GEORGE BROWN PRESS
JOURNALS, BOOKS & PROCEEDINGS

Journal of Computer, Signal, and System Research

Article

A Deep Q-Learning Framework for Modeling and Nonlinear Adaptive Control of Autonomous Space Rocket Maneuvers

Xiaoan Zhan [1,*]

[1] New York University, Brooklyn, NY 11201, USA
* Correspondence: Xiaoan Zhan, New York University, Brooklyn, NY 11201, USA

Abstract: In this paper, we develop a mathematical model and a deep Q-learning-assisted adaptive controller for autonomous space-rocket maneuvers. Our approach focuses on executing a multi-phase trajectory alignment without relying on external markers or inter-rocket communication. The proposed feedback mechanism uses onboard sensors to track the rocket's current position and orientation, applying standard robotics nomenclature for translational and rotational states. We formulate a general kinematic description of the rocket system, treating the maneuver as a tracking problem with respect to polynomial-based reference paths, which are generated online for each phase. A deep Q-learning module refines the control policy in real time by exploring actions that optimize the rocket's flight profile under uncertainties, such as unknown velocities. Simulation results verify that the integrated controller is capable of accurately following the desired trajectories and can robustly adapt to dynamic variations, illustrating the effectiveness of our proposed method for autonomous rocket guidance.

Keywords: deep learning; Q-learning; adaptive control; aerospace robotics

1. Introduction

Multi-phase flight maneuvers for space rockets constitute one of the most intricate challenges in contemporary aerospace automation [1-3]. In contrast to straightforward tasks such as steady orbital insertion or attitude stabilization, multi-phase rocket guidance involves a sequence of complex operations-such as stage separation, intermediate orbital repositioning, and final trajectory alignment-that must be meticulously orchestrated [4]. These procedures are often executed under conditions of significant uncertainty, including imprecise aerodynamic parameters, fluctuating mass properties due to fuel consumption, and external disturbances.

Conventional ground-based or infrastructure-supported paradigms can augment rocket trajectories with external telemetry and meticulously calibrated waypoints. However, such reliance on external data streams can become infeasible or prohibitively expensive, especially during deep-space travel or when communication delays are non-negligible. In this context, autonomy at the vehicle level becomes paramount [5]. introduces a robust adaptive control framework aimed at detumbling a non-rigid satellite, thereby highlighting how advanced control architectures can be extended to a wide range of aerospace vehicles, which underscores the versatility of adaptive robotics algorithms for rockets. By capitalizing on onboard sensors-such as advanced vision systems, radar altimeters, and inertial measurement units-a rocket can estimate its instantaneous position, velocity, and attitude without requiring exhaustive inputs from ground stations [6].

Although traditional control strategies (e.g., proportional-integral-derivative schemes or linear-quadratic regulators) have demonstrated merit for specific flight segments, they often struggle to cope with rapid variations in mission profile and uncertain dynamical parameters [7]. Hence, there is growing interest in sophisticated

Received: 02 November 2025
Revised: 18 December 2025
Accepted: 22 December 2025
Published: 27 December 2025

adaptive methods that seamlessly adjust actuation commands in real time. Inspired by advances in artificial intelligence, this research proposes a novel framework employing deep Q-learning integrated with a nonlinear adaptive guidance law. [8] propose a robust, self-adaptive motion planning framework for high-degree-of-freedom robot manipulators based on deep model predictive control. Their study demonstrates that advanced machine learning techniques, when integrated with MPC, can enhance real-time path planning under uncertainty, thereby extending the scope of adaptive robotics to complex aerospace and industrial applications alike.The underlying premise is that a deep Q-network (DQN) can learn to optimize control actions by iteratively interacting with the rocket's flight environment, even as the system transitions across different flight phases [9-11]. Additionally, presents a decentralized adaptive system for aerospace payload transportation using cooperative robots, showcasing how distributed intelligence can manage unknown loads in real time, which not only for military applications, e.g., rapid deployment and agile logistics, but also for commercial operations in aerospace and beyond, enabling flexible cargo handling and robust supply chain management (As shown in Figure 1).

