AFRL-AFOSR-VA-TR-2022-0529

Optimal Sensor Tasking through Deep Reinforcement Learning for Space Situational Awareness

Gebre-Egziabher, Demoz
REGENTS OF THE UNIVERSITY OF MINNESOTA
200 OAK ST SE # 224
MINNEAPOLIS, MN, 55455
USA

08/11/2022
Final Technical Report

DISTRIBUTION A: Distribution approved for public release.

Air Force Research Laboratory
Air Force Office of Scientific Research
Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
20220811	Final	20180201	20220131

4. TITLE AND SUBTITLE
Optimal Sensor Tasking through Deep Reinforcement Learning for Space Situational Awareness

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
	FA9550-18-1-0115	61102F
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**

6. AUTHOR(S)
Demoz Gebre-Egziabher

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
REGENTS OF THE UNIVERSITY OF MINNESOTA
200 OAK ST SE # 224
MINNEAPOLIS, MN 55455
USA

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)
Air Force Office of Scientific Research 875 N. Randolph St. Room 3112 Arlington, VA 22203	AFRL/AFOSR RTB1	AFRL-AFOSR-VA-TR-2022-0529

12. DISTRIBUTION/AVAILABILITY STATEMENT
A Distribution Unlimited: PB Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Recent advancements in Deep Reinforcement Learning (DRL) have demonstrated groundbreaking results across a number of domains. In particular, DRL has been used to successfully develop an artificial intelligence approach that can defeat expert human GO players. The generality of DRL and the groundbreaking results motivate the exploration of these approaches for SSA applications. Deep learning approaches mimic the function of the human brain by learning nonlinear hierarchical features directly from data where each hierarchy builds in abstraction. In an end-to-end fashion, DRL approaches can be used to process data directly to “learn” a control policy from training data. Under this work, DRL approaches are developed for space situational awareness (SSA) applications. Our research is divided into five interrelated research areas; development of DRL algorithms for SSA sensor tasking application, usage of deep learning techniques for space object (SO) classification and characterization, development of optimal constellation architecture for improved Low Earth Orbit's orbital capacity, development of DRL techniques for SO intent identification and anomalies detection, and development of Koppman Operator theories for space applications.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	c. THIS PAGE		
U	U	U	UU	50

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
MICHAEL YAKES	00000000

DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Final Report for
Grant FA9550-18-1-0115

OPTIMAL SENSOR TASKING THROUGH DEEP REINFORCEMENT LEARNING FOR SPACE SITUATIONAL AWARENESS

PRINCIPAL INVESTIGATOR:
Richard Linares
Assistant Professor
Email: linaresr@mit.edu
Phone: (617) 258-6738

Demoz Gebre-Egziabher
Professor
Email: gebre@umn.edu
Phone: (612) 624-2305

Period of Performance:
February 2018 to January 2022

April 1, 2022

1 Summary: Objectives and Status of Effort

In this report, we summarize our accomplishments under Grant FA9550-18-1-0115. This project has five interrelated research areas. The first of these focuses on developing reinforcement learning approaches for Space Situational Awareness (SSA) applications. We developed Deep Reinforcement Learning (DRL) algorithms to solve the SSA sensor management problem. Ground-based optical sensors are typically restricted to a narrow field of view (FoV) and must be slew at a finite rate from object to object as they observe them. Since there are many objects that a narrow FoV sensor could choose to observe, it must algorithmically create a schedule that dictates which direction to point its sensor; a combinatorial optimization problem known as the sensor tasking problem. Deep neural networks were used to capture high-level features while overcoming the curse of dimensionality for optimal SSA sensor tasking. Under this thrust, a suite of SSA simulation environments was developed where the DRL agents can be trained to perform sensor management for ground-based and space-based optical sensors. We also developed a framework for the training of DRL agents to perform catalog maintenance for SSA with ground-based and space-based sensors while considering the long-term reward.

The second thrust looked into the usage of deep learning techniques for space objects classification and characterization. The current simplified description of space objects (SOs) in the SSA SO catalog limits the dynamic propagation model that can be used for predicting the state of the motion of SO. The SSA catalog assumes a cannonball shape with generic surface properties for all tracked SOs. The future SSA SO catalog and SSA systems need to be able to build a more detailed picture of the SO characteristics to allow for more precise propagation models. The external forces — drag and solar radiation pressure – acting on a SO can differ significantly depending on the SO's geometry and physical properties. Deep learning techniques are leveraged to extract the physical properties of the SO from traditional light curve measurements. A framework to classify and characterize space objects via light curves using deep learning was developed under this thrust.

The third thrust focuses on improving the orbital capacity for the Low Earth Orbit (LEO) regime. We looked into the generation of satellite constellation architectures that allow for closely packed orbital shells for increased orbital capacity. With the ever-increasing space activity and multiple large constellations that have been planned for the LEO regime, the orbital conjunction risk and operational burden associated with planning and coordinating collision avoidance maneuvers for active satellites have been significantly increased. With the deployment of these large constellations, it was hypothesized that a significant portion of the conjunction risk in the future will stem from active versus active cross-operator conjunction events. A more efficient satellite constellation architecture is necessary to ensure safe segregation between orbital shells while improving the orbital capacity. Under this thrust, a methodology to generate closely packed constellations for increased orbital capacity in the Low Earth Orbit (LEO) regime was developed.

Our fourth thrust looked into the usage of deep reinforcement learning techniques to infer SO intent and anomalies in real-time. Inverse Reinforcement Learning (IRL) techniques are used to determine the behavior of SOs by estimating the reward function that the SO is using for control. These IRL techniques can be used to analyze the maneuvering of SOs from observational data in real-time and predict their future maneuvers. We leveraged IRL techniques to determine the optimal reward function that a SO is using while maneuvering with random disturbances by assuming that the observed trajectories are optimal with respect to the SO's own reward function.

Our fifth thrust focuses on developing the Koopman Operator theories for space application. Space dynamical systems (celestial mechanics) can be highly nonlinear and complex with no analytical solutions. One such example is the modeling of the effects of zonal harmonics induced by non-spherical celestial bodies on the motion of a satellite. The Koopman Operator is leveraged to formulate the space dynamical systems in terms of their observable and transform this nonlinear system with a finite number of dimensions into a linear system with an infinite number of dimensions. The linear system with infinite dimensions is approximated by projecting into a subspace to provide an accurate linearized model of the complex space dynamical system. Our research thrusts have clear and direct relevance to Space Force and more broadly DoD missions in safeguarding our space assets and the space environment for the future generation.

The principal investigator for this effort is Assistant Professor Richard Linares and Professor Demoz Gebre-Egziabher. They are assisted in the conduct of this research by several post-docs and graduate research assistants. A brief summary of our research is provided in the Section 2. The individuals involved in this effort are documented in Section 4 and publications supported by this effort are listed in Section 5. In section 6, we discuss transitions of our research to commercial and government agencies.

2 Accomplishments and Technical Updates

In this section, a summary of the accomplishments and technical updates for each thrust is provided. Readers are referred to the publications listed at the end of the report for detailed developments.

2.1 Large Scale Reinforcement Learning for Sensor Tasking in Space Situational Awareness

Space Situational Awareness (SSA) involves detecting, tracking, cataloging, and identifying artificial objects or Space Objects (SO) orbiting the Earth. SSA is crucial in protecting our space assets and ensuring the availability of space for future generations. This task is becoming increasingly difficult with the ever-increasing number of artificial objects orbiting the Earth and the improvement in SSA sensing capabilities [1]. In the United States, the Space Surveillance Network (SSN) is tasked with achieving SSA by maintaining information on over 22,000 objects, 1,100 of which are active, with a collection of optical and radar sensors.

Currently, the field of SSA has progressed over the past few decades with the development of new sensors for making more detailed SO observations and algorithms for processing the observation data [2–11]. Even with the development of these new approaches, significant work must be made in the translation of SSA goals to sensor and resource allocation which is known as the Sensor Management (SM) problem.

The SM problem involves allocating a limited number of resources (satellite observers) to observe a large number of SO targets. The number of SO targets to be tracked can be a few orders of magnitude larger than the available sensing resources. The SM problem is a general challenge across many engineering applications and has been extensively studied including tracking multiple objects using surveillance cameras [12] and mobile robotic applications [13]. The sensor tasking problem has been formulated using both an information-theoretic framework [14] and the general Bayesian framework [15]. Additionally, work has been investigated for sensor tasking in the context of dynamical systems [16, 17]. In recent years, the optimal sensor tasking problem has been investigated by the control systems and robotics communities in the area of information-theoretic control [18], active sensing [19], and dual control [20]. Although the sensor tasking problem has been studied extensively, there have been very few works that address the long-term tasking problem. In addition, most tasking formations are based on very specific user-defined metrics and there has not been any work on developing general tasking approaches that can address a diverse set of objectives. Finally, most of the sensor tasking-related work focuses on linear systems and the nonlinear problem has not been addressed adequately. These challenges, among others, are addressed.

Recent advancements in Deep Reinforcement Learning (DRL) [21, 22] have demonstrated groundbreaking results across a number of domains. In particular, DRL has been used to successfully develop an artificial intelligence approach that can defeat expert human GO players [21]. The generality of DRL and the groundbreaking results motivate the exploration of these approaches for SSA applications. Here the term "deep" refers to any neural network learning approach with more than one hidden layer. Deep learning approaches mimic the function of the human brain by learning nonlinear hierarchical features directly from data where each hierarchy builds in abstraction [23]. In an end-to-end fashion, DRL approaches can be used to process data directly to "learn" a control policy from training data.

However, the training of DRL agents still suffers from sensitivity to empirical choices of hyperparameters such as model architecture, loss function, and training hyperparameters. Traditionally, hyperparameters tuning is done by doing a grid search or random search over all feasible configurations. However, due to the increase in training times and hyperparameter search space with the increase in DRL agents' depth and complexity, it is typically computationally infeasible to conduct a grid search on all possible configurations for DRL agents. In recent years, various hyperparameters tuning algorithms have been introduced such as Population Based Training (PBT) [24], HyperBand [25], Bayesian Optimization and HyperBand (BOHB) [26], and Asynchronous Successive Halving Algorithm (ASHA) [27]. These hyperparameters tuning algorithms serve to maximize the RL agents' performance while minimizing the training cost. These algorithms use information from prior training to guide the hyperparameter turning process. PBT was shown to improve the performance of state-of-the-art RL baselines across a suite of challenging reinforcement learning problems.

To this end, a suite of SSA environments was developed to train the DRL agents and evaluate their performance. Each environment shares a similar process flow as shown in Figure 1, but has minor implementation differences such as the population of target objects, action slew and settle time,

observation formulation, and minimum viewing horizon. The output for each module is listed in blue. During the SSA environment initialization, the SOs orbital elements and covariance are initialized. A grid-based observation of the environment is then generated by the observation generation module based on the initialized variables. The grid-based observation is then fed as an input to the DRL agent, which then generates a pointing direction (an integer). The action slew time is then calculated based on the pointing direction and the SOs states and covariance are propagated forward in time using an Unscented Kalman Filter (UKF) framework. SOs located within the sensor's field of view are then identified. A noisy measurement is then generated for each SO within the sensor's field of view and is then used to update the SOs states and covariance. A new observation grid is then generated using the updated states and covariance and the whole process is repeated.

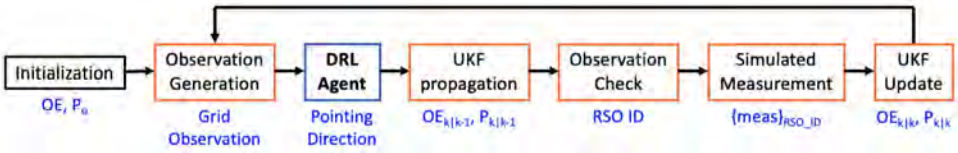


Figure 1: Process flow of the SSA environment.

The SSA environment provides functionality to simulate ground-based and space-based optical telescopes. It can simulate scenarios for a single and a collection of optical telescopes. It uses Simplified General Perturbation 4 (SGP4) and UKF for SO states and covariance propagation and update. Some of the user-controllable configurations include:

- SOs initialization: Random initialization, guided initialization, loading orbital elements from a database, uncertainties initialization, SO detection probability
- Sensor location: Ground-based (latitude, longitude, and altitude), space-based (orbital elements)
- Number of sensors: Single, multiple
- Sensor parameters: Action slew time, action settle time
- UKF noise parameters: Measurement noise, process noise

The DRL architectures studied in our work are inspired by DRL architectures used in solving Atari video games [22, 28]. All DRL agents use an actor-critic network with no shared weights, where two identical neural networks are set up and optimized concurrently. All layers use a rectified linear unit (ReLU) activation, except for the final output layer, where a soft-max activation function is used. Layer normalization is also used in between convolution layers to normalize each layer's output to a standard normal distribution. This stabilizes the hidden state dynamics and improves the performance of the DRL agents. The output of the final layer is then post-processed using an action elimination layer. The action elimination layer is formulated based on concepts presented in [29], where the action probabilities corresponding to the empty field of view grids are driven to zero. This reduces the action space to be searched by the agent and speeds up the convergence of the agent. An example of the DRL agent's architecture is shown in Figure 2.

