SPACE ROBOTS

Optimality principles in spacecraft neural guidance and control

Dario Izzo[1]*, Emmanuel Blazquez[1], Robin Ferede[2], Sebastien Origer[2], Christophe De Wagter[2], Guido C. H. E. de Croon[2]

This Review discusses the main results obtained in training end-to-end neural architectures for guidance and control of interplanetary transfers, planetary landings, and close-proximity operations, highlighting the successful learning of optimality principles by the underlying neural models. Spacecraft and drones aimed at exploring our solar system are designed to operate in conditions where the smart use of onboard resources is vital to the success or failure of the mission. Sensorimotor actions are thus often derived from high-level, quantifiable, optimality principles assigned to each task, using consolidated tools in optimal control theory. The planned actions are derived on the ground and transferred on board, where controllers have the task of tracking the uploaded guidance profile. Here, we review recent trends based on the use of end-to-end networks, called guidance and control networks (G&CNets), which allow spacecraft to depart from such an architecture and to embrace the onboard computation of optimal actions. In this way, the sensor information is transformed in real time into optimal plans, thus increasing mission autonomy and robustness. We then analyze drone racing as an ideal gym environment to test these architectures on real robotic platforms and thus increase confidence in their use in future space exploration missions. Drone racing not only shares with spacecraft missions both limited onboard computational capabilities and similar control structures induced from the optimality principle sought but also entails different levels of uncertainties and unmodeled effects and a very different dynamical timescale.

INTRODUCTION

The design of a space exploration mission heavily relies on optimality principles. Given the absence of a safe harbor in space, every onboard resource, including propellant mass, available energy, and computing capabilities, must be used parsimoniously to ensure the highest possible mission return. Even with safety margins built into the mission plan, executing suboptimal plans can result in the failure of the entire mission. This is in contrast to many Earth-based industrial applications, where optimality is often not the main concern. To address this challenge, since the early days of space flight, optimal control models have been developed that capture the optimality principles relevant to different mission phases and translate them into elaborate system behaviors. In practice, the optimal guidance profile that follows the application of abstract optimality principles is carefully derived on the ground, well ahead of the mission launch, for most flown and planned missions. The plan is then uploaded on board and acted upon by the dedicated control system that, using the specific actuators available, tracks the planned profile by continuously canceling incurred deviations. This approach is common to interplanetary trajectory phases, landing phases, surface exploration phases, formation flying missions, and more (see Fig. 1), thanks to the abstract nature of the well-established optimal control theory, which allows capture of different dynamics, actuator models, and timescales. It also has the advantage of being a well-tested and validated approach with a history of successfully embedding optimality principles in the onboard control system, allowing missions to meet their requirements.

This common approach is, however, known to be suboptimal. It violates a "minimal intervention" principle, well described by Todorov and Jordan (*1*): Efforts to correct deviations from an average path should be made only when interfering with task performance. In our context, in the case of a deviation from the preplanned trajectory, the onboard controller should not try to steer the system back to the trajectory because the new situation may require other actions for optimality. The approach also carries higher risks in missions where substantial unmodeled effects, noise, uncertainties, and unforeseen events are present, which could result in strong deviations from the original plan. In such cases, a new optimal guidance profile may need to be developed on the ground and uploaded to the spacecraft, greatly hindering its autonomy.

This raises the question of why space missions do not continuously replan and compute the optimal guidance profile on board, for example, by using model predictive control (MPC). Recent efforts have been made to develop MPC approaches for various mission profiles, aiming to overcome these limitations. MPC has been studied in the past decades as a promising control approach for aerospace systems [see (*2*) for a review of MPC in this context] allowing onboard automated transformations of high-level optimality principles into actions. It relies on the availability of numerical methods to reliably solve some form of an optimal control problem, starting from the information on the current system state and the time. This online optimization returns an optimal sequence of open-loop predicted actions, the first of which is considered the best current control action. Despite great advances in associated numerical techniques and theory (*3*, *4*), the uptake of modern MPC approaches in space missions remains limited by the available onboard computational capabilities and the reliability of existing numerical solvers.

A different, albeit related, approach appeared more recently in the robotics and the aerospace fields. It is loosely inspired by models that interpret sensorimotor action in humans in terms of optimal control theory (*5*), thus suggesting how optimality principles, in the mathematical sense, are deeply embedded directly in the neuromusculoskeletal system. The new approach attempts to mimic this structure

[1]Advanced Concepts Team, European Space Research & Technology Centre, Keplerlaan 1, 2200 AG Noordwijk, Netherlands. [2]Micro Air Vehicle Lab, Faculty of Aerospace Engineering, Delft University of Technology, 2629 HS Delft, Netherlands.
*Corresponding author. Email: dario.izzo@esa.int

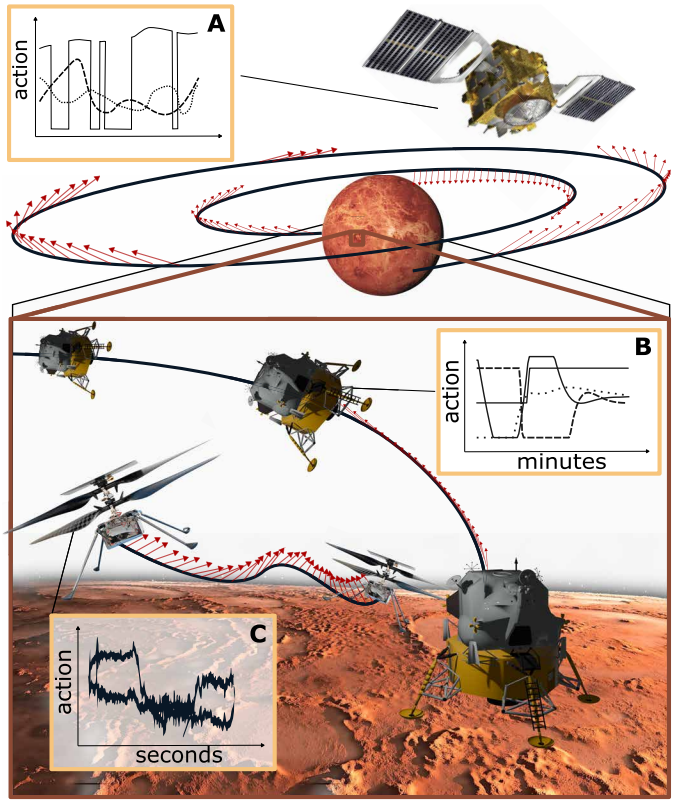

Fig. 1. Optimality principles determine the decision-making during different phases of exploration missions. (**A**) During an interplanetary phase, the spacecraft dynamics are well identified. Uncertainties are limited, and the departure from a theoretical mass optimal guidance is of less importance because of the relatively slow dynamics involved. Please note the timescale of the *x* axis in years and the bang-bang profile of the thrust (solid line). (**B**) During a landing phase, according to the specific mission profile, the adaptiveness and robustness of the planned actions have a larger effect on the mission success, also considering that human operators are typically too far away to allow replanning within an acceptable timeframe. Please note the timescale of the *x* axis in minutes and the discontinuous actions. (**C**) During a planetary exploration phase (for instance, rovers or flying drones), uncertainties are larger, and optimality principles, such as careful use of available onboard energy, need to be embedded into highly disturbed and fast dynamics (timescale in seconds). Depending on the phase of the missions, control systems may have a lot or very little time to recover from errors and cope with noise. The Ingenuity Mars helicopter is used here to visualize this case.

in the guidance and control architecture of space systems by training a deep artificial neural network (DNN) to represent directly the relation between the system state and its action under some predefined task and optimality principle. In this context, the network depth refers to the use of multilayer perceptrons that are capable, already with a few layers, of a high representativity (*6*, *7*). Following this approach, the guidance and control blocks of a typical space mission control architecture are substituted with one end-to-end DNN that does not make use of a reference trajectory and is thus also called a guidance and control network (G&CNet) (*8*, *9*). In comparison with the MPC approach, the step of having to solve an optimal control problem on board is bypassed and substituted by a single, computationally less expensive, neural inference (see Fig. 2). Most of the desirable properties of MPC [see (*2*) for a comprehensive review] are instead retained, with the new architecture requiring orders of magnitude fewer onboard computational resources. The treatment of constraints, often quoted as an advantage in MPC approaches, is, for example, also possible when training the DNN weights by introducing the corresponding task and optimality principle in the training pipeline. Ongoing research on this approach is focused on designing an efficient procedure to train the weights of artificial neural networks to accurately represent the desired task execution.

In this work, we review the use of G&CNets as a viable and promising path toward achieving onboard optimal decision-making in different space mission scenarios. We discuss past works that made use of two main techniques to train such networks in the context of space missions: behavioral cloning, an imitation learning approach that learns the optimal policy directly (essentially supervised learning), and reinforcement learning (RL). After evaluating the current state of the art for both approaches, we conclude by showing the use of drone flight racing as a safe stepping stone toward the onboard implementation of such networks for real missions and present concrete examples of the capabilities of embedded G&CNet implementations using drones as a model system. The success of G&CNets on extremely resource-restricted drones illustrates their potential to bring real-time optimal control within reach of a wider variety of robotic systems, both in space and on Earth.