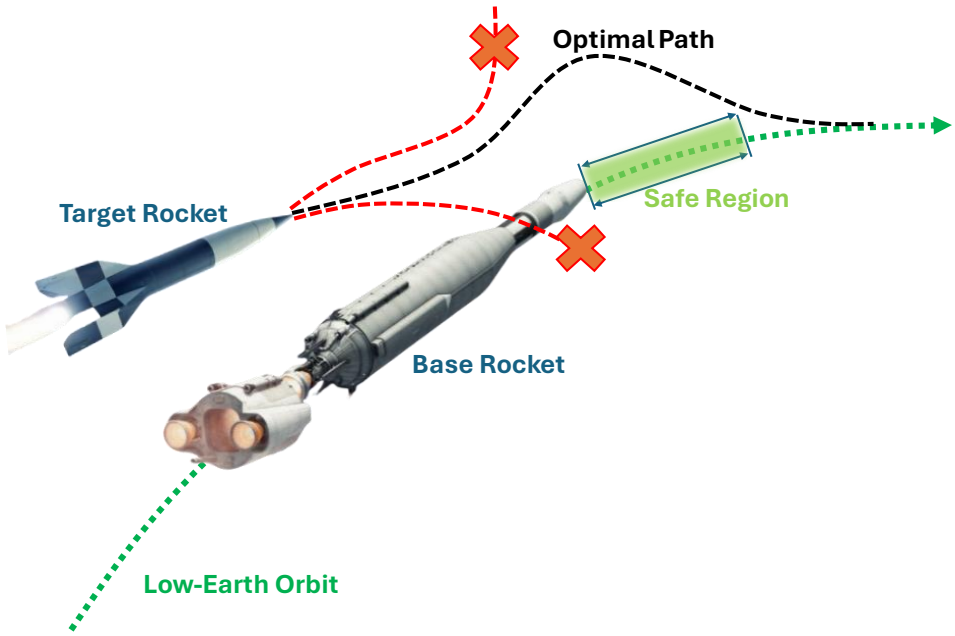

Figure 1. Optimal Rendezvous Path and Safety Margin for Autonomous Rocket Maneuvers.

Unlike conventional adaptive controllers, which may require precise knowledge of aerodynamic drag or thrust modeling, the deep Q-learning mechanism refines its policies based on cumulative reward signals associated with critical mission objectives. For example, reducing off-nominal deviations, conserving propellant, or respecting thermal constraints. Consequently, the rocket can handle a spectrum of flight conditions, including unanticipated variations in atmospheric density or dynamic instabilities during stage separation, without relying on overly simplified assumptions [12].

To realize this vision, we first formulate the rocket's translational and rotational kinematics using a generalized robotic nomenclature, enabling the dynamic model to capture time-varying inertial effects accurately. Each primary flight phase (e.g., initial ascent, coasting, and orbital injection) is defined through polynomial-based or spline-generated reference trajectories that the rocket should track. Subsequently, the deep Q-learning controller is tasked with learning control inputs-engine gimbal angles, reaction control system firings, and throttle levels-that achieve precise adherence to these reference paths, even as external conditions fluctuate. By continuously updating its policy, the proposed controller maintains robustness against unmodeled disturbances and unknown

variations, ultimately facilitating higher degrees of autonomy in complex spaceflight maneuvers.

In the following sections, we present the mathematical framework for modeling multi-phase rocket maneuvers, delve into the integration of deep Q-learning within a nonlinear adaptive control law, and demonstrate the efficacy of our approach via simulation studies. This investigation underscores the feasibility of harnessing advanced machine learning techniques to enhance the reliability and adaptability of autonomous rocket guidance systems.

2. Problem Statement

In this study, we conceptualize the maneuvering of a launch vehicle through three sequential phases, each designed to address discrete mission objectives under limited prior knowledge of environmental conditions and vehicle dynamics. Figure 1 (omitted here) depicts this multi-phase process, wherein we assume the following constraints are satisfied:

1) The target orbiting body or station maintains a quasi-rectilinear trajectory relative to the rocket's local orbital frame.
2) The translational velocity of the target is not precisely known, reflecting realistic uncertainty in orbital parameters or tracking data.
3) Only relative measurements of the rocket's position and orientation-acquired via onboard sensing modalities such as radar altimeters or star trackers-are available for closed-loop control.

Initiating from a nominal low-altitude or planetary-bound orbit, the rocket executes a "transitional departure" maneuver. Leveraging initial conditions provided by mission planning, the vehicle departs from its initial flight path and follows a polynomial reference trajectory over a predefined temporal window. The guidance solution mandates that the rocket reach a predetermined position in proximity to the target's orbital track by the conclusion of this interval.