The DRL agents are implemented using Tensorflow [30] and the Ray Tune toolbox [31]. All DRL agents are trained using proximal policy optimization with population-based training. The training of DRL agents is sensitive to the choices of training hyperparameters, such as learning rate, training batch size, and mini-batch size. Non-optimal training hyperparameters can lead to slow learning and sometimes even catastrophic failure and divergence of the DRL agent. Population-based training is used to overcome this issue and has previously been shown to improve convergence and achieve a higher final reward for a suite of challenging DRL problems [24]. Under the population-based training formulation, a population of DRL agents is trained concurrently with different training hyperparameters. After a fixed number of training intervals, the performance of the population of trained DRL agents is ranked. The training hyperparameters and neural network parameters (weight and bias parameters) for the top-performing agents are exploited to replace the lowest-performing agents. The training hyperparameters of the lowest-performing agents are either perturbed based on the hyperparameters of the top-performing agents or are re-sampled from the provided mutation sampling range. Meanwhile, the neural network parameters from the top-performing agents are transferred to the lowest-performing agents.

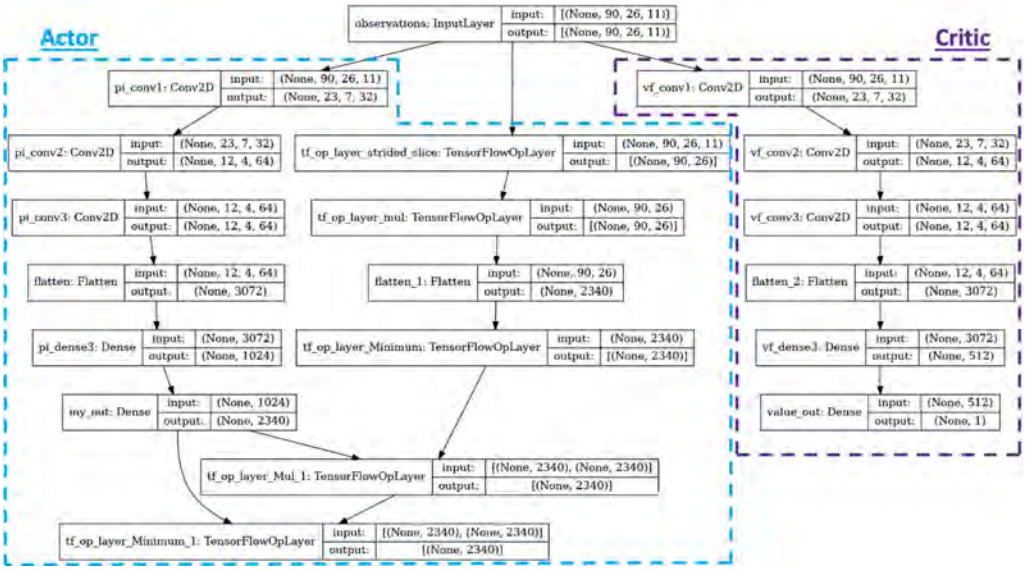


Figure 2: Example DRL architecture.

We also created two myopic policies/schedulers to benchmark the performance of the DRL agents. The first myopic scheduler uses a greedy policy that picks the SO with the largest uncertainties within the sensor's field of regard to be observed, whereas the second myopic scheduler (advanced greedy policy) scales the SO mean uncertainties with the required action slew time before picking the most rewarding SO to be observed. These myopic policies are computationally efficient and easy to implement, however, they only consider the reward gained from the current time step and do not take into account any long-term effects of their current action. The action policy for the greedy and advanced greedy policies are given by Equations (1) and (2) respectively.

$$a^* = \arg\max_a \left(tr[P^x_{i,k}]\right) \tag{1}$$

where $tr[P^x_{i,k}]$ is the trace of a posteriori state covariance for SO i at the current time step k.

$$a^* = \arg\max_a \left(tr[P^x_{i,k}] \cdot \delta t_i^{-1/m}\right) \tag{2}$$

where $tr[P^x_{i,k}]$ is the trace of a posteriori state covariance for SO i at the current time step k, δt_i is the action slew time to move from the current pointing direction to observe SO i, and m is a constant. The advanced greedy policy is evaluated with different values of m (ranging from 1 to 10) to identify the optimum scaling factor.

2.1.1 Sensor Tasking for Ground-based Optical Sensor Using Deep Reinforcement Learning

The ground-based optical sensor used in our SSA environments is modeled based on the Zimmerwald SMall Aperture Robotic Telescope (ZimSMART) with the addition of a Light Detection And Ranging (LiDAR) system to produce Cartesian coordinate measurements for all SOs within the current field of view. ZimSMART is an amateur grade 20cm survey telescope operated by the Astronomical Institute of the University of Bern, Switzerland [32]. The optical sensor parameters are shown in Table 2. The slew time is approximated such that it takes 7.7s for the first 4°, while all subsequent grid takes 4.55s. The exposure time is assumed to be 1.3s. Due to the action slew rate, the propagation time for a single time step can vary between 9s to 209.2s depending on the distance between the current pointing direction and the selected action. Two separate motors for the azimuth and elevation control of the mount are used. A high-level overview of the differences between the suite of ground-based SSA environments is shown in Table 1.

The problem is simplified by discretizing the continuous action and observation space. The field of regard of the sensor is discretized into patches with the size of the field of view of the sensor. The field

Table 1: High-Level Overview of the Different SSA Environments

	SSA-v1	**SSA-v2**	**SSA-v3**	**SSA-v4**	**SSA-v5**	**SSA-v6**	**SSA-v7**
RSO # (n)	50	50	50	100	100	100	500
RSO orbit	LEO + MEO	LEO + MEO	LEO + MEO	LEO + MEO	LEO + MEO	LEO + MEO	GEO
Propagation time	Fixed (30s)	Fixed (30s)	Fixed (30s)	Fixed (30s)	Fixed (30s)	Varies (9s -209.2s)	Varies (9s -209.2s)
Observation	Cart state and cov.	OE and cov.	OE and cov.	Grid	Grid	Grid	Grid
Observation Space	42n	42n	42n	80x16x8	80x16x11	90x19x12	90x19x11
Action	RSO ID	RSO ID	RSO ID	Pointing dir.	Pointing dir.	Pointing dir.	Pointing dir.
Action space	n	n	n	1280	1280	1710	1710
Viewing Restrictions	None	None	Alt $> 18°$	Alt $> 18°$	Alt $> 18°$	Alt $> 14°$	Alt $> 14°$

Table 2: Parameters of Optical Sensor.

Sensor Parameters	Values
Field of View	$4° \times 4°$
Minimum Viewing Horizon	$14°$
Action Slew Rate	$7.7s$/first $4°$
	$4.55s$/subsequent $4°$
Exposure time	$1.3s$

of regard spans the full range of possible azimuth angles (0 to 360 degrees), whereas the elevation range is constrained by the minimum viewing horizon and is thus limited between 14 degrees and 90 degrees. Using a sensor's field of view of 4 by 4 degrees, the field of regard is discretized into 90 discrete azimuth grids and 19 discrete elevation grids. This results in a combination of 1710 different patches of the sky that the sensor can observe, i.e. a finite action space with 1710 possible pointing directions. However, the effective azimuth viewing angle increases with the increase in elevation angle. This effect becomes larger at a higher elevation, causing overlapping of the field of view at high elevation angles. The relationship between the effective azimuth viewing angle γ with elevation angle θ is given by Equation (3).

$$\gamma = \arccos\left[\frac{\cos^2\theta - 1 + \cos 2°}{\cos^2\theta}\right] \tag{3}$$

The same discretization is used when formulating the observation array. Thus, the observation array takes the form of a three-dimensional array with the dimensions: 90 by 19 by $\boldsymbol{x}$, where the first two dimensions represent all 1710 possible pointing directions and $\boldsymbol{x}$ is the number of information layers. For our studies, 11 information layers are used and are listed in Table 3. When multiple SOs are located within the same grid, only the values corresponding to the SO with the highest uncertainties are used for layers 2 - 7.

Each row in the observation array represents a pointing azimuth, while each column represents a pointing elevation. The observation array is populated such that the current pointing direction is always in the 45^{th} row, i.e. the center row of the observation array corresponds to the current pointing azimuth. The other rows are filled accordingly based on their relative azimuth angle from the current pointing direction.

However, the large range of action slew time (9s - 209.2s) causes a large variation in the SO position while the observer is slewing. Therefore, the observation grid is partitioned into seven regions where each region uses a different propagation time based on the approximate action slew time. The grid observation array is formed using SO states from multiple time steps, offset by 30s each (15s, 45s, 75s, 105s, 135s, 165s, and 195s) as shown in Figure 3. The white cell corresponds to the current pointing direction and the regions are color-coded by the propagation time used to populate the data for that region. The propagation time step increases as a function of the action slew angle. This ensures that the observation array accurately reflects what the agent is expected to observe while keeping the computational time low.

Table 3: Observation information for each grid in the observation.

Layer	Data
1	Number of SOs in grid
2	Elevation fraction location of SO in grid
3	Azimuth fraction location of SO in grid
4	Range of SO in grid
5	Rate of change in elevation of SO in grid
6	Rate of change in azimuth of SO in grid
7	Rate of change in range of SO in grid
8	Max of SO's trace of covariance in grid
9	Sum of SO's trace of covariance in grid
10	Mean of SO's trace of covariance in grid
11	Current pointing direction (boolean)

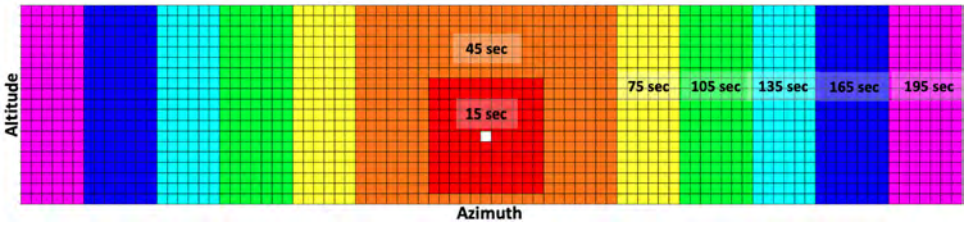


Figure 3: Discretized observation space and SO propagation time is used to account for action slew time.

DRL agents with different architectures are trained with the in-house SSA environment on a population of 100 randomly initialized GEO SOs using proximal policy optimization [33] and population-based training [24]. Both convolution neural network based (*CNN*) and fully connected layer based (*Dense*) DRL agents are studied in this work. The DRL architectures studied in this work are summarized in Table 4 and are inspired by DRL architectures used in solving Atari video games [22, 28]. All DRL agents use an actor-critic network with no shared weights, where two identical neural networks are set up and optimized concurrently. *Conv2d(16,8,4)* corresponds to using a kernel with 16 filters and a sliding window of 8×8 with a 4×4 stride. On the other hand, *FCL(256)* corresponds to a fully connected layer with 256 nodes. All layers use a rectified linear unit (ReLU) activation, except for the final output layer, where a soft-max activation function is used. Layer normalization is also used in between convolution layers to normalize each layer's output to a standard normal distribution. This stabilizes the hidden state dynamics and improves the performance of the DRL agents. The output of the final layer is then post-processed using an action elimination layer. The action elimination layer is formulated based on concepts presented in [29], where the action probabilities corresponding to the empty field of view grids are driven to zero. This reduces the action space to be searched by the agent and speeds up the convergence of the agent.

The DRL agents are optimized using the reward function defined by Equation (4) that is constructed based on the objective function discussed in Blackman (1986) [34]. The action reward is allocated based on the Fisher Information Gain using the Kalman filter equation and a utility function. The reduction in position uncertainties is used to calculate the reward of an observation.

$$rew = \arg\max_{a_k} \left[tr\left(P_{k|k-1}^{(a_k)}\right) - tr\left(P_{k|k}^{(a_k)}\right) \right] \tag{4}$$

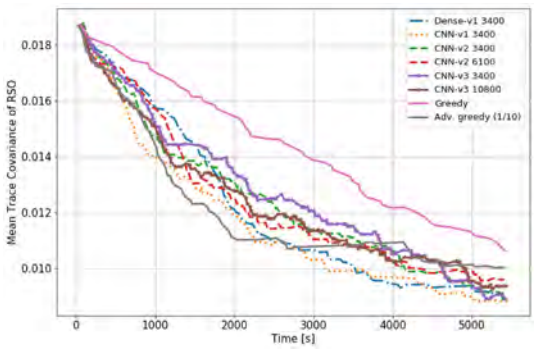
where $P_{k|k-1}^{(a_k)}$ and $P_{k|k}^{(a_k)}$ are the a priori and a posteriori covariance of SO a_k respectively, and a_k is the set of SOs within the sensor's field of view. The DRL agents are trained using a population size of 20 concurrent agents with 100 near GEO SOs until convergence in the mean episodic reward.

Two different metrics are used to quantify the agent's performance: the number of unique SOs observed and mean SO uncertainties. The trained DRL agents are able to outperform both myopic policies in both performance criteria. Figures 4(a) and 4(b) show the mean SO covariance and number of unique SOs observed over a single-seeded rollout respectively.

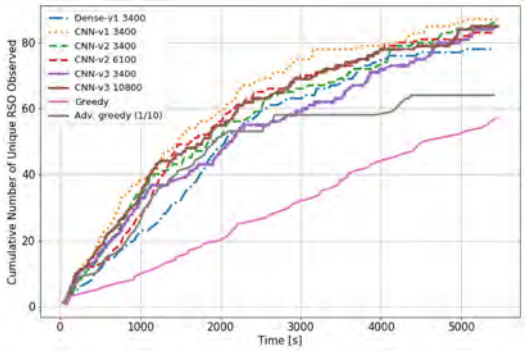
With 3400 training iterations, all DRL agents were able to outperform the myopic policies in terms of final mean trace covariance at the end of the 90-minute observation window. The advanced greedy

Table 4: CNN based neural network architectures.