OPTIMALITY IN GUIDANCE AND CONTROL OF SPACE SYSTEMS

Space systems are typically well characterized and tested extensively on the ground before launch. Their mass, inertia tensor, flexibility, and thrusters are all subject to thorough testing, and precise models of the entire system behavior are developed well in advance of the mission launch. However, in some cases, the development of the control system of a space system still requires the consideration of stochastic terms in its dynamics. Uncertainties can arise from unmodelable (or difficult-to-model) effects originating either from the system itself or from the environment in which it is designed to operate. Examples of the former include fuel sloshing (*10*, *11*), misthrust events (*12*–*14*), or corrupted sensor measurements (*15*). Uncertainties coming from the environment arise, for example, in deep space missions to asteroids where the body shape is known only to a limited extent during most mission phases (*16*, *17*) or in situations where the atmosphere of some celestial body plays a role, such as reentry, aerocapture, or exploratory drone flights (*18*–*20*). Uncertainties about solar activity or other space-environment quantities also result in unmodeled effects.

In any case, ignoring the stochastic contribution and using the sophisticated mathematical framework for deterministic optimal control, which emerged from Pontryagin's seminal work on optimal control theory (*21*), offers a starting point to understand the mathematical structure rising from chosen optimality principles. Under this framework, it is well known how the optimal feedback is already a discontinuous and nonlinear function of the system state for most simple low-dimensional and linear systems (*21*). An informing example is that of the time optimal precession angle control of a spinning satellite (*22*) where the analytical solution is available and allows exceptionally to observe the optimal control-switching structure in the full state space, revealing the extremely nonlinear and discontinuous nature of such a function (see Fig. 3). This is the case not only for time optimality but also, and to a larger extent, for higher-dimensional cases and propellant mass optimality: the primary optimality principle for most deep-space mission phases. In

deep-space missions, thrust is mostly achieved through the ejection of propellant mass, which causes the spacecraft to lose mass and accelerate in the opposite direction. This specific form of control for mass-varying systems creates a distinct structure of the resulting optimal actions that is the subject of a large body of works from the aerospace community [see, for example, (*23*)]. Thrust is also often modeled as a sequence of impulsive velocity changes rather than as a continuous action, and dedicated concepts such as the primer vector introduced by Lawden (*24*) have been used to expand on Pontryagin's work and help to answer complex questions on the resulting intricate structure of the optimal sequence of impulsive maneuvers (*25*–*28*).

When stochastic terms are considered in the system dynamics, the mathematical structure of the optimal control problem undergoes a substantial change. As a result, Pontryagin's theory is no longer applicable, and Bellman's dynamic programming methods (*29*) offer a more appropriate tool based on the concept of the optimal cost to go or value function (*30*). This added complexity is captured by a set of nonlinear, second-order, partial differential equations known as the Hamilton-Jacobi-Bellman equations (*31*). Obtaining a solution to these equations determines the value function and, consequently, the optimal policy. However, even in the simplest cases, solving these equations can be challenging, especially for problems relevant to this context. These problems often lead to discontinuous structures for optimal actions, requiring a high level of accuracy in the pursued final solution.

Most importantly, Bellman's mathematical framework allows for demonstrating the existence and uniqueness of solutions in terms of the value function and thus the existence (outside of singular corner cases) of optimal feedback in the form $\mathbf{u}^*(\mathbf{x})$ where $\mathbf{u}^*$ denotes the optimal feedback and $\mathbf{x}$ denotes the system state. In summary, for any deterministic or stochastic task, the current system state and an associated optimality principle are sufficient to decide the action to be taken. However, such an optimal state-action mapping is nonlinear and discontinuous and has an extremely intricate differential structure. Consequently, executing these actions becomes more challenging because of the requirement for precise state estimation and timing. Of course, these characteristics also pose additional difficulties when learning from the optimal feedback using deep neural networks, which is discussed profusely in the next section.

EMBEDDING OPTIMALITY PRINCIPLES INTO NEURAL MODELS

Artificial neural networks are highly versatile in their ability to approximate complex and discontinuous functions, even in their simplest shallow feed-forward architecture, as recently rediscussed in depth by Calin (*32*). They have been demonstrated to represent the optical characteristics of complex three-dimensional scenes with remarkable detail, as evidenced by the work of Mildenhall *et al.* (*33*), as well as to model the gravitational field of irregular bodies in the solar system with a precision exceeding that of classical methods, such as spherical harmonic expansions (*34*). Therefore, DNNs are an obvious choice for representing the complex structure of the optimal policies $\mathbf{u}^*$ (or the value function) often needed in space missions. A DNN (here denoted generically with $\mathcal{N}_\theta$) can formally approximate, within some tolerance ϵ, a parametric optimal feedback of the form

$$\mathbf{u}^*(\mathbf{x}, \mathbf{p}) = \mathcal{N}_\theta(\mathbf{x}, \mathbf{p}) + \epsilon \qquad (1)$$

where the parameters $\mathbf{p}$ may capture different tasks, objectives, and environmental properties, such as constraints and unknown gravitational effects. As a result, the biases and weights θ of the neural model incorporate multiple optimality principles, which are subsequently converted into control commands through onboard inference. This inference process, eventually to be carried out on board space agents, is becoming increasingly efficient because of the development of dedicated artificial intelligence (AI) accelerators for on-the-edge computations, leading to the creation of new space-qualified dedicated hardware (*35*). AI-focused processors were embarked in the Φ-Sat-1 (*36*) mission, field-programmable gate arrays on OPS-SAT (*37*), and HYPer-spectral Smallsat

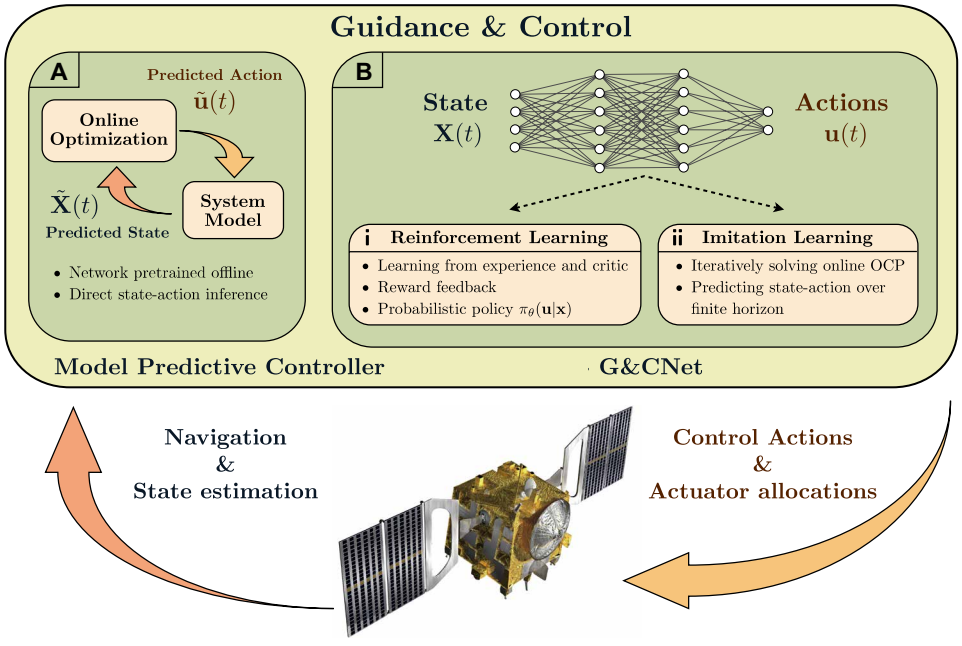

Fig. 2. G&CNets have a similar role to MPC in the architecture of an autonomous mission. (**A**) MPC iteratively solves onboard optimal control problems predicting state and actions over a defined time horizon on the basis of an existing system model. This results in possible optimality guarantees with full predictive information at the expense of heavy onboard computational burden determined by the complexity of the system model and the optimal control problem to be solved. (**B**) A G&CNet inference directly transforms the system state into actions. (i) When trained using RL, an agent learns from experience the final probabilistic policy on the basis of a critical reward-feedback loop with the environment. The resulting architecture can be resilient to stochastic disturbances but is often based on engineered reward functions that depart from the original optimality principle assigned. (ii) When trained via supervised learning, the network captures a clear optimality principle in its structure, directly inferring optimal actions from the state feedback at high frequency. Such a solution allows for fast direct inference with limited hardware requirements and is possibly subject to instability and lack of robustness when the state falls outside the set used to train the network.