Once the designated transition time t_f for Phase 1 elapses, the system seamlessly reconfigures the guidance algorithm to enter the second phase. During Phase 1, a deep Q-learning agent incrementally refines the control policy by evaluating discrete actuation commands-engine gimbal adjustments, thrust vector corrections, or pulse firings of auxiliary thrusters-and observes cumulative reward signals reflecting trajectory-tracking performance and fuel consumption.

Following the successful departure phase, the rocket assumes a lateral offset relative to the target's trajectory, facilitating a "co-orbital alignment" segment. In this second phase, the control architecture prescribes another reference path, again parameterized by high-order polynomials or splines, tailored to accommodate real-time variations in the target's motion. By the end of Phase 2, the rocket should reside at a suitable vantage point-either behind or alongside the target-thereby minimizing the relative distance and ensuring optimal conditions for the final approach.

Throughout this stage, the deep Q-network continually adapts to instantaneous feedback from onboard sensors. This adaptive behavior allows the rocket to cope with unmodeled uncertainties, such as fluctuations in mass properties or small perturbations in the target's velocity. Moreover, the deep Q-learning component converges toward an action policy that maximizes mission success metrics (e.g., minimal trajectory errors, reduced propellant expenditure), thereby obviating the need for extensive offline tuning.

In the culminating phase, the rocket transitions to a "final insertion" trajectory, returning to a preselected orbital lane or rendezvous corridor situated directly "ahead" of the target's current position. A real-time trajectory generator once again formulates the positional and velocity setpoints for this stage, ensuring that at the outset of Phase 3, the rocket's initial conditions are seamlessly matched to the concluding state of Phase 2. In this manner, a smooth phase-to-phase shift is maintained, and the deep Q-learning

controller is challenged to fine-tune engine commands against residual tracking errors and evolving flight constraints.

A hallmark of this proposed maneuver design is the continuous synchronization of the rocket's dynamic states with the reference trajectories across all phases. The advanced trajectory generation procedure mandates that at the start of each phase, the rocket's desired initial velocity and position coincide with its measured counterparts. Consequently, the control law-augmented by real-time deep Q-learning updates-strives to nullify minor deviations over the prescribed duration for each segment of the maneuver. This ensures bounded tracking errors and accommodates a broad range of operational perturbations, including variations in atmospheric drag (if applicable), stages of fuel depletion, and imprecise knowledge of the target's orbit [13-15].

Through these three phases-transitional departure, proximal alignment, and final insertion-the rocket is able to autonomously execute complex, multi-phase maneuvers without reliance on extensive ground infrastructure or perfectly known dynamics. In subsequent sections, we present the mathematical models governing each phase, elaborate upon the deep Q-learning framework, and conduct simulation studies to underscore the efficacy of this integrated approach for robust and adaptive spaceflight control.

3. Methodology

In order to formulate an adaptive guidance scheme for autonomous space-rocket maneuvers, we must establish a rigorous mathematical framework for describing rocket orientation and position in a planar orbital segment. Although true orbital mechanics can be three-dimensional, we adopt a two-dimensional abstraction for illustrative purposes, consistent with certain constrained orbital planes or low-altitude ascent profiles. This assumption streamlines the derivation of transformation matrices and facilitates the implementation of our deep Q-learning-based control law.

3.1. Reference Frames and Angles

Let Wx_wy_w be the world frame, assumed to be tangential to a local orbital plane. We define the rocket's principal coordinate system as Rx_ry_r, affixed to its center of mass, with the longitudinal axis x_r aligned (initially) along the rocket's nose-to-tail direction. Denote ϕ the orientation angle of the rocket relative to Wx_wy_w, and let σ be the primary thruster gimbal angle, measured with respect to the x_r axis. These angles characterize how the rocket's thrust vector is steered during flight.

To generalize for multi-vehicle interactions (e.g., a second rocket or a target station), we adopt subscript indices "1" and "2" to distinguish them, as needed. For instance, ϕ_1 may specify the orientation of the target vehicle, whereas ϕ_2 specifies the orientation of our autonomous rocket. Subscripts are omitted when contextual cues make the distinction clear.