Architecture	Input Size	Output Size	Layers	Number of Parameters
Mnih et al. (2013) [28]	$(84 \times 84 \times 4)$	$4 \sim 18$	Conv2d(16,8,4), Conv2d(32,4,2), FCL(256), FCL(Output Size)	1,004,852 $\sim$ 1,008,450
Mnih et al. (2015) [22]	$(84 \times 84 \times 4)$	$4 \sim 18$	Conv2d(32,8,4), Conv2d(64,4,2), Conv2d(64,3,1), FCL(512), FCL(Output Size)	4,045,476 $\sim$ 4,052,658
Dense-v1	$(90 \times 19 \times 12)$	1710	Conv2d(3,1,1), FCL(768), FCL(768), FCL(Output Size)	5,846,229
CNN-v1	$(90 \times 19 \times 12)$	1710	Conv2d(48,8,4), Conv2d(80,4,2), Conv2d(80,3,1), FCL(2048), FCL(Output Size)	9,560,190
CNN-v2	$(90 \times 19 \times 12)$	1710	Conv2d(32,8,4), Conv2d(64,4,2), Conv2d(64,3,1), FCL(1024), FCL(Output Size)	4,207,438
CNN-v3	$(90 \times 19 \times 12)$	1710	Conv2d(16,8,4), Conv2d(32,4,2), Conv2d(32,3,1), FCL(700), FCL(Output Size)	2,035,586



(a) Mean trace covariance of all SOs for one rollout episode (lower is better).

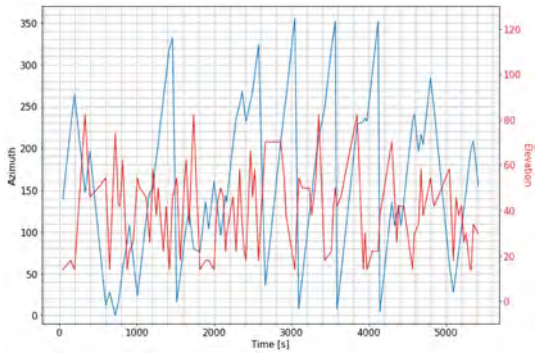


(b) Cumulative number of unique SOs observed for one rollout episode (higher is better).

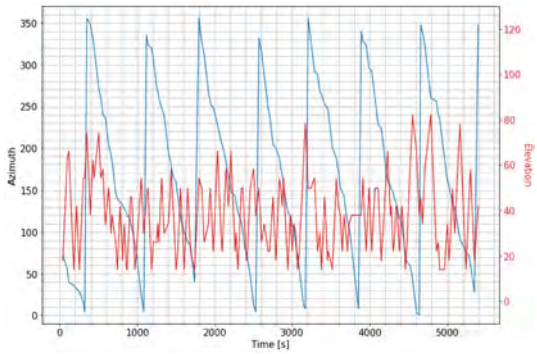
Figure 4: Performance of DRL agents with different architectures trained using 100 GEO SOs, evaluated on a single-seeded rollout of 90 minutes. Note: *CNN-v0 3400* represents a CNN based DRL agent trained for 3400 training iterations.

policy initially outperformed some of the DRL agents due to its myopic nature, but towards the end of the observation window, the DRL agents benefited from their long-term planning capabilities and resulted in a lower final mean trace covariance. The myopic policies have a lower performance in terms of the number of unique SO observed compared to the DRL agents as they are not formulated for this performance criteria. Here, it is shown that through the usage of DRL agents, it is feasible to have a single policy that optimizes multiple performance criteria or operator objectives. The performance of the DRL agents could be further improved with additional training iterations as shown by the *CNN-v2*

and *CNN-v3* DRL agents, where their final mean SO uncertainties are further reduced with additional training.



(a) Azimuth and elevation pointing direction selected by the *Dense-v1 3400* DRL agent.



(b) Azimuth and elevation pointing direction selected by the *CNN-v1 3400* DRL agent.

Figure 5: Azimuth and elevation pointing directions selected by DRL agent.

Figures 5(a) and 5(b) show the actions selected by the *Dense-v1* and *CNN-v1* DRL agents over the same seeded rollout of 90 minutes respectively. All CNN based DRL agents show a similar trend in their selected pointing directions, where they sweep the field of regard systematically and only move in a consistent Azimuth direction. The *Dense-v1* DRL agent that only uses fully connected layers shows an erratic behavior in its pointing direction. CNN based DRL agents can better exploit the spatial relationship between the SOs through the use of convolution layers, thus allowing them to plan out a more efficient scan path.

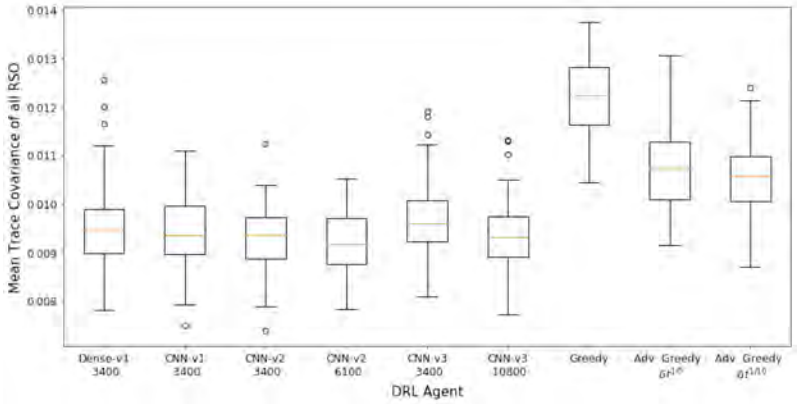
After training convergence, 100 Monte Carlo simulations are also carried out to obtain a better statistical performance comparison between the trained DRL agents and the two myopic policies. Each Monte Carlo run uses a population of 100 GEO SOs with random orbital elements and random initial covariance. This ensures that a diverse population of GEO SOs and scenarios are represented in the performance evaluation. The box plot in Figures 6(a) and 6(b) show the aggregate performance of the DRL agents and myopic policies. All DRL agents outperformed the myopic policies with the *CNN-v2* DRL agent achieving the best performance for both performance criteria. All CNN based DRL agents have somewhat similar performance. On average, the CNN based DRL agent observed more than 85% of the 100 SOs in the SSA environment. The *Dense-v1* DRL agent, based on fully connected layers, has the worst performance among all DRL agents. Among the myopic policies, the advanced greedy policies performed better than the naive greedy policy as they take the action slew time into consideration when selecting which pointing direction to observe.

We also evaluated the robustness of the DRL agents to changes in SO orbital regime, number of SOs to be tracked, length of the observation window, and observer location. The DRL agents have shown to be robust to all these changes and still maintain a superior performance compared to both myopic policies.

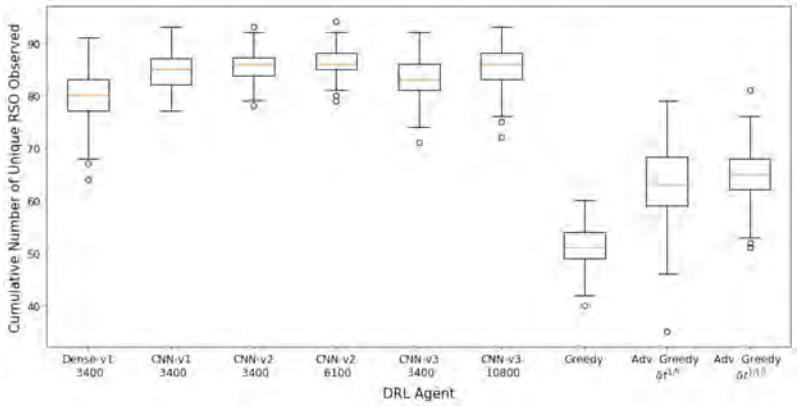
In conclusion, we have successfully developed and trained a DRL-based scheduler for ground-based narrow-FOV sensors to efficiently observe SOs orbiting overhead using deep reinforcement learning. Our DRL agents have shown to be able to outperform myopic policies—those in which only the benefits of a small number of estimated, future observations are considered—across several figures of merit, including SO covariance and the number of unique SOs observed during the study period, and are robust to variation in the environment.

2.1.2 Sensor Tasking for Space-based Optical Sensor Using Deep Reinforcement Learning

We subsequently adapted our framework to be applicable to space-based SSA sensors, where the sensor is in a constant motion in Earth's low Earth orbit (LEO). Space-based sensors in any orbital regime offer clearer, more frequent, and more diverse space object observations than ground-based sensors. When a sensor is in orbit, its instantaneous FOR is larger than that of a ground ground-based sensor, as it can observe objects with negative elevation angles, as shown in Figure 7. Due to its position above the Earth's atmosphere, most portions of the sky can be observed such that the line of sight does not intersect the Earth or its atmosphere. Due to its orbital velocity, a space-based sensor's aggregate FOR



(a) Distribution of the final mean trace covariance at the end of the observation window (lower is better).



(b) Distribution of the cumulative number of unique SOs observed at the end of the observation window (higher is better).

Figure 6: Aggregate performance of the DRL agents with 100 randomly initialized GEO SOs, evaluated over 100 seeded rollouts of 90 minutes each. Note: *CNN-v0 3400* represents a CNN-based DRL agent trained for 3400 training iterations.

over the course of one orbital period is the entire sky. With these advantages at play, space-based sensors allow SSA operators to collect more observations of more space objects at shorter timescales.

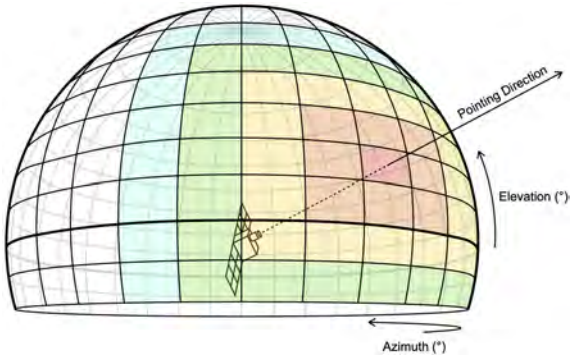


Figure 7: Simplified geometry of the nominal field of regard for the space-based sensor, which is used to generate the observational array for the trained agent

The space-based SSA environment shares a similar architecture to the SSA environment used for the ground-based SSA environment. The orbit of the space-based sensor is modeled based on Sapphire's orbit with the orbital elements (OEs) described in Table 5.

The FOR for the space-based sensor evaluated as part of this study spans the full range of possible

Table 5: Orbital elements of the Sapphire space-based sensor on August 4, 2021

Orbital Element	Value
Inclination (°)	98.4446
Right ascension of ascending node (°)	55.3814
Argument of perigee (°)	133.3296
Eccentricity	0.0012220
Mean motion (revolutions/day)	14.34248711

azimuth angles (0 to 360°), but only a fraction of all possible elevation angles (-14 to 90° as opposed to -90 to 90°), due to the Earth limb exclusion zone. The -14° elevation limit means a sensor in a circular orbit at 500 km altitude can point 14° away from its velocity vector in the radial direction, towards Earth. The FOR is discretized into non-overlapping FOVs, such that each FOV spans 4° in both azimuth and elevation. This discretization simplifies the continuous action space, by representing it as a discrete space, resulting in a finite number of pointing directions. This also results in a fixed observation space regardless of the number of SOs in the SSA environment. Under this discretization, the FOR is fully covered by a spatial grid with 90 discrete azimuth angles and 26 discrete elevation angles for a total of 2,340 possible pointing directions. The same observation array formulation used in the ground-based SSA environment is used for the space-based SSA environment.

Several neural network formulations were explored for the space-based DRL agent based on the actor-critic neural network architectures used in the ground-based DRL agent. The information on the explored neural network designs is listed in Table 6. For all neural network layers, zero padding is used

Table 6: Comparison of explored neural network architectures

Architecture	Input Size	Output Size	Actor Model	Critic Model
`CNN_v1`	$90 \times 26 \times 11$	2340	`Conv2d(32,8,4)`, `Conv2d(64,4,2)`, `Conv2d(64,3,1)`, `FCL(1024)`, `FCL(2340)`	`Conv2d(32,8,4)`, `Conv2d(64,4,2)`, `Conv2d(64,3,1)`, `FCL(512)`, `FCL(1)`
`CNN_v2`	$90 \times 26 \times 11$	2340	`Conv2d(16,8,4)`, `Conv2d(32,4,2)`, `Conv2d(32,3,1)`, `FCL(700)`, `FCL(2340)`	`Conv2d(16,8,4)`, `Conv2d(32,4,2)`, `Conv2d(32,3,1)`, `FCL(350)`, `FCL(1)`
`CNN_v3`	$90 \times 26 \times 11$	2340	`Conv2d(48,8,4)`, `Conv2d(80,4,2)`, `Conv2d(80,3,1)`, `FCL(2048)`, `FCL(2340)`	`Conv2d(48,8,4)`, `Conv2d(80,4,2)`, `Conv2d(80,3,1)`, `FCL(1024)`, `FCL(1)`
`CNN_v4`	$90 \times 26 \times 11$	2340	`Conv2d(48,8,4)`, `Conv2d(80,4,2)`, `FCL(2048)`, `FCL(2340)`	`Conv2d(48,8,4)`, `Conv2d(80,4,2)`, `FCL(1024)`, `FCL(1)`

to avoid losing information at the boundaries of the data. The rectified linear unit (ReLU) activation

function is used for all layers except the last fully connected layer of the actor and critic models. The actor model also consists of an action-masking function that penalizes actions that do not contain any SOs. The action-masking function encourages the action policy to focus on choosing actions that lead to more significant rewards.

