Available online at www.sciencedirect.com

ScienceDirect

Advances in Space Research 67 (2021) 3667–3682

ADVANCES IN SPACE RESEARCH (a COSPAR publication)

www.elsevier.com/locate/asr

Propulsionless planar phasing of multiple satellites using deep reinforcement learning

Brenton Smith *, Rasit Abay, Joshua Abbey, Sudantha Balage, Melrose Brown, Russell Boyce

School of Engineering and Information Technology, UNSW Canberra, Northcott Drive, Campbell, ACT, Australia

Received 4 November 2019; received in revised form 10 September 2020; accepted 13 September 2020
Available online 1 October 2020

Abstract

This work creates a framework for solving highly non-linear satellite formation control problems by using model-free policy optimisation deep reinforcement learning (DRL) methods. This work considers, believed to be for the first time, DRL methods, such as advantage actor-critic method (A2C) and proximal policy optimisation (PPO), to solve the example satellite formation problem of propellantless planar phasing of multiple satellites. Three degree-of-freedom simulations, including a novel surrogate propagation model, are used to train the deep reinforcement learning agents. During training, the agents actuated their motion through cross-sectional area changes which altered the environmental accelerations acting on them. The DRL framework designed in this work successfully coordinated three spacecraft to achieve a propellantless planar phasing manoeuvre. This work has created a DRL framework that can be used to solve complex satellite formation flying problems, such as planar phasing of multiple satellites and in doing so provides key insights into achieving optimal and robust formation control using reinforcement learning.

Keywords: Reinforcement learning; Formation control

1. Introduction

The growth in miniaturised satellite utilisation has been and continues to be driven by reductions in cost. The sustainability of the business models of emerging space entities, such as Planet, rely on the cost-effectiveness of CubeSats to deliver cutting-edge capabilities. An example of cutting-edge technology is intersatellite quantum key distribution for secure communications (Naughton et al., 2019). The realisation of these capabilities often necessitates the use of miniaturised satellite formations or constellations. For example, Planet[1] operate dozens of CubeSats in multiple coplanar constellations to attain frequently updated Earth imagery. Additionally, more than half of the formation flying spacecraft that were, or will be, launched between the year 2000 and 2025 have a mass of less than 100 kg (Di Mauro et al., 2018).

Orbital perturbations make satellite formations unstable. Therefore, a mechanism is required to actively maintain the relative position of formation flying spacecraft. However, a large proportion of CubeSat platforms exclude propulsion systems due to the added cost and complexity that is not compatible with the vertically integrated and low-cost agile development philosophy of modern CubeSat operators. Environmental accelerations, such as aerodynamic drag (Leonard et al., 1989), aerodynamic lift (Smith et al., 2017b; Smith, 2019), ionospheric drag (Smith et al., 2019), coulomb forces (Seubert and Schaub, 2009), the Lorentz force (Huang et al., 2015), and solar radiation pressure (Williams and Wang, 2002; Smirnov et al., 2007; Hou et al., 2016), have been proposed methods for maintaining formations of miniaturised spacecraft.

* Corresponding author.
E-mail address: brenton.smith@unsw.edu.au (B. Smith).
[1] https://www.planet.com/.

https://doi.org/10.1016/j.asr.2020.09.025
0273-1177/

Planet utilise the difference in the drag accelerations (differential drag) to phase multiple spacecraft within a low Earth orbit (LEO) coplanar constellation using a rules-based formation control methodology (Foster et al., 2017). Here, pre-defined rules, computed on the ground, command transitions in operational modes of the satellites (Foster et al., 2017). Rules-based control methodology to phase satellites within a LEO constellation using differential drag is common. For example, it is the approach used to control the phase of satellites within NASA's CYGNSS constellation (Bussy-virat et al., 2019). Generally, rule-based autonomy involves a human-in-the-loop, and is not robust to environmental changes, or optimised for competing operational priorities (Harris et al., 2019). For example, simplifications in initial rule-based propellantless formation control algorithms assumed the realistically varying atmospheric density was constant (Leonard et al., 1989; Bevilacqua and Romano, 2008; Miele and Venkataraman, 1984; Horsley et al., 2013; Smith et al., 2017b) which was shown to produce residual errors after formation manoeuvring (Kumar and Ng, 2008; Lambert et al., 2012). Classical adaptive control techniques have since been developed to account for atmospheric density uncertainties, but these rule-based approaches do not account for mission priorities other than manoeuvring (Pérez and Bevilacqua, 2013; Pérez and Bevilacqua, 2014; Harris and Açikmese, 2014; Dell'Elce and Kerschen, 2015; Mazal et al., 2016). Further, many of these rules-based propellantless formation control algorithms require cooperative actions to instantiate relative motion using environmental accelerations. For example, in order to create differential drag to control a formation of two satellites, both spacecraft have to control their attitude to mismatch their ballistic coefficients. Therefore, many of these algorithms are designed to control two-satellites at a time and are not optimally suited to controlling multi-agent (more than two satellite) systems because of the non-linear increase in complexity that arises from coordinating differential drag across multiple pairs of agents at the same time. This is despite many current and future formations of miniaturised satellites being comprised of up to hundreds of spacecraft (Foster et al., 2017) which will demand formation control cognition that scales beyond the abilities of human operators without elements of automation. Alternative to rule-based autonomy, optimisation-based autonomy is achieved by setting up competing operational priorities as a constrained optimisation problem which is solved using considerable computing power; adding cost and precluding the implementation of these techniques on-board CubeSats (Harris et al., 2019).

Deep reinforcement learning (DRL) has been proposed to achieve satellite autonomous decision making that is more robust to uncertainties while reducing operational costs, and maximising mission returns (Harris et al., 2019). Deep reinforcement learning is a sub-discipline of machine learning where agents independently learn to complete a task, dictated by a reward function, based on iterative experience within a process called training. A DRL agent learns to map actions with resulting consequences without needing to know, or model, the underlying physical phenomenon causing the consequences. For example, DRL agents have been considered to map actuator thrusting to the resulting spacecraft motion in order to control powered descent onto celestial bodies (Gaudet and Furfaro, 2012; Furfaro et al., 2018b; Furfaro et al., 2018a; Gaudet et al., 2020; Gaudet and Furfaro, 2014; Willis et al., 2016). Also, DRL has been used in applications of rendezvous (Broida and Linares, 2019). Once trained, DRL agents are computationally efficient enough to implement onto miniaturised platforms. For example, agents have been successfully implemented onto small quadrotor unmanned aerial vehicles to avoid collisions (Zhang et al., 2016). Therefore, it may be feasible to integrate a DRL agent on-board a CubeSat, with limited computing power, to achieve more autonomous satellite formation control and reduce associated operational costs by eliminating computationally intense and latent rule-based, or optimisation based autonomy. Further, the ability of DRL to learn from experience better equips it to deal with uncertainties and can be used to optimise spacecraft behaviour to deal with competing mission priorities (Harris et al., 2019). Further, the ability for DRL to solve highly complex problems may allow it to be used to control highly constrained formations of multiple spacecraft.

Given the aforementioned benefits of DRL for satellite autonomy, and how common the phasing of miniaturised satellites using differential drag is (Bussy-virat et al., 2019; Foster et al., 2015; Foster et al., 2017), this work investigates the high-level feasibility of DRL to optimally rephase the relative argument of latitude of multiple coplanar miniaturised satellites (multi-agents) using propellantless means in LEO. By applying DRL to rephase multiple satellites, this work will seek to create a framework for data driven discovery of unforeseen ways to optimally conduct long-term and complex satellite formation flying problems. The agents will be trained and tested under uncertain and changing environmental conditions by using propagation methods such as numerical propagators and surrogate models. The work is laid out as follows: Section 2 will outline the methodology used to perform this first effort assessment of the feasibility of the DRL method of rephasing the multi-agent system of spacecraft. Section 3 will provide simulation results of multi-satellite planar rephasing using the DRL method, as well as, discuss the implications, practicalities, and performance of the methodology and results.

2. Methodology

2.1. The formation flying manoeuvre

The aim of this work is to gain insight into the practicability of DRL to train an agent to provide data driven optimal solutions to complex satellite formation control problems. The complex satellite formation control problem

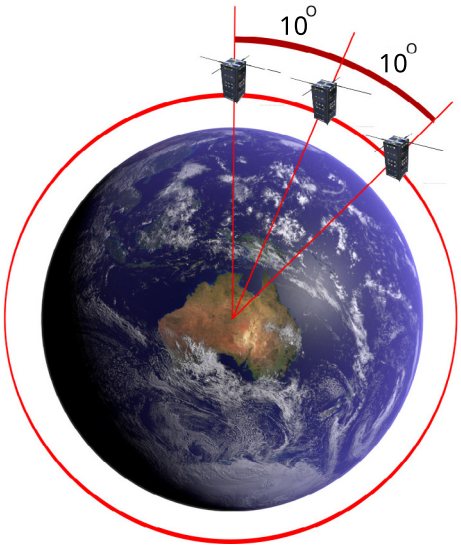

Fig. 1. The desired formation configuration.

considered in this work is the propellantless phasing of multiple spacecraft operating in coplanar orbits.