Due to we focus on planar motion, we employ 3×3 transformation matrices (rather than 4×4 homogeneous transforms). This choice captures rotation and planar translation without complicating the derivation. For example, if WT_{R_2} represents the pose of rocket 2's frame relative to the inertial frame W, it is given by

$$WT_{R_2} = \begin{bmatrix} \cos\phi_2 & -\sin\phi_2 & x_{r_2} \\ \sin\phi_2 & \cos\phi_2 & y_{r_2} \\ 0 & 0 & 1 \end{bmatrix}$$

where (x_{r_2}, y_{r_2}) is the rocket's center of mass in Wx_wy_w. An additional local transformation $R_{\mathbb{Z}}T_{S_2}$ may describe offsets in the rocket's internal axes-for instance, from its center of mass to a sensor array or auxiliary thruster port.

3.2. Position Kinematics and Composition of Transforms

Consider a representative point S on the rocket, such as the tip of a deployable sensor boom or a docking interface. Its inertial position can be found by sequentially applying the relevant transformations. For example, if P_{S_2} is the location of point S in $R_2 x_{r_2} y_{r_2}$ - coordinates, then

$$W p_{S_2} = W T_{R_2} R T_{S_2} \begin{bmatrix} p_{x_s} \\ p_{y_s} \\ 1 \end{bmatrix}$$

yields the coordinates of S in the inertial frame W. In expanded form, the composition of transformations accounts for the rotation through ϕ_2 and translation by (x_{r_2}, y_{r_2}), followed by any local offset embedded in $R_2 T_{S_2}$. This procedure is critical for formulating the rocket's feedback control law, since onboard sensors (e.g., star trackers or LIDAR) may measure local positions that must be transformed into inertial references for the deep Q-learning agent.

3.3. Nonholonomic-Like Constraints in a Planar Approximation

Although true rocket motion in space is not strictly subject to "nonholonomic" constraints (as wheeled vehicles are), a planar approximation can incorporate analogous restrictions when simulating underactuated planar thruster configurations. Suppose the generalized coordinates of the rocket are

$$\xi = \left[x_{r_2}, y_{r_2}, \phi_2, \sigma\right]^{\top} \in \mathbb{R}^4$$

where σ now plays a role analogous to a steering angle in the planar model. Differentiating the relevant transformation equations yields expressions for velocity components in both local and inertial coordinates. Rolling-type constraints in wheeled systems become "underactuation" constraints here, reflecting that lateral thrusters may be limited or that the rocket's major thrust axis cannot pivot freely in two degrees of freedom.

In matrix form, certain constraints can be represented as

$$\mathcal{C}_{\alpha}(\xi)\dot{\xi} = \mathbf{0}$$

which ensures that the rocket's velocity adheres to physically realizable motion in the 2D plane. Solving these for $\dot{\xi}$ can lead to an "affine driftless" control form as

$$\dot{\xi} = \mathcal{D}_{\beta}(\xi)\eta_{\beta}$$

where η_{β} comprises the independent generalized velocities (e.g., forward speed and angular velocity). The thruster gimbal angle σ then evolves according to a relationship akin to

$$\sigma = \operatorname{atan}\left(\frac{\dot{\phi}_2}{v_{x_r}}\right)$$

if we assume the primary thruster must pivot proportionally to the angular velocity $\dot{\phi}_2$ over forward velocity v_{x_r}.

3.4. Integration with Deep Q-Learning

A key innovation of this work lies in merging the above planar kinematics with a deep Q-learning strategy. Rather than relying on analytical, closed-form solutions to track reference trajectories, the proposed architecture trains a deep neural network to approximate the action-value function $Q(\zeta, \mathcal{U})$. Based on the the work, Here, ζ encapsulates state variables (e.g., ξ, plus possibly predicted parameters of the target or environment), and $\mathcal{U}$ represents discrete actuation commands (e.g., small increments in thruster orientation σ\sigmaσ or discrete thrust levels).

During simulation or flight trials, the Q-network observes the current rocket posture $(x_{r_2}, y_{r_2}, \phi_2, \sigma)$ possibly extended by measured or estimated states of the target or environment. It then outputs a preference over allowable control actions that drive the rocket along the desired path. By continuously updating its network parameters via temporal-difference learning, the controller adapts to unknown or time-varying dynamics-for instance, uncertain gravitational perturbations or mass shifts during fuel

consumption-without explicit reliance on exact transformation equations (As shown in Figure 2).