Two reward functions are explored in this work. The first reward function `rew1` is based on the time-discounted maximum trace reduction. The second reward function `rew2` is simply the first reward without the time-discount factor. These rewards $R[t]$ at timestep k are shown in Equations 5 and 6, respectively.

$$R_{\texttt{rew1}}[k] = \frac{\underset{a_k}{\text{argmax}} \left[tr\left(P_{k|k-1}^{(a_k)}\right) - tr\left(P_{k|k}^{(a_k)}\right)\right]}{T[k] - T[k-1]} \tag{5}$$

$$R_{\texttt{rew2}}[k] = \underset{a_k}{\text{argmax}} \left[tr\left(P_{k|k-1}^{(a_k)}\right) - tr\left(P_{k|k}^{(a_k)}\right)\right] \tag{6}$$

where $tr\left(P_{k|k-1}^{(a_k)}\right)$ is the trace of the covariance for SOs within the FOV selected by action a_k and $T[k]$ is the time at timestep k.

The neural network architectures outlined in Table 6 were trained on an SSA environment with 400 near-GEO-altitude SOs using reward function `rew1`. The trained DRL agents were evaluated over 100 Monte Carlo simulations. Two different metrics are used to quantify the agent's performance: the number of unique SOs observed and mean SO uncertainties. Their performances are shown in Figure 8. The DRL agent `CNN_v1` outperformed all other DRL agents and both myopic policies across both figures of merit. The DRL agent `CNN_v2` has a smaller, narrower network than that of `CNN_v1` which limits its expressiveness. Thus it is not able to fully capture the physics behind the high-dimensional environment.

The DRL agent `CNN_v3` has a larger, wider network than that of `CNN_v1` but is more susceptible to over-fitting and convergence to local minima or saddle points and thus resulting in a sub-par performance. The performance of the DRL agent `CNN_v3` could be further improved by introducing additional regularization into the neural network architecture and training using a larger population. The DRL agent `CNN_v4` has a shallower network than `CNN_v3` in order to reduce the effects of data over-fitting, however, the issue was not fully eliminated and the agent would benefit from additional regularization.

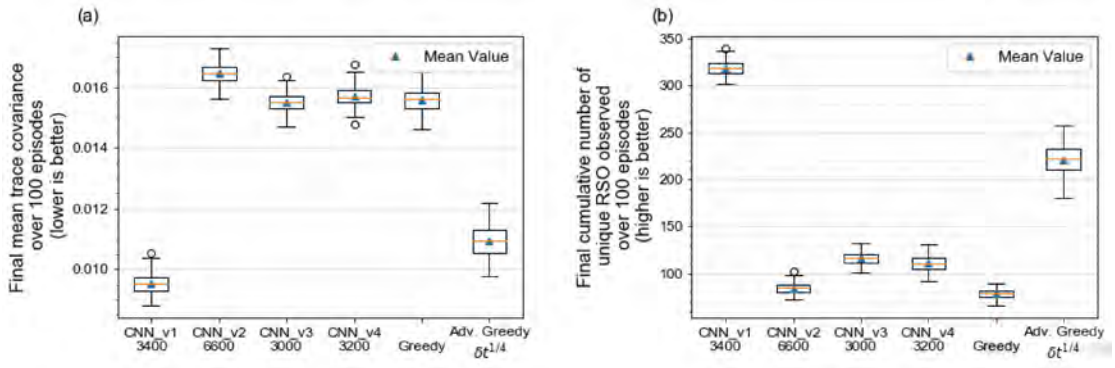


Figure 8: Comparison of results for agents trained with different neural network architectures; (a) statistics of final mean trace covariance over 100 Monte Carlo episodes; (b) statistics of the final cumulative number of unique SO observed over 100 Monte Carlo episodes

We also evaluated the robustness of the DRL agents to changes in the number of SOs, sensor properties (action slew time, dwell time, and settle time), and sensor orbital parameters. The DRL agents showed robustness toward these parameters and still outperformed the myopic policies for all scenarios. We then applied the DRL agent to a multi-agent environment, where a collection of three space-based optical sensors work in tandem. Again, the DRL agents are able to outperform the myopic policies in both performance criteria.

In our work related to sensor tasking for space-based SSA sensors to monitor SOs in the extended geosynchronous orbital regime (GEO) regime, we showed that a DRL-enabled solution to the sensor tasking problem for a space-based SSA can be trained to perform well compared to baseline myopic policies. The DRL agents are robust to changes in sensor properties, the number of targets in the environment, and the orbital plane of the space-based sensors. We also show that a set of space-based

sensors can perform well using the single-sensor agent in a parallel manner, allowing for flexibility in the multi-agent formulation. The agent can be also trained in one LEO orbit and then placed in another without loss of performance—this robustness allows for ease of use in varying scenarios for simulations as well as live sensor-tasking of multiple space-based sensors.

2.2 Space Object Classification via Light Curve Measurements Using Deep Learning

Researchers working on SSA are interested in providing a detailed understanding of SO population behavior, which must go beyond the currently SO catalog comprising simplified SO characteristics, such as solar radiation pressure and drag coefficients. To provide a more realistic and reliable understanding of the SO dynamics, future catalogs must include detailed SO characteristics (e.g., shape and state of motion). The latter can be employed in dynamical propagation models to predict SO trajectory and behavior accurately.

Developing a detailed understanding of the SO population is a fundamental goal of SSA. Such a simplified description limits the dynamic propagation model used for predicting the state of motion of SO to models that assume cannonball shapes and generic surface properties. The future SO catalog and SSA systems will have to be capable of building a detailed picture of SO characteristics. Traditional measurement sources for SO tracking, such as radar and optical, provide information on SO characteristics.

Optical sensors are generally employed to track near-geosynchronous SO. Such sensors provide both astrometric and photometric measurements. Consequently, SO properties can be estimated from astrometry (e.g., trajectories) and photometric data (e.g., shape and state of motion). More specifically, light curves, i.e., the flux of photons across a wavelength reflected by the SO and collected by optical sensors, play an important role in determining the SO attitude and state of motion. Indeed, attitude estimation and extraction of other characteristics using light curve data have been demonstrated in Ref. [7,11,35–37].

Traditional measurement sources for SO tracking (e.g. radar and/or optical measurements) have been shown to be sensitive to shape [4, 11], attitude [5, 8, 11], angular velocity [6], and surface parameters [7, 38]. A literature review shows that recent advancements have been made to estimate SO properties. Such techniques heavily rely on estimation theory and include the development of multiple model [10, 11], nonlinear state estimation [5, 6, 8], and full Bayesian inversion [9] approaches for SO characterization. Although grounded in a solid theoretical background, the above-mentioned methods tend to be computationally expensive. New techniques are sought that can provide a higher degree of accuracy, computational efficiency, and reliability.

Generally, classifying SO is a challenging task. State-of-the-art methods rely on well-established physical models that are embedded in an inversion scheme capable of processing the data and estimating the model parameters. For example, Reference [11] used a Multiple Model Adaptive Estimation classification approach to model the dynamics and physics, estimate relevant parameters, and finally classify SOs. Although such a method is one of the most promising available in the literature, the inversion process requires the estimation of a large number of parameters. As a result, the computational burden is significant and may not be practical for a catalog comprising a large number of objects. Here, we are interested in exploring a data-driven classification approach that employs both simulated and real data to learn the functional relationship. between observed light curves and SO class.

The Convolutions Neural Network (CNN) has achieved remarkable performance on image processing tasks. Examples include 1) object classification [39], 2) scene classification [40], and 3) video classification [41]. Importantly, the key enabling factor for the success of CNN is the development of techniques that can optimize large-scale networks, comprising tens of millions of parameters, as well as massive labeled datasets. Inspired by these results, our work studies the classification of observational data generated by both simulated and real dynamical systems. The dynamical system investigated here is the rotational dynamics of SOs. The physical attributes of the SOs, such as shape and mass distribution, are also included in the classification process. The challenging aspect of the application of CNNs to physical dynamical systems is the generation of labeled training data. In a dual-fold fashion, we first use physical models to simulate observations by sampling randomly from the distribution of physical attributes and dynamic states. Light curve measurements are used as inputs, and classes of the SOs are used as outputs for training the CNN approach. The 1D-CNN then learns convolutional kernels that look for characteristic features in the light curve data. As opposed to manually specified features, the features are adaptively learned, given the training data. Subsequently, available and labeled real light curves are employed to train a specified CNN architecture directly.

We explored the usage of Convolutional Neural Network (CNN) with two machine learning techniques that represent somewhat the state of the art of automatic classification techniques. The first method is called Bagged Trees, and it is an ensemble method. The idea behind ensemble methods is to combine many weak learners into a highly accurate single ensemble algorithm. Thus, we can consider a classification ensemble as a machine learning model comprising a weighted combination of many individual classification algorithms. We considered a type of ensemble algorithm called Bagging (where Bag stands for Bootstrap. aggregation [42]) combined with a basic decision tree algorithm [43]. The bagging process on a decision tree works by generating many bootstraps – replicas of the training dataset– and then growing decision trees on such replicas. Each of the bootstrap replicas is obtained by randomly choosing N observations and considering N replacements (where N is the training set size). The method works by simply training each of the individual weak learners on resampled versions of the training set. The overall response of the model is obtained by average prediction over the ensemble of individual learners. The second set of techniques is the well-known Support Vector Machines (SVM, [44]) adapted for the multi-class case. SVM generally classify data by finding the hyperplane that separates two classes by the largest margin. Non-linear transformations are generally employed when the classes are non-separable by a simple (linear) hyperplane. In this case, a variety of kernels are available to execute the transformation. The data-driven deep learning approach is compared with more conventional machine learning techniques (e.g., random forest [45] and SVM [44]) to show that hierarchical feature learning is the key to building successful discriminative models.

There are several models used for simulating light curve measurements in literature and these models differ in the physics that they represent and their level of complexity, but for SOs applications, the ability to model specular reflection and complex shapes while converting energy is desirable. The Ashikhmin-Shirley [46] (AS) model has all the desirable properties while producing realistic SO light curves. This model is based on the bidirectional reflectance distribution function (BRDF), which models light distribution scattered from the surface due to the incident light. The BRDF at any point on the surface is a function of two directions, the direction from which the light source originates and the direction from which the scattered light leaves the observed surface.

The labeled training data are generated using the AS model with the model parameters being sampled from a predefined distribution. We considered four categories, i.e., fragments, rocket bodies, regular polygon prisms, and rectangular cuboids. The SO parameter models associated with shape and surface are randomly generated out of a uniform distribution. Importantly, the regular polygon prisms are then further divided into equilateral triangular prisms, square prisms, and regular hexagonal prisms. The regular polygon prisms are prisms whose ends (i.e. top. and bottom) are regular shapes. The shape of a regular polygon prism is defined by the number of sides n, side length s, and height h. These parameters are sampled from the following distribution:

$$h_{regular} = (h_{min} + 0.01) + (h_{max} - h_{min} - 0.01)\nu[0, 1] \tag{7}$$

$$s_{regular} = (s_{min} + 0.01) + (s_{max} - s_{min} - 0.01)\nu[0, 1] \tag{8}$$

The models are generated by sampling side lengths and heights from a uniform distribution on the interval [0.01, 5] m. For the regular polygon prisms, the number of sides is also selected randomly on the interval [3, 6], with all instances of 5 sides being set to 4 as pentagonal prism models are not included. In addition to the model geometry, the material properties also need to be defined. For each model, all facets are assumed to have the following: $R_{spec} = 0.7$, R_{diff} =0.3, η =0.5. The Phong parameters n_u and n_v are each taken to be equal to 1000 for all facets of every model. The mass of the SO is randomly sampled using the following

$$m_{so} = m_{min} + (m_{max} - m_{min})\nu[0, 1] \tag{9}$$

The rocket body models are generated using octant triangulation of a sphere, which divided the surface of a sphere into N facet normal. Then the rocket body models are generated by connecting two hemisphere ends of radius r with a cylinder of height l. This model is not exact for all rocket bodies but is close enough to approximate the types of light curves seen for rocket bodies. Finally, the fragment shapes use the cuboid model but with much small aspect ratios than payload shapes. Figure 9 shows the training data generated using the process discussed above where the labels are shown using colored data points.