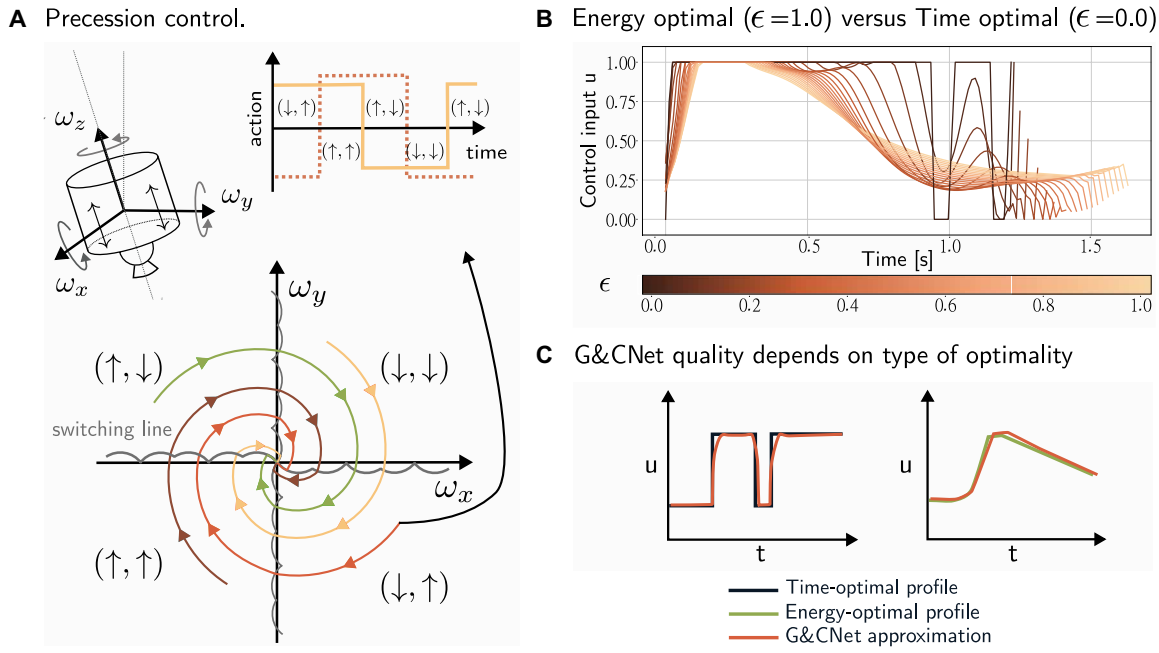

Fig. 3. Challenges in approximating optimal feedback with a G&CNet. (**A**) Optimal control tasks can have very high-dimensional solutions. Already in simple precession control, a complex structure emerges. In this case, the task is to lead in the shortest possible time a precessing satellite to a uniform rotation around its symmetry axis, thus canceling the components ω_x and ω_y of its angular velocity. The resulting deterministic optimal control problem is one rare case where an analytical solution can be derived, allowing us to peek into the structure of the optimal policy over the entire state space. According to the values of ω_x and ω_y, the thrusters are switching direction in correspondence to a complex and discontinuous switching line. The resulting time-optimal trajectories are shown in color. (**B**) The optimality principle pursued affects the resulting control profile and its gradient. Here, the optimal control commands from energy-optimal to time-optimal quadcopter flights are shown. The control profiles were obtained by solving optimal control problems with a direct method. The cost function used in this case (*59*) is $J(\mathbf{u}, t_f) = (1-\epsilon)\,t_f + \epsilon \int_0^{t_f} \|\mathbf{u}(t)\|^2 dt$, where $\mathbf{u}$ are the control inputs, t_f is the final time, and ϵ is a term allowing to gradually go from time to energy optimality. (**C**) Smooth control profiles result in smaller errors compared with nonsmooth bang-bang profiles when approximated by a G&CNet.

for ocean Observation (HYPSO-1) (*38*, *39*), and graphics processing units are being considered for future non–mission-critical applications (*40*, *41*). The use of DNNs for onboard systems with limited computational resources, such as spacecraft, cubesats, and planetary drones, has been only recently proposed in the context of space missions and is attracting the attention of a growing number of scientists. The term G&CNets (*8*, *9*), introduced in early studies at the European Space Agency, is here used to indicate such DNNs promising to replace traditional control and guidance architectures in future space missions. G&CNets' main attractiveness stems from their promise to bypass problems connected to the onboard solution of optimal control problems, typical, for example, of classic MPC schemes (*2*), at the cost of introducing the need to pretrain robust neural models on the ground. Trivially, in the limit case in which both an MPC and a G&CNet can compute the actual true optimal-feedback $\mathbf{u}^*$ without errors and with similar computational complexity, they correspond to equivalent guidance and control architectures. Two main approaches are mostly being studied in the context of G&CNet training: behavioral cloning and RL. In both cases, as part of the algorithmic framework, a simulation of the space system considered is required. The large variety of spacecraft and mission profiles limits the possibility to construct a single simulator able to capture all aspects, as opposed to a recent attempt for the specific case of quadrotors (*42*). Simulators are thus built and used on a case-by-case basis, typically involving the high-fidelity numerical solution of initial value problems defined upon differential equations describing the system dynamics, possibly including some augmented states. Typically, the absence of contact dynamics and complex aerodynamics effects and the highly precise identification of actuator models done for most space missions result in the capability of simulators to capture the real system dynamics with a high degree of fidelity.

Approaches based on behavioral cloning

The model parameters θ of the DNN approximating the parametric optimal feedback $\mathcal{N}_\theta(\mathbf{x}, \mathbf{p})$ can be learned via a supervised learning approach, provided some dataset

$$\mathcal{D} \coloneqq \{(\mathbf{x}, \mathbf{p}), \mathbf{u}^*\} \tag{2}$$

is available containing optimal state-action pairs for one or more tasks represented by $\mathbf{p}$. This approach is also referred to as imitation learning (*7*) or, more precisely, behavioral cloning (*43*) because the expert policy (an optimal pilot in this case) is learned directly. After the successful demonstration over a range of simulated landing tasks (*6*), a large number of works independently tested the capabilities of imitating the optimal control in diverse space contexts such as lunar and Mars landings (*43*, *44*), irregular asteroid landings (*45*), low-thrust missions and orbital transfers (*46*–*48*), solar sailing (*49*), proximity operations (*50*), drone flights (*7*), and misthrust problems (*51*). Although many of the cases reported achieved convincing results, much research still needs to be done to tackle the issues connected to the behavioral cloning approach: the efficient

creation of a dataset $\mathcal{D}$, the verification of requirements on the resulting onboard control system (*52*), and the inclusion of mechanisms able to cope with unmodeled components (*51*).

The creation of a dataset $\mathcal{D}$ requires running numerical optimal control solvers over a large set of initial states. This demanding computational effort can be alleviated by the use of techniques leveraging the proximity to previous solutions found [such as homotopy or continuation; (*53*)] but remains a limiting factor, although not one burdening the onboard inference speed. As a consequence, the number of optimal trajectories needed to build $\mathcal{D}$ for these specific space tasks has been, so far, mostly limited to the order of tens of thousands (*48*, *49*, *51*, *54*). Small datasets are unable to harness the full potential of a deep supervised learning pipeline and only allow for partial investigations of the potential of G&CNets, restricting possible results to only small portions of the state space. In cases where Pontryagin's theory can be used to derive necessary conditions for optimality via the introduction of a two-point boundary value problem defined on the augmented dynamics (indirect methods), a technique called backward generation of optimal examples (BGOE; see Fig. 4) has been recently introduced to allow the creation of datasets $\mathcal{D}$ that are orders of magnitude larger than what is otherwise possible. BGOE has been used in the context of studies on Earth-Venus mass optimal interplanetary transfers (*46*), as well as time-optimal asteroid belt mining missions (*47*), showing promising results and allowing the creation of public datasets containing millions of optimal trajectories rather than only a few thousand.

Studies on the use of BGOE are still preliminary, and the technique cannot be used in general, for example, in contexts where indirect optimization methods fail. As a consequence, most of the past works using the behavioral cloning approach to train a G&CNet for a spacecraft suffer from an insufficient representativity of the trained network, resulting in the system state propagating outside the training set and thus the onboard closed loop becoming unstable. In the presence of strong perturbations and uncertainties, this can be the case even if the dataset size is appropriate. To tackle this issue, approaches of data augmentation implementing different variations of the dataset aggregation method (*55*) technique have been not only proposed and used (*43*, *44*, *51*, *54*) but also criticized because they increase the burden of generating additional optimal state-action pairs.

An alternative approach to creating the dataset $\mathcal{D}$ proposed outside of the traditional space domain involves the use of optimal controllers as the guiding supervisory signal, as explored in the works by Nubert *et al.* (*56*) and recent developments in near-optimal rapid MPC methods (*57*). This strategy enables real-time inference of an approximate MPC policy, thereby circumventing, similarly to G&CNETs, the computational overhead associated with solving optimization problems at each time step. However, it introduces an additional layer of approximation, which may be superfluous when access to the actual optimal control is readily available as a teacher signal. To address concerns arising from the use of such an approximate policy, a "dual policy" learning scheme has been proposed [as documented in (*57*)] that conducts an online assessment of the controller's optimality. In cases where the proposed controls are found to be suboptimal, a cautious fallback mechanism is triggered using established controllers, such as proportional-integral-derivative or linear quadratic regulator, intentionally sacrificing optimality to ensure constraint satisfaction.