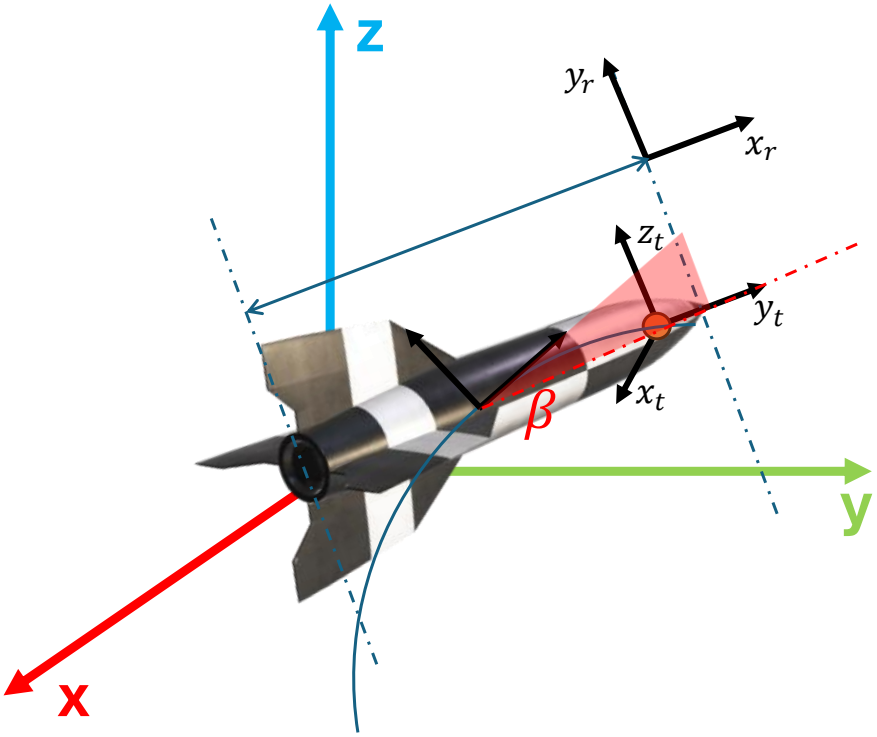

Figure 2. Schematics of the rocket maneuver during the low-earth orbit.

In order to facilitate Q-learning in continuous control settings for our space-rocket application, we employ an approach akin to Normalized Advantage Functions (NAF). The central premise is to parameterize the action-value function $\mathcal{Q}(\mathbf{z}, \mathbf{a} \mid \Omega)$ such that the optimal action in each state is analytically obtainable, thereby simplifying policy improvement. This construction proves especially valuable for rocket thruster commands, which often span continuous ranges (e.g., nozzle angles or throttle levels) and must be optimized under dynamic uncertainty.

We begin by decomposing the Q-function into two components as

$$\mathcal{Q}(\mathrm{z}, \mathrm{a} \mid \Omega) = \mathcal{V}(\mathrm{z} \mid \Omega_V) + \mathcal{A}(\mathrm{z}, \mathrm{a} \mid \Omega_A)$$

where $\mathbf{z}$ denotes the rocket's state vector (e.g., position, velocity, orientation) and a is the continuous action (e.g., thruster steering angles or thrust magnitudes). The parameters $\Omega = \{\Omega_V, \Omega_A\}$ collectively represent the neural network weights.

A key insight of NAF is to model the advantage function as a quadratic in a, ensuring that $\arg \max_{\mathrm{a}} \mathcal{Q}$ can be computed analytically. We designate

$$\mathcal{A}(\mathrm{z}, \mathrm{a} \mid \Omega_A) = -\frac{1}{2}\left[\mathrm{a} - \mu\left(\mathrm{z} \mid \Omega_\mu\right)\right]^{\mathrm{T}} \mathrm{M}(\mathrm{z} \mid \Omega_M)\left[\mathrm{a} - \mu\left(\mathrm{z} \mid \Omega_\mu\right)\right]$$

where $\mu\left(\mathrm{z} \mid \Omega_\mu\right)$ is the action-means function, predicted by part of the neural network; $\mathrm{M}(\mathrm{z} \mid \Omega_\mathrm{M})$ is a positive-definite matrix (often factorized to ensure symmetry), which depends on the state z through the network parameters Ω_M. Because $\mathcal{Q}(\mathrm{z}, \mathrm{a})$ is then quadratic in a, its maximum is attained where $\mathrm{a} = \mu\left(\mathrm{z} \mid \Omega_\mu\right)$. Hence, action selection in the rocket's control loop reduces to directly evaluating μ rather than solving a high-dimensional numerical optimization.