Real light curve observations taken from the Multichannel Monitoring Telescope were also used as our training data [47]. This data source is publicly available through astroguard.ru. For this work, the training data was developed using segments of 500 measurement samples taken from the MMT dataset for objects with TLE information. Their TLE numbers and TLE names label the objects in the MMT

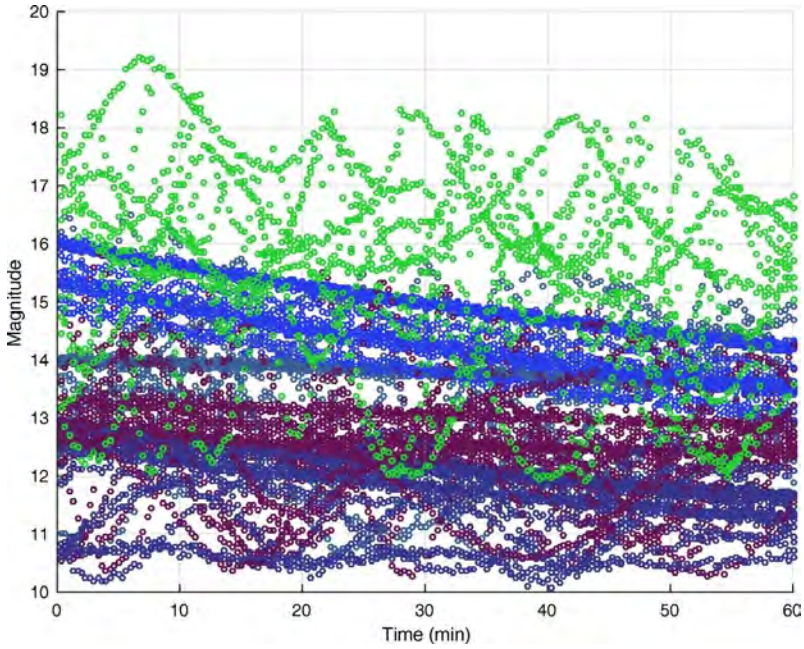


Figure 9: Labeled simulated training data where each label is given a unique color.

dataset, and from the TLE names, class information can be extracted. Three classes are used in this work, and these classes are Debris, Rocket Bodies, and Satellite. Figure 10 shows some representative examples of the MMT data used for training. From this figure, a clear difference can be seen between Debris, Rocket Bodies, and Satellite classes.

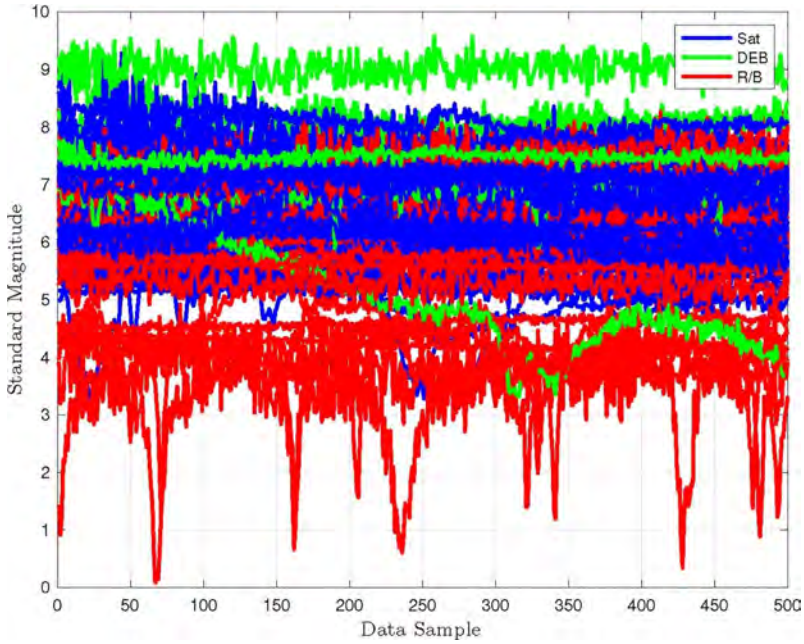


Figure 10: Labeled real training data from the MMT database, where each label is given a unique color.

A CNN is trained to map the measurement vector, $\mathbf{x}$, to classes, $\mathbf{y}$, using a set of training examples. The vector $\mathbf{y}$ is the class label for each light curve input $\mathbf{x}$. In this work, the classes used are rocket bodies, payloads, and debris. For example, $\mathbf{y}$ in this case is a 3×1 vector and the elements represent the probability that the input light curve belongs to the rocket bodies, payloads, and debris classes, respectively. Then $\mathrm{y}=[0.95,0.05,0]^T$ indicates that the input light curve belongs to the rocket bodies

class with probability 0.95, to the payload class with probability 0.05, and debris with probability 0. Since the labeled training set provides supervised examples, it has labels that indicate probability 1 for the class of the input light curve. However, the neural network may have uncertainty in the class output and therefore does not typically have output labels of probability 1.

The CNN designed for this work to learn the functional relationship between measurements and SO classes consists of 1D convolutional layers with rectified linear unit (ReLU) activation, dropout, max-pooling, and two fully connected layers with ReLU activation as shown in Figure 11. The output layer uses the softmax function to map to classification states. The cost function used for this work is the

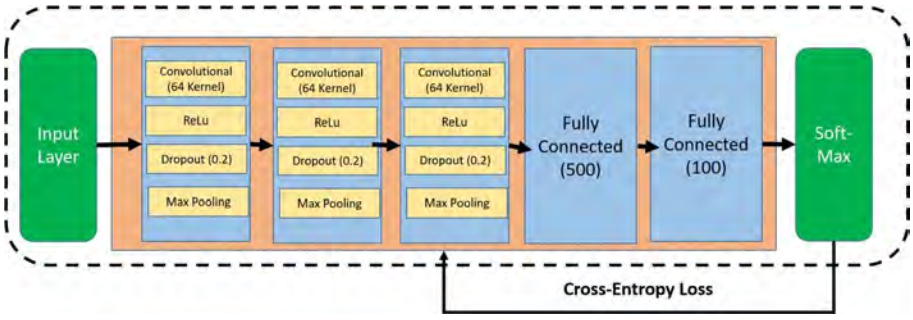


Figure 11: CNN network architecture.

cross-entropy loss. This loss function minimizes the cross-entropy loss between training outputs and the CNN outputs. The CNN classification approach is trained by stochastic gradient descent by minimizing the cross-entropy loss from the outputs compared to the labeled data.

Figure 12 shows the results of the training process on the simulated data. The CNN has been trained on 8000 samples comprising the training set and tested on 2000 samples during the training process. Figure 12a shows the behavior of the cross-entropy loss as a function of the epoch. Figure 12b shows the model accuracy as a function of the epoch. Both test and training sets have similar accuracy results. We report that the CNN exhibits an accuracy of 97.83% on the test set. Training time is reported to be 2000 sec.

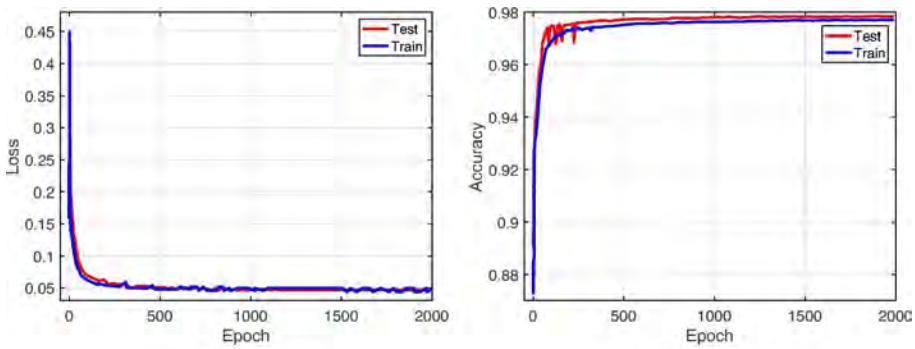
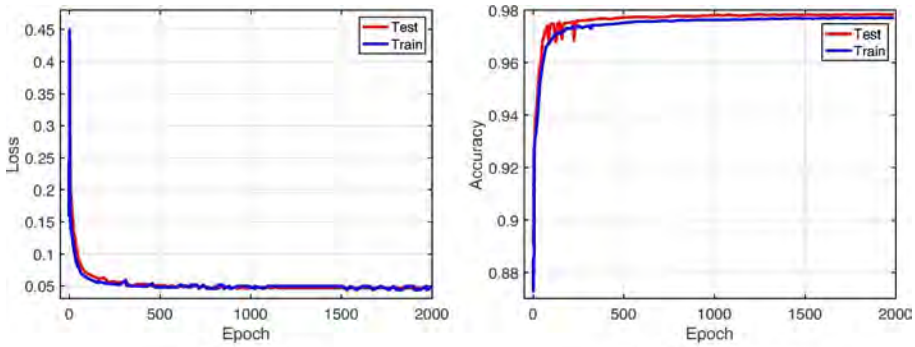


Figure 12: CNN classifications result on simulated data.

On the other hand, Figure 13 shows the results of the training process on the real light curve data taken from the MMT database. Figure 13a shows the behavior of the cross-entropy loss as a function of the epoch. Figure 13b shows the model accuracy as a function of the epoch. The real dataset is more comprehensive as it accounts for classes of objects with different observing conditions and different orbits (i.e., GEO, LEO, and MEO). Thus, it is expected that the separation boundaries between classes are highly non-linear. Here, we show that accuracy over the training set and test set is markedly different. At the end of the training, accuracy on the sequence of mini-batches is as large as 87.5%. Conversely, the final accuracy on the test set is 75.4%. Training time is about 4000 sec and a single GPU with 1500 processors.

We then compared the performance of the CNN to conventional state-of-the-art machine learning methods (e.g., bagged trees and SVM) on classifying SO based on the observed light curve. We also



(a) CNN Loss (b) CNN Accuracy

Figure 13: CNN classifications result on real data from the MMT database.

utilized triplet-distributed Stochastic Neighbor Embedding (t-SNE) to help us visualize the underlying structures of our simulated and real light curve dataset. For the simulated light curve data, we were able to use a 2-D embedding to represent the data. We sampled 3600 vectors (i.e., 400 per class) and processed the sample via the t-SNE algorithm for four selected distances, i.e., Mahalanobis, cosine, Chebyshev, and Euclidean. The resulting 2-D embedding is visualized in Fig. 14a, b, c, and d. The structure of the data and clustering is readily apparent in all cases but the one employing the Mahalanobis distance. All classes seem to be fairly separated in the 2-D embedding, which reflects the clustering of the 182-dimensional space of the original light curves. The separation between the different classes may be responsible for the high performances achieved with a non-deep machine learning technique (e.g., bagged decision trees), which is able to find the decision boundaries between the different classes easily. Although the deep CNN achieves similar performances, the deep architecture does not yield any significant advantage for the case generated by the simulated data.

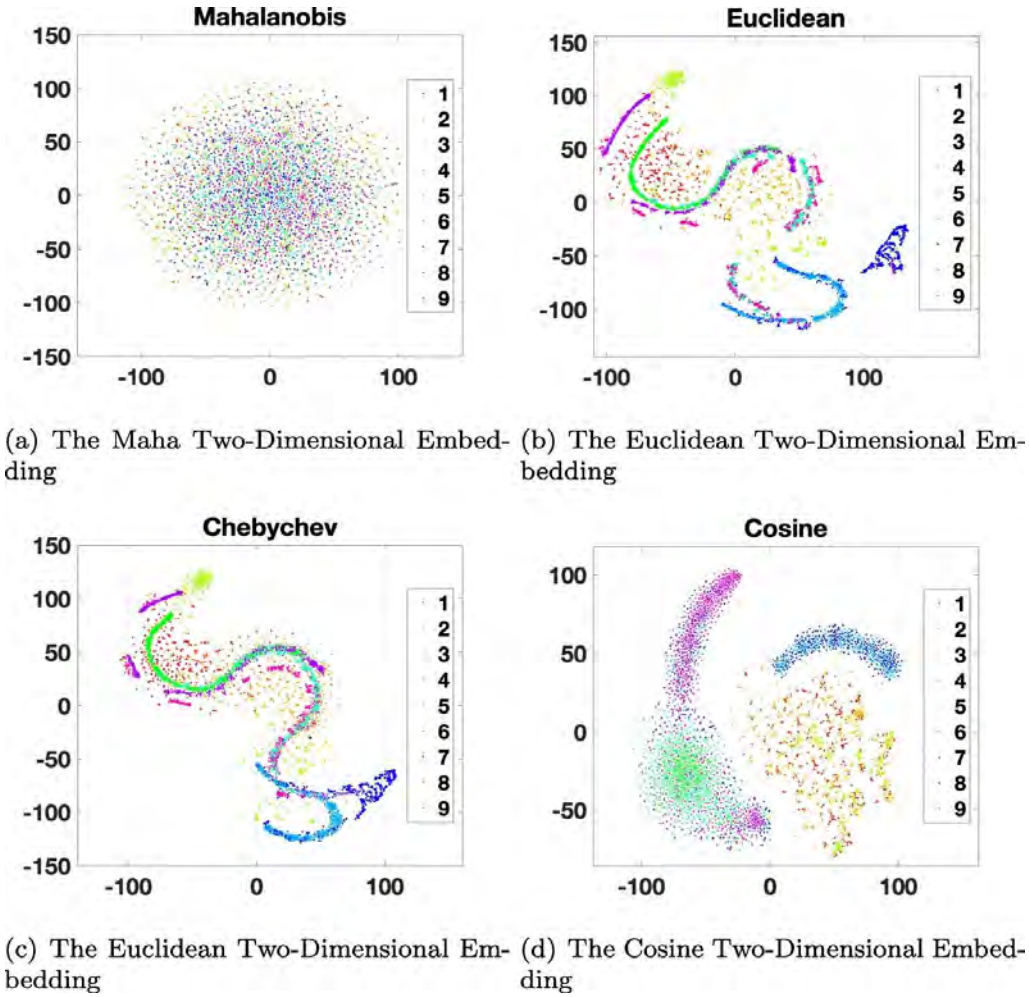
For the real light curve data from the MMT dataset, we were not able to obtain a useful 2D embedding to represent the dataset due to the higher dimensionality of light curve data (vector dimension of 500 for real light curve data compared to 182 for simulated light curve data). Hence, 3D embeddings were used. We sampled 1200 vectors (i.e., 400 per class) and processed the sample via the t-SNE algorithm for four selected distances, i.e., Mahalanobis, Cosine, Chebyshev, and Euclidean. The resulting 3-D embedding is visualized in Fig. 15a, b, c, and d. As in the previous case, the structure of the data and the corresponding clustering in 3-D is readily apparent in all cases but the one employing the Mahalanobis distance. Here the clustering of data in the 3D embedding is much more complex, and decision boundaries are not well defined as in the previous case. As seen in the previous sections, the non-deep machine learning techniques (i.e., bagged decision trees and SVM) dramatically underperform the deep CNN architecture. Due to the many-layer configuration, the CNN can learn complex decision boundaries and discriminate between the three classes in a much more efficient fashion.

