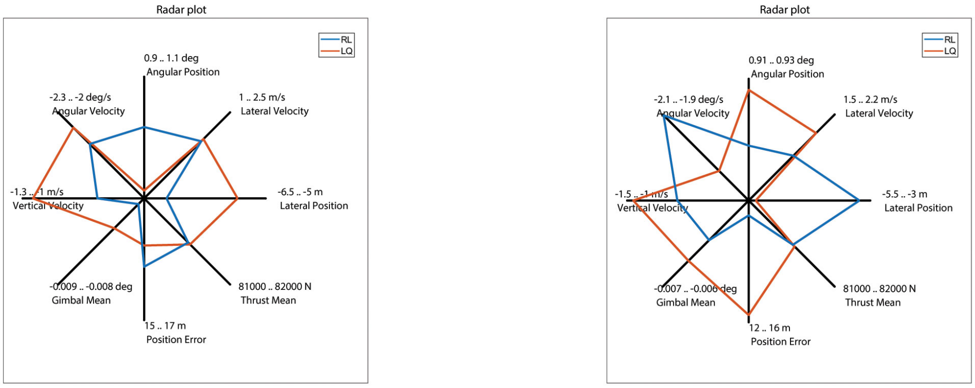


Fig. 8 Vertical velocity at landing - step disturbance in the gimbal.

4.6.2 LQR based on the wrong model

Center of mass position It was established that the distance from the centre of mass to the thrust application point is -4.6 m (the minus sign is due to the reference frame orientation). By changing this distance, the linear model is also changed. The vertical and lateral impact velocity as well as thrust power are lower with the RL controller for any CM position considered (range from -6.2 m to -3.8 m). The average value of the gimbal angle is slightly higher with the RL controller. Regarding the lateral position error and the mean position error along the trajectory it is found that when the CM is closer to the point of thrust application point, the performance of the RL controller is worse than the LQR one. When the CM is further away from the thrust application point, the trend is the inverse. This can be seen in the graphs of figure 9.



(a) CM further from the thrust application point.

(b) CM closer to the thrust application point.

Fig. 9 Radar plot: nominal scenario with wind - comparison between RL and LQ controllers.

Initial mass Consider an initial mass increase of 30%, that corresponds to an increase of 3157,2 kg. For an initial mass bigger than the one considered, the RL controller has a smaller position and lateral speed error on landing for any initial mass value considered as can be verified in figure 10. Regarding the vertical speed at landing, when the initial mass is 15% higher than the considered one, the LQR performance worsens, presenting a higher modulus speed than the RL. The angular position error also undergoes an inversion at 15% mass, with RL presenting a lower value. Again, it is verified that the position error accumulated along the trajectory for RL is always lower than for LQR. Both thrust and gimbal angle are similar for both controllers.

For a lower initial mass (until 15%), the system controlled with LQR starts to instabilize presenting oscillations. This situation represents an advantage of the RL controller: in this specific case it is more robust and adapts better.

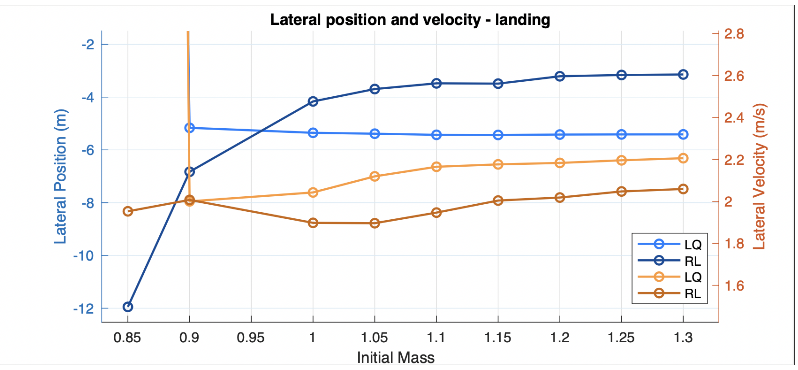


Fig. 10 Lateral position and velocity at landing (zoom) - change of the rocket initial mass.

4.6.3 Modification of actuator gains

Thrust The RL controller is only able to stabilize the system up to a gain of 0.8. Again, it is found that for the position and lateral velocity error as well as the mean position error along the trajectory, the RL controller shows better results. The vertical impact velocity is lower with the RL controller when the actuator gain is lower. Regarding the angular motion, the performance of the LQR controller is better than the RL one

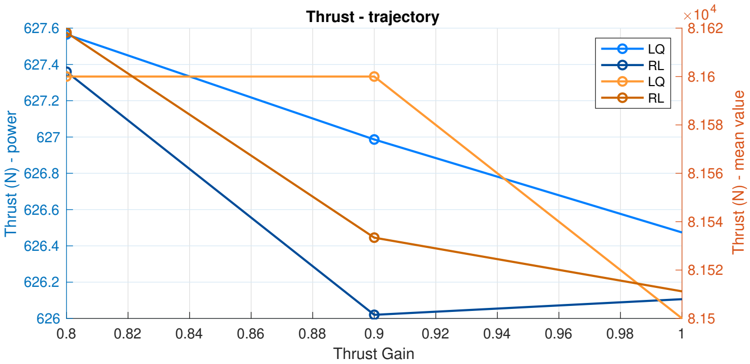


Fig. 11 Thrust metrics - modification of the thrust actuator's gain.

with exception of the 0.8 gain where the trend reverses. The same is true for the average thrust value (figure 11). The average gimbal angle is always lower with the LQR controller. For a gain smaller than 0.8, the vertical motion controller cannot estimate the gains well making the system unstable and the actuators saturate. By increasing the learning time of the vertical motion controller, the RL controller stabilises the system. For a gain of 0.6, a learning time of 100 s stabilizes the system. This is an example of how the short trajectory duration complicates the problem.