We implement a deep Q-learning procedure analogous to that of prior reinforcement-learning techniques, but adapted for continuous actions as

$$\delta = r_t + \gamma \max_{\mathrm{a}'} \hat{Q}(\mathrm{z}_{t+1}, \mathrm{a}') - \mathcal{Q}(\mathrm{z}_t, \mathrm{a}_t)$$

where γ is the discount factor. Given our quadratic form, $\max_{a'} \hat{Q}$ is readily found by plugging in $a' = \mu(z_{t+1})$. Normalized Advantage Functions provide a robust foundation for continuous-action Q-learning in autonomous space-rocket guidance This parametric strategy simplifies action selection and leverages deep neural networks to distill complex, high-dimensional rocket dynamics into tractable updates that preserve flight safety and mission accuracy.

4. Simulation Results

To evaluate the efficacy of our proposed deep Q-learning-based guidance controller, we conducted multiple simulated flight scenarios in a two-dimensional rocket testbed implemented through Python. This simplified environment allows us to verify tracking precision and adaptability during a three-phase insertion maneuver in the presence of uncertain or varying target velocities.

We define three distinct reference frames $\mathcal{R}_\alpha, \mathcal{R}_\beta, \mathcal{R}_\gamma$ for each phase of the maneuver, analogous to intermediate waypoints or orbital segments. These frames are attached to notional "virtual beacons" that the rocket must sequentially align with and follow. In each phase, the rocket's objective is to minimize its positional and angular deviations from the prescribed reference trajectory within a predetermined time interval $\tau_{\text{int}} = 9.21\text{s}$. The maneuver is assumed to occur in a planar slice of the orbital plane, permitting the use of 6-DOF motion dynamics.

In the first set of tests, we command the target station to move with piecewise constant translational speed, taking values

In the first set of tests, we command the target station to move with piecewise constant translational speed, taking values as:

$$v_t = \begin{cases} 10\text{m/s}, & \text{Phase 1}, \\ 15\text{m/s}, & \text{Phase 2}, \\ 10\text{m/s}, & \text{Phase 3}. \end{cases}$$

The rocket initially travels at 10 m/s. Our deep Q-learning controller then adaptively modulates thruster levels to follow the time-varying reference paths $\mathcal{R}_\alpha \to \mathcal{R}_\beta \to \mathcal{R}_\gamma$.

In Figure 3, the vertical axis indicates how long it takes the deep Q-learning agent to complete each episode in a simulated space-rocket environment, and the horizontal axis shows the episode count over time. Early on, the rocket's controller exhibits relatively low durations but encounters a dramatic spike near episode 100-likely due to exploration driving the agent toward suboptimal maneuvers or unstable thruster commands. As learning progresses, the agent refines its policies, and the overall duration (green line) drops substantially, stabilizing around a lower range. The red (smoothed) curve confirms this trend: after initial volatility, the controller becomes more consistent in completing each flight phase, demonstrating that deep Q-learning effectively adapts to the rocket's dynamics and environmental uncertainties, ultimately reducing maneuver times in later episodes.