From our work, it was shown that CNNs are highly accurate classifiers whenever trained on simulated data, yielding 98% accuracy for the selected test set. However, whenever trained on simulated light curves, the CNN tends to lose the advantage over more conventional state-of-the-art machine learning methods (e.g., bagged trees, SVM). Nevertheless, the CNN significantly outperforms the other techniques whenever real data are considered. Indeed, on real light curve test sets, CNNs achieve 75% accuracy, whereas bagged trees report an accuracy of 62% and SVM an accuracy of 44%.

2.3 Space Constellation for Increased Orbital Capacity

Very large satellite constellations totaling 100,000 satellites or more have been proposed for Low Earth Orbit (LEO), including nearly 40,000 proposed to the Federal Communications Commission (FCC) last November alone. The nation of Rwanda, on behalf of the company E-Space, recently submitted an International Telecommunications Union (ITU) filing for a single constellation consisting of more than 300,000 satellites across 27 shells. Not all of these constellations will make it to orbit and fewer still will be completed. Nevertheless, this more than order-of-magnitude increase in the active satellite population will necessitate numerous technical, operational, and policy changes to ensure the long-term sustainability of the space environment in the face of denser operations.

Worryingly, multiple large constellations have been proposed with limited orbital separation or over-



(a) The Maha Two-Dimensional Embedding

(b) The Euclidean Two-Dimensional Embedding

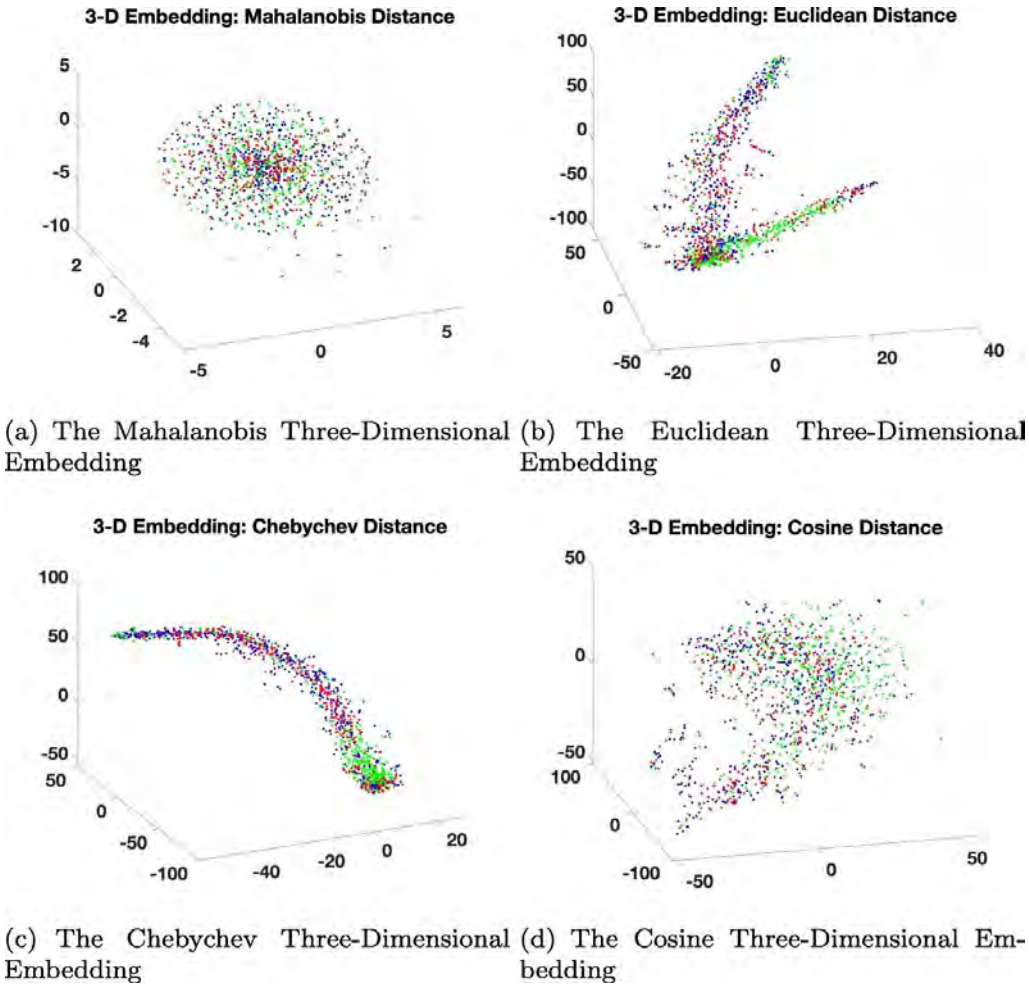
(c) The Euclidean Two-Dimensional Embedding

(d) The Cosine Two-Dimensional Embedding

Figure 14: Two-dimensional embedding of the simulated light curve data using the t-SNE algorithm.

lapping nominal semi-major axes. Large constellations with overlapping orbital altitudes have the potential to significantly increase orbital conjunction risk and the operational burden associated with planning and coordinating collision avoidance maneuvers for active spacecraft controlled by different operators. Reference [48] found that, for the region from approximately 500-600km, the addition of planned constellations would result in significantly increased conjunction risk with a large fraction of that risk coming from active versus active cross-operator conjunction events. This finding may understate risk given the numerous additional constellations proposed subsequent to that study. Furthermore, because collision avoidance is not perfect, overlapping large constellations increases the residual collision risk and thus reduces long-term orbital capacity, limiting future constellation placement options if the risk is not adequately mitigated.

We looked into developing a potential solution to impose safe segregation between orbital shells while improving orbital capacity relative to these alternatives by relying on the use of frozen orbits and taking advantage of the approximate latitude-based altitude dependency that these shells exhibit. Many Earth-observing missions have used frozen orbits to minimize radial orbit variation, beginning with Seasat [49], ERS-1 and ERS-2 [50], and TOPEX/Poseidon [51], and continuing to more modern Earth observation missions that are required to maintain a nearly fixed radial distance to a ground site [52]. Reference [53] suggested using frozen orbits and specifically optimizing for minimum radial distance variation as a function of latitude in order to avoid cross-shell collision risk. The paper computes these orbits numerically in the presence of zonal and tesseral harmonics, lunisolar third-body perturbations, solar radiation pressure, and atmospheric drag. The same frozen orbit technique was used to define satellite locations within a single shell in Reference [54] and shown to reduce intra-shell conjunction frequency. Nevertheless, these methods do not explicitly enforce periodicity or provide methods to



(a) The Mahalanobis Three-Dimensional Embedding

(b) The Euclidean Three-Dimensional Embedding

(c) The Chebychev Three-Dimensional Embedding

(d) The Cosine Three-Dimensional Embedding

Figure 15: Two-dimensional embedding of the simulated light curve data using the t-SNE algorithm.

evaluate orbital capacity, and neither paper discusses how to nest these frozen shells.

We studied the nesting of concentric frozen orbital shells and to generate these shells using methods that, subject to numerical error and quasi-stability of frozen orbits, enforce periodicity and connect the generated shells to previous theoretical results on 2D Lattice Flower Constellations (2D-LFC) [55] that allow estimation of orbital capacity, reconfiguration, and optimization for minimum separation distances [56–59]. Rather than optimize individual satellite trajectories, we approached the problem by defining operational volumes, or slots, that evolve in a quasi-periodic manner and in which particular satellites can operate according to whatever control laws and internal positions they see fit. Accordingly, we incorporated only the effect of the Earth's geopotential to preserve the evergreen satellite-agnostic nature of slot designs within a shell and cross-plane quasi-periodicity which can be used to provide useful collision avoidance guarantees for 2D-LFCs.

A consequence of this decision is that shell designs are not necessarily as compact as satellite-optimized trajectories that may fit within them, but better connect to existing 2D-LFC slotting work, can be computed in reasonable time periods without access to supercomputing resources, and do not make assumptions about the drag area, coefficient of reflectivity, mass, or other general properties of the slotted satellites during shell design (although values do need to be assumed for post-hoc full-force modeling for stability analysis). We further explored several important implications for shell design. These include the usefulness of constellation visualization in the latitude-altitude plane as an analysis tool for ensuring non-conjunction between adjacent shells, the influence of shell inclination on possible shell tessellation, and an assessment of simplified shell geometry estimation methods. Last, we applied this methodology for potentially overlapping or adjacent shells in LEO. These results and conclusions likely have relevance even if other methods are used to generate frozen shells.

Results generated using either of the presented 2D-LFC definition methods or a simplified shell model can be visualized in the latitude-altitude plane. The separation between shells in this space provides intuitive assurance that over the time period propagated, the shells with not intersect and do not pose an inter-shell conjunction hazard. The shape of shells is primarily a function of inclination and eccentricity. As seen in Figure 16 and demonstrated mathematically in Reference [53], shells generated to be classically frozen under a $J_{3,0}$ model are arbitrarily stable and thin, subject only to numerical error. However, as seen in Figure 17, $J_2 - J_3$ shells are not sufficiently stable when propagated under

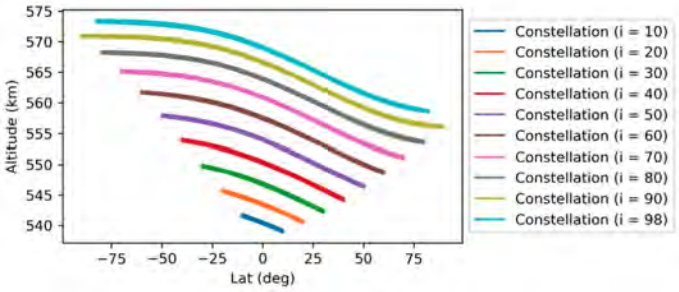


Figure 16: $J_2 - J_3$ Frozen shells propagated under a $J_{3,0}$ model for 30 days. The shells show an inclination-dependent shape and tessellate well.

more realistic geopotential and rapidly lose coherency. While shells can be defined to account for higher-order and/or degree geopotentials, as described previously, this leads to additional shell thickness that increases with time and magnifies differences in shell shape as a result of inclination.

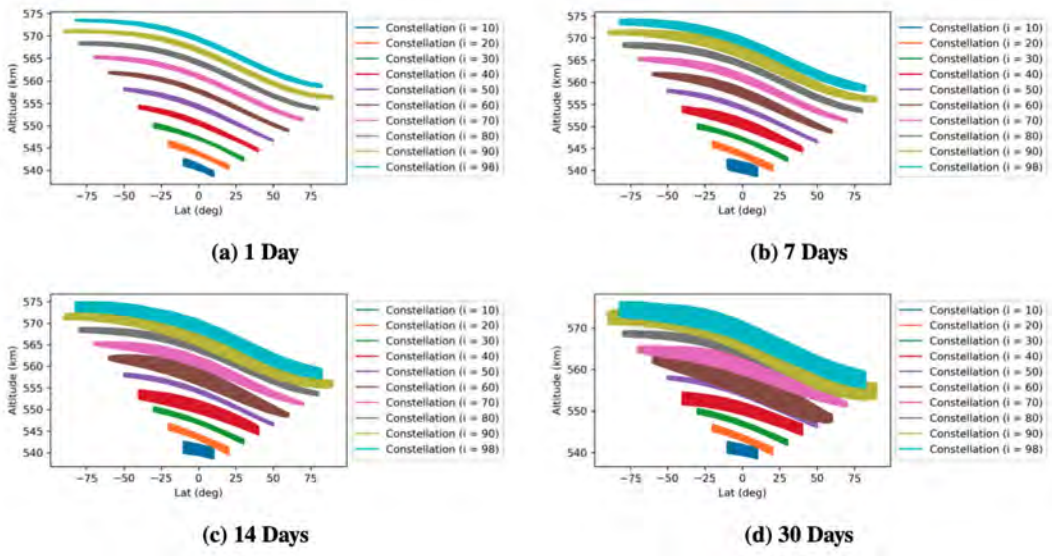


Figure 17: $J_2 - J_3$ Frozen shells propagated under a $J_{21,21}$ model for 30 days. The shells lose coherence over time.

Shells designed using either of the above methods are stable over a longer period and preserve similar separation geometry. As seen in Figure 18, frozen coordinated shells can under any circumstances be placed closer than an arbitrary Keplerian minimum separation distance between shells while preserving the same actual minimum separation distance between shells! This is a very important result for ensuring simultaneous orbital efficiency and safety.

