The Journal of the Astronautical Sciences (2024) 71:10
https://doi.org/10.1007/s40295-023-00428-1

ORIGINAL ARTICLE

Learning-Based Parameter Optimization for a Class of Orbital Tracking Control Laws

Gianni Bianchini[1] · Andrea Garulli[1] · Antonio Giannitrapani[1] · Mirko Leomanni[2] · Renato Quartullo[1]

Accepted: 6 December 2023

Abstract
This paper presents a learning algorithm for tuning the parameters of a family of stabilizing nonlinear controllers for orbital tracking, in order to minimize a cost function which combines convergence time and fuel consumption. The main feature of the proposed approach is that it achieves performance optimization while guaranteeing closed-loop stability of the resulting controller. This property is exploited also to restrict the class of admissible controllers and hence to expedite the training process. The learning algorithm is tested on three case studies: two different orbital transfers and a rendezvous mission. Numerical simulations show that the learned control parameters lead to a significant improvement of the considered performance measure.

Keywords Orbital tracking · Control design · Parameter optimization · Learning

✉ Renato Quartullo
quartullo@diism.unisi.it

Gianni Bianchini
giannibi@diism.unisi.it

Andrea Garulli
garulli@diism.unisi.it

Antonio Giannitrapani
giannitrapani@diism.unisi.it

Mirko Leomanni
mirko.leomanni@unipg.it

[1] Dipartimento di Ingegneria dell'Informazione e Scienze Matematiche, Università di Siena, Via Roma 56, 53100 Siena, Italy

[2] Dipartimento di Ingegneria, Università di Perugia, Via Goffredo Duranti 93, 06125 Perugia, Italy

Published online: 06 February 2024

1 Introduction

Recent years have witnessed an increasing interest towards the use of learning techniques in aerospace applications. The steadily growing research activity in this area is testified by several surveys, classifying a variety of solutions for guidance [1, 2], navigation [3] and control [4]. In particular, rendezvous and docking (RVD) problems have been tackled by machine learning techniques in combination with model-based methods [5–7], as well as by reinforcement learning approaches [8–11]. A common feature of these works is that the control scheme includes an artificial neural network, possibly coupled with other types of controllers, which is trained by using experimental or simulation data. Among the motivations behind these techniques, there is the potential of neural networks to approximate complex maps and the possibility of designing the controller even without an explicit model of the physical system. Moreover, they allow one to optimize meaningful cost functions involving state and input variables. Unfortunately, providing a rigorous stability analysis of Neural Feedback Loops (NFLs), i.e., closed-loop schemes including neural networks as feedback controllers, is a hard task. In most works, stability and performance are evaluated only a posteriori, by means of simulation campaigns. Furthermore, the training of the neural controller may not consider some relevant points of the flight envelope, thus leading to unexpected behaviors of the control scheme or even instability.

Current research trends are tackling the above problem from different perspectives. Several works apply Lyapunov analysis to guarantee closed-loop stability of a control scheme combining a nonlinear feedback controller (based, e.g., on sliding mode or backstepping) and a neural network. This type of approach has been explored for aerospace control problems like attitude control [12], formation flying [13] and rendezvous and docking [14]. However, in these works the neural network is used only to adapt the controller parameters against model uncertainty and external disturbances. More recently, a remarkable effort has been devoted to studying closed-loop stability of NFLs, by resorting to classical control analysis paradigms (see, e.g., [15–17]). A limitation of these approaches is that the involved computational burden tends to grow considerably with the number of neurons and layers of the neural controller. A third line of research exploits learning tools to select the parameters of a controller belonging to a pre-specified class, whose structure is designed in order to guarantee the desired stability properties. In this context, [18] is one of the first works enforcing specific parameterizations of the controller (including the Youla-Kucera one) and then estimating the control parameters using the REINFORCE algorithm [19], a classical tool in machine learning. The Youla parameterization is also adopted in [20], while PID controllers are considered in [21]. Learning within a family of robustly stabilizing controllers has been addressed in [22].

A key feature of spacecraft control systems is that well-established and reliable models of the orbital dynamics are available [23, 24]. Therefore, a large body of literature is focused on the design of model-based control schemes for such problems (see, e.g., [25–31] and references therein). A common challenge these techniques have to face is that it is by no means trivial to tune the controller parameters in order to optimize specific performance indexes, such as fuel consumption and maneuver completion time. This motivates the adoption of a two-step design procedure, along the lines suggested in works such as [18] and [21]: first, a class of control laws guaranteeing closed-loop stability is chosen; then, learning techniques are employed to tune the parameters of the control law so as to optimize performance. This type of strategy has been already adopted in the aerospace field, either to optimize the parameters of feedback control laws for powered descent landing [32] or to tune a Lyapunov-based Q-law for trajectory design [33]. Such works adopt actor-critic reinforcement learning algorithms, whose training process is usually computationally demanding.

In this paper, the approach outlined above is adopted in the context of orbital tracking. The objective is to design an optimal control law that achieves closed-loop stability while minimizing a mixed time-fuel performance index. This is a challenging problem, being the orbital tracking dynamics nonlinear and the cost function nonsmooth. To this aim, the family of almost globally stabilizing feedback controllers proposed in [34] is considered. A specialized version of the REINFORCE algorithm, known as Augmented Random Search (ARS) [35], is employed to learn the values of the controller parameters which minimize the desired cost function. The learning procedure requires only the computation of the cost value associated to each episode within a batch of simulations of the closed-loop control system. The novelty of the proposed approach with respect to control schemes based on NFLs is that closed-loop stability is always guaranteed during the exploration of the parameter space, and hence also for the optimized controller. This allows one to remarkably speed-up the training process, by avoiding to consider parameter combinations that would lead to system instability. Numerical simulations on three different missions, involving orbital transfer and rendezvous maneuvers, confirm that the learning algorithm converges to a control law that optimizes a trade-off between settling time and fuel consumption. In particular, it is shown that the proposed technique can be exploited to tune the control system performance with respect to a set of initial mission configurations. Moreover, thanks to the simplicity of the ARS learning algorithm, the parameter tuning process takes seconds for a single mission, thus making the proposed approach computationally attractive with respect to other learning techniques proposed in the literature.

The paper is organized as follows. Section 2 reviews the dynamic model used for orbital tracking. The considered class of stabilizing controllers along with the performance optimization problem is introduced in Sect. 3, and the learning algorithm is presented in Sect. 4. The results of numerical simulations are discussed in Sect. 5, while Sect. 6 contains conclusions and future developments.