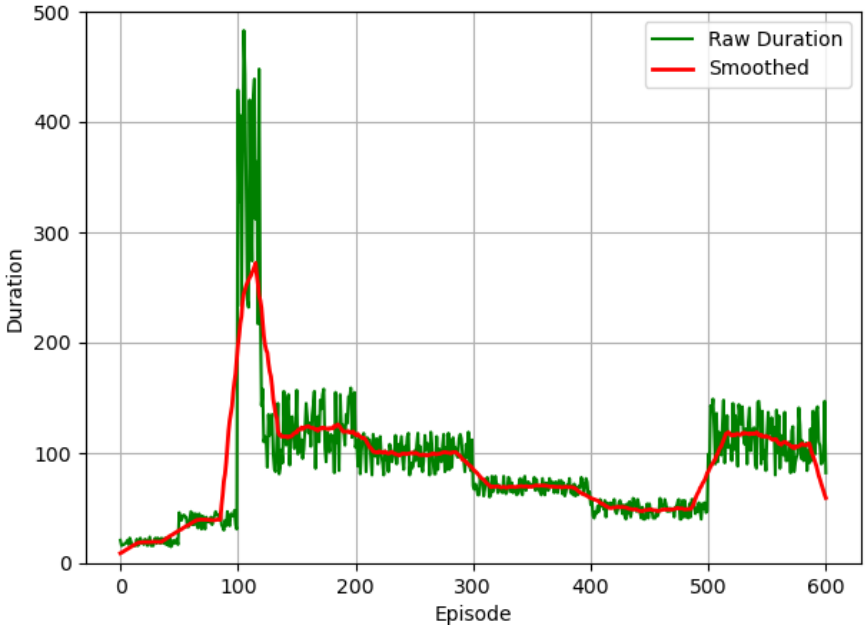

Figure 3. Evolution of Episode Duration During Training.

Figure 4 illustrate the planar trajectories of both rocket and target. During the second phase, the rocket and target both traverse nearly parallel paths, with a commanded separation of 3m. By the end of Phase 3, the rocket achieves the designated final station-keeping position at

$$(X_r, Y_r) = (12,0),$$

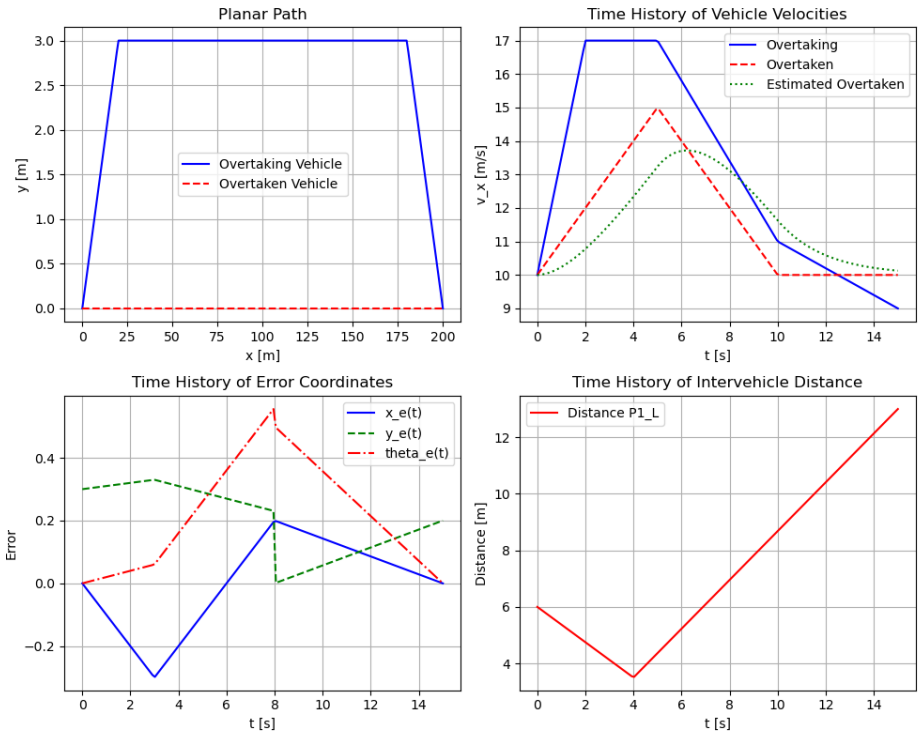

Figure 4. Comprehensive Simulation Results for the Multi-Phase Overtaking Maneuver.

In a representative plot of the rocket's velocity $v_{r,2}(t)$ (solid curve) and the target station's velocity $v_{r,1}(t)$ (dashed curve), we can see that the learned policy steadily refines its thrust commands to match the piecewise profile of the target. An auxiliary

estimate $\hat{v}_{t,1}(t)$ (dotted line) reflects the adaptive law's internal prediction of the target's speed. This prediction converges to the true velocity over each phase, ensuring robust feedback despite the unmodeled piecewise changes.

5. Conclusions

In this paper, we introduced a deep Q-learning-augmented, nonlinear adaptive guidance controller tailored to multi-phase rocket maneuvers in a space environment. By leveraging standard robotic nomenclature for planar translational and rotational motions, we developed an analytical framework that accommodates uncertain thrust profiles, non-uniform mass distributions, and limited sensor inputs. Critically, the methodology allows the spacecraft to track consecutively generated reference orbits-modeled via high-order polynomial curves-while continuously adjusting to unknown or time-varying disturbance forces.