Planar phasing of multiple spacecraft is a common manoeuvre that occurs to establish or change the relative angle between two spacecraft operating within the same orbital plane. Typically, the desired state of the spacecraft is to have identical orbit periods such that the spacecraft follow each other along approximately the same trajectory. This way the relative phase angle between the spacecraft is maintained as can be seen in Fig. 1. Propellantless planar phasing involves establishing and maintaining the configuration described above without creating acceleration by ejecting propellant. Instead accelerations are generated using interactions with the space environment such as that arising from aerodynamic drag.

2.1.1. Initial conditions

For the propellantless planar phasing problem studied in this work, the initial states were assumed to mimic a close-proximity satellite formation as would be the case after initial deployment from a launch vehicle. A formation comprised of three spacecraft was considered. Of the initial orbital elements of the three spacecraft, the semi-major axis, inclination, eccentricity, right ascension of the ascending node, and argument of perigee were kept constant for all iterations and are shown in Table 1.

A reference initial true anomaly was generated randomly between 0° and 360° based on a uniform distribution. A random value between 0° and 0.5°, based on a uniform distribution, was added to the reference initial true anomaly such that the initial true anomaly was different for each spacecraft. For the purpose of this work, the phase angle is defined as the mean argument of latitude. The mean argument of latitude is the sum of the mean argument of perigee and the mean true anomaly. The relative phase angle is the difference in mean argument of latitude between two spacecraft.

Table 1
Initial conditions controlled during simulations.

Variable
Semi-Major Axis
Eccentricity
Inclination
Right ascension of the ascending node
Argument of perigee

2.1.2. Desired final conditions

The desired final formation configuration was chosen to be an equispaced formation where the relative mean argument of latitude was 10° between each pair of satellites as is shown in Fig. 1.

The performance of the DRL agent to conduct the propellantless planar phasing of multiple spacecraft will be based on:

1. The accuracy of the propellantless planar phasing manoeuvre based on the ability for the agents to achieve a mean relative argument of latitude within one degree of the desired value and a relative semi-major axis less than 1 km. These conditions are the convergence criteria that is used for the DRL networks within this work.
2. The time taken to achieve the change in relative phasing manoeuvres as defined by the first instance that the success criteria described above is reached; and
3. The speed in which the agents can be trained to find a solution.

The success criteria, determined by the tolerance in the mean relative argument of latitude and mean relative semi-major axis error, considered in this work, is very lenient. This was a deliberate choice in order to provide the DRL agent a problem that it could easily solve, and thus, allowing insight into the general nature of how DRL can be used to solve the propellant planar phasing of multiple satellites. If the manoeuvre tolerances are too stringent, the agents developed within this first effort analysis will struggle to solve the problem leading to very little insight into the behaviour of the agents in undertaking propellantless planar phasing.

2.2. Conducting propellantless planar phasing

This first analysis into the applicability of DRL to conduct propellantless planar phasing of multiple satellites will train the DRL agent within a simulation environment where the dynamics of the spacecraft are modelled by orbit propagations. A simulation environment has been chosen because during initial training of an untrained DRL agent, the naive actions taken by the agent may lead to a collision of spacecraft, or other counter-productive behaviour.

Additionally, DRL agents can take thousands of episodes to converge upon an adequate solution. A phasing manoeuvre performed in space can take days or weeks to complete (Foster et al., 2017). Therefore, the time taken to perform the thousands of phasing manoeuvres in space required to train the DRL agent would likely exceed the mission life of a formation in LEO. By modelling the dynamics with propagations, the learning rate, and therefore, time taken to converge upon a solution is fast enough to allow frequent test cases to be executed. This leads to greater insight into the performance of DRL to conduct planar phasing of multiple spacecraft. Orbit propagators that account for perturbations such as aerodynamics, solar radiation pressure, third-body gravity and aspherical Earth gravity are commonly used to model the dynamics of space objects such as orbit debris (Serra et al., 2018). Considering the goal of this first effort analysis is to gain general insight into the practicability of DRL agents to solve complex satellite formation control problems such as propellantless planar phasing, it is valid to substitute actual spacecraft motion with orbit propagations.

2.3. Dynamics model

This paper models the dynamics of propellantless planar phasing manoeuvres using orbit propagations. In order for this analysis to be valid, the models used to simulate the motion of the spacecraft need to be representative of reality. However, typically there is a correlation between orbit propagation accuracy and propagation computational latency. DRL agents are not sample efficient. That is, DRL agents can take thousands of propellantless planar phasing iterations, modelled by orbit propagations, to converge upon a solution. Therefore, the latency of the orbit propagations will strongly determine the time taken for the DRL agent to learn to solve the problem. Within this analysis, a DRL agent was trained using a surrogate model of a high-fidelity numerical orbit propagator as a computationally efficient mechanism for predicting the future orbit state. Efforts within this work to train the DRL agent using the high-fidelity numerical propagator was also undertaken in this work.

Fast convergence upon a solution is desired because this increases the rate at which an agent can be trained within the simulation realm. Therefore, it also increases the rate at which the DRL agent can be tested in simulated or real conditions, assessed and improved upon to better solve a satellite formation flying problem. Fast training times are also ideal for cases where an existing agent needs to be trained, with little prior warning, how to handle new anomalous environments. This is particularly important if the successful handling of anomalies is critical to the health of the mission.

Table 2
Constants used within the high-fidelity propagation model.

Parameter	Value
gravitational harmonic degree	20
gravitational harmonic order	20
$F_{10.7}$	150
A_p	5

2.3.1. High-fidelity propagation model

A numerical three degree-of-freedom orbit propagation code, written in Cython and incorporating the OreKit[2] low-level spacecraft dynamics library, was tested to model the spacecraft dynamics. The high-fidelity propagator has been verified against the official release of OreKit (Smith, 2019). The high-fidelity propagator includes acceleration models for the Earth aspherical gravity, classical solar radiation pressure, drag, and third-body gravity accelerations. The high-fidelity propagator makes extensive use of OreKit for the acceleration models; and date/time, and coordinate system transformations. The NRLMSISE-00 atmospheric model is used to model the thermosphere. The acceleration models are superimposed and then numerically integrated using the Prince-Dormand 8(53) method native to the SciPy Python package.[3] The constants used within high-fidelity orbit propagator are shown in Table 2. The propagation parameters common to all propagation methods are shown in Table 3.

2.3.2. Propagator surrogate model

The high-fidelity orbit propagator is computationally intensive because of its numerical integration algorithm. Training a DRL agent can take thousands of episodes, and therefore, the time taken to train an agent is sensitive to the latency of the underlying model used to represent the orbital dynamics. Therefore, a surrogate model of the high-fidelity propagator was developed to test its ability to speed up the training of the DRL agent in converging upon an optimal solution.

The surrogate model of the high-fidelity propagation model was created as follows:

1. An episode was defined to be 30 days of propagation.
2. A total of 150,000 samples were generated for an episode using the high-fidelity propagator. Each spacecraft within each sample contained a randomly allocated cross-sectional area within the domain of three possible values that could be taken by the agents as defined in Section 2.4.
3. The semi-major axis and argument of latitude, that is the sum of the true anomaly and argument of perigee, were calculated from a Cartesian state propagated using the high-fidelity numerical propagator in increments of a day. The initial Cartesian state was generated based on

[2] https://www.orekit.org/.

[3] https://www.scipy.org/.

Table 3
Constants used within the high-fidelity propagation model

Parameter	Value
Drag coefficient, c_d	2.2
Minimum reference area, S_{min}	0.2 m^2
Medium reference area, S_{med}	0.4 m^2
Maximum reference area, S_{max}	0.6 m^2
Mass, m	3 kg
Epoch	2018–03-02T18:54:15.662
Radius of the Earth, R_{Earth}	6378.135 km

constant orbital elements other than the mean argument of latitude which was randomly generated between 0° and 360° based on a uniform distribution.

4. After each day, the difference between the propagated state and the equivalent unperturbed elements (two-body dynamics); and the cross-sectional areas of the spacecraft are stored for training the propagator surrogate model.
5. The fully-connected neural networks were trained to map inputs, namely semi-major axis, argument of latitude and cross-sectional area of the satellites, to deviations in semi-major axis and argument of latitude (compared to two-body dynamics) due to perturbations for the next day.

Thus the surrogate model takes a state estimated using two-body dynamics and then uses the surrogate model to transform the calculated two-body state into its estimate of what the state would be if the dynamics were modelled using the high-fidelity numerical propagator.

Fig. 2 shows the architecture of the neural networks with inputs and outputs. Fig. 3 and Fig. 4 show the error evolution for the semi-major axis and phase angle (argument of latitude) between the state propagated by the high-fidelity propagator and that predicted by the surrogate model over 30 days for the least accurate case tested. For this worst case sample, the cumulative error was at most approximately 600m for the semi-major axis and 6° for the argument of latitude. These errors are an improvement on the errors that would be generated if only the two-body dynamics were considered but are still large. The error in the state of the spacecraft between propagations made by the surrogate model and high-fidelity numerical propagator after 30 days were approximately the same as the equivalent error between the high-fidelity propagator and two-body dynamics after one day of propagation. Additionally, the accumulating error displayed in Figs. 3 and 4 somewhat stabilises over the thirty day period used to train the surrogate model. The absolute propagation error resulting from the surrogate model propagator is applied to all propagated spacecraft meaning that the error in their relative state is less than that shown in Figs. 3 and 4. The surrogate propagator was four orders of magnitude faster than the high-fidelity numerical propagator. Therefore, the surrogate propagator trades off accuracy compared to the high-fidelity numerical propagator for computational speed. The implications of this will be further discussed within Section 3.