Last, the question of how many different tasks and optimality principles can be embedded into a single DNN using a behavioral cloning approach remains open because works addressing this issue were only carried out preliminarily for interplanetary trajectories (*58*) and drones (*59*, *60*). These early results show how the introduction of multiple tasks and environment variables encoded in the extra parameters $\mathbf{p}$ seems to help in regularizing the discontinuous behavior resulting from aggressive optimality principles and in coping with unforeseen, unmodeled external perturbations (see the later section on onboard drone flight implementation).

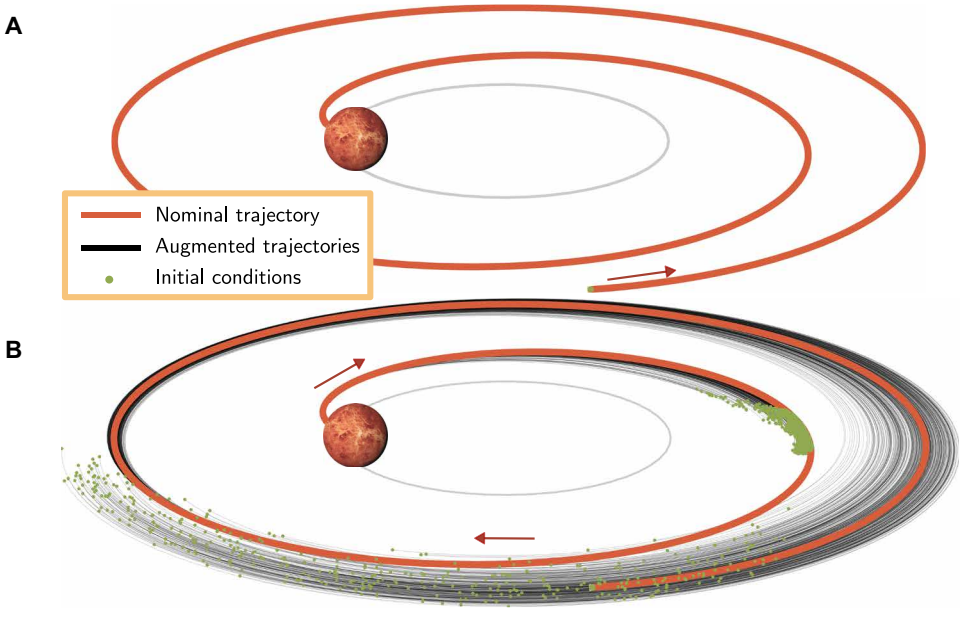

Fig. 4. The BGOE technique allows generation of orders-of-magnitude larger datasets by perturbing one nominal solution. (**A**) The nominal solution for the case of a time-optimal transfer from the asteroid belt to Earth (visualized in a rotating frame). (**B**) Two bundles of 200,000 optimal trajectories were found by applying BGOE to the nominal solution. Larger perturbations of the nominal trajectory result in better coverage of conditions close to the Earth (short bundle), which reduces the likelihood that the spacecraft lands outside of the training data (*47*). For comparison, the generation of all 400,000 trajectories uses the same numerical resources used to generate one nominal optimal solution. For clarity, we show only a portion of this dataset here.

Approaches based on deep reinforcement learning

A second approach suitable for learning the parameters θ of a G&CNet is deep RL (DRL). DRL captures a large variety of approaches and numerical methods concerned with the problem of an agent learning a policy to perform a specified task in its environment from experience and not from expert demonstrations. The policy, often indicated with the symbol $\pi_\theta(\mathbf{u}|\mathbf{x})$, is represented by a DNN

and returns the probability to choose the control $\mathbf{u}$ given the agent state $\mathbf{x}$. In a deterministic setting, such a policy, when optimal, must correspond to the solution of the related optimal control problem so that one can formally write

$$\pi_\theta^*(\mathbf{u}\,|\,\mathbf{x}) = \delta(\mathbf{u} - \mathbf{u}^*(\mathbf{x})) \tag{3}$$

where the Dirac delta has been used to indicate certainty over choosing the optimal feedback u^*. In a stochastic setting, the DNN typically outputs some statistical property of the policy, most commonly its mean value—the variance being often considered constant. To use DRL to train the parameters of a G&CNet, the continuous control problem has to be modeled as an agent that learns through a sequential decision-making process, a Markov decision problem, where the simulated environment includes all types of uncertainties relevant to the particular mission phase considered. Early works (*61*, *62*), for example, applied this approach to the problem of controlling a spacecraft hovering over an asteroid with an uncertain gravity field and considering a stochastic dynamics perturbed by solar radiation pressure acceleration as well as accounting for sensor noise. The approach was later extended and refined to interplanetary transfers (*9*, *63*), rendezvous and docking (*64*, *65*), planetary landing problems (*66*), and drone flights (*67*, *68*). In all of these cases, the trained DNN was suitable for onboard use and proved to be robust to different levels and types of stochastic effects. Although very promising, the DRL approach has much to prove in terms of actual optimality. Optimality principles and terminal and path constraints are all encapsulated in the so-called reward function driving most of the agent learning. Engineering an appropriate reward function turns out to be problematic in most cases, and, when successful, the optimality principle that it corresponds to is unclear. Not surprisingly, the suboptimality resulting from the DRL approach was noted and quantified by the authors of successful implementations (*9*, *63*, *64*).

In a different context, specifically in the domain of drone racing, recent advancements in RL approaches have yielded noteworthy performances. In drone racing, the challenge associated with defining a dense reward function is considerably mitigated, if not entirely eliminated. This distinctive characteristic emerges from the ability to regard a reduction in distance to the targeted gate as approaching optimality. This feature stands in stark contrast, for example, to the interplanetary transfer case, where the task of formulating a dense reward function is more complex because of the absence of inherent trajectory-related metrics. Recent work proposed the use of DRL to learn Lyapunov functions as well as the domain-specific Q-law (*69*) in multiple revolution interplanetary transfers and showed the difficulty of finding such dense rewards in general.

Song *et al.* (*70*) have recently argued that modern DRL efforts are superior to methods based on optimal control because they do not seek to optimize a given objective function better; instead, they intrinsically define a better objective. We agree that this may be the case but only for systems where uncertainties are prevailing and approaches grounded on the underlying optimal control theory (Bellman equations) fail to provide sufficient performances. The recent paper from Kumar *et al.* (*71*) is also relevant to this discussion because advantages of behavioral cloning over DRL were found in cases where noise levels were low and suboptimal demonstrations were absent. Many of the space applications where G&CNETs have been proposed fall into this category. A common criticism of the DRL approach is its high computational requirements, which can result in computational times three orders of magnitude larger than those required for running one single optimal control solver to find a policy (*9*). Efforts have also been made to tackle this issue. One example is to integrate knowledge about a linear quadratic regulator (here the piecewise affine structure of the control law and the maximal control invariant sets) into the RL algorithm (*72*), thus enhancing the efficiency of learning the control policy. Last, although behavioral cloning methods have been used to seamlessly learn a variety of policies and thus integrate several tasks into the same deep neural network, doing the same in a DRL framework requires a careful design of the reward function and adds further complexity to the resulting pipeline.

TESTING ON FLYING DRONES

The field of onboard guidance and control of space systems using DNNs, specifically the G&CNet architecture, is in its infancy. To gain acceptance within the aerospace community as a potential improvement over current guidance and control schemes, convincing evidence of its embedded capabilities is necessary, as well as the possibility of offering guarantees on the resulting system behavior. Unfortunately, space missions are also extremely expensive and, with the exception of a few technology development platforms, do not allow for extensive testing of mission-critical software in space. With much of the current research on G&CNets focused on simulations, concerns are raised that the reality gap may be overlooked (*73*–*77*).