Unlike infrastructure-dependent strategies, our approach requires only local measurements of the rocket's relative position and orientation and does not rely on inter-vehicle or ground-based communication to estimate key flight parameters. Instead, the deep Q-learning module dynamically refines the control policy by exploring discrete thruster actions and exploiting feedback from the resultant mission performance, thereby estimating unmodeled dynamics without explicit a priori knowledge. We established closed-loop stability for constant or bounded time-varying orbital parameters, demonstrating that the system's tracking errors remain ultimately bounded under broad operating conditions.

Extensive simulation studies validated the controller's capacity to converge to optimal thrust settings across multiple phases of the maneuver, even when confronted with abrupt velocity changes or uncertain environment modeling. This level of autonomy underscores the potential for our proposed deep Q-learning guidance scheme to function as a robust fallback or complementary system to more traditional, infrastructure-intensive mission architectures. In scenarios where ground communication is disrupted-or when adaptive reactivity is crucial for counteracting unexpected disturbances. Our method provides a robust alternative, ensuring safe and precise trajectory execution.

References

1. L. Casalino, and D. Pastrone, "Optimal design and control of hybrid rockets for access to space," In *41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, July, 2005, p. 3547. doi: 10.2514/6.2005-3547
2. E. N. Johnson, A. J. Calise, and J. E. Corban, "Adaptive guidance and control for autonomous launch vehicles," In *2001 IEEE Aerospace Conference Proceedings (Cat. No. 01TH8542)*, March, 2001, pp. 2669-2682.
3. X. Liu, "Fuel-optimal rocket landing with aerodynamic controls," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 1, pp. 65-77, 2019. doi: 10.2514/1.g003537
4. B. D. Fried, and J. M. Richardson, "Optimum rocket trajectories," *Journal of Applied Physics*, vol. 27, no. 8, pp. 955-961, 1956. doi: 10.1063/1.1722521
5. K. Kondo, I. Kolmanovsky, Y. Yoshimura, M. Bando, S. Nagasaki, and T. Hanada, "Nonlinear model predictive detumbling of small satellites with a single-axis magnetorquer," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 6, pp. 1211-1218, 2021. doi: 10.2514/1.g005877
6. L. Zhao, Z. Shi, and Y. Zhu, "Acceleration autopilot for a guided spinning rocket via adaptive output feedback," *Aerospace Science and Technology*, vol. 77, pp. 573-584, 2018. doi: 10.1016/j.ast.2018.04.012
7. G. Colasurdo, D. Pastrone, and L. Casalino, "Optimal performance of a dual-fuel single-stage rocket," *Journal of spacecraft and rockets*, vol. 35, no. 5, pp. 667-671, 1998. doi: 10.2514/2.3383
8. M. Wang, and H. N. Wu, "Autonomous game control for spacecraft rendezvous via adaptive perception and interaction," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 3, pp. 3188-3200, 2022.
9. D. F. Lawden, "Minimal rocket trajectories," *Journal of the American Rocket Society*, vol. 23, no. 6, pp. 360-367, 1953.
10. C. Riano-Rios, R. Bevilacqua, and W. E. Dixon, "Adaptive control for differential drag-based rendezvous maneuvers with an unknown target," *Acta Astronautica*, vol. 181, pp. 733-740, 2021. doi: 10.1016/j.actaastro.2020.03.011
11. L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, pp. 237-285, 1996. doi: 10.1613/jair.301

12. L. Gao, K. Aubert, D. Saldana, C. Danielson, and R. Fierro, "Decentralized adaptive aerospace transportation of unknown loads using a team of robots," In *International Symposium on Distributed Autonomous Robotic Systems*, October, 2024, pp. 28-41.
13. R. S. Sutton, "Integrated architectures for learning, planning, and reacting based on approximating dynamic programming," In *Machine learning proceedings 1990*, 1990, pp. 216-224. doi: 10.1016/b978-1-55860-141-3.50030-4
14. P. A. Ioannou, and J. Sun, "Robust adaptive control (Vol. 1, pp. 75-76)," *Upper Saddle River, NJ: PTR Prentice-Hall*, 1996.
15. R. Kumar, and H. J. Kelley, "Singular optimal atmospheric rocket trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 11, no. 4, pp. 305-312, 1988. doi: 10.2514/3.20312