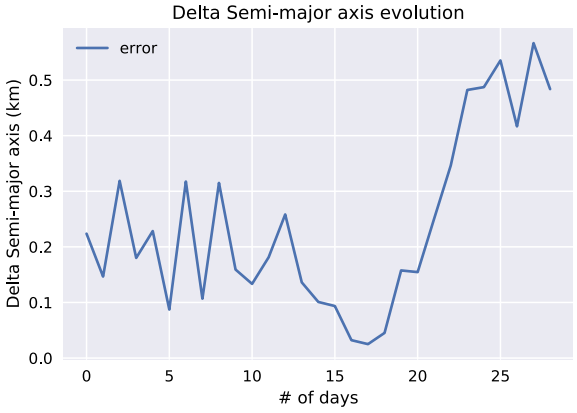

Fig. 3. Semi-major error evolution for surrogate propagator.

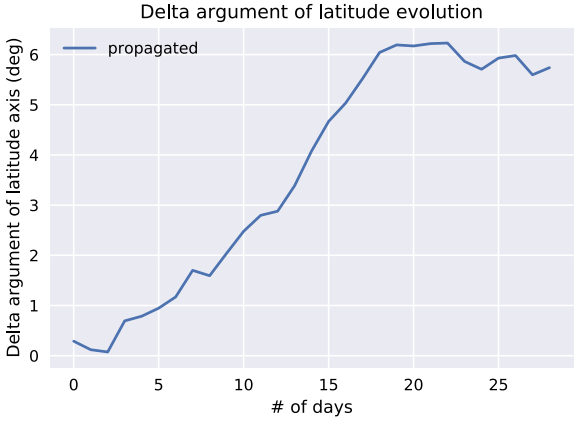

Fig. 4. Argument of latitude error evolution for surrogate propagator.

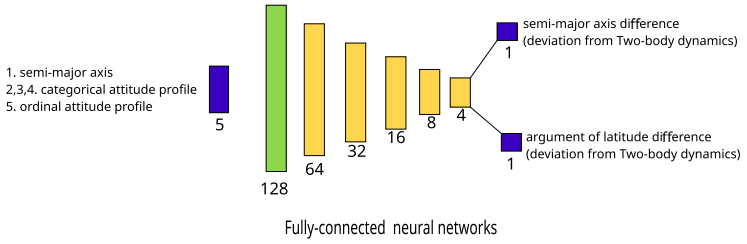

Fig. 2. The architecture of neural network for surrogate propagator.

The surrogate model is trained based on data from the high-fidelity orbit propagator which takes an initial state of the spacecraft, as well as, initial epoch. Therefore, the accuracy of the high-fidelity propagator is a significant factor in determining the accuracy of the surrogate model. Therefore, the surrogate model is specialised for the reference orbit and epoch used to train the surrogate model and its accuracy is not guaranteed for orbits that deviate from the orbit used to train the surrogate model. Therefore, in using the surrogate model to train the DRL agent, the surrogate model needs to be trained based on data that is representative of the orbit that the formation flying agents will operate in. However, the surrogate model can be applied to predict the future position of the satellite that it is trained on, and therefore, does not need to be retrained for different formation flying problems so long as the satellite being propagated has the same mass and cross-sectional area.

The computational cost of training the surrogate model within this work can be quantified as follows:

1. It takes approximately 60 h to generate 1050000 propagation samples using the high-fidelity orbit propagator. However, a 60-core CPU cluster (AMD Opteron Processor 6376 1400.0 MHz) was utilised in this work resulting in the data generation only taking one hour.
2. Training the surrogate model on the generated data took 30 min.

To the author's knowledge, no publicly available pre-trained network exists that solves a similar formation flying problem to the one analysed within this work. Therefore, a transfer learning approach, where a third-party pre-trained network could be taken as a starting point and then refined to solve the particular problem within this work could not be utilised.

2.3.3. Formation configuration

Most modern constellations consist of more than two spacecraft. For example, 88 spacecraft were launched as part of Planet's Flock-3p constellation (Foster et al., 2017). When a formation is comprised of more than two spacecraft, the changes in relative motion between each agent is coupled. That is, a deliberate change in relative state between two spacecraft will also alter their relative state compared to all other spacecraft. Further, in multi-satellite systems, actions taken by some spacecraft impose constraints on other spacecraft. This is the case for formation control algorithms, such as those used by Planet (Foster et al., 2015; Foster et al., 2017) that consider only the three differential drag states (a ternary set) caused by permutations of the maximum and minimum possible magnitudes of absolute drag, as shown in Fig. 5. For example, if a reference spacecraft within a formation of multiple identically shaped spacecraft attains the maximum absolute drag state, all other spacecraft in the system can only attain a lower or equal magnitude of absolute drag. In this situation, the differential drag that can be achieved among all spacecraft relative to the reference spacecraft is constrained to act in one direction. A benefit of the DRL method is its ability, due to the underlying deep neural network, to solve highly non-linear and constrained problems. Therefore, this analysis will apply the ability for DRL to solve highly non-linear problems by testing its practicability to perform satellite formation control of multiple satellites simultaneously. Specifically, the propellantless planar phasing of three spacecraft will be conducted because this is the minimum number of spacecraft that captures the complexity of multi-spacecraft systems. By limiting the system to three spacecraft, the computational cost associated with propagating the orbit of each spacecraft during each learning episode is minimised which increases the learning rate of the DRL agent. However, limiting the number of spacecraft to three comes with limitations. For example, the solution found by a DRL agent to a particular problem within a particular environment is unlikely to be an optimal solution to a similar, but different, problem within the same environment, or the same problem in a different environment. This is because the DRL agent's learnt mapping of inputs to corresponding outputs for one environment is unlikely to correspond to the same mapping in another environment. Therefore, by constraining the number of spacecraft to three, the specific DRL agent trained within this analysis is unlikely to optimally solve a system consisting of a different number of spacecraft. Further, the DRL agent applied within this work controls the multi-agent system in a centralised fashion. That is, the agent takes observations from each spacecraft and subsequently controls the actions of all spacecraft. Therefore, as the number of spacecraft increases, the decision and/or observation space of the DRL agent grows exponentially. This exponential growth in the decision and/or observation space will ultimately limit the number of spacecraft that this control method can practically handle.

The purpose of this paper is to gain insight into the process of training a DRL agent to solve and make data driven discoveries into complex satellite formation control problems such as the propellantless planar phasing of multiple spacecraft. The insights uncovered in this work are valuable, despite the aforementioned constraints, because the process may be used to control the highly complex tasks of performing satellite formation manoeuvring of multiple spacecraft amongst unpredictable space environments and other uncertainties.

2.4. Actions

Limiting the differential drag configurations to a ternary set, as shown in Fig. 5, constrains the sign of the differential drag value acting on combinations of spacecraft at any one point in time. However, when considering the propellantless coplanar phasing of multiple satellites, Smith et. al. (Smith et al., 2017a) demonstrated that both positive and negative values of differential drag can be achieved at the same time relative to a reference spacecraft if its default

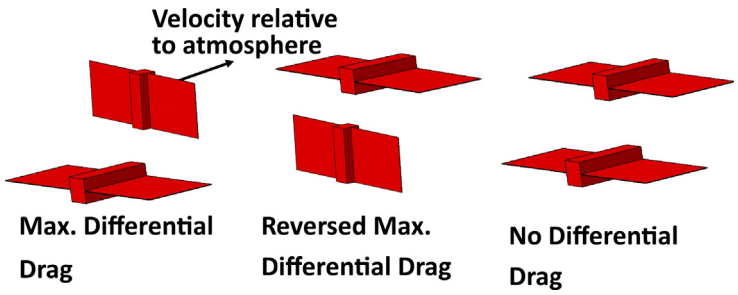

Fig. 5. The ternary set of differential drag states.

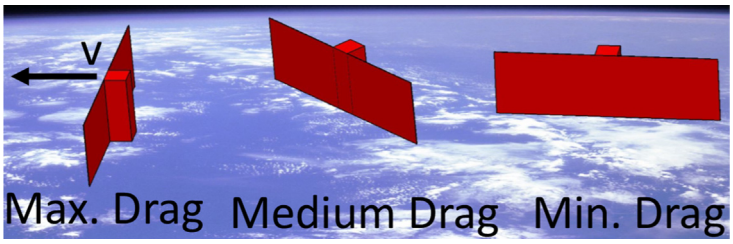

Fig. 6. Depiction of maximum, medium, and minimum absolute drag (Smith et al., 2017a).

magnitude of absolute drag is a medium value as shown in Fig. 6. That is, when a reference spacecraft has a medium magnitude of absolute drag, the other identically shaped spacecraft still have the capacity to generate more or less drag than the reference to achieve a positive or negative differential drag relative to the reference spacecraft. This was shown to improve the speed of differential drag-based planar phasing by allowing all other spacecraft to manoeuvre independently and simultaneously relative to the reference spacecraft. Therefore, the actions considered within the DRL environment used within this work consisted of permutations of maximum, medium, and minimum drag states. In reality, a change in the drag acting on a satellite would be achieved by changing the incidence angle of the satellite itself (Foster et al., 2015) or its solar arrays. Such a change in incident angle would change both the cross-sectional area exposed to the oncoming gas flow and the drag, lift, and side-force coefficients. Within this work, the maximum, medium, and minimum drag states were enacted by changing the cross-sectional area of the spacecraft within the simulations. The values of the minimum, medium, and maximum cross-sectional areas used within the simulations were $0.02\ m^2, 0.04\ m^2$, and $0.06\ m^2$. Therefore, there are 27 permutations of the cross-sectional area which form possible input actions for the DRL agent. The drag coefficient within the simulations was assumed to be a constant value of 2.2, a commonly assumed value for low Earth orbit (Moe et al., 1998). The lift and side-force coefficients within the propagations were assumed to be zero. Assuming that the magnitude of absolute drag can be controlled by pure alterations in the cross-sectional area of the spacecraft is not reflective of reality as spacecraft rarely are able to expand and contract their size while keeping their drag coefficient completely constant. However, this work concerns the ability of

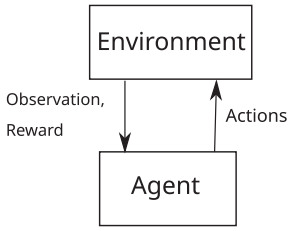

Fig. 7. The layout of the interaction between the agent and the environment in the reinforcement learning framework.