To address this issue, we propose that G&CNets can be studied on the challenging real-world task of drone racing. Drones have differences with spacecraft. For instance, time-optimal drone flights operate on a different timescale compared with spacecraft interplanetary transfers or landings. Moreover, unmodeled effects and disturbances tend to be larger, particularly for smaller vehicles like micro air vehicles. System identification for drones is also often challenging and may not achieve the level of accuracy typically required in space robotics systems. However, drones also share strong similarities with space systems because they too have limited onboard resources and require careful optimization of their use. The time optimality principle in drone racing, coupled with the complex dynamics of the drones, provides a valuable comparison to the mass optimality principles preeminent in space missions. Hence, drones represent a hard use case in terms of latency requirements, reality gaps, and resource constraints. Successfully embedding neural-based optimal control systems onboard drones will provide increasing confidence for the use of this method on space systems. An ongoing effort is underway to integrate G&CNets into quadcopter flight control systems, aiming to achieve end-to-end control with optimal flight time and to increase the trust in the use on board spacecraft. Building on research that used simplified quadcopter models in simulations (*6*, *7*) and performed real-world flight tests where G&CNets were used for longitudinal control (*78*), tests have been conducted where a high-dimensional quadcopter model with 16 degrees of freedom was used to compute the optimal-feedback for the networks to imitate. The networks were then trained for end-to-end control, which involved sending motor commands directly without any intermediate controllers (*59*, *60*). The reason to send direct motor commands is to grant the network complete control authority, allowing it to handle actuator saturation directly. Consequently, the learned solution is not constrained by either the temporal

dynamics of an inner-loop controller or the saturation priorities assigned in motor mixing as in previous attempts. However, this direct approach also posed new challenges because the time optimality principle pursued resulted in aggressive maneuvers that were highly dependent on the network's ability to accurately replicate the ground truth optimal actions. Because of the short dynamical timescale of the quadcopter flights considered (on the order of seconds), even minor discrepancies in the network's performance considerably affect the overall control accuracy. In hover-to-hover flights, unmodeled moments caused the drone to take a suboptimal path, diving down and overshooting its target. Additionally, modeling errors accumulated and caused deviations from the optimal plan, which destabilized the trajectory (*60*).

Despite the difficulty in modeling the system dynamics, sensor measurements on board can be used to detect discrepancies between predicted and actual forces and moments. This suggests that the G&CNet can be modified by adding to its inputs estimated unmodeled effects. A modified network $\mathcal{N}_\theta(\mathbf{x}, \mathbf{p})$ can then be trained to approximate the solution of a parametric optimal control problem, where $\mathbf{p}$ represents the discrepancy between the modeled dynamics and the observed behavior. By comparing the predicted and measured moments on board and using their difference as input to the network (*60*), G&CNets have been shown to adapt to unexpected moments, delivering a considerable improvement in stability and optimality, as illustrated during real flights of a Parrot AR 2.0 drone in Fig. 5A.

In (*59*), a similar approach was adopted, where the G&CNet was modified to instead handle unexpected actuator saturations. It was demonstrated that the knowledge of the actuator's limits was crucial to remaining on the optimal path during high-speed successive waypoint flights. By estimating a model parameter on board (in this case, the maximum motor revolutions per minute) and feeding it back into an additional neural network input, G&CNets can thus cope with unmodeled effects, as shown in Fig. 5B.

Additional parameters fed into a G&CNet can also represent different tasks. For instance, in (*59*), a G&CNet was also trained to fly through two waypoints, where the waypoint positions in space, which act as boundary conditions to the optimal control problem, can vary. When computing the state-action pairs to use in the training dataset, different distances between waypoints are used, and the network thus learns to represent the solution of the optimal control problem specific to the parametric geometry that it receives as additional input. During the flight, the network can then optimally plan a path through two waypoints, where the second waypoint position is displaced, as illustrated in Fig. 6.

G&CNets face serious challenges when highly aggressive or acrobatic maneuvers are pursued on drones. The G&CNets here showcased during real flights were trained to imitate energy optimal control (*60*) or a mixture of energy and time optimal control (*59*). Energy optimality was preferred because it resulted in smoother trajectories, which were simpler to learn and execute. However, moving toward more time-optimal trajectories also creates a challenge as the optimal solution approaches a bang-bang profile, which is increasingly difficult to learn (see Fig. 3C), particularly for the small networks used (*59*). Moreover, the aggressive commands will steer the drone toward the edge of its flight envelope, where the dynamics are most unpredictable. Therefore, the reality gap problem remains the biggest obstacle to this approach in the context of aggressive maneuvering. Although the adaptive networks have shown important

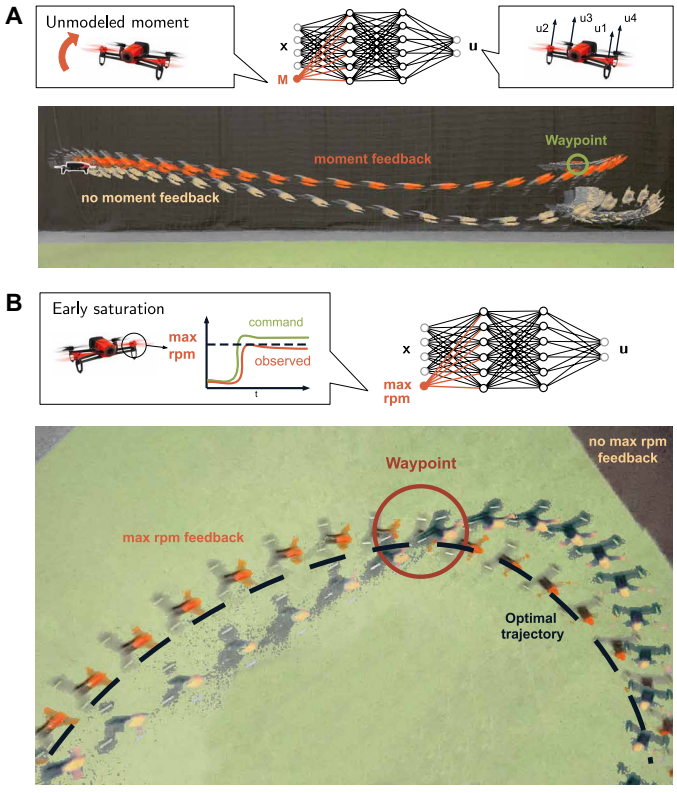

Fig. 5. G&CNets' robustness to model mismatch. Two examples of real autonomous flights of a Parrot AR drone 2.0 in the TU Delft Cyberzoo: (**A**) Unmodeled moments are detected in real time on board and fed back to the G&CNet. The initial drone position is also shown with a white border. (**B**) Unexpected early saturation of the motor revolutions per minute is present. The saturation is estimated on board and fed back to the G&CNet. In both cases, the G&CNet learns a class of optimal control policies and selects the one to enact according to the detected mismatch. This results in improved trajectories: (A) The drone reaches the waypoint without unnecessarily losing altitude. (B) The drone follows the optimal trajectory more closely (yellow dashed line).

improvement in robustness, additional alterations are necessary to guarantee more general robustness for modeling errors, sensor noise, and delays.

These findings demonstrate that relatively small G&CNets, in this case, a feed-forward neural network with three hidden layers consisting of 120 neurons each, are capable of representing a broad range of optimal control problems and tasks effectively. The resulting G&CNets can cancel out rather than accumulate the errors made in flight, both in simulation and on the real quadcopter. This is, in part, possible not only because they can be made adaptive but also because they can be inferred at a high rate (450 Hz) in real time on board the drone (as measured on the Parrot P7 dual-core Cortex A9 CPU). A dual-core 800-MHz ARM Cortex-A9 processor is also now orbiting on board the European Space Agency's OPS-SAT satellite (*79*), thus making the results achieved of particular interest here because they map closely to the space systems with similar computational capabilities.

As discussed previously, recent success in drone racing highlights the benefits of RL when compared with MPC (*70*), showing how RL excels in directly addressing model uncertainties using domain randomization, which bolsters the controller's robustness.

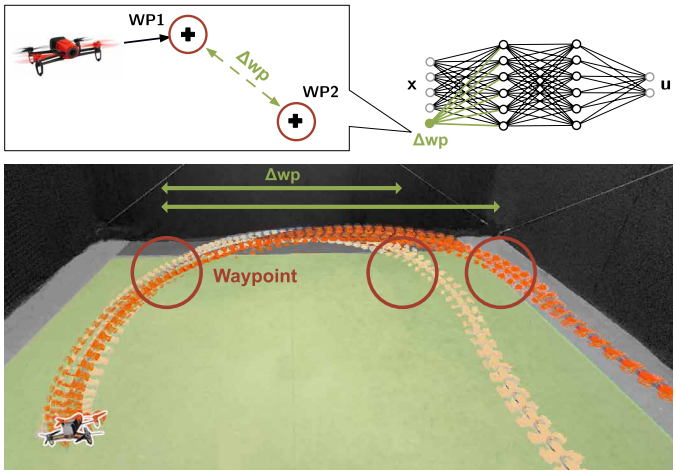

Fig. 6. Embedding multiple tasks in one G&CNet. Real autonomous flights of a Parrot AR drone 2.0 in the TU Delft Cyberzoo. During this specific test, the same G&CNet is shown to have learned to imitate the optimal feedback in two distinct tasks differing in the final waypoint position. The tasks are different because the optimal approach to a given waypoint depends on the position of the next waypoint. The position of the next waypoint is fed to the DNN as an extra parameter. The initial drone position is also shown with a white border.

This success exemplifies the benefits of integrating guidance and control within a neural network, in contrast to separating the control problem into planning and tracking. A particularly impressive follow-up is the achievement in (*80*), where an RL controller's performance surpasses that of three human drone racing champions. In this work, the authors crossed the reality gap through the fusion of abstraction, embodied in an inner-loop controller that tracks thrust and rate commands, with the precise modeling of the closed-loop system using a learned residual model (*81*). In other work (*82*), it was demonstrated that performance could be improved further by using end-to-end RL without abstraction layers. This serves as a compelling testament to RL's remarkable performance in the context of drone racing, where the availability of a dense reward function and other specificities of the task make the DRL approach much more efficient.