1.1 Notation

The symbol $0_{n\times m}$ denotes a null $n \times m$ matrix, while the identity matrix of order n is denoted by I_n. The partial derivative $\partial f / \partial x$ is expressed as a row vector. To save space,

$\cos(\cdot)$ and $\sin(\cdot)$ are abbreviated with $\mathrm{c}(\cdot)$ and $\mathrm{s}(\cdot)$, respectively. Moreover, we define the rotation matrix

$$R(\phi) = \begin{bmatrix} \mathrm{c}(\phi) & -\mathrm{s}(\phi) \\ \mathrm{s}(\phi) & \mathrm{c}(\phi) \end{bmatrix}.$$

Finally, for $v \in \mathbb{R}^n$, $\epsilon \in \mathbb{R}$, $\max\{v, \epsilon\}$ denotes the vector whose components are the maximum between the components of v and ϵ.

2 Orbital Tracking

In this paper, the dynamics of an orbiting spacecraft are described in terms of the six Equinoctial Orbital Elements

$$\psi = \left[\psi_1 \ \dots \ \psi_6\right]^T = \left[L, p, e_X, e_Y, h_X, h_Y\right]^T,$$

where L is the true longitude, p is the orbit semi-parameter, e_X, e_Y are the components of the eccentricity vector, and h_X, h_Y are the components of the inclination vector [36]. The dynamics are given by

$$\dot{\psi} = f(\psi) + g(\psi)u,$$

where $u = \left[u_r, u_\theta, u_h\right]^T$ is the control vector (radial, transverse and normal forcing accelerations, respectively),

$$f(\psi) = \sqrt{\frac{\mu}{\psi_2^3}}\left[\, (1+\zeta_X)^2 \ 0 \ 0 \ 0 \ 0 \ 0\right]^T,$$

$$g(\psi) = \frac{\sqrt{\psi_2}}{\sqrt{\mu}(1+\zeta_X)} \begin{bmatrix} 0 & 0 & \eta \\ 0 & 2\,\psi_2 & 0 \\ (1+\zeta_X)\mathrm{s}(\psi_1) & q_X & -\,\eta\,\psi_4 \\ -(1+\zeta_X)\mathrm{c}(\psi_1) & q_Y & \eta\,\psi_3 \\ 0 & 0 & \dfrac{(1+h^2)}{2}\mathrm{c}(\psi_1) \\ 0 & 0 & \dfrac{(1+h^2)}{2}\mathrm{s}(\psi_1) \end{bmatrix},$$

with

$$\begin{aligned} \zeta_X &= \psi_3\mathrm{c}(\psi_1) + \psi_4\mathrm{s}(\psi_1), \\ q_X &= \psi_3 + (2+\zeta_X)\mathrm{c}(\psi_1), \\ q_Y &= \psi_4 + (2+\zeta_X)\mathrm{s}(\psi_1), \\ \eta &= \psi_5\mathrm{s}(\psi_1) - \psi_6\mathrm{c}(\psi_1), \\ h^2 &= \psi_5^2 + \psi_6^2, \end{aligned}$$

and μ is the gravitational parameter of the central body. On any unforced orbit, only the true longitude ψ_1 varies in time.

The considered control task is to track a target reference trajectory $\psi^r(t) = [\psi_1^r(t), \psi_2^r, \psi_3^r, \psi_4^r, \psi_5^r, \psi_6^r]^T$ where $\psi^r(t)$ satisfies the unforced periodic dynamics $\dot{\psi}^r = f(\psi^r)$ with given initial conditions $\psi^r(0)$. In order to ease the control design, the dynamics of the tracking error $\tilde{\psi} = \psi - \psi^r$ are modeled as in [34] using the transformed variables:

$$\begin{aligned}
x_1 &= \tilde{\psi}_1 \\
x_2 &= \sqrt{1 + \frac{\tilde{\psi}_2}{\psi_2^r}} - 1 \\
\begin{bmatrix} x_3 \\ x_4 \end{bmatrix} &= \begin{bmatrix} \frac{\psi_2^r}{\tilde{\psi}_2 + \psi_2^r} & 0 \\ 0 & \sqrt{\frac{\psi_2^r}{\tilde{\psi}_2 + \psi_2^r}} \end{bmatrix} R(\tilde{\psi}_1 + \psi_1^r) \begin{bmatrix} \tilde{\psi}_3 + \psi_3^r \\ -\tilde{\psi}_4 - \psi_4^r \end{bmatrix} + \begin{bmatrix} -\frac{\tilde{\psi}_2}{\tilde{\psi}_2 + \psi_2^r} \\ 0 \end{bmatrix} - \begin{bmatrix} \zeta_X^r \\ \zeta_Y^r \end{bmatrix} \\
x_5 &= \tilde{\psi}_5 \\
x_6 &= \tilde{\psi}_6,
\end{aligned} \tag{1}$$

where $[\zeta_X^r, \zeta_Y^r]^T = R(\psi_1^r)[\psi_3^r, -\psi_4^r]^T$. The transformation (1) is such that $x = 0$ if and only if $\tilde{\psi} = 0$. The corresponding dynamic model is given by:

$$\dot{x} = \begin{bmatrix} F(\chi, \psi^r) \\ 0_{2\times 1} \end{bmatrix} + \begin{bmatrix} G(\chi, \psi^r) \\ 0_{2\times 2} \end{bmatrix} \begin{bmatrix} u_r \\ u_\theta \end{bmatrix} + H(x, \psi^r)\, u_h, \tag{2}$$

where $\chi = [x_1 \dots x_4]^T$,

$$F(\chi, \psi^r) = \begin{bmatrix} 0 & F_{12} & F_{13} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -F_{33} & -F_{12} \\ 0 & F_{42} & F_{12} + F_{43} & 0 \end{bmatrix} \chi, \; G(\chi, \psi^r) = \begin{bmatrix} 0 & 0 \\ 0 & G_{22} \\ 0 & 0 \\ G_{41} & 0 \end{bmatrix},$$

$$H(x, \psi^r) = \frac{G_{22}}{(x_2 + 1)} \begin{bmatrix} (x_5 + \psi_5^r)\mathrm{s}(x_1 + \psi_1^r) - (x_6 + \psi_6^r)\mathrm{c}(x_1 + \psi_1^r) \\ 0_{3\times 1} \\ \frac{1 + (x_5 + \psi_5^r)^2 + (x_6 + \psi_6^r)^2}{2}\mathrm{c}(x_1 + \psi_1^r) \\ \frac{1 + (x_5 + \psi_5^r)^2 + (x_6 + \psi_6^r)^2}{2}\mathrm{s}(x_1 + \psi_1^r) \end{bmatrix},$$