The use of coordinated sequential inclinations can result in significant increases in the number of admissible shells when compared to an alternative with randomly chosen inclinations spaced to ensure a non-overlap between shells. In Figure 19, two notional configurations for the equatorial altitude range from 400-800 km are considered to demonstrate this effect, each preserving separation between shell center-lines. Shell center-lines are estimated rather than explicitly calculated and propagated. In the left figure, each shell has a randomly selected inclination (assumed to be in the interval [10, 100] degrees),

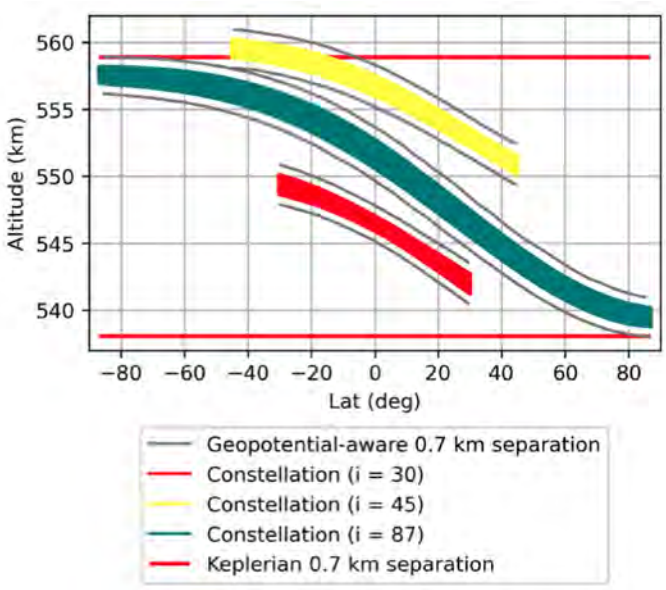


Figure 18: Shells generated using frozen orbits can nest closer than the osculating variation in the semi-major axis for each shell and still maintain coherency and safety (zonal freezing method, 30-day propagation, $J_{21,21}$ model)

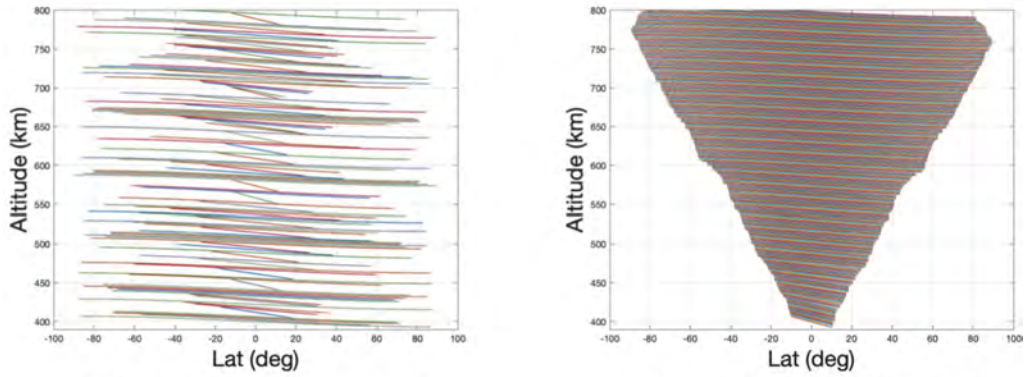


Figure 19: A comparison of Random and Sequential Inclination Shell Layering Approaches.

while in the right figure, shells are placed in a strictly ascending order of inclination. Ten satellites were modeled for each shell, with each satellite having random per-satellite mean eccentricity values drawn from a uniform random distribution between 0.00102 and 0.0012, a random inclination offset from the shell inclination of between 0.1 and 1 degrees (drawn from a uniform random distribution), and an argument of perigee of 90 degrees with per satellite error of between 2 and 20 degrees (again sampled from a uniform random distribution). The sequential strategy is able to fit 103 more shells (196 vs. 93) when compared to the random approach. Assuming every shell uses a frozen eccentricity rather than a random small eccentricity would reduce but not eliminate the penalty associated with random shell order. This ordering and spacing is not presented as a proposal that we believe should be specifically implemented in LEO, but rather to be indicative of gains to capacity from coordination of shell inclinations.

2.4 Deep Reinforcement Learning Techniques for SO Intent Identification and Anomaly Detection

Space Situational Awareness (SSA) has many definitions depending on the goal at hand, but in general, it involves collecting and maintaining knowledge of all space objects (SOs) orbiting the Earth and the space environment. This task is becoming more difficult as the number of objects currently tracked by the U.S. increases due to breakup events and improving tracking capabilities [1]. The Space Surveillance

Network (SSN) is tasked with maintaining information on over 22,000 objects, 1,100 of which are active, with a collection of optical and radar sensors. Determining physically significant characteristics, i.e. attributes, that go beyond simple orbital states is a key objective that is required for protecting space capabilities and achieving SSA. For example, the SSN catalog currently includes a radar cross-section and a non-conservative force parameter, analogous to a ballistic coefficient, which provides additional SO characterization information beyond position and velocity. Future SSA systems will have to be capable of building a much more detailed picture of SO attributes in order to maintain better knowledge of their characteristics, which ultimately may lead to better tracking capabilities.

Inverse Reinforcement Learning (RL) is used to learn the behavior of Space Objects (SOs) from observed orbital motion. The behavior of SOs is estimated using inverse RL to determine the reward function that each SO is using to control. Since SOs having the capability to maneuver are controlled to achieve a particular mission-driven goal, maneuvering can be very subjective and only a data-driven learning approach can reveal the true goal. It is also important to determine what type of behavior a SO is using and if this behavior changes. Inverse RL approaches use optimal control principles to learn what reward function is being used by an agent given observations.

The simplest inverse RL approach, discussed in Ref. [60], solves for the reward function using a weighted sum of features. This is the Feature Matching Approach (FMA). The weights determined from the inverse RL calculation are the representation of the expert reward function. The estimated reward function weights can be used to determine the type of behavior mode the SO is following and to classify the model based on libraries of behavior models. These weight vectors can be added to the state of SOs as a way to represent the policy that the SO is currently following and allow for the change of this policy over time and as the behavior changes. Using the inverse RL approach, the optimal control problem is formed as a Markov Decision Process (MDP) where the reward function is not explicitly given. Rather, observations of expert demonstrations for a given task and the goal is given to estimate the reward function that the expert used to derive the demonstration trajectories. It is common to assume that the expert's actions are optimal with respect to the expert's reward function. Unfortunately, several problems exist with this early approach. First, the formulation is limited to discrete MDP system models which contain discrete state-action pairs at some time. Many problems exist with continuous state-action (i.e. a low-thrust transfer orbit problem), thus, this formulation is limited to just discrete MDPs. Another problem is the ambiguity between different policies and their optimality to numerous reward functions. When sub-optimal trajectories are observed, many different policies are needed to match the feature counts, and thus, many policies satisfy the feature matching. The ambiguity is not settled by this theory.

Another earlier method to inverse RL solves for a reward function using maximum margin planning (MMP) [61]. The main idea of this method is to find the state expectation that is closest to the expert demonstrations. The weights are found which form policies that have a higher expected reward than all the other policies by some margin. The demonstrated trajectories can have different feature maps, start states, and goal states which allow for more accurate reward function solutions compared to the expert. While FMA assumes that the expert is acting optimally (or close to optimally) and the reward function found by inverse RL should match the expert features closely, MMP relaxes this assumption to mimicking the expert while questioning the expert's specific reward function. Even with MMP's advantage over FMA, MMP has similar problems to FMA. Although the assumption about the expert's optimality is weakened, this assumption is still in place. Thus, for sub-optimal expert behaviors, the feature matching expectation can still be ambiguous. MMP is also formulated for discrete state and action spaces. No extension to continuous spaces has been made.

An approach that resolves the ambiguity in the FMA and MMP assumption and has extensions to continuous states and actions is the maximum entropy formulation [62]. This approach uses the principle of maximum entropy which resolves the ambiguity of sub-optimal expert demonstrations to find a single stochastic policy. Specifically, maximum entropy allows for distribution over behaviors while matching feature expectations with no duty to any specific stochastic expert trajectory. Maximum entropy has the benefit of determining solutions with a sequence of side information which are variables that are not predicted but are related to these predicted variables. If the side information is dynamic (changing through time), an extension called maximum causal entropy can be used [63]. The original maximum entropy approach assumes all side information is available a priori (i.e. current known knowledge) while the side information in maximum causal entropy is revealed through time. Thus, the future side information has no causal influence on past variables. Therefore, this extension can naturally handle MDPs and state spaces involving stochasticity and continuous state and action variables. An inverse RL framework using maximum causal entropy and a linear quadratic regulator (LQR) setting is discussed

in [63–65] and provides a solution to predicting reward functions from LQR-controlled SO trajectories.

Other methods based on maximum entropy exist including nonlinear inverse RL using Gaussian processes [66], maximum entropy deep inverse RL [67], and generative adversarial imitation learning [68] which can solve for rewards that are complex. The main idea of inverse RL using Gaussian processes is to use a Gaussian process to express the reward function as a nonlinear function. This allows for the prediction of reward functions that involve complex behaviors from sub-optimal expert trajectories. Maximum entropy deep inverse RL uses neural networks approximate complex nonlinear reward functions and sub-optimal expert trajectories. Generative adversarial imitation learning enables the learning of complex expert behaviors by combining maximum causal entropy with RL techniques. By using an occupancy measure, the expert behavior is compared to the behavior learned by the RL algorithm. The expert trajectories from the SO orbit problem are formed as a quadratic reward, so these extensions are unnecessary, and a basic maximum causal entropy method is used to predict the reward function.

2.4.1 Physically-Constrained Inverse Optimal Control for Satellite Maneuver Detection

We looked into the usage of inverse optimal control (IOC) to develop a physically-constrained behavior estimation approach for SOs. Optimal Control refers to the process of computing state and control trajectories that are optimal with respect to a cost function subject to constraints, and in the context of this work, we call this the Forward Optimal Control (FOC) problem. The Inverse optimal control (IOC) refers to the process of determining the cost function implied by observed state and control trajectories. The study of IOC problems has been an active area for more than 50 years, [69] and has been successfully applied to many fields [70]. Kalman [69] first studied the problem of finding all performance indices that a given control law is optimal with respect to. This theory has seen many applications for the development of control laws [70, 71] and more recently in reinforcement learning [60, 62, 67, 72, 73].

IOC is also known as Inverse Reinforcement Learning (IRL) [72] but our work makes a distinction between these two approaches. Our work uses the term IRL to refer to computing IOC solutions without knowledge of underlying dynamics and in a stochastic setting. There are many emerging applications for both IRL and IOC which include transferring expert demonstrations to robotic systems [60], humanoid robot control, autonomous driving [74], human-robot cooperation [75], and stabilization of rigid spacecraft systems [76]. IRL is often applied to the problem of learning control policies from demonstrations; this process is referred to as learning or apprenticeship learning. The demonstration data is usually considered to be state and control trajectories that are captured from expert demonstrations.

In general, both the IRL and IOC problems fall under a more general framework of imputing objective function for optimization problems from observed solutions. In this context, [77] developed an approach for imputing the objective function given known constraints and observed trajectories. The general concept for finding the cost function is to set up a set of optimality equations that should hold for the observed trajectories and then minimize the residual of these equations. [77] used the Karush-Kuhn-Tucker (KKT) optimality conditions to determine the unknown cost function. Additionally, [77] proposed that this approach could be applied to the IOC problem where the known constraints are the system dynamics and the unknown objective function is the control cost function. These ideas were then extended to solving the deterministic discrete-time IOC problem [78], IOC problems for hybrid systems [79], additive cost functions and linear constraints for aircraft intent determination [80], and for analysis of human locomotion [78]. Finally, for continuous-time systems, these concepts can be extended using Pontryagion's Minimum principle where the Euler Lagrange equations in continuous cases are employed [81].

By using IOC, a set of reward functions can be found for which a given policy is optimal. Unfortunately, this set may contain degenerate reward functions (i.e. reward functions that are zero). To mitigate this, additional constraints are imposed on the cost function [77]. Another problem that arises is large or infinite state spaces in which a tabular form of the reward function is unattainable. By assuming the reward function is a linear combination of fixed basis functions, the IOC stays in the class of linear programs that can be solved efficiently [82]. Nevertheless, these problems should be considered while formulating the IOC algorithm.

The estimated reward function provided by IOC can be used to determine the type of behavior mode the unit is following and to classify the mode based on libraries of behavior modes. This concept was demonstrated for aircraft intent estimation using IOC solutions [80]. Our work also investigates using feature weight vectors determined by the IOC approach for maneuver prediction and classification for satellites. Our work follows the standard assumption made for both IOC and IRL approaches that the cost function is a linear combination of features [83]. These weight vectors can be added to the state of

SOs as a way to represent the policy that the SO is currently following and allow for the change of this policy over time as the behavior changes. Rather, we are given observations of expert demonstrations for a given task, and the goal is to estimate the reward function that the expert used to derive the demonstration trajectories. It is common to assume that the expert's actions are optimal with respect to the reward function the expert is using and our work makes this assumption. Our work investigates the Pontryagin's minimum principle for solving for the expert's reward function [81].