FUTURE WORKS

Small neural networks have demonstrated, in several scenarios relevant to space exploration, their ability to capture optimality principles and to perform various guidance and control tasks requiring high levels of autonomy. As the development of specialized accelerators for onboard inference continues to grow (*35*), there is increasing interest in exploring which additional capabilities can be integrated into larger networks. However, because of the data-intensive nature of deep architectures, the efficient generation of training examples is imperative for realizing the potential of larger networks. End-to-end training of large models, when performed on board spacecraft, remains an open question, subject to the availability of onboard acceleration and the development of distributed learning systems as a key enabling technology (*83*). This could result in considerable autonomy benefits for future deep space missions, and recent interest in satellite swarms and constellations may rapidly lead to new developments in this field.

The question of whether the training procedure of G&CNets should imitate optimal examples or use a RL paradigm remains open and can today only be answered on a case-by-case basis that considers the user's desired trade-off between computational efficiency, robustness, quest for optimality, and interpretability. Exploring the synergies between these two approaches may lead to innovative, hybrid solutions, yet this topic will surely continue to fuel scientific discussions in the years to come.

Unresolved challenges remain regarding the complete qualification and validation of artificial neural networks when embedded into safety-critical systems like the guidance, navigation, and control subsystems. In this regard, it is crucial to differentiate between neural-based control schemes on the basis of the presence or absence of an online learning-based component. In the first scenario, extensive research conducted in the robotics, automotive, and aeronautic domains advocates for a paradigm shift in the certification process (*84*–*87*). It also highlights the difficulty in making meaningful assertions about system behavior during the learning process. Safe learning for control applications aims at ensuring the stability and robustness of the proposed data-driven solutions in the face of system uncertainties. This can be achieved through learning-based model augmentation and reduction of the uncertainty envelope, encouraging robustness considerations in learning policies via analytical or heuristic-based methods, or applying so-called control certification filters at the output of a neurocontroller to either constrain the control policy or offer redundancy via the switch to robust controllers in the case of predicted system destabilization. Conversely, in the second scenario, where learning is performed a priori, neural-based systems can be treated as fixed black boxes or mathematical functions, similar to other embedded control algorithms. The majority of existing work on G&CNets falls into this second category, although incorporating an online learning component into the proposed architectures is a naturally anticipated future development.

Similarly to other embedded control algorithms, to achieve certification, end-to-end neural guidance and control implementations thus need to demonstrate robustness to stochastic parametric and dynamic system uncertainties, as well as compatibility with modern fault detection, isolation, and recovery routines that require both explainability and adaptability. Fortunately, recent developments in explainable AI offer new possibilities for developing more comprehensive deep neural models that can provide greater transparency and understanding of their capabilities and limitations, which could ultimately unlock mission-critical certifications (*88*). It is also crucial to ensure that neural systems perform as intended under a range of operating conditions and environments while meeting safety and performance requirements. To this end, and specifically for G&CNets, a first valid approach to the stability analysis has been proposed on the basis of the complete high-order expansion of the resulting closed-loop dynamics (*52*), allowing proof of the stability of a system with multiple degrees of freedom controlled by an end-to-end artificial neural network. Although more work is needed to generalize the results obtained, this result shows the possibility of approaching the study of G&CNet stability as any other controlled closed-loop system.

Neuromorphic technologies have recently emerged as promising enablers for onboard edge computing and learning applications in space. These technologies propose innovatory software and hardware designs inspired by biological neural systems, focusing on low

power and energy efficiency, which are highly synergetic with space applications. The widespread availability of neuromorphic dynamic vision sensors (*89*) and chips (*90*) has led to a recent surge in publications on the use of event-based cameras and spiking neural networks on board spacecraft (*91*–*94*), with an event-based sensor having recently been sent to the International Space Station (*95*). Of particular interest for G&CNets is the neuromorphic promise to enable real-time continuous learning, with the potential to revolutionize neural guidance and control capabilities and robustness at the cost of developing and accepting new certification procedures for learning-based safety-critical systems.

CONCLUSIONS

In this study, we have reviewed a nascent trend in neural spacecraft guidance and control, inspired by the presence of optimality principles in human sensorimotor actions, that leverages an onboard end-to-end DNN to capture the relation between the spacecraft's state and its optimal actions. The reviewed G&CNet architecture is computationally efficient and suitable for real-time onboard processing and has shown great potential in representing complex optimal state-feedback relations in various scenarios of interest while maintaining many of the favorable attributes of more classical control methods. Preliminary attempts to build trust in this approach have been presented that successfully embed G&CNet on board drones on hardware compatible with modern space CPUs using time-optimal flights as a proxy for a real space guidance and control system. Looking ahead, we foresee the adoption of onboard neural guidance and control as a concrete option in future space missions because it enables the necessary autonomy to meet the ambitious goals of future space missions while parsimoniously using onboard available resources.