with

$$
\begin{aligned}
F_{12} &= \sqrt{\tfrac{\mu}{(\psi_2^r)^3}}\left(x_3 + 1 + \zeta_X^r\right)^2,\\
F_{13} &= \sqrt{\tfrac{\mu}{(\psi_2^r)^3}}\left(x_3 + 2 + 2\zeta_X^r\right),\\
F_{42} &= \sqrt{\tfrac{\mu}{(\psi_2^r)^3}}\left(x_2 + 2\right)\left(x_3 + 1 + \zeta_X^r\right)^3,\\
F_{33} &= F_{13}\,\zeta_Y^r,\\
F_{43} &= F_{13}\,\zeta_X^r,\\
G_{22} &= \sqrt{\tfrac{\psi_2^r}{\mu}}\tfrac{1}{(x_3+1+\zeta_X^r)},\\
G_{41} &= \sqrt{\tfrac{\psi_2^r}{\mu}}.
\end{aligned}
$$

It is worth noticing that the vector fields in (2) are periodically time-varying, with the same period as the reference trajectory.

3 Controller Class and Performance Assessment

By using the results in [34], we define a parametric family of stabilizing controllers for system (2), as follows:

$$
\begin{aligned}
u_r(x, \psi^r; K) &= -\frac{1}{G_{41}}\left(F_{43}\,x_3 - \dot{\xi}\right) - K_4(x_4 - \xi)\\
u_\theta(x, \psi^r; K) &= -K_1\frac{G_{41}F_{12}}{G_{22}}\mathrm{s}(x_1) - \frac{F_{42}}{G_{22}}(x_4 - \xi) - K_2\frac{G_{41}}{G_{22}}x_2\\
u_h(x, \psi^r; K) &= -K_5\frac{1}{G_{41}}\frac{\partial V}{\partial x}H,
\end{aligned}
\tag{3}
$$

where $K = [K_1, \dots, K_5]^T$ is a vector of constant parameters,

$$
V(x) = K_1 G_{41}(1 - \mathrm{c}(x_1)) + \frac{1}{2}(x_2^2 + x_3^2 + (x_4 - \xi)^2 + x_5^2 + x_6^2),
\tag{4}
$$

and

$$
\xi = \frac{1}{F_{12}}\left(K_1 G_{41} F_{13}\mathrm{s}(x_1) - F_{33}x_3 + K_3 G_{41} x_3\right).
\tag{5}
$$

The explicit expressions of $\dot{\xi}$ and $\frac{\partial V}{\partial x}H$ in (3) are not reported for brevity. The control law introduced above exploits backstepping and damping control techniques. In particular, ξ in (5) plays the role of a virtual input to the dynamics of χ in (2), while the last equation in (3) represents a damping term. The following result states the stability properties of the control law (3), which can be proved by adopting $V(x)$ in (4) as a Lyapunov function (see [34] for details).

Proposition 1 *Let $K_i > 0,\ i = 1, \dots, 5$. Then, the origin of closed-loop system (2)-(5) is almost globally asymptotically stable.*

This result defines the set of parameters K guaranteeing that the proposed control law stabilizes the tracking error system. However, it is well known that tuning the performance of nonlinear control laws is far from trivial. Indeed, a misguided choice of the control parameters of the closed-loop system (2)-(5) may lead, for example, to extremely slow tracking of the reference trajectory or to an excessive control effort. The goal of this paper is to tune the parameters K of the stabilizing control law (3) so as to optimize the performance of the closed-loop system in terms of a trade-off between the settling time and the fuel consumption. To this purpose, we denote by y the distance between the actual and reference spacecraft position, expressed in Cartesian coordinates. This can be seen as an output signal of system (2), i.e.,

$$y = Y(x, \psi_r), \tag{6}$$

where the mapping Y is obtained from (1) and the transformation which relates the satellite Equinoctial elements to the corresponding inertial cartesian states [37]. In order to learn the controller parameters K from the input–output behavior, system (2), (6) with control law (3) is simulated over a horizon of length T_e (each simulation is called an *episode*). The input and output values collected at sampling times kT_s, $k = 0, \dots, H$, with $T_e = HT_s$, are denoted as $u(k)$ and $y(k)$, respectively. Then, the performance index to be minimized is specified as

$$J(x(0); K) = H_{conv} + \rho \sum_{k=0}^{H_{conv}-1} ||u(k)||, \tag{7}$$

where $x(0)$ denotes the initial state vector,

$$H_{conv} = \min\{\bar{k} \;:\; y(k) \leq \epsilon, \;\; \forall k \geq \bar{k}\}, \tag{8}$$

and ϵ is a threshold assessing practical convergence. The parameter ρ is used to trade-off the two conflicting requirements of minimizing the maneuver completion time H_{conv} and the fuel consumption.

In the following, the problem of minimizing the cost (7) with respect to the controller parameter vector K is addressed. Being (7) a discontinuous function of K, a gradient-free optimization method is required. In the next section, a learning-based approach is proposed.

4 Learning Procedure

A classical approach to minimize a function $J(K)$ with respect to $K \in \mathbb{R}^q$ is the so-called *random search*, which amounts to computing a numerical approximation of the function gradient along a random search direction.

Recently, an enhanced version of this approach, namely the *Augmented Random Search (ARS)* method, has been proposed in [35]. It is a derivative-free stochastic optimization method which explores the parameter space of a family of deterministic control policies, by simulating episodes with randomly perturbed parameter vectors K. The ARS algorithm improves with respect to the basic random search, by adopting several heuristics which have proven to be effective in speeding up the learning process. First, multiple random search directions $\delta_j \in \mathbb{R}^q$, $j = 1, \dots, N$, are selected in order to enhance the exploration of the parameter space. This is done by generating N random vectors δ_j sampled from a normal distribution with zero mean and covariance matrix Σ_δ. Notice that the latter plays a significant role in scaling the exploration appropriately for each element of the parameter space. Then, the parameter vector is updated along a direction which is a weighted average of the random search vectors, according to the cost variation along each δ_j. The update step is scaled by the standard deviation σ_J of the $2N$ cost values $J_+^{(j)} = J(K_+^{(j)})$, $J_-^{(j)} = J(K_-^{(j)})$, $j = 1, \dots, N$, evaluated by simulating the closed-loop system with the corresponding control law parameter values

$$K_+^{(j)} = \max\{K + \sigma\delta_j,\, \epsilon_K\} \tag{9}$$

$$K_-^{(j)} = \max\{K - \sigma\delta_j,\, \epsilon_K\}, \tag{10}$$

until practical convergence of the trajectory y is achieved. In (9)-(10), σ is a positive scaling constant and $\epsilon_K > 0$ is a small quantity, instrumental to guaranteeing positivity of the controller parameters, as required by Proposition 1. The scaling by σ_J is useful to adapt the step sizes according to the local sensitivity of the cost with respect to perturbations of the control parameters [35]. Then, the parameter update step is performed as

$$K^{(i+1)} = \max\{K^{(i)} - \frac{\alpha}{N\sigma_J}\sum_{j=1}^{N}(J_+^{(j)} - J_-^{(j)})\delta_j,\, \epsilon_K\}. \tag{11}$$