DRL agents to enact formation flying manoeuvres based on differential drag. Therefore, the mechanism used to enact the differential drag does not matter so long as the magnitudes of differential drag acceleration are reflective of what can be achieved in reality. Further, the magnitude of differential drag is typically an order of magnitude larger than differential lift and side-force (Sentman, 1961). Therefore, the effects on the relative motion of differential lift and side-force can be controlled for this first effort analysis of the performance of DRL to enact satellite formation control.

2.5. Deep reinforcement learning

Deep reinforcement learning is a subfield of machine learning. It can be used to learn the optimal decisions required to achieve a particular task from experience, even in complex and uncertain environments. Although DRL includes a neural network to learn patterns between an input and output, it does not require labelled data as reference to the solution during training as is the case for supervised machine learning. Deep reinforcement learning models learn optimal behaviour through supervision by

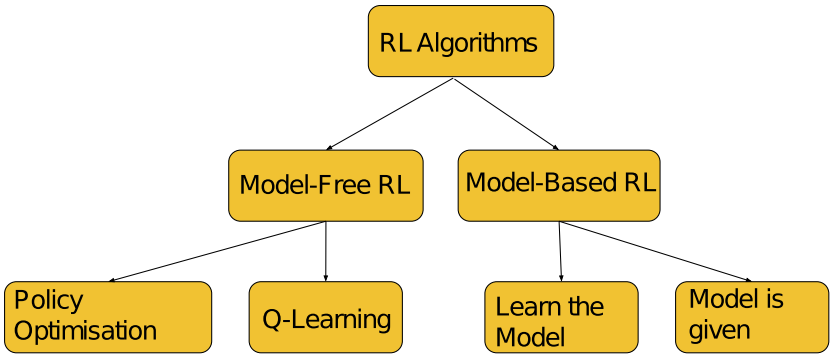

Fig. 8. The taxonomy of reinforcement learning algorithms.

observation and cumulative reward feedback from the environment. Fig. 7[4] shows the interaction between the agent and the environment in the reinforcement learning (RL) framework. In addition to the agent and the environment, a reinforcement learning model includes other critical components such as a policy, a reward signal and a value function (Sutton and Barto, 1998). A policy is a mapping between observations and actions that subsequently determines the agent's behaviour. A reward signal is a scalar value that is provided to the agent after it interacts with the environment. The reward value reflects the performance of the agent in achieving its prescribed goal. The agent tries to maximise the scalar reward value in order to optimally solve its designated problem. While a reward signal defines the immediate reward for the current action, the value function is established after repeated experiences to predict the probable reward for a given action (Sutton and Barto, 1998).

Different types of reinforcement learning algorithms are shown in Fig. 8.[5] This paper considers multiple model-free policy optimisation methods, namely advantage actor-critic agents (A2C) and proximal policy optimisation (PPO), to calculate the automated, time optimal planar phasing of miniaturised satellites using environmental perturbations. Multiple model-free policy optimisation methods are considered in this work to test whether the framework developed in this paper is capable of supporting multiple policy-based reinforcement learning methods. Model-free approaches (trial and error learning) is chosen over model-based RL algorithms (learning by intentional planning) due to their faster convergence and simplicity. The model-based methods require a model of the environment that helps the agent to plan ahead before it interacts with the actual environment. The policy optimisation methods are preferred over Q-learning algorithms because the direct optimisation of policies is more efficient for problems with stochastic environments and large number of actions. In addition, Q-learning methods utilises the value, which is the discounted total reward of action and state pairs, to decide how to act at each step, and the proposed problem does not require the approximation of the values (Sutton and Barto, 1998). The details about the methods used to investigate the proposed problem are provided in the following subsections along with a subsection that explains the setup of the problem in the Gym toolkit, by OpenAI,[6] which is designed for developing and comparing reinforcement algorithms.

2.5.1. Advantage actor-critic method

Asynchronous Advantage Actor-Critic Method (A2C) (Wang et al., 2016) is a type of actor-critic reinforcement learning method that is more stable and has faster convergence compared to simple policy gradient (PG) methods. The actor is a neural network that conducts actions in the environment, and the critic is another neural network that facilitates the learning of the agent by approximating the value function ($V(s)$). The convergence of the agent towards an optimal solution is dependent on the variance of the gradients that update the policy ($\pi(s)$). The actor-critic methods reduce the variance of the gradient by scaling the policy gradient, utilising a function called the advantage function ($A(a,s)$). (see Fig. 9).

2.5.2. Proximal policy optimisation method

The Proximal Policy Optimisation Method (PPO) (Schulman et al., 2017) can be considered as an extension of A2C (Schulman et al., 2017; Wang et al., 2016) based on the approach that it utilises to update the policy. The objective function of PPO directly optimises the policy. The objective functions for A2C and PPO are given in Eqs. (1) and (2) respectively.

$$J_\theta = \mathbb{E}[\nabla_\theta log\pi_\theta(s)A(a,s)] \tag{1}$$

$$\begin{aligned} J_{\theta_{clip}} &= \mathbb{E}[min(R_\theta A(a,s), clip(R_\theta, 1-\epsilon, 1+\epsilon)A(a,s))] \\ R_\theta &= \frac{\pi_\theta(s)}{\pi_{\theta_{old}}(s)} \end{aligned} \tag{2}$$

[4] http://rail.eecs.berkeley.edu/deeprlcourse/.
[5] https://spinningup.openai.com.
[6] https://gym.openai.com/.

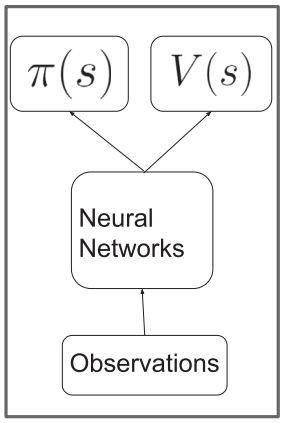

Fig. 9. The higher level architecture of A2C.

where $\pi_\theta(s)$ is a policy, $A(a,s)$ is an estimator of advantage function, and ϵ is a hyperparameter to limit the update of the policy. The ϵ parameter has been introduced to stabilise the updates of the policy during training. PPO methods may provide faster and more stable convergence during training due to the extensions added to the policy update scheme.

2.5.3. OpenAI Gym

The OpenAI Gym framework was utilised within this work to train the DRL agents. The OpenAI Gym framework is designed for developing and comparing reinforcement learning algorithms. The two main elements of a reinforcement learning model are the agent and the environment, the Gym toolkit provides various environments with utilities regarding actions that can be conducted by the agents and the observations that can be obtained from the environment. In addition, custom environments can be developed within the Gym framework. Fig. 10 represents the higher level architecture of an environment. A custom planar phasing for miniaturised satellites environment is developed using the Gym framework for this paper.

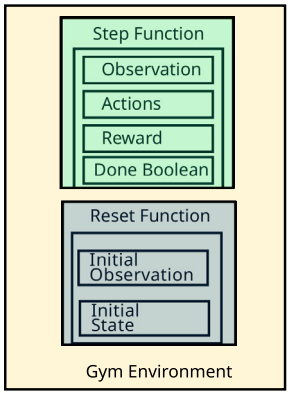

Fig. 10. The higher level architecture of the Gym toolkit.

2.5.4. Reward function and observations

In reinforcement learning, agents learn by interacting with the environment to achieve a goal. The reward signal is the feedback that the agents receive to guide them in accomplishing a task. The reward signal is generated using a user defined reward function. The reward function is a mathematical function that encapsulates the desired performance of the agent, after interacting with its environment, in solving a problem. The DRL agent tries to find an optimal solution to the reward function which then corresponds to solving the underlying problem. The DRL agent solves the problem by training a deep neural network to map actions to desired outcomes through "trial and error". Typically, during the initial training process, the actions taken by the DRL agent have a high probability of being suboptimal, or counter-productive. The agent progressively learns the optimal actions required to solve the problem within its environment as it experiences better performing decisions.