REFERENCES AND NOTES

1. E. Todorov, M. I. Jordan, Optimal feedback control as a theory of motor coordination. *Nat. Neurosci.* **5**, 1226–1235 (2002).
2. U. Eren, A. Prach, B. Bahadır Koçer, S. V. Raković, E. Kayacan, B. Açıkmeşe, Model predictive control in aerospace systems: Current state and opportunities. *J. Guid. Control Dynam.* **40**, 1541–1566 (2017).
3. B. Açıkmeşe, J. M. Carson, L. Blackmore, Lossless convexification of nonconvex control bound and pointing constraints of the soft landing optimal control problem. *IEEE Trans. Control Syst. Technol.* **21**, 2104–2113 (2013).
4. G. Frison, M. Diehl, HPIPM: A high-performance quadratic programming framework for model predictive control. *IFAC-Pap. OnLine* **53**, 6563–6569 (2020).
5. E. Todorov, Optimality principles in sensorimotor control. *Nat. Neurosci.* **7**, 907–915 (2004).
6. C. Sánchez-Sánchez, D. Izzo, Real-time optimal control via deep neural networks: Study on landing problems. *J. Guid. Control Dynam.* **41**, 1122–1135 (2018).
7. D. Tailor, D. Izzo, Learning the optimal state-feedback via supervised imitation learning. *Astrodynamics* **3**, 361–374 (2019).
8. D. Izzo, M. Märtens, B. Pan, A survey on artificial intelligence trends in spacecraft guidance dynamics and control. *Astrodynamics* **3**, 287–299 (2019).
9. A. Zavoli, L. Federici, Reinforcement learning for low-thrust trajectory design of interplanetary missions. arXiv:2008.08501 (2020).
10. J. Vreeburg, Spacecraft maneuvers and slosh control. *IEEE Control. Syst. Mag.* **25**, 12 (2005).
11. M. Reyhanoglu, J. Rubio Hervas, Nonlinear control of a spacecraft with multiple fuel slosh modes, in *2011 50th IEEE Conference on Decision and Control and European Control Conference* (IEEE, 2011), pp. 6192–6197.
12. P. Servidia, R. Pena, Spacecraft thruster control allocation problems. *IEEE Trans. Automat. Contr.* **50**, 245–249 (2005).
13. W. Cai, X. H. Liao, Y. D. Song, Indirect Robust adaptive fault -tolerant control for attitude tracking of Spacecraft. *J. Guid. Control Dynam.* **31**, 1456–1463 (2008).
14. N. Zhou, Y. Kawano, M. Cao, Neural network-based adaptive control for spacecraft under actuator failures and input saturations. *IEEE Trans. Neural Netw. Learn. Syst.* **31**, 3696–3710 (2020).
15. S. Yin, B. Xiao, S. X. Ding, D. Zhou, A review on recent development of spacecraft attitude fault tolerant control system. *IEEE Trans. Ind. Electron.* **63**, 3311–3320 (2016).
16. J. Melman, E. Mooij, R. Noomen, State propagation in an uncertain asteroid gravity field. *Acta Astronaut.* **91**, 8–19 (2013).
17. Y. Ren, J. Shan, Reliability-based soft landing trajectory optimization near asteroid with uncertain gravitational field. *J. Guid. Control Dynam.* **38**, 1810–1820 (2015).
18. H. F. Grip, D. P. Scharf, C. Malpica, W. Johnson, M. Mandic, G. Singh, L. A. Young, Guidance and control for a Mars helicopter, in *2018 AIAA Guidance, Navigation, and Control Conference* (American Institute of Aeronautics and Astronautics, 2018), 10.2514/6.2018-1849.
19. H. F. Grip, J. Lam, D. S. Bayard, D. T. Conway, G. Singh, R. Brockers, J. H. Delaune, L. H. Matthies, C. Malpica, T. L. Brown, A. Jain, A. M. San Martin, G. B. Merewether, Flight control system for NASA's Mars helicopter, in *AIAA Scitech 2019 Forum* (American Institute of Aeronautics and Astronautics, 2019), 10.2514/6.2019-1289.
20. J. Balaram, M. Aung, M. P. Golombek, The Ingenuity helicopter on the Perseverance rover. *Space Sci. Rev.* **217**, 56 (2021).
21. L. S. Pontryagin, V. G. Botlyanskii, R. V. Gamkrelidze, E. F. Mishchenko, *The Mathematical Theory of Optimal Processes* (John Wiley & Sons, 1962).
22. D. Izzo, Internal mesh optimization and Runge-Kutta collocation in a direct transcription method applied to interplanetary missions, in *Proceedings of the 55th International Astronautical Congress of the International Astronautical Federation* (International Astronautical Federation, 2004), 10.2514/6.IAC-04-A.6.04.
23. B. A. Conway, *Spacecraft Trajectory Optimization*, Cambridge Aerospace Series (Cambridge Univ. Press, 2010).
24. D. Lawden, *Optimal Trajectories For Space Navigation* (Butterworth, 1963).
25. T. Edelbaum, How many impulses, in *3rd and 4th Aerospace Sciences Meeting* (American Institute of Aeronautics and Astronautics, 1967), 10.2514/6.1966-7.
26. J. E. Prussing, Optimal impulsive linear systems: Sufficient. *J. Astronaut. Sci.* **43**, 195–206 (1995).
27. T. E. Carter, Necessary and sufficient conditions for optimal impulsive rendezvous with linear equations of motion. *Dyn. Control* **10**, 219–227 (2000).
28. E. Taheri, J. L. Junkins, How many impulses redux. *J. Astronaut. Sci.* **67**, 257–334 (2020).
29. R. Bellman, R. E. Kalaba, *Dynamic Programming and Modern Control Theory* (Elsevier Science, 1965).
30. E. Todorov, Optimal control theory, in *Bayesian Brain: Probabilistic Approaches to Neural Coding*, K. Doya, S. Ishii, A. Pouget, R. P. N. Rao, Eds. (MIT Press, 2006), pp. 260–298.
31. M. Bardi, I. C. Dolcetta, *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations* (Birkhäuser, 1997).
32. O. Calin, Universal approximators, in *Deep Learning Architectures: A Mathematical Approach*, Springer Series in Data Science (Springer, 2020), pp. 251–284.
33. B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, R. Ng, Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* **65**, 99–106 (2021).
34. D. Izzo, P. Gómez, Geodesy of irregular small bodies via neural density fields. *Commun. Eng.* **1**, 48 (2022).
35. G. Furano, G. Meoni, A. Dunne, D. Moloney, V. Ferlet-Cavrois, A. Tavoularis, J. Byrne, L. Buckley, M. Psarakis, K.-O. Voss, L. Fanucci, Towards the use of artificial intelligence on the edge in space systems: Challenges and opportunities. *IEEE Aerosp. Electron. Syst. Mag.* **35**, 44–56 (2020).
36. G. Giuffrida, L. Fanucci, G. Meoni, M. Batič, L. Buckley, A. Dunne, C. Van Dijk, M. Esposito, J. Hefele, N. Vercruyssen, G. Furano, M. Pastena, J. Aschbacher, The Φ-Sat-1 mission: The first on-board deep neural network demonstrator for satellite earth observation. *IEEE Trans. Geosci. Remote Sens.* **60**, 5517414 (2022).
37. G. Labrèche, D. Evans, D. Marszk, T. Mladenov, V. Shiradhonkar, T. Soto, V. Zelenevskiy, OPS-SAT spacecraft autonomy with TensorFlow lite, unsupervised learning, and online machine learning, in *2022 IEEE Aerospace Conference* (IEEE, 2022). 10.1109/AERO53065.2022.9843402.
38. A. S. Danielsen, T. A. Johansen, J. L. Garrett, Self-organizing maps for clustering hyperspectral images on-board a CubeSat. *Remote Sens.* **13**, 4174 (2021).
39. R. Pitonak, J. Mucha, L. Dobis, M. Javorka, M. Marusin, CloudSatNet-1: FPGA-based hardware-accelerated quantized CNN for satellite on-board cloud coverage classification. *Remote Sens.* **14**, 3180 (2022).
40. B. Denby, B. Lucia, Orbital edge computing: Nanosatellite constellations as a new class of computer system, in *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems* (ACM, 2020), pp. 939–954.
41. F. C. Bruhn, N. Tsog, F. Kunkel, O. Flordal, I. Troxel, Enabling radiation tolerant heterogeneous GPU-based onboard data processing in space. *CEAS Space J.* **12**, 551–564 (2020).
42. P. Foehn, E. Kaufmann, A. Romero, R. Penicka, S. Sun, L. Bauersfeld, T. Laengle, G. Cioffi, Y. Song, A. Loquercio, D. Scaramuzza, Agilicious: Open-source and open-hardware agile quadrotor for vision-based flight. *Sci. Robot.* **7**, eabl6259 (2022).