The update (11) is repeated iteratively for $i = 1, \dots, M$, where M is the total number of iterations (note that, each iteration requires to perform $2N$ episodes and cost evaluations). The outcome of the learning procedure is the final parameter vector $K^* = K^{(M)}$. The overall procedure is summarized in Algorithm 1. Note that, rather than employing a predefined maximum number of iterations, alternative stopping criteria can be adopted for the proposed algorithm. For instance, the learning procedure can be terminated when the cost J does not decrease significantly anymore. This is typically done by smoothing the cost value with a moving average and then checking if its decrease is below a given threshold. In the simulations presented in Sect. 5, we let the learning procedure evolve over a predefined number of iterations in order to test the numerical stability of the method.

It is worth stressing that asymptotic convergence of the closed-loop system trajectories is guaranteed for all learning episodes by the global stability property of the control law (3) and the fact that in Algorithm 1 all the generated K_+, K_- and updated $K^{(i)}$ are strictly positive. This feature turns out to be crucial to streamline the learning procedure. Indeed, the occurrence of divergence or other unstable behaviors would prevent a meaningful computation of the costs $J_+^{(j)}$, $J_-^{(j)}$, thus leading to high variance of the local cost values and, in turn, of the parameter updates.

Algorithm 1 Augmented Random Search (ARS)

```
Hyperparameters: number $M$ of iterations, number $N$ of sampled directions per iteration, covariance matrix $\Sigma_\delta$ of normal distribution for sampling vectors $\delta_j$, perturbation step $\sigma$, maximum length $H$ of each episode, stepsize $\alpha$, tolerance $\epsilon_K$.
Initialize: parameter vector $K^{(1)} \in \mathbb{R}^q$.
for each iteration $i = 1, 2, \dots, M$ do
    Sample independent vectors $\delta_j \in \mathbb{R}^q$, $j = 1, \dots, N$, from normal distribution with zero mean and covariance matrix $\Sigma_\delta$
    for $j = 1, \dots, N$ do
        Define perturbed parameter vectors
            $K_+^{(j)} = \max\{K^{(i)} + \sigma\delta_j, \epsilon_K\}$
            $K_-^{(j)} = \max\{K^{(i)} - \sigma\delta_j, \epsilon_K\}$
        Simulate system (2)-(5) with parameters $K_+^{(j)}$, $K_-^{(j)}$
        For each episode, compute costs $J_+^{(j)}$, $J_-^{(j)}$
    end for
    Compute standard deviation $\sigma_J$ of the $2N$ cost values $J_+^{(j)}$, $J_-^{(j)}$, $j = 1, \dots, N$
    Update control parameters $K^{(i+1)}$ as in (11)
end for
```

5 Numerical Simulations

In this section, Algorithm 1 is exploited to tune the parameter vector $K = \left[K_1, \dots, K_5\right]^T$ of the control law (1) within different case-studies, in order to demonstrate its suitability for performance optimization in the context of space applications. In particular three different scenarios are considered: (A) an orbital transfer from a low Earth orbit (LEO) to a geostationary transfer orbit (GTO); (B) an orbital transfer from a GTO to a geostationary Earth orbit (GEO); (C) a rendezvous mission in LEO. The first two case studies are representative of orbit control problems characterized by strong nonlinearities, which raise the challenge of optimizing a complex transient response. The latter application focuses on a scenario in which feedback control is essential to achieve a sufficient level of mission autonomy.

The implementation of the proposed algorithm utilizes the C++ programming language and runs on a 3.10 GHz CPU with 16 cores, using OpenMP constructs to enable parallel computing. The parallelization is applied to both episodic exploration directions and initial conditions so as to improve the computational efficiency. The hyperparameters chosen for the learning algorithm are reported for each scenario in the corresponding subsection.

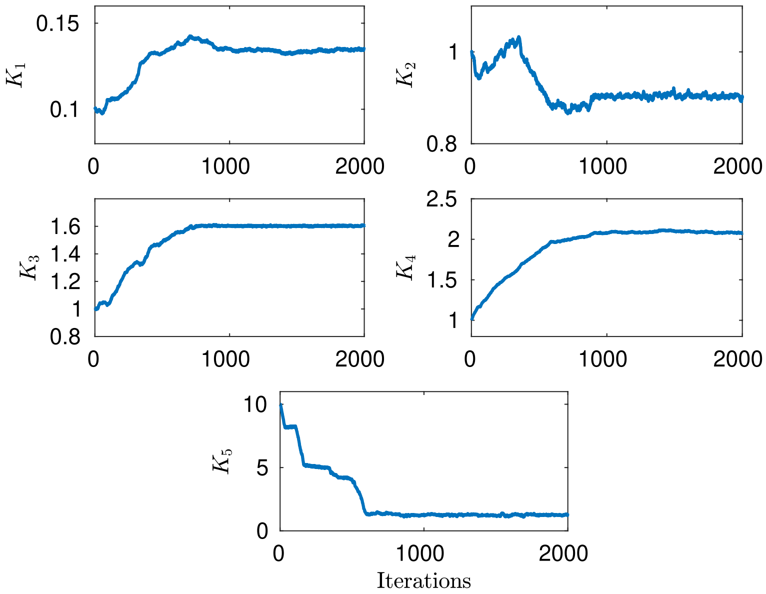

Fig. 1 Scenario A. Evolution of the parameter vector $K^{(i)}$ during the learning process

5.1 Orbital Transfer: LEO-GTO

In this transfer mission, which is inspired by [38], the initial orbit is an equatorial circular orbit with a semi-major axis equal to 6778 km, while the reference orbit is a higher altitude elliptic orbit. The initial and reference orbital elements are reported in Table 1.