Therefore, shaping the reward signal to properly guide the agents in completing the correct task well, is essential for reinforcement learning. The agent's fully-connected neural networks in this work (Fig. 11) learn the desired actions in a sequence based on observations with the help of the reward signal. Eq. (3) shows the reward function used for this work.

$$Reward = \Phi + \Phi^{(2-\frac{N}{30})}\Theta \tag{3}$$

$$\Phi = \frac{1}{\sqrt{1 + E_{maol}^{T} E_{maol}}} \tag{4}$$

$$\Theta = \frac{1}{\sqrt{1 + E_{sma}^{T} E_{sma}}} \tag{5}$$

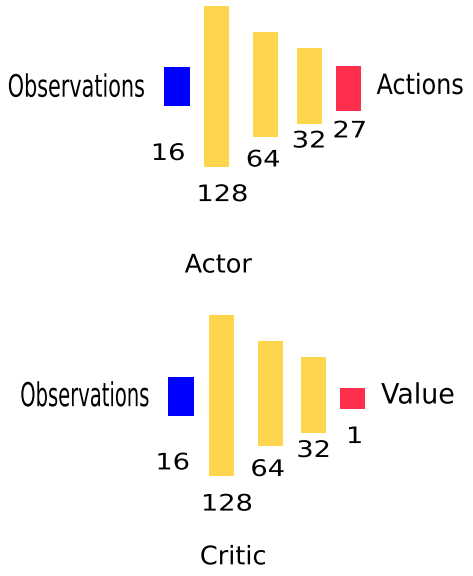

Fig. 11. The architecture of the policy (top) and value (bottom) networks.

Algorithm 1. Algorithm for a step in the training process.

Input: *action*: the commanded action, $state_i$: starting state of all spacecraft;
Output: *done*: true if formation achieved else false, *reward*: value of reward function, $state_{i+1}$: state propagated a day into the future;
1 initialise the propagator with the starting $state_i$ for this step (previous state from last step);
2 assign cross-sectional areas to each spacecraft based on the commanded *action*;
3 $state_{i+1}$ = propagate the state of each spacecraft a day into the future;
4 $\delta\lambda$ = the relative phase angle calculated for each pair of spacecraft from $state_{i+1}$;
5 δa = the relative mean semi-major axis calculated for each pair of spacecraft from $state_{i+1}$;
6 $done = \delta\lambda < 1^\circ$ and $\delta a < 1km$ for all pairs of spacecraft (true if the formation meets the convergence criteria else false);
7 *reward* = the reward calculated from the reward function defined in Eq. (3);

In Eqs. (4) and (5), E_{maol} is the error between the desired (10°) and current relative phase angle (mean argument of latitude) of the satellites, E_{sma} is the error between the zeroed relative semi-major axes and current relative semi-major axes of the satellites, N is the number of days passed since the epoch. The observations provided to the agents and used as input into the reward function are given in Table 4. The observations were made at a frequency of one day. This is a conservative value representative of the frequency that a miniaturised satellite operator utilising a single ground-station can downlink observational data and issue new commands. The reward function in Eq. (3) has two components which are related to relative phase angle separation (Φ) and relative semi-major axis decrease (Θ). This is to address these two coupled aspects of propellantless planar phasing. That is, in order to achieve a relative phase angle change between any two spacecraft there must be a difference in the semi-major axes of the spacecraft.

Therefore, within the initial stage of the manoeuvre, the agents are enticed to separate their relative phase angles by rewarding them more for behaviour that separates their relative phase angle compared to holding their semi-major axes fixed. This can be seen in the reward function chosen in Eq. (3) where in the first few days of operation, that is when $N \ll 30$, the exponent in the $\Phi^{(2-\frac{N}{30})}\Theta$ term of Eq. (3) returns a higher reward signal for actions that minimises the error in mean argument of latitude (i.e. maximises Φ) relative to actions that minimise the relative semi-major axis (i.e. maximise Θ). This causes the agent to take actions that separate the mean argument of latitude of the spacecraft by increasing their relative semi-major axis. However, as time, and the manoeuvre, progresses $(2-\frac{N}{30}) \to 1$ as $N \to 30$ in Eq. (3) such that less weight is given to the minimisation of the mean argument of latitude. This coupled with the fact that as the desired relative mean argument of latitude is established, Φ becomes its minimum value of 1 such that the reward signal is then maximised by maximising Θ through minimising the relative semi-major axis. This in effect entices the agent to learn the behaviour that minimises the relative semi-major axis once the desired relative mean argument of latitude is established. The reward function chosen within this work encapsulates the behaviour that is desired of the specific formation considered in this work. The selected reward function is adequate for this first effort analysis into the feasibility of DRL to control a multi-agent formation of spacecraft. The reward function can be interchanged with different reward functions that may better dictate the desired behaviour. The implementation of the observations and reward function during each step of the training process can be seen in Algorithm 1.

2.6. Model description

The A2C and PPO model-free policy optimisation DRL methods were used in this work to train agents to solve propellantless planar phasing of multiple spacecraft using orbit propagators to model the underlying orbital dynamics. The model-free policy optimisation DRL methods considered in this work contain separate neural networks called the actor and critic as was explained in Section 2.5. The architecture of the actor and critic networks can be seen in Fig. 11. The model has three layers. The output of the actor is a 27 dimensional soft probabilities which

Table 4
Observations provided to the agents.

Parameters (a: agents, t_n : n^{th} *step*)	Value Range
Difference in current angle ($a_2 - a_1$)	$0 - 360^\circ$
Difference in current angle ($a_3 - a_2$)	$0 - 360^\circ$
Difference in desired and current angle ($a_2 - a_1$)	$0 - 360^\circ$
Difference in desired and current angle ($a_3 - a_2$)	$0 - 360^\circ$
Difference in desired and current angle (unit vector) ($a_2 - a_1$)	0 – 1
Difference in desired and current angle (unit vector) ($a_3 - a_2$)	0 – 1
Difference in semi-major axes ($a_2 - a_1$)	0 – 5 km
Difference in semi-major axes ($a_3 - a_2$)	0 – 5 km
Rates of all above parameters ($t_n - t_{n-1}$)	–

Table 5
Hyperparameters of the networks within this work.

Parameter	A2C	PPO
Hidden Layers	Actor: 128–64-32 Critic: 128–64-32	Actor: 128–64-32 Critic: 128–64-32
Neural Network Architecture	Fully Connected	Fully Connected
Learning Rate	Actor: 0.001 Critic: 0.001	Actor: 1e-4 Critic: 0.001
Batch Size	32	64
Done Criteria: Planar Phase Angle Relative Semi-Major Axis	 10° < 1 km	 10° < 1 km

is identical to the dimensions of the action space as described in Section 2.4. Softmax is used to pick an action. The critic has one dimensional output corresponding to the value output. The actor and critic networks considered within this paper do not share any layer. The hyperparameters used can be seen in Table 5. A grid search method was used to optimise the hyperparameters. The convergence criteria for the network can be seen in Table 5 and is described in more detail within Section 2.1.2.

3. Results and discussion

3.1. Importance of propagation speed

This work seeks to test the feasibility of model-free deep reinforcement learning to control, and provide data driven discoveries into, complex satellite formation flying problems. Within this work, the ability of a high-fidelity numerical propagator and a surrogate propagation model were tested to simulate orbital dynamics to train a DRL agent to complete propellantless planar phasing of multiple spacecraft. The propagation methods had varying propagation accuracy and computational latency. The propagation method used to simulate the dynamics of the spacecraft heavily influenced the rate at which the DRL could converge upon a solution. This was because of the computational latency inherent to each propagation method that was encompassed within the training environment. The propagation latency translated to latency in performing an episode of learning.

The suitability of the propagation method was quantified by the minimum amount of time it took to propagate an orbit a day into the future from the same initial conditions. Values for each propagation method can be seen in Table 6.

Table 6
Time taken to propagate an orbit a day into the future.

Propagation Model	Propagation time (s)
High-fidelity Numerical Propagator	10
Surrogate Model	0.003

When training using the high-fidelity propagator, it took in the order of weeks for a solution to be found using the A2C and PPO training methods. The latency of the high-fidelity propagator was inhibiting as it prevented frequent feedback of the suitability of the DRL setup for solving the problem and subsequent iterative improvement. For example, the fitness of the reward function for solving the problem could not be assessed until after weeks of training had elapsed. Often the reward function would be found to be suboptimal for solving the formation manoeuvring problem meaning that weeks of training had been wasted. Training DRL agents using the high-fidelity propagator within this work was so onerous that no agent was trained to adequate convergence. Therefore, simulating the spacecraft dynamics using computationally latent propagators is unattractive for initially solving new satellite formation flying problems where feedback and optimisation of the reward function and training setup is required.

The surrogate model propagated orbits four orders of magnitude faster than the high-fidelity numerical propagator as can be seen in Table 6. This translated to much faster times taken to converge upon a solution to the propellantless planar phasing problem. When using the surrogate propagation model, the A2C solution took 2 min to converge, and the PPO took 11 min. It was not expected that the A2C method would converge faster than the PPO method as the PPO method is typically more efficient than the A2C method. This result may be due to the stochastic nature of the convergence time for A2C and PPO methods. Additionally, this result may be due to suboptimal hyperparameters within the specific PPO networks considered in this work leading it to perform below its potential.

Therefore, the ability to successfully train DRL agents to perform satellite formation flying is drastically improved by low latency orbit propagations. This is particularly true for the propellantless planar phasing manoeuvre selected as the subject of this work. This is because the large desired relative mean argument of latitude values took days of real time manoeuvring to achieve under the subtle differential drag accelerations which corresponded to intensive work required by the underlying orbit propagators.