43. O. Mulekar, R. Bevilacqua, H. Cho, Metric to evaluate distribution shift from behavioral cloning for fuel-optimal landing policies. *Acta Astronaut.* **203**, 421–428 (2023).
44. R. Furfaro, I. Bloise, M. Orlandelli, P. Di Lizia, F. Topputo, R. Linares, Deep learning for autonomous lunar landing, in *2018 AAS/AIAA Astrodynamics Specialist Conference*, vol. 167 of *Advances in the Astronautical Sciences* (Univelt, 2018), pp. 3285–3306.
45. L. Cheng, Z. Wang, Y. Song, F. Jiang, Real-time optimal control for irregular asteroid landings using deep neural networks. *Acta Astronaut.* **170**, 66–79 (2020).
46. D. Izzo, E. Öztürk, Real-time guidance for low-thrust transfers using deep neural networks. *J. Guid. Control Dyn.* **44**, 315–327 (2021).
47. D. Izzo, S. Origer, Neural representation of a time optimal, constant acceleration rendezvous. *Acta Astronaut.* **204**, 510–517 (2023).
48. H. Li, H. Baoyin, F. Topputo, Neural networks in time-optimal low-thrust interplanetary transfers. *IEEE Access* **7**, 156413–156419 (2019).
49. L. Cheng, Z. Wang, F. Jiang, C. Zhou, Real-time optimal control for spacecraft orbit transfer via multiscale deep neural networks. *IEEE Trans. Aerosp. Electron. Syst.* **55**, 2436 (2018).
50. L. Federici, B. Benedikter, A. Zavoli, Deep learning techniques for autonomous spacecraft guidance during proximity operations. *J. Spacecr. Rockets* **58**, 1774–1785 (2021).
51. A. Rubinsztejn, R. Sood, F. E. Laipert, Neural network optimal control in astrodynamics: Application to the missed thrust problem. *Acta Astronaut.* **176**, 192–203 (2020).
52. D. Izzo, D. Tailor, T. Vasileiou, On the stability analysis of deep neural network representations of an optimal state feedback. *IEEE Trans. Aerosp. Electron. Syst.* **57**, 145 (2020).
53. E. Trélat, Optimal control and applications to aerospace: Some results and challenges. *J Optim. Theory Appl.* **154**, 713–758 (2012).
54. O. Mulekar, H. Cho, R. Bevilacqua, Six-degree-of-freedom optimal feedback control of pinpoint landing using deep neural networks, in *AIAA Scitech Forum* (American Institute of Aeronautics and Astronautics, 2023), p. 0689.
55. S. Ross, G. Gordon, D. Bagnell, A reduction of imitation learning and structured prediction to no-regret online learning, in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, JMLR Workshop and Conference Proceedings (MLResearch Press, 2011), pp. 627–635.
56. J. Nubert, J. Köhler, V. Berenz, F. Allgöwer, S. Trimpe, Safe and fast tracking on a robot manipulator: Robust MPC and neural network control. *IEEE Robot. Autom. Lett.* **5**, 3050–3057 (2020).
57. X. Zhang, M. Bujarbaruah, F. Borrelli, Near-optimal rapid MPC using neural networks: A primal-dual policy learning framework. *IEEE Trans. Control Syst. Technol.* **29**, 2102–2114 (2021).
58. C. I. Sprague, D. Izzo, P. Ögren, Learning dynamic-objective policies from a class of optimal trajectories, in *2020 59th IEEE Conference on Decision and Control (CDC)* (IEEE, 2020), pp. 597–602.
59. S. Origer, C. De Wagter, R. Ferede, G. C. de Croon, D. Izzo, Guidance & control networks for time-optimal quadcopter flight. arXiv:2305.02705 (2023).
60. R. Ferede, G. C. de Croon, C. De Wagter, D. Izzo, An adaptive control strategy for neural network based optimal quadcopter controllers. arXiv:2304.13460 (2023).
61. B. Gaudet, R. Furfaro, Robust spacecraft hovering near small bodies in environments with unknown dynamics using reinforcement learning, in *AIAA/AAS Astrodynamics Specialist Conference* (American Institute of Aeronautics and Astronautics, 2012), 10.2514/6.2012-5072.
62. S. Willis, D. Izzo, D. Hennes, Reinforcement learning for spacecraft maneuvering near small bodies, in *AAS/AIAA Space Flight Mechanics Meeting*, vol. 158 of *Advances in the Astronautical Sciences* (American Astronautical Society/American Institute of Aeronautics and Astronautics, 2016), pp. 1351–1368.
63. D. Miller, J. A. Englander, R. Linares, Interplanetary low-thrust design using proximal policy optimization, in *2019 AAS/AIAA Astrodynamics Specialist Conference*, no. GSFC-E-DAA-TN71225 in the NASA STI Repository (American Astronautical Society/American Institute of Aeronautics and Astronautics, 2019).
64. H. Yuan, D. Li, Deep reinforcement learning for rendezvous guidance with enhanced angles-only observability. *Aerosp. Sci. Technol.* **129**, 107812 (2022).
65. C. E. Oestreich, R. Linares, R. Gondhalekar, Autonomous six-degree-of-freedom spacecraft docking with rotating targets via reinforcement learning. *J. Aerosp. Inf. Syst.* **18**, 417–428 (2021).
66. R. Furfaro, R. Linares, Waypoint-based generalized ZEM/ZEV feedback guidance for planetary landing via a reinforcement learning approach, in *3rd International Academy of Astronautics Conference on Dynamics and Control of Space Systems, DyCoSS 2017* (Univelt Inc., 2017), pp. 401–416.
67. Y. Song, M. Steinweg, E. Kaufmann, D. Scaramuzza, Autonomous drone racing with deep reinforcement learning, in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2021), pp. 1205–1212.
68. A. T. Azar, A. Koubaa, N. Ali Mohamed, H. A. Ibrahim, Z. F. Ibrahim, M. Kazim, A. Ammar, B. Benjdira, A. M. Khamis, I. A. Hameed, G. Casalino, Drone deep reinforcement learning: A review. *Electronics* **10**, 999 (2021).
69. H. Holt, R. Armellin, N. Baresi, Y. Hashida, A. Turconi, A. Scorsoglio, R. Furfaro, Optimal Q-laws via reinforcement learning with guaranteed stability. *Acta Astronaut.* **187**, 511–528 (2021).
70. Y. Song, A. Romero, M. Müller, V. Koltun, D. Scaramuzza, Reaching the limit in autonomous racing: Optimal control versus reinforcement learning. *Sci. Robot.* **8**, eadg1462 (2023).
71. A. Kumar, J. Hong, A. Singh, S. Levine, When should we prefer offline reinforcement learning over behavioral cloning? arXiv:2204.05618 (2022).
72. S. Chen, K. Saulnier, N. Atanasov, D. D. Lee, V. Kumar, G. J. Pappas, M. Morari, Approximating explicit model predictive control using constrained neural networks, in *2018 Annual American Control Conference (ACC)* (IEEE, 2018), pp. 1520–1527.
73. N. Jakobi, P. Husbands, I. Harvey, Noise and the reality gap: The use of simulation in evolutionary robotics, in *Advances in Artificial Life: European Conference on Artificial Life 1995*, vol. 929 of *Lecture Notes in Computer* Science, F. Morán, A. Moreno, J. J. Merelo, P. Chacón, Eds. (Springer, 1995), pp. 704–720.
74. J. C. Zagal, J. Ruiz-del Solar, P. Vallejos, Back to reality: Crossing the reality gap in evolutionary robotics. *IFAC Proc. Vol.* **37**, 834–839 (2004).
75. S. Koos, J.-B. Mouret, S. Doncieux, The transferability approach: Crossing the reality gap in evolutionary robotics. *IEEE Trans. Evol. Comput.* **17**, 122 (2012).
76. K. Y. Scheper, G. C. de Croon, Abstraction as a mechanism to cross the reality gap in evolutionary robotics, in *From Animals to Animats 14: 14th International Conference on Simulation of Adaptive Behavior, SAB 2016, Aberystwyth, UK, August 23–26, 2016, Proceedings 14*, vol. 9825 of *Lecture Notes in Computer Science*, E. Tuci, A. Giagkos, M. Wilson, J. Hallam, Eds. (Springer, 2016), pp. 280–292.
77. S. Höfer, K. Bekris, A. Handa, J. C. Gamboa, M. Mozifian, F. Golemo, C. Atkeson, D. Fox, K. Goldberg, J. Leonard, C. K. Liu, J. Peters, S. Song, P. Welinder, M. White, Sim2Real in robotics and automation: Applications and challenges. *IEEE Trans. Autom. Sci. Eng.* **18**, 398–400 (2021).
78. S. Li, E. Öztürk, C. De Wagter, G. C. De Croon, D. Izzo, Aggressive online control of a quadrotor via deep network representations of optimality principles, in *2020 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2020), pp. 6282–6287.
79. D. Evans, M. Merri, OPS-SAT: A ESA nanosatellite for accelerating innovation in satellite control, in *SpaceOps 2014 Conference* (American Institute of Aeronautics and Astronautics, 2014), p. 1702.
80. E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, D. Scaramuzza, Champion-level drone racing using deep reinforcement learning. *Nature* **620**, 982–987 (2023).
81. G. de Croon, Drone-racing champions outpaced by AI. *Nature* **620**, 952–954 (2023).
82. R. Ferede, C. De Wagter, D. Izzo, G. C. de Croon, End-to-end reinforcement learning for time-optimal quadcopter flight. arXiv:2311.16948 (2023).
83. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics* (MLResearchPress, 2017), pp. 1273–1282.
84. J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, C. J. Tomlin, A general safety framework for learning-based control in uncertain robotic systems. *IEEE Trans. Automat. Contr.* **64**, 2737–2752 (2018).
85. F. Tambon, G. Laberge, L. An, A. Nikanjam, P. S. N. Mindom, Y. Pequignot, F. Khomh, G. Antoniol, E. Merlo, F. Laviolette, How to certify machine learning based safety-critical systems? A systematic literature review. *Autom. Softw. Eng.* **29**, 38 (2022).
86. L. Hewing, K. P. Wabersich, M. Menner, M. N. Zeilinger, Learning-based model predictive control: Toward safe learning in control. *Annu. Rev. Control Robot. Auton. Syst.* **3**, 269–296 (2020).
87. L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, A. P. Schoellig, Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annu. Rev. Control Robot. Auton. Syst.* **5**, 411–444 (2022).
88. A. Barredo Arrieta, N. Díaz-Rodríguez, J. del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **58**, 82–115 (2020).
89. C. Brandli, R. Berner, M. Yang, S.-C. Liu, T. Delbruck, A 240 × 180 130 dB 3 µs latency global shutter spatiotemporal vision sensor. *IEEE J. Solid-State Circuits* **49**, 2333–2341 (2014).
90. M. Davies, N. Srinivasa, T. H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C. K. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y. H. Weng, A. Wild, Y. Yang, H. Wang, Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro* **38**, 82–99 (2018).
91. T.-J. Chin, S. Bagchi, A. Eriksson, A. van Schaik, Star tracking using an event camera, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (IEEE, 2019), pp. 1646–1655.
92. O. Sikorski, D. Izzo, G. Meoni, Event-based spacecraft landing using time-to-contact, in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (IEEE, 2021), pp. 1941–1950.
93. S. McLeod, G. Meoni, D. Izzo, A. Mergy, D. Liu, Y. Latif, I. Reid, T.-J. Chin, Globally optimal event-based divergence estimation for ventral landing, in *Computer Vision – ECCV 2022*

Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part I (Springer, 2023), pp. 3–20.
94. L. Azzalini, E. Blazquez, A. Hadjiivanov, G. Meoni, D. Izzo, Generating a synthetic event-based vision dataset for navigation and landing, in *9th International Conference on Astrodynamics Tools and Techniques* (*ESA*, 2023).
95. M. G. McHarg, R. L. Balthazor, B. J. McReynolds, D. H. Howe, C. J. Maloney, D. O'Keefe, R. Bam, G. Wilson, P. Karki, A. Marcireau, G. Cohen, Falcon Neuro: An event-based sensor on the International Space Station. *Opt. Eng.* **61**, 085105 (2022).

Acknowledgments
Author contributions: D.I. conceived and directed the project, produced the first paper draft, and supervised the simulation and experimental work. E.B. contributed by reviewing satellite onboard control methods and supervising the space simulations. R.F. and S.O. performed the drone experiments. C.D.W. and G.C.H.E.d.C. contributed by reviewing drone-based onboard methods (neural based) and supervising the drone experiments. D.I., R.F., and S.O. wrote all the software to perform the dedicated simulations, train the neural networks, and solve the underlying optimal control problems. All authors contributed to writing the paper, assembling the figures, and discussing the overall content/structure. **Competing interests:** The authors declare that they have no competing interests.

Submitted 12 May 2023
Accepted 24 May 2024
Published 19 June 2024
10.1126/scirobotics.adi6421