The sampling time is $T_s = 16$ min and the parameters characterizing the performance index in (7)-(8) are set to $\rho = 50$ and $\epsilon = 10$ km. The algorithm hyperparameters are chosen as follows: $M = 2000$, $\alpha = 5 \cdot 10^{-3}$, $\sigma = 2 \cdot 10^{-3}$, $N = 16$, $H = 320$, corresponding to 10 consecutive target orbits and $T_e = 86.4$ hours. The initial parameter vector is selected as $K^{(1)} = [0.1,\ 1,\ 1,\ 1,\ 10]^T$. The covariance matrix is specified as $\Sigma_\delta = \mathrm{diag}\{0.1,\ 1,\ 1,\ 1,\ 10\}$, which ensures an appropriate scaling of the perturbation direction for the vector K. The choice of the scaling values is the

Table 1 Scenario A: orbital elements of the initial and reference orbits

Orbital element	Initial orbit	Reference orbit
True longitude	$L(0) = 4.1015$ rad	$L^r(0) = 3.9270$ rad
Semi-parameter	$p = 6778$ km	$p^r = 19455$ km
Eccentricity vector	$e_X = 0$ $e_Y = 0$	$e^r_X = -0.0776$ $e^r_Y = 0.2898$
Inclination vector	$h_X = 0$ $h_Y = 0$	$h^r_X = 0.3046$ $h^r_Y = 0.0816$

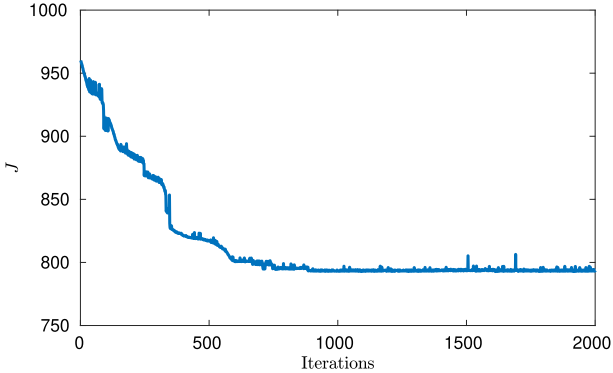

Fig. 2 Scenario A. Evolution of the maneuver cost J

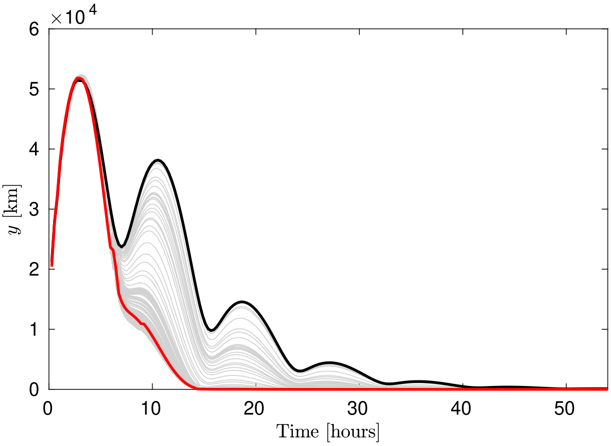

Fig. 3 Scenario A. Evolution of the output signal $y(t)$ resulting from the application of Algorithm 1: first iteration (black line) and final iteration (red line)

outcome of a trial-and-error selection process based on the experimentation on different datasets. In this setting, the computation time required for tuning the controller parameters amounts to 12 s. This corresponds on average to 6 ms per iteration.

The evolution of the parameter vector $K^{(i)}$ is depicted in Fig. 1, while Fig. 2 displays the overall cost J defined by (7). A cost reduction of about 18% with respect to the initial non-optimized cost is achieved in less than 1000 iterations. It can be observed that the cost is not monotonically decreasing during the learning process, due to the stochasticity of the search algorithm. In fact, the finite number of the explored directions in the parameter space may lead to a local increase in the cost at some iterations. Fig. 3 shows the output $y(t)$ defined by (6) for all the iterations of the learning algorithm. The black and red lines denote the trajectories corresponding to the initial and the final parameter vector of the controller, respectively. It can be seen that optimizing such parameters leads to a remarkable reduction of the flight time and that all the trajectories generated during the learning phase achieve converge towards the origin. To qualitatively illustrate these results, Fig. 4 shows a three-dimensional plot of the resulting spacecraft trajectories in the Earth centered inertial (ECI) frame. It can be seen that the learned trajectory accomplishes notably less revolutions than the initial one, confirming the aforementioned cost reduction. The control input signals obtained in the first and last iterations of the learning process are reported in Fig. 5. The optimization of the selected cost function allows for a reduction of the peak value of the normal acceleration u_h and a shorter activation of the radial one u_r.

5.2 Orbital Transfer: GTO-GEO

In this case-study, the proposed approach is tested on a GTO to GEO transfer. The target orbit is an equatorial GEO with semi-major axis of 42,165 km, while the initial GTOs are characterized by a semi-major axis of 24364 km, eccentricity of 0.7306 and initial true longitude of $\pi/6$. The inclination i, right ascension of the ascending node (RAAN) Ω, and argument of periapsis ω of the GTO are randomly drawn from a uniform distribution on the interval $[\pi/4, \pi/2]$. In particular, a set of 50 different initial GTO configurations has been considered. The hyperparameters of the learning algorithm are chosen as in the previous case study, except for $T_s = 45$ min, $H = 1280$ (corresponding to 40 target orbits), and $T_e = 957.4$ hours. The computation time required by the proposed approach to optimize the controller parameters for the entire set of initial configurations amounts to about 40 min.

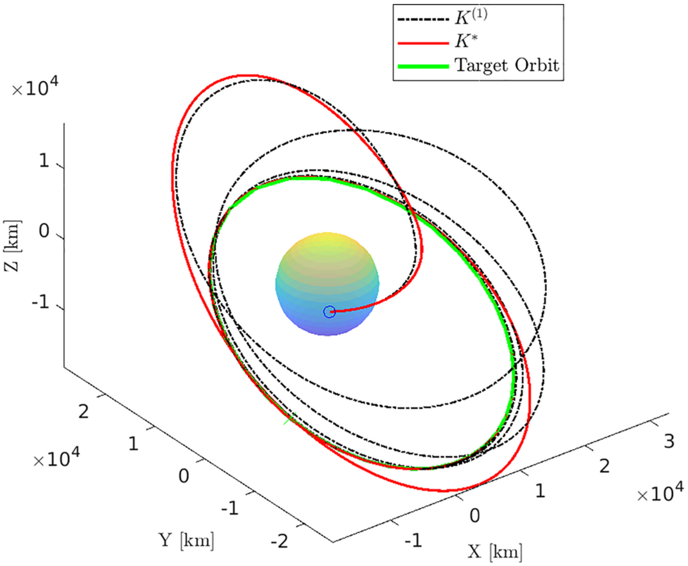

Fig. 4 Scenario A. Three-dimensional trajectories corresponding to the initial (black) and optimized (red) parameters of the control law (3). The black circle marks the intial condition, while the target orbit is colored green