3.2. Numerical error accumulation

Numerical propagators are considered to be the most precise propagation technique for short-term simulations. However, numerical propagators suffer from the accumulation of numerical error. Numerical error occurs because numerical propagators only approximate the true integration of the differential equations of motion of a spacecraft. The numerical error compounds for every step that the orbit is numerically propagated causing a patterned accumulation in error. The high-fidelity numerical propagator used within this work was no exception and was found to have accumulating numerical error. Accumulating numerical error results in the numerically propagated position of the spacecraft diverging from the position the spacecraft

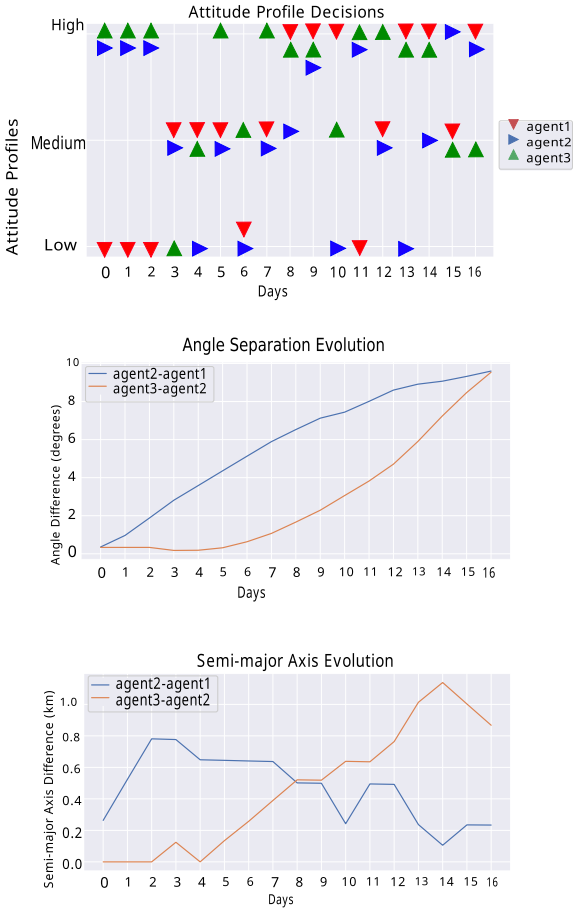

Fig. 12. The performance of agents using A2C method.

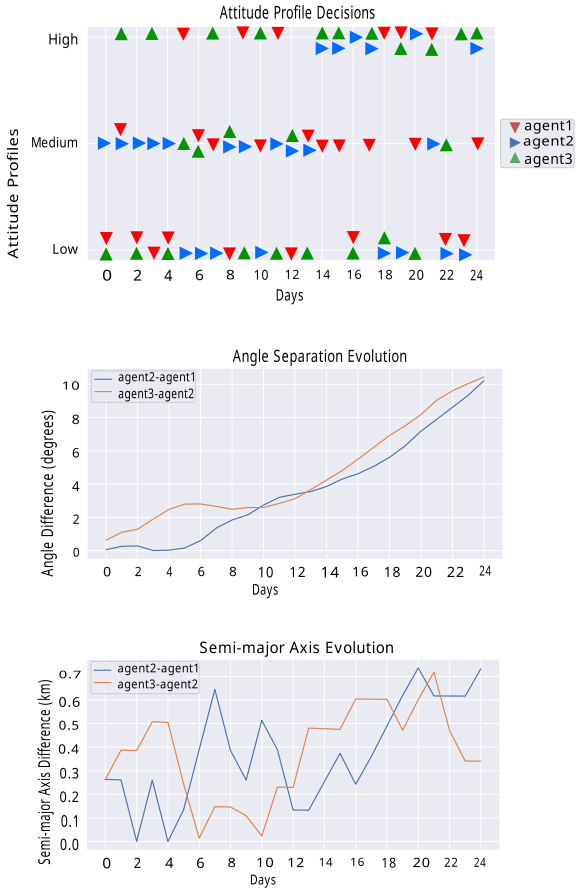

Fig. 13. The performance of agents using PPO method.

should be based if it naturally translated according to its acceleration. Therefore, the accumulating numerical error is a non-realistic artefact and does not occur in true orbital motion. This is a problem for training the DRL agents within this work using numerical propagators because the learned orbital motion of the spacecraft included non-realistic numerical artefacts.

Even though the surrogate model was trained using the high-fidelity numerical propagator, the numerical errors were limited within the surrogate model propagations. This is because only numerical propagations a day into the future from a given input state were used to train the surrogate model. Recalling that numerical error accumulates over time, by limiting the propagations used to train the surrogate model to a day, a day's worth of numerical error was learned by the surrogate model compared to 30 day's worth of numerical error that accumulated when training the agent with numerical propagators. The agent trained using the surrogate model propagator within this work was tested in an environment modelled by the high-fidelity propagator to product Fig. 12 and Fig. 13. Therefore, an element of the manoeuvre error maybe attributed to numerical error in the environment simulation that was not learnt by the agent trained using the surrogate model.

3.3. Performance of the policy-gradient methods

This work considers the A2C and PPO policy-gradient reinforcement learning methods to rephase multiple spacecraft, not to benchmark the methods against each other, but to assess whether the pipeline developed in this work can work with various policy-based DRL methods. The DRL methods trained using the surrogate model were tested in an environment simulated by the high-fidelity propagator.

For A2C, the success criteria of the propellantless planar phasing of three satellites, outlined in Section 2.1.2, is achieved in 16 days. The mean angle error in the mean relative argument of latitude is 0.5 degrees. The error in the mean relative semi-major axis is 0.7 km. Fig. 12 shows the relative mean phase angle, relative mean semi-major axis evolution, and attitude (cross-sectional area) profile decisions for three satellites.

For PPO, the success criteria for the manoeuvring of three satellites is achieved in 24 days. The error in mean relative argument of latitude is 0.3 degrees. The error in the mean relative semi-major axis is 0.5 km. Fig. 12 shows the relative phase angle, relative semi-major axis separation evolution, and attitude (cross-sectional area) profile decisions for three satellites.

Both the A2C and PPO methods satisfy the desired terminal conditions. However, the A2C method outperformed the PPO method in this work by completing the manoeuvre in a shorter amount of time. The residual relative mean semi-major axis at the end of manoeuvring for the A2C case was also less than that of the PPO case. These results demonstrate that the framework developed in this work is able to support multiple reinforcement learning methods to solve complex formation flying problems.

3.4. General performance of the agent

The purpose of this paper was to propose a DRL framework that can be used to find optimal solutions to highly complex formation flying problems. The DRL agent trained in this work to demonstrate planar phasing of multiple spacecraft was trained based on orbital dynamics modelled by a surrogate orbit propagator. This surrogate orbital propagator was itself trained on numerical propagation data from initial state and epoch defined within limited bounds. The agent adequately solved the formation flying problem when the initial orbital conditions were defined within the original bounds in which the agent was trained as were detailed in Section 3.3. To investigate the general performance of the same agent as analysed in Section 3.3, the PPO agent was applied to solve the same rephasing problem of three spacecraft as defined in Section 2.1.2 when the initial orbital elements were slightly deviated. Table 7 shows the final error in actual semi-major axis and mean argument of latitude compared to the desired values respectively with isolated deviations in initial conditions. The deviated initial conditions in Table 7 can be compared to the original values that the agent was trained on in Table 1.

The results in Table 7 show that for small deviations in the initial orbital conditions, the agent is still capable of rephasing the spacecraft. However, the performance of the agent degraded as the initial orbital conditions deviated from those that the agent was originally trained with. This is because the agent is trained to solve a problem based on spacecraft dynamics which are a function of a particular orbital regime. By applying the agent in a different orbital regime, the fundamental dynamics encapsulated within the agent are no longer reflective of the actual dynamics experienced by the spacecraft leading to poor performance. Therefore, the degradation in the performance of the agent is more sensitive to initial conditions that change the fundamental dynamics of the spacecraft such as semi-major axis and eccentricity. As is typical of most data-driven solutions, the DRL agent trained using the framework proposed in this paper will work most optimally when applied to solve problems that closely resemble the problem it was originally trained to solve. A new agent should be trained using the method outlined in this paper to solve formation flying problems where that problem deviates significantly from existing agents.

3.5. Importance of the reward function

During the training of a DRL agent to perform satellite formation flying, the definition of the reward function is critical for achieving constructive behaviour. For example, in this analysis, it was difficult to define a reward function that guided the DRL agent to both achieve the relative mean argument of latitude while also reducing the residual relative mean semi-major axis to zero at the completion of the manoeuvre. The residual error in the relative mean semi-major axis at the end of manoeuvring for the A2C and PPO cases shown in Figs. 12 and 13 was large such that the relative mean phase angle achieved by the machine learning agents will be unstable in the long-term.

It is common to optimise formation flying algorithms to minimise the time taken to complete a manoeuvre in order to minimise the "down-time" that the satellite cannot perform its primary mission. It was challenging to define a reward function that encouraged the successful completion of the propellantless planar phasing manoeuvre in the minimum amount of time as possible. The elapsed manoeuvre time can be an input observation signal to the reward function that could encourage time-based behaviour in order to attempt to minimise the manoeuvre time. The DRL methods tested in this paper would greatly benefit from further optimisation of the reward function to encourage more timely manoeuvring. An evolution in the reward value is plotted over the training step for each of the PPO and A2C methods in Fig. 14 and Fig. 15 respectively.