Gimbal Angle In the range of values considered (from 0.7 to 1), the value of the vertical speed at ground contact is higher (in modulus) when controlling with RL (opposite to the trend seen so far). The average gimbal angle (figure 12), the average thrust value and the angle position error at landing are lower for the LQR. The remaining variables do not show a definite trend.

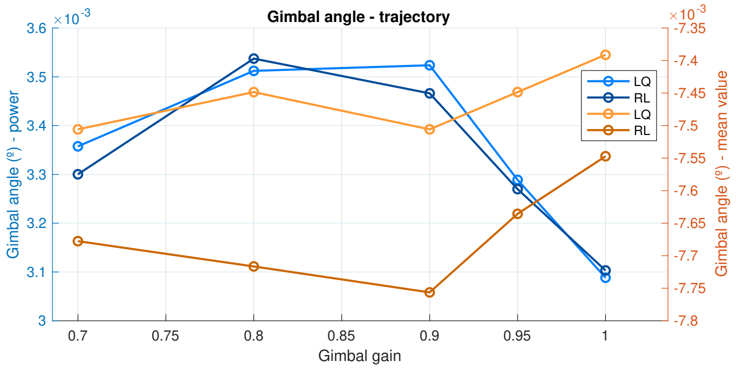


Fig. 12 Gimbal metrics - modification of the gimbal actuator's gain.

5 Conclusions

This study investigates the application of a reinforcement learning based adaptive controller, using the Q learning algorithm, to the vertical landing of a rocket, by assessing the performance levels and robusteness of the algorithm comparing with the ones obtained for the LQR controller.

In order to obtain a controller with RL, several parameters must be defined. The major difficulties in estimating the controller gains consisted in the fact that the trajectory duration was very short, the actuators have maximum changing rates and a cascade controller was used. In order to speed up the estimation, the estimation algorithms should have a good parameter initialization - the initialization of the covariance matrices and the initial vector of the gains is fundamental.

This study has shown that RL can outperform classical approaches, such as LQR, for the landing problem of reusable launchers. However, this is done at the cost of a higher design effort, when compared to classical LQR or to gain scheduling approaches, in which an LQR is designed for each linearization point along the trajectory, leading to similar levels of performance of RL. Nevertheless, it is expected that RL requires significantly less effort to adapt the design to another launcher, when compared to classical approaches, thanks to its model-free nature.

In terms of future work, onboard optimization-based guidance is foreseen to be used together with RL, in order to improve the overall robustness to exogenous disturbances and to the uncertainty in the initial state. Another very relevant topic that requires further research is the numerical implementation of the methods to compute the RL gains. In this work, a simplified approach using median filters to avoid large gains variation was adopted to avoid numerical instability, being this a sub-optimal approach.

Acknowledgments

Part of this work was supported by INESC-ID under project UIDB/50021/2020, financed by Fundação para a Ciência e a Tecnologia (Portugal).

References

[1] NASA Ames Research Center Jones, Harry W. The recent large reduction in space launch cost. 48th International Conference on Environmental Systems ICES-2018-81, Albuquerque, New Mexico, 8-12 July 2018.

[2] Goodwin and Sin. *Adaptive Filtering, Prediction and Control*. Addison Wesley, 3º edition, 1984. ISBN 13: 9780130040695.

[3] S. A. A. Rizvi and Z. Lin. Output feedback q-learning control for the discrete-time linear quadratic regulator problem. *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS*, 30(5):1–14, 2019. DOI: 10.1109/TNNLS.2018.2870075.

[4] Inês Zagalo. Adaptive and reconfigurable control of landing manoeuvres in uncertain environments. *IST - Instituto Superior Técnico*, May 2023.

[5] Vrabie D. Lewis, F. L. and V. L. Syrmos. *Optimal Control*. John Wiley Sons, Inc., Hoboken, New Jersey, 3º edition, 2012. ISBN: 978-0-470-63349-6.

[6] F. L. Lewis and D. Vrabie. Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE Circuits Syst. Mag*, 9(3):32–50, 2009. DOI: 10.1109/MCAS.2009.933854.

[7] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. ISBN: 9780262039246.

[8] Marie Albisser. *Identification of aerodynamic coefficients from free flight data*. PhD thesis, Doctorat de l'Université de Lorraine, Jully 2015.

[9] Marcos A. Simplicio, P. V. M. and S. Bennani. Reusable launchers: Development of a coupled flight mechanics, guidance and control benchmark. *Journal of Spacecraft and Rockets*, (2):1–34, 2019. DOI: 10.2514/1.A34429.

[10] O. C. Okwo A. B. Kisabo, A. F. Adebimpe and S. O. Samuel. State-space modelling of a rocket for optimal control system design. *Journal of Aircraft and Spacecraft Technology*, page 10, 2019.