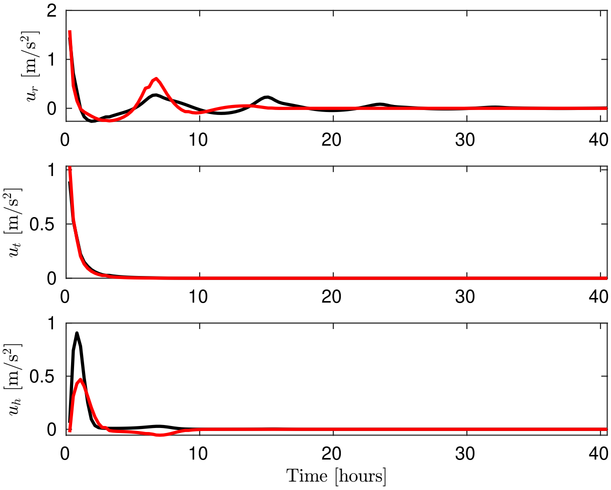

Fig. 5 Scenario A. Radial, transverse and normal control signals: first iteration (black line) and final iteration (red line)

Fig. 6 shows the trajectories obtained for the considered set of initial configurations, before and after the optimization performed by Algorithm 1. In particular, the black and red trajectories represent the evolution of the output $y(t)$ resulting from the application of the control law (3) with parameters $K^{(1)}$ and K^*, respectively. It can be seen that the optimized trajectories display a much better envelope profile than the initial ones, especially in terms of convergence time. Table 2 summarizes the results obtained by applying the proposed learning algorithm (Algorithm 1), with respect to the total cost, the convergence time and the fuel efficiency. These show a remarkable improvement in the cost-related metrics.

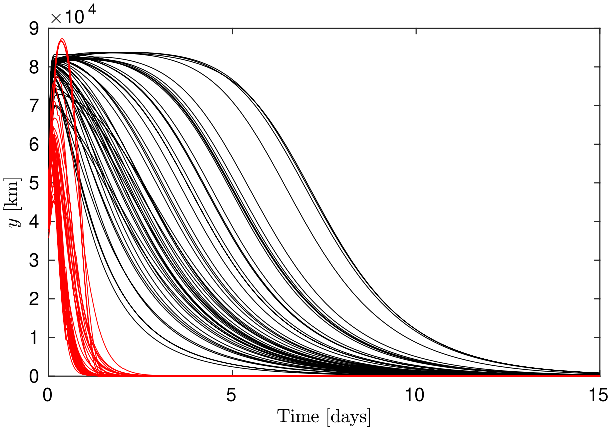

Fig. 6 Scenario B. Trajectories of the output $y(t)$ for the considered set of initial conditions, before (black) and after (red) the optimization performed using Algorithm 1

Table 2 Results for Scenario B

	Cost reduction (%)		
	Total cost	Settling time	Fuel consumption
Average	78.4	86.0	68.8
Minimum	65.4	67.5	40.5
Maximum	83.6	91.8	76.1

5.3 Rendezvous

In this case study, we consider a terminal rendezvous scenario, in which a controlled spacecraft (referred to as the chaser) must intercept an uncontrolled target. The purpose of this study is to assess the performance obtained by using a mean parameter vector $\hat{K}$ computed by averaging the results of the learning process over a sufficiently representative set of initial conditions. The motivation is the potential application to rendezvous missions. In these scenarios, the initial condition is not known accurately beforehand, being the result of a previous transfer mission. Moreover, online learning of the best controller tuning for a specific initial condition may not be possible due to computational constraints. To overcome this limitation while still achieving an acceptable performance, pre-computing mean tuning parameters turns out to be a viable option. A performance analysis of the controller tuned in this way is presented hereafter.

In the considered setting, the target moves along a near-circular LEO with an altitude of 1000 km above the Earth, an inclination of 81 deg and an initial true longitude of 45 deg. The chaser is assumed to initially lie in the neighborhood of the target following a preliminary coarser orbit injection maneuver. To account for this feature, a set of 50 random initial conditions $x(0)$ are generated through a normal distribution centered at the target equinoctial elements ψ^r, using the covariance matrix $\sigma_\psi = \text{diag}\,\{0.5\text{ deg}, 20\text{ km}, 3 \cdot 10^{-5}, 3 \cdot 10^{-5}, 2 \cdot 10^{-3}, 2 \cdot 10^{-3}\}$. The resulting initial inter-satellite separation is about 60 km on average. The hyperparameters used in the learning procedure are as follows: $T_s = 3$ min, $\epsilon = 1$ km, $\rho = 400$ and $M = 5000$. The overall computation time required to apply Algorithm 1 to the entire set of initial conditions is about 90 min.

Fig. 7 shows the evolution of parameter vector $K^{(i)}$ during the iterations of the learning process, for a selected subset of the considered initial conditions. The final mean parameter vector resulting from the optimization is equal to $\hat{K} = [1.22, 5.41, 0.72, 5.29, 0.40]^T$.

Figure 8 depicts the trajectories of the output y provided by the control law (3) with the parameters $K^{(1)}$, K^* and $\hat{K}$, for a single realization of the initial conditions. It can be seen that the controller employing mean tuning parameters achieves a considerable reduction of the convergence time, which is comparable to the one provided by the dedicated tuning K^*. An equally good behavior is observed for the entire set of initial states $x(0)$. Table 3 presents statistics on this experiment and confirms that the performance achieved by the parameters K^* and $\hat{K}$ is on a similar level. It is concluded that tuning the controller (3) with the learned mean parameter