Initially within the training setup, the manoeuvring time was capped at 30 days. In this case, the DRL agent was trained over 30 days after which its solution was assessed based on the accuracy of the final state of the spacecraft. However, it was discovered that waiting for 30 days to

Table 7
Performance of the agent from deviations in initial orbital conditions.

Deviated Variable	Deviated Value		E_{sma}	E_{maol}
Semi-Major Axis	6800.00 km	agent2-agent1	−2.99km	−0.75°
		agent3-agent2	0.97 km	−0.39°
Eccentricity	0.001	agent2-agent1	0.27*km*	−0.27°
		agent3-agent2	−0.87 km	−0.01°
Inclination	90°	agent2-agent1	−0.74*km*	−0.13°
		agent3-agent2	−0.01 km	−0.27°

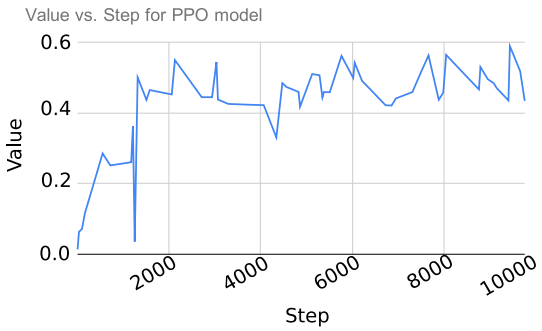

Fig. 14. The value of the reward function with each step using PPO.

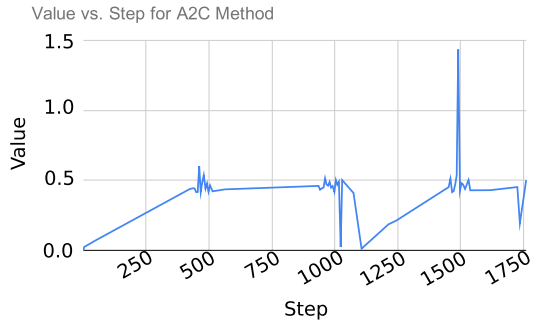

Fig. 15. The value of the reward function with each step using A2C.

elapse before assessing the solution based on the success criteria was inadvertently training the agents to complete the manoeuvres in exactly 30 days. Therefore, the strategy was changed such that the solution was assessed after each day long timestep and the training was stopped as soon as the DRL agent converged on an adequate solution even if this was less than the 30 day upper limit.

3.6. Comparison with an expert

The problem solved within this work is fundamentally different to similar problems existing in literature, such as that recently published (2019) to rephase NASA's CYGNSS constellation (Bussy-virat et al., 2019), because the problem within this work does not consider the spacecraft to have an advantageous separation rate in planar phase angle due to the direction of deployment from the dispenser which was the case for CYGNSS (Bussy-virat et al., 2019). Also, this problem takes the improved step of rephasing spacecraft by considering a set of low, medium, and high drag states giving pairs of spacecraft greater flexibility to attain positive or negative differential drag accelerations at a given point in time (Smith et al., 2017a). This is contrast to other methods where only low and high drag states are considered (Bussy-virat et al., 2019; Foster et al., 2015; Foster et al., 2017). Additionally, this work is a first effort analysis into the use of deep reinforcement learning to conduct satellite formation control and, is therefore, not yet an optimised solution to the formation flying problem considered. The purpose of this work is to generate a framework that can be built upon to make data driven discoveries of optimal ways to solve formation flying problems. Therefore, a direct comparison to existing satellite rephasing strategies and performance is not valid such as that described for CYGNSS (Bussy-virat et al., 2019). However, it is valid to consider some high-level differences between the solution arrived at by experts and that arrived at by the DRL agent.

The main difference between the expert solution and the DRL agent is that the differential drag configurations are not held constant throughout the multiple day long manoeuvre, as was the case for the control of CYGNSS (Bussy-virat et al., 2019), but instead transitions between low, medium, and high drag configurations frequently as can be seen in Figs. 12 and 13. This can be attributed to the stochastic nature of the DRL agent in finding a near globally optimal solution to the formation flying problem dictated by the reward function and the complex environment in which the agent is trained and tested. The unintuitive results in Figs. 12 and 13 may reveal a more optimal way to solve a given problem, dictated by the reward function, beyond the intuitive solution reached by experts as per the status quo. In some cases, the solution reached by experts may only be a locally optimal solution. However, as stated in Section 3.5, the accuracy and relevance of the solution found by the DRL agent is dependent on how accurately the reward function used to train the DRL agent captures the nature of the formation flying problem to be solved.

4. Conclusion

Model-free policy optimisation deep reinforcement learning (DRL) is a method of mapping inputs to outcomes without modelling the underlying environment, and therefore, can solve highly non-linear and uncertain problems. Given the benefits of deep reinforcement learning, this work investigates the feasibility of DRL, believed to be for the first time, to conduct and make data driven discoveries of the optimal solutions to complex satellite formation control problems. The complex formation flying problem studied in this work was the propellantless planar phasing of multiple miniaturised spacecraft.

The agents were trained based on orbital dynamics modelled using a series of orbit propagators of varying fidelity and computational latency. The time taken to train the deep reinforcement learning agents was very sensitive to the computational latency of the orbit propagation method used to simulate the spacecraft dynamics. The deep reinforcement learning agents took in the order of weeks to train using high-fidelity numerical propagation algorithms which inhibited the rate of iterative insights and improvements to the framework setup. Further, the DRL agent learned the numerical error arising from prolonged numerical propagations of the underlying dynamics introducing inherent error to the agent's formation flying solution compared to an environment with realistic orbital dynamics. A

novel surrogate propagation model, generated within this work, improved the propagation computation time by four orders of magnitude allowing the DRL agent to be trained within minutes. By training the surrogate model based on orbital dynamics from numerical propagations of at most a day long, the numerical error learned by the surrogate model was contained and did not grow with the length of the propagation using the surrogate model. Therefore, the DRL agent trained using dynamics modelled using the surrogate model did not learn unrealistic numerical propagation errors.

In addition to the propagation speed, careful definition of the reward function used to guide the behaviour of the DRL agent was found to be very important. For the planar phasing manoeuvre, the reward function had to provide incentive for the DRL agent to separate its relative planar phase angle during the initial period of the manoeuvre and then stabilise it within the later stages. Poorly defined reward functions led to the DRL trying to solve a problem which corresponded to failed planar phasing of multiple spacecraft.

Using the surrogate model and a well defined reward function to train the DRL agent, the agent trained within this work successfully found a solution to the propellantless planar phasing of three spacecraft. The framework developed within this work is not limited to controlling three spacecraft and can be scaled to a greater number of spacecraft up to a limit dictated by the size of the observation and action space. Further, the general framework developed within this work is not limited to planar phasing of multiple spacecraft but can be applied to other complex satellite formation flying problems by updating the reward function.

This work sets the foundation for further expansion of the DRL method for conducting satellite formation control, and thus, is a step toward realising autonomous and robust satellite formation flight.

5. Future work

This work analyses the feasibility of model-free policy optimisation deep reinforcement learning (DRL) methods to achieve propellantless planar phasing of multiple satellites, believed to be for the first time. An output from this first effort analysis is the identification and insight into many areas of future work that could expand and improve the application of DRL to perform satellite formation flying which, for the sake of conciseness, have not been included in this paper. These areas of future work are:

1. An experiment analysing the performance of satellite formation control using the DRL method, trained using simulated satellite dynamics, orbiting in an actual space environment.
2. Testing the limits of the maximum number of spacecraft that can be practically controlled using this DRL control method.
3. Improvements in the precision of the final desired relative state of spacecraft by further optimising the reward function used to dictate the goal state of the trained DRL agent.
4. Further optimisation the reward function used to determine the formation flying problem solved by the agent to include a time parameter for time optimal manoeuvring.
5. Comprehensive profiling of the gradient learning methods, A2C and PPO, used within this work to see which is more performant over a greater range of formation control scenarios.
6. Testing a transfer learning approach which involves pretraining the agent using a computationally efficient surrogate model propagator before further improving the accuracy of the agent by training it with computationally inefficient, yet accurate high-fidelity propagators.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Bevilacqua, R., Romano, M., 2008. Rendezvous maneuvers of multiple spacecraft using differential drag under j2 perturbation. J. Guidance, Control, Dynam. 31, 1595–1607. https://doi.org/10.2514/1.36362.

Broida, J., Linares, R., 2019. Spacecraft Rendezvous Guidance in Cluttered Environments Via Reinforcement Learning. In: 29th AAS/AIAA Space Flight Mechanics Meeting January. Ka'anapali, HI: American Astronautical Society, Univelt.

Bussy-virat, C.D., Ridley, A.J., Masher, A., Nave, K., Intelisano, M., 2019. Assessment of the Differential Drag Maneuver Operations on the CYGNSS Constellation. IEEE J. Select. Top. Appl. Earth Observ. Remote Sens. 12, 1–9. https://doi.org/10.1109/JSTARS.2018.2878158.

Dell'Elce, L., Kerschen, G., 2015. Optimal propellantless rendez-vous using differential drag. Acta Astronaut. 109, 112–123. https://doi.org/10.1016/j.actaastro.2015.01.011.

Di Mauro, G., Lawn, M., Bevilacqua, R., 2018. Survey on Guidance Navigation and Control Requirements for Spacecraft Formation-Flying Missions. J. Guidance, Control, Dynam. 41, 581–602. https://doi.org/10.2514/1.g002868.