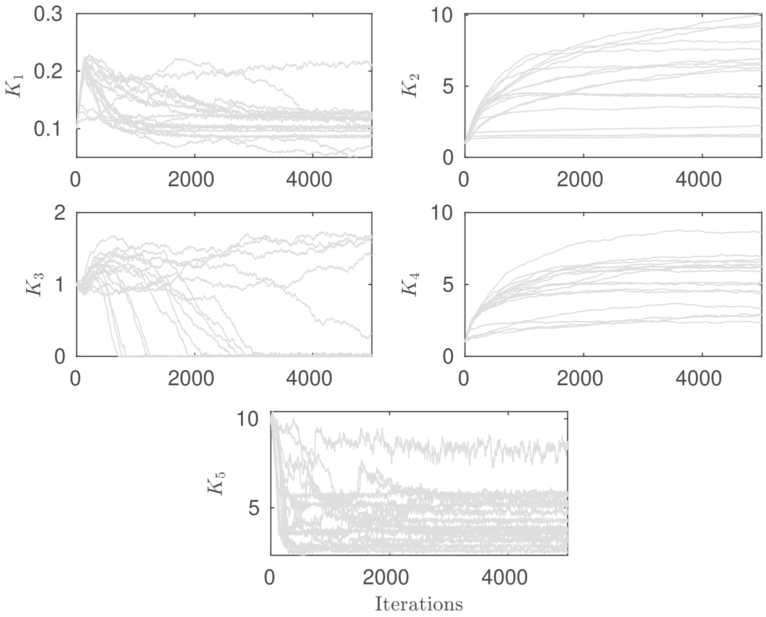

Fig. 7 Scenario C. Evolution of the parameter vector during learning for a subset of the considered initial conditions

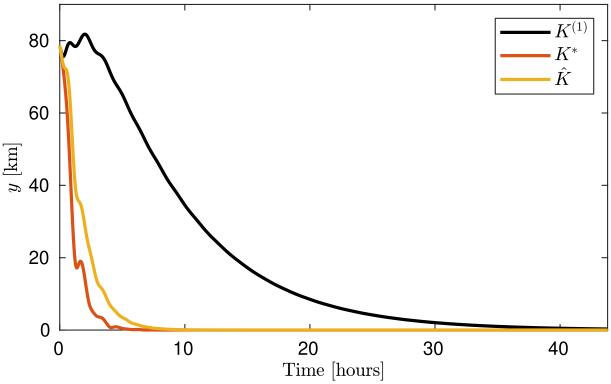

Fig. 8 Scenario C. Distance output y obtained with the control parameter vectors $K^{(1)}$ (black line), K^* (red line) and $\hat{K}$ (yellow line)

vector $\hat{K}$ is an advantageous strategy for terminal rendezvous maneuvers, allowing for achieving near-optimal performance whenever on-board optimization is not viable.

In Fig. 9, the cost evolutions, smoothed by a 50-sample moving average, are reported for the considered initial conditions. It can be seen that the cost converges in all the learning tests, even in the cases in which some parameter value does not reach a steady state value (thus suggesting a low sensitivity of the cost with respect

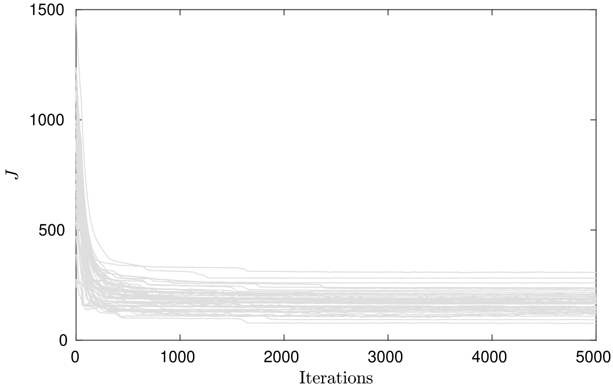

Fig. 9 Scenario C. Evolution of the smoothed cost J for the considered initial conditions

Table 3 Results for Scenario C

	Total cost reduction (%)	
	K^*	$\hat{K}$
Average	81.3	78.4
Minimum	72.6	68.9
Maximum	87.2	85.9

to such parameters). By using the stopping criterion discussed in Sect 4, i.e., terminating the learning procedure when the smoothed cost does not decrease more than a predefined threshold (here set to 10^{-5}), one has that the procedure converges on average after approximately 1800 iterations.

6 Conclusions

Optimization of performance measures in orbital tracking is a challenging task due to the complexity of the dynamic models and the necessity to guarantee fundamental requirements such as stability, robustness and constraint satisfaction. This work has shown that a simple learning technique, based on the ARS algorithm, can be successfully employed to tune the parameters of a family of stabilizing controllers for orbital tracking, in order to optimize a cost function accounting for both settling time and fuel consumption. The approach combines the benefits of model-based control design to those of simulation-based learning techniques. A major advantage of the proposed approach lies in its computational efficiency, which makes it compatible with on-board implementation. It is believed that the proposed learning procedure can be successfully employed to optimize the parameters of other families of control laws, while guaranteeing specific stability/performance properties, during the parameter exploration phase. In perspective, the proposed methodology can also be useful to analyze the sensitivity of the performance metrics with respect to the control parameters. Future research may concern the comparison of the proposed algorithm with other learning approaches (e.g., policy optimization) and the inclusion of state/input constraints or parametric uncertainties in the control synthesis problem.

Funding Open access funding provided by Università degli Studi di Siena within the CRUI-CARE Agreement.

References

1. Izzo, D., Sprague, C.I., Tailor, D.V.: Machine learning and evolutionary techniques in interplanetary trajectory design. In: Fasano, G., Pinter, J.D. (eds.) Modeling and Optimization in Space Engineering: State of the Art and New Challenges, pp. 191–210. Springer, New York (2019)
2. Izzo, D., Märtens, M., Pan, B.: A survey on artificial intelligence trends in spacecraft guidance dynamics and control. Astrodynamics **3**, 287–299 (2019)
3. Song, J., Rondao, D., Aouf, N.: Deep learning-based spacecraft relative navigation methods: A survey. Acta Astronaut. **191**, 22–40 (2022)
4. Shirobokov, M., Trofimov, S., Ovchinnikov, M.: Survey of machine learning techniques in spacecraft control design. Acta Astronaut. **186**, 87–97 (2021)
5. Ueda, S., Noumi, A.: Precise rendezvous guidance in low earth orbit via machine learning. In: Proceedings of SICE International Symposium on Control Systems, SICE ISCS (2019)
6. Li, H., Dong, Y., Li, P.: Real-time optimal approach and capture of ENVISAT based on neural networks. Int. J. Aerospace Eng. (2020). https://doi.org/10.1155/2020/8165147
7. Ikeya, K., Liu, K., Girard, A., Kolmanovsky, I.: Learning reference governor for constrained spacecraft rendezvous and proximity maneuvering. J. Spacecraft Rockets (2023). https://doi.org/10.2514/1.A35483
8. Gaudet, B., Linares, R., Furfaro, R.: Spacecraft rendezvous guidance in cluttered environments via artificial potential functions and reinforcement learning. In: Advances in the Astronautical Sciences, Vol. 167, pp. 813–828 (2018)
9. Wang, X., Wang, G., Chen, Y., Xie, Y.: Autonomous rendezvous guidance via deep reinforcement learning. In: 2020 Chinese Control And Decision Conference (CCDC), pp. 1848–1853 (2020). IEEE
10. Hovell, K., Ulrich, S.: On deep reinforcement learning for spacecraft guidance. In: AIAA Scitech 2020 Forum (2020)
11. Federici, L., Scorsoglio, A., Zavoli, A., Furfaro, R.: Meta-reinforcement learning for adaptive spacecraft guidance during finite-thrust rendezvous missions. Acta Astronaut. **201**, 129–141 (2022)
12. Leeghim, H., Choi, Y., Bang, H.: Adaptive attitude control of spacecraft using neural networks. Acta Astronaut. **64**(7–8), 778–786 (2009)
13. Bae, J., Kim, Y.: Adaptive controller design for spacecraft formation flying using sliding mode controller and neural networks. J. Franklin Inst. **349**(2), 578–603 (2012)

14. Xia, K., Huo, W.: Robust adaptive backstepping neural networks control for spacecraft rendezvous and docking with uncertainties. Nonlinear Dyn. **84**, 1683–1695 (2016)
15. Yin, H., Seiler, P., Arcak, M.: Stability analysis using quadratic constraints for systems with neural network controllers. IEEE Trans. Autom. Control **67**(4), 1980–1987 (2021)
16. Wang, R., Barbara, N., Revay, M., Manchester, I.R.: Learning over all stabilizing nonlinear controllers for a partially-observed linear system. arXiv preprint arXiv:2112.04219 (2021)
17. Newton, M., Papachristodoulou, A.: Stability of non-linear neural feedback loops using sum of squares. In: 2022 IEEE 61st Conference on Decision and Control (CDC), pp. 6000–6005 (2022)
18. Roberts, J.W., Manchester, I.R., Tedrake, R.: Feedback controller parameterizations for reinforcement learning. In: 2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL), pp. 310–317 (2011)
19. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach. Learn. **8**(3), 229–256 (1992)
20. Friedrich, S.R., Buss, M.: A robust stability approach to robot reinforcement learning based on a parameterization of stabilizing controllers. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 3365–3372 (2017)
21. Lawrence, N.P., Stewart, G.E., Loewen, P.D., Forbes, M.G., Backstrom, J.U., Gopaluni, R.B.: Reinforcement learning based design of linear fixed structure controllers. IFAC-PapersOnLine **53**(2), 230–235 (2020)
22. Holicki, T., Scherer, C.W., Trimpe, S.: Controller design via experimental exploration with robustness guarantees. IEEE Control Syst. Lett. **5**(2), 641–646 (2020)
23. Sullivan, J., Grimberg, S., D'Amico, S.: Comprehensive survey and assessment of spacecraft relative motion dynamics models. J. Guid. Control Dyn. **40**(8), 1837–1859 (2017)
24. Leomanni, M., Garulli, A., Giannitrapani, A., Quartullo, R.: Satellite relative motion modeling and estimation via nodal elements. J. Guid. Control Dyn. **43**(10), 1904–1914 (2020)
25. Di Cairano, S., Park, H., Kolmanovsky, I.: Model predictive control approach for guidance of spacecraft rendezvous and proximity maneuvering. Int. J. Robust Nonlinear Control **22**(12), 1398–1427 (2012)
26. Feng, W., Han, L., Shi, L., Zhao, D., Yang, K.: Optimal control for a cooperative rendezvous between two spacecraft from determined orbits. J. Astron. Sci. **63**, 23–46 (2016)
27. Eren, U., Prach, A., Koçer, B.B., Raković, S.V., Kayacan, E., Açıkmeşe, B.: Model predictive control in aerospace systems: current state and opportunities. J. Guid. Control Dyn. **40**(7), 1541–1566 (2017)
28. Leomanni, M., Bianchini, G., Garulli, A., Giannitrapani, A.: State feedback control in equinoctial variables for orbit phasing applications. J. Guid. Control Dyn. **41**(8), 1815–1822 (2018)
29. Mammarella, M., Capello, E., Park, H., Guglieri, G., Romano, M.: Tube-based robust model predictive control for spacecraft proximity operations in the presence of persistent disturbance. Aerospace Sci. Technol. **77**, 585–594 (2018)
30. Pagone, M., Boggio, M., Novara, C., Vidano, S.: A Pontryagin-based NMPC approach for autonomous rendez-vous proximity operations. In: 2021 IEEE Aerospace Conference (50100), pp. 1–9 (2021). IEEE
31. Galullo, M., Bucchioni, G., Franzini, G., Innocenti, M.: Closed loop guidance during close range rendezvous in a three body problem. J. Astronaut. Sci. **69**(1), 28–50 (2022)
32. Furfaro, R., Scorsoglio, A., Linares, R., Massari, M.: Adaptive generalized ZEM-ZEV feedback guidance for planetary landing via a deep reinforcement learning approach. Acta Astronaut. **171**, 156–171 (2020)
33. Holt, H., Armellin, R., Baresi, N., Hashida, Y., Turconi, A., Scorsoglio, A., Furfaro, R.: Optimal q-laws via reinforcement learning with guaranteed stability. Acta Astronaut. **187**, 511–528 (2021)
34. Leomanni, M., Bianchini, G., Garulli, A., Giannitrapani, A.: A class of globally stabilizing feedback controllers for the orbital rendezvous problem. Int. J. Robust Nonlinear Control **27**(18), 4607–4621 (2017)
35. Mania, H., Guy, A., Recht, B.: Simple random search of static linear policies is competitive for reinforcement learning. Adv. Neural Inf. Process. Syst. **31** (2018)
36. Walker, M.J.H., Ireland, B., Owens, J.: A set of modified equinoctial orbit elements. Celest. Mech. **36**(4), 409–419 (1985)
37. Battin, R.H.: An Introduction to the Mathematics and Methods of Astrodynamics. AIAA, Reston (1999)

38. Leomanni, M., Bianchini, G., Garulli, A., Giannitrapani, A.: Nonlinear orbit control with longitude tracking. In: Proceedings of the 55th IEEE Conference on Decision and Control, Las Vegas (USA), pp. 1316–1321 (2016)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.