Foster, C., Hallam, H., Mason, J., 2015. Orbit determination and differential-drag control of planet labs cubesat constellations. arXiv preprint arXiv:1509.03270, pp. 1–13. URL: http://arxiv.org/abs/1509.03270. arXiv:1509.03270.

Foster, C., Mason, J., Vittaldev, V., Leung, L., Beukelaers, V., Stepan, L., Zimmerman, R., 2017. Constellation phasing with differential drag on planet labs satellites. J. Spacecr. Rock. 55, 1–11. https://doi.org/10.2514/1.A33927.

Furfaro, R., Bloise, I., Orlandelli, M., Di, P., 2018a. Deep learning for autonomous lunar landing. In: 2018 AAS/AIAA Astrodynamics Specialist Conference. Snowbird, UT, pp. 1–22. URL: http://arclab.mit.edu/wp-content/uploads/2018/12/2018_astro03.pdf.

Furfaro, R., Bloise, I., Orlandelli, M., Di Lizia, P., Topputo, F., Linares, R., et al., 2018b. A recurrent deep architecture for quasi-optimal feedback guidance in planetary landing. In: IAA SciTech Forum on Space Flight Mechanics and Space Structures and Materials. Moscow, Russia, pp. 1–24. URL: https://re.public.polimi.it/retrieve/handle/11311/1069166/324860/FURFR02-18.pdf.

Gaudet, B., Furfaro, R., 2012. Robust spacecraft hovering near small bodies in environments with unknown dynamics using reinforcement learning. In: AIAA/AAS Astrodynamics Specialist Conference. Minneapolis, Minnesota, p. 5072. URL: https://arc.aiaa.org/doi/10.2514/6.2012-5072. doi:10.2514/6.2012-5072.

Gaudet, B., Furfaro, R., 2014. Adaptive pinpoint and fuel efficient mars landing using reinforcement learning. IEEE/CAA J. Autom. Sin. 1, 397–411. https://doi.org/10.1109/JAS.2014.7004667.

Gaudet, B., Linares, R., Furfaro, R., 2020. Deep reinforcement learning for six degree-of-freedom planetary landing. Adv. Space Res. 65, 1723–1741. https://doi.org/10.1016/j.asr.2019.12.030.

Harris, A., Teil, T., Schaub, H., 2019. Spacecraft decision-making autonomy using deep reinforcement learning. In: 2019 AAS/AIAA Astrodynamics Specialist Conference. Portland, Maine, p. 5072. URL: https://hanspeterschaub.info/Papers/Harris2019.pdf.

Harris, M.W., Açikmese, B., 2014. Minimum time rendezvous of multiple spacecraft using differential drag. J. Guid., Control, Dynam., 37, 365–373. URL: http://arc.aiaa.org/doi/10.2514/1.61505. doi:10.2514/1.61505.

Horsley, M., Nikolaev, S., Pertica, A., 2013. Small satellite rendezvous using differential lift and drag. J. Guidance, Control, Dynam. 36, 445–453. https://doi.org/10.2514/1.57327.

Hou, Y.G., Zhang, M.J., Zhao, C.Y., Sun, R.Y., 2016. Control of tetrahedron satellite formation flying in the geosynchronous orbit using solar radiation pressure. Astrophys. Space Sci. 361. https://doi.org/10.1007/s10509-016-2732-1.

Huang, X., Yan, Y., Zhou, Y., 2015. Optimal lorentz-augmented spacecraft formation flying in elliptic orbits. Acta Astronaut. 111, 37–47. https://doi.org/10.1016/j.actaastro.2015.02.012.

Kumar, B.S., Ng, A., 2008. A Bang-Bang Control Approach to Maneuver Spacecraft in a Formation with Differential Drag. In: Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit August. Honolulu, Hawaii, pp. 1–11. doi:10.2514/6.2008-6469.

Lambert, C., Kumar, B.S., Hamel, J., Ng, A., 2012. Implementation and performance of formation flying using differential drag. Acta Astronaut. 71, 68–82. https://doi.org/10.1016/j.actaastro.2011.08.013.

Leonard, C., Hollister, W., Bergmann, E., 1989. Orbital formationkeeping with differential drag. J. Guidance, Control, Dynam. 12, 108–113. https://doi.org/10.2514/3.20374.

Mazal, L., Pérez, D., Bevilacqua, R., Curti, F., 2016. Spacecraft Rendezvous by Differential Drag Under Uncertainties. J. Guidance, Control, Dynam. 39, 1721–1733. https://doi.org/10.2514/1.G001785.

Miele, A., Venkataraman, P., 1984. Optimal trajectories for aeroassisted orbital transfer. Acta Astronaut. 11, 423–433. https://doi.org/10.1016/0094-5765(84)90083-3.

Moe, K., Moe, M.M., Wallace, S.D., 1998. Improved Satellite Drag Coefficient Calculations from Orbital Measurements of Energy Accommodation. J. Spacecr. Rock., 35, 266–272. URL: https://arc-aiaa-org.wwwproxy1.library.unsw.edu.au/doi/pdf/10.2514/2.3350. doi:10.2514/2.3350.

Naughton, D., Bedington, R., Barraclough, S., Islam, T., Griffin, D., Smith, B., Kurtz, J., Alenin, A., Vaughn, I., Ramana, A., Dimitrijevic, I., Sheng Tang, Z., Kurtsiefer, C., Ling, A., Boyce, R., 2019. Design considerations for an optical link supporting intersatellite quantum key distribution. Opt. Eng. 58, 1. https://doi.org/10.1117/1.oe.58.1.016106.

Pérez, D., Bevilacqua, R., 2013. Differential drag spacecraft rendezvous using an adaptive lyapunov control strategy. Acta Astronaut. 83, 196–207. https://doi.org/10.1016/j.actaastro.2012.09.005.

Pérez, D., Bevilacqua, R., 2014. Lyapunov-based adaptive feedback for spacecraft planar relative maneuvering via differential drag. J. Guidance, Control, Dynam. 37, 1678–1684. https://doi.org/10.2514/1.G000191.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O., 2017. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

Sentman, L.H., 1961. Comparison of the Exact and Approximate Methods for Predicting Free Molecule Aerodynamic Coefficients. ARS J. 31, 1576–1579.

Serra, R., Hoshi, K., Vasile, M., Yamakawa, H., 2018. Study of Lorentz Force on Debris with High Area-to-Mass Ratios. J. Guidance, Control, Dynam. 41, 1675–1686. https://doi.org/10.2514/1.g003317.

Seubert, C.R., Schaub, H., 2009. Tethered coulomb structures: Prospects and challenges. J. Astronaut. Sci. 57, 347–368. https://doi.org/10.1007/BF03321508.

Smirnov, G.V., Ovchinnikov, M., Guerman, A., 2007. Use of solar radiation pressure to maintain a spatial satellite formation. Acta Astronaut. 61, 724–728. https://doi.org/10.1016/j.actaastro.2007.03.009.

Smith, B., 2019. A Comprehensive Examination of Low Earth Orbit Aerodynamic Accelerations for Satellite Formation Control. Ph.D. thesis University of New Sourth Wales Canberra. doi:10.13140/RG.2.2.31390.89922.

Smith, B., Boyce, R., Brown, M., 2017a. Fast Aerodynamic Establishment of a Constellation of CubeSats. In: 7th European Conference for Aeronautics and Space Sciences. Milan. doi:10.13009/EUCASS2017-289.

Smith, B., Boyce, R., Brown, M., Garratt, M., 2017b. An Investigation into the Practicability of Differential Lift Based Spacecraft Rendezvous. J. Guidance, Control, Dynam. 40, 2682–2689. https://doi.org/10.2514/1.G002537.

Smith, B., Capon, C., Brown, M., 2019. Ionospheric Drag for Satellite Formation Control. J. Guid., Control, Dynam., 42, 2590–2599. URL: https://arc.aiaa.org/doi/pdf/10.2514/1.G004404. doi:10.2514/1.G004404.

Sutton, R.S., Barto, A.G., 1998. Introduction to reinforcement learning, volume 135. The MIT press, Cambridge, Massachusetts.

Wang, J.X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J.Z., Munos, R., Blundell, C., Kumaran, D., Botvinick, M., 2016. Learning to reinforcement learn. arXiv preprint arXiv:1611.05763.

Williams, T., Wang, Z.-S., 2002. Uses of solar radiation pressure for satellite formation flight. Int. J. Robust Nonlinear Control 12, 163–183. https://doi.org/10.1002/rnc.681.

Willis, S., Izzo, D., Hennes, D., 2016. Reinforcement learning for spacecraft maneuvering near small bodies. In: Advances in the Astronautical Sciences (pp. 1351–1368). Napa, CA volume 158. URL: http://www.esa.int/gsp/ACT/doc/MAD/pub/ACT-RPR-MAD-2016-NAPA-HoveringOnSmallBodies.pdf.

Zhang, T., Kahn, G., Levine, S., Abbeel, P., 2016) Learning deep control policies for autonomous aerial vehicles with mpc-guided policy search. In: 2016 IEEE International Conference on Robotics and Automation (ICRA) (pp. 528–535). Stockholm, Sweden. URL: https://ieeexplore.ieee.org/document/7487175. doi:10.1109/ICRA.2016.7487175.