Our work utilizes IOC to develop a physically-constrained behavior estimation approach for Space Objects (SOs). Our work uses the dynamical equations of motion for SOs to determine constraints that can be imposed on the IOC solution. Our work discusses the use of IOC to learn the behavior of Space Objects (SOs) from observed orbital motion. The behavior of SOs is estimated using IOC to determine the reward function that each SO is using for determining its control. In general, SOs are controlled to achieve a particular goal that is determined by their mission, and therefore only a data-driven learning approach can reveal the true goal. It is also important to determine what type of behavior SOs are using and if this behavior changes. IRL and IOC approaches use optimal control principles to learn what reward function or control objective function is being used by an agent given observations [60].

We applied the IOC approach to three simulated cases of maneuvering SOs. The first two cases use an SO in GEO which is maneuvering to maintain a given GEO stationary box. The control objective function was specified, and simulation data was generated for the hypothetical SOs. The first two cases differ in the features used to represent the control objective function. The control objective function was estimated using a linear combination of features. The first case uses simple second-order polynomial features while the second case demonstrates complex features based on neural networks. The nonlinear basis functions used for our work included polynomial terms and extreme learning machines (ELMs). ELMs are effective for the SO behavior problem because it has been shown to work well for high dimensional problems. Machine learning techniques have been successfully used in learning functional relationships, that only require a limited amount of data. Most of such techniques (e.g. NNs, Support Vector Machines (SVM)) are faced with many challenges including, slow learning speed, poor computational scalability as well as the requirement of ad-hoc human intervention. Extreme Learning Machines have been recently established as an emergent technology that may overcome some of the above-mentioned challenges by providing better generalization, faster learning speed, and minimum human intervention [84]. ELMs work with "generalized" Single Layer Forward Networks (SLFN, Figure 20). SLFNs are computationally designed to have a single hidden layer (which can be either Radial Basis Function (RBF) or other activation functions) coupled to a linear output layer. The key point is that the hidden neurons need not to be tuned and their weights (training parameters) can be sampled from a random distribution. Theoretical studies [85] show that feed-forward networks with minimum output weights tend to achieve better generalization. ELMs tend to reach a) the minimum training error and b) the smallest norm of output weights with consequently improved generalization. Importantly, since the hidden nodes can be selected and fixed, the output weights can be determined via least-square methodologies.

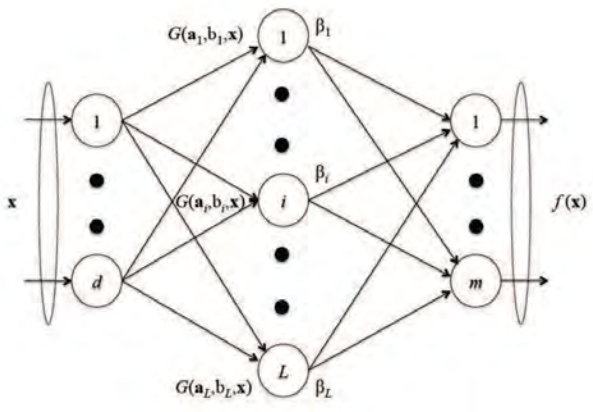
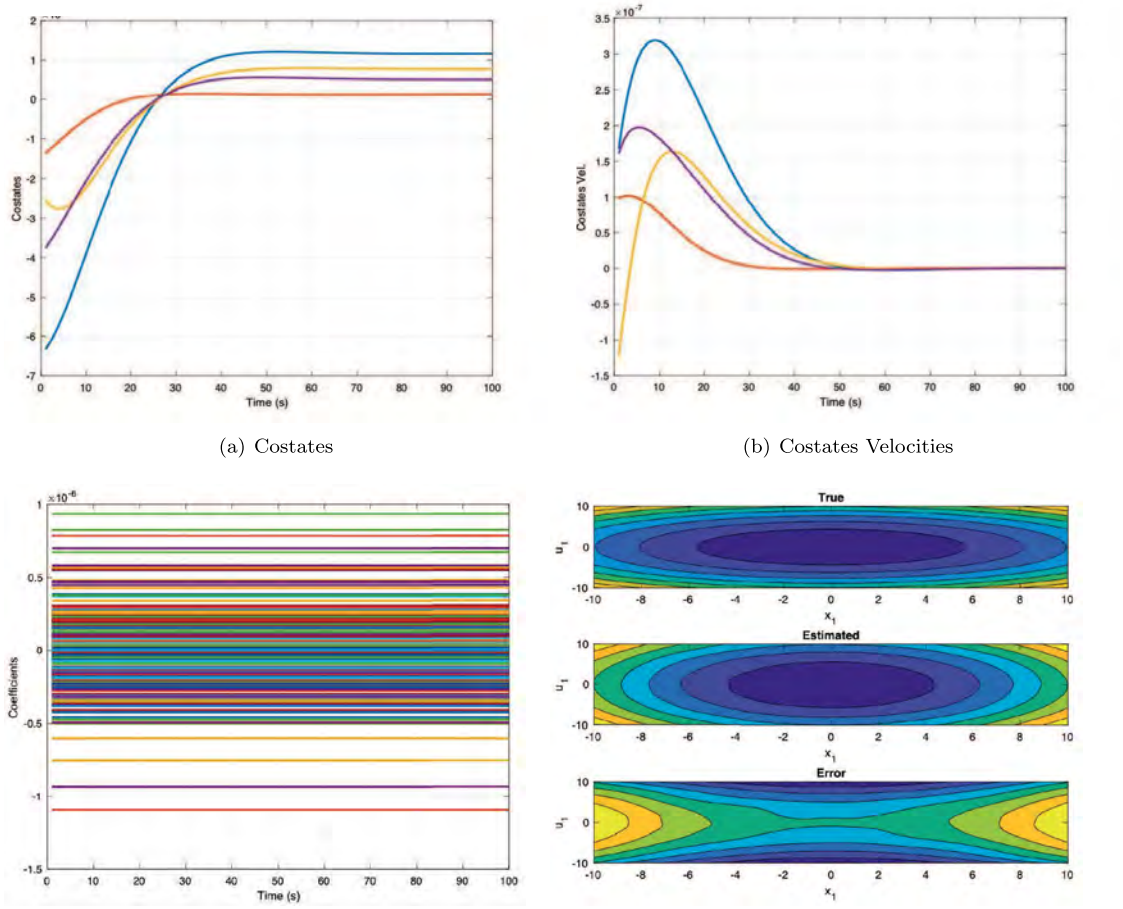


Figure 20: Typical Architecture of a Single Layer Forward Network (SLFN) Which Is the Most Fundamental ELM

Figure 21 shows the results for the case that uses ELM features. From this figure it can be seen that the ELM features are flexible enough to capture the cost function, which is based on polynomial features. Additionally, this case had 400 ELM features which were easily solvable using the IOC approach. The

estimated ELM weight parameters are shown in Figure 21(c), where the costate and costate velocities are shown in Figures 21(a) and 21(b), respectively. These results show promise for extending our work to high-dimensional systems.



(a) Costates

(b) Costates Velocities

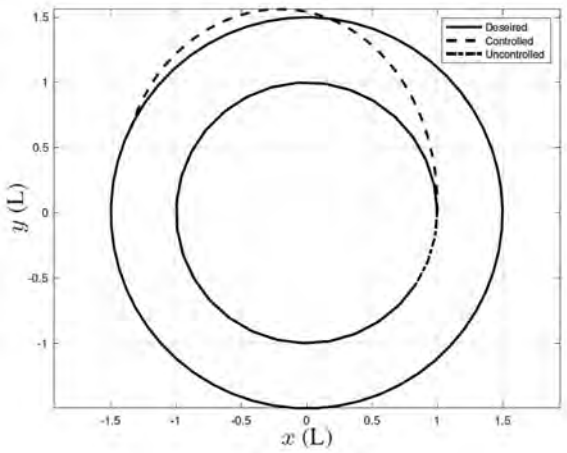
(c) Weight Parameters

(d) Estimated vs Control Objective Function

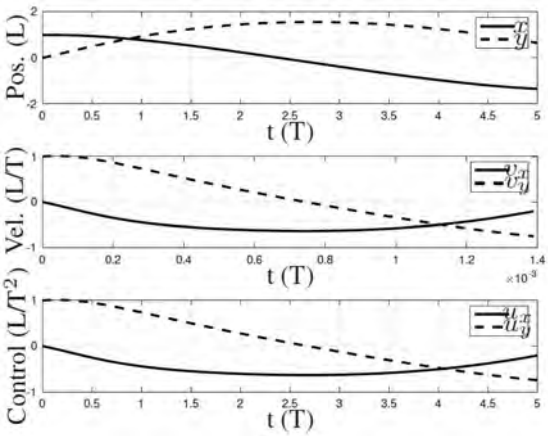
Figure 21: ELM features for Relative Motion Example

We then study the behavior estimation problem for optimal orbit raising with continuous thrust. The estimated control objective function approximated the true control objective function well, and good performance was shown for the proposed approach. The trajectory used for this case is shown in Figure 22. The solution for the forward problem is shown in Figure 22(a) and 22(b). In this case, the forward problem is nonlinear since nonlinear two-body dynamics are used. However, the inverse problem is still convex and solvable using the LQR-based approach. Figure 22(c) shows the solution of the Riccati Differential Equation for the initial $P(t_0)$ and once this is obtained, the coefficients can be solved for using the Quadratic Programming problem. The estimated weight vector using five observed trajectories recovered the control cost function with good accuracy. The norm error for this case was $\|c_{\text{true}} - c_{\text{est5}}\|_2 = 7e^{-2}$.

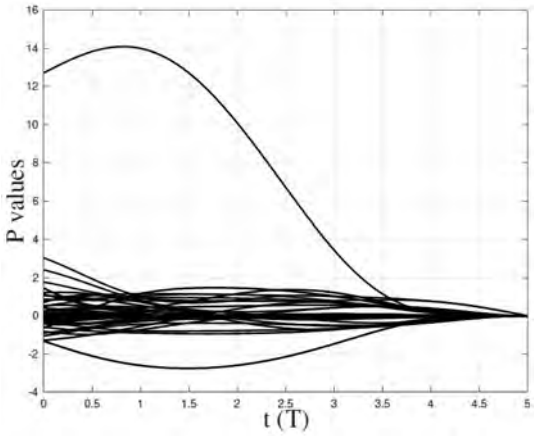
In conclusion, given the observed trajectories of states and controls, the IOC approach can be used to estimate the control objective function that a given SO is using. Our work considers three simulated cases of maneuvering SOs. The first two cases use an SO in GEO which is maneuvering to maintain a given GEO stationary box. The third case studies the behavior estimation problem for optimal orbit raising. The control objective function was specified, and simulation data was generated for the hypothetical SOs. Good performance was shown for the first two cases, and it was observed that the accuracy of the IOC solution improves with the number of observed trajectories used. For the third and final case, the estimated control objective function approximated the true control objective function well, and good performance was shown for the proposed approach.



(a) Transfer Orbit



(b) Orbit Raising Control



(c) P Values Orbit Raising

Figure 22: Continuous Thrust Orbit Optimal Raising Example

2.4.2 Space Objects Maneuvering Prediction via Maximum Causal Entropy Inverse Reinforcement Learning

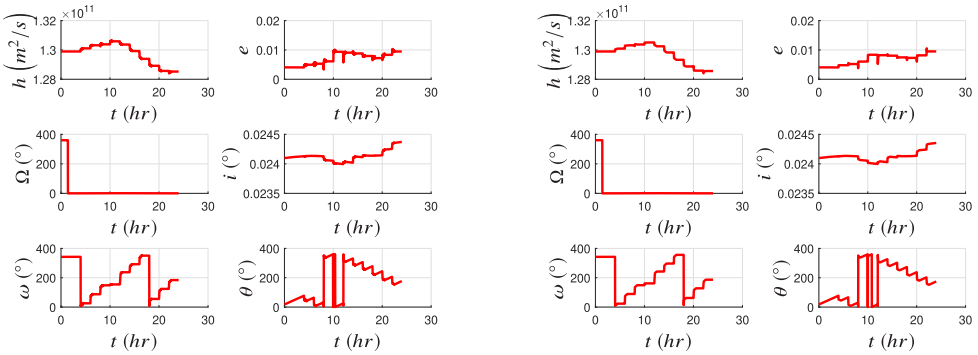
We also investigated the maximum causal entropy approach [60] for predicting the expert's reward function. Inverse RL results for Low Earth Orbit (LEO) station-keeping and Geostationary Orbit (GEO) station-keeping are presented using the maximum causal entropy. Specifically, LEO and GEO expert trajectories are formed with an LQR quadratic reward, and inverse RL using maximum causal entropy is used to predict the reward function. This approach determines the optimal reward function that a SO is using while maneuvering with random disturbances by assuming that the observed trajectories are optimal with respect to the SO's own reward function.

FMA and MMP are formulated by matching feature counts. This is fundamentally ill-posed because no true optimal policy will match the feature counts of a stochastic problem. Many different policies will satisfy the constraint. The principle of maximum entropy determines the least committed probability distribution to stay within the stochastic problem constraints [86]. The causally conditioned probability is an extension that is used when side (state) information is determined after each time step [87]. If future information on the distribution is used, the additional information decreases the causal entropy because there is less uncertainty present. The maximum causal entropy approach predicts a policy (actions) based on the state information provided, thus yielding an option to predict a reward function.

For the GEO scenario, the objective of the SO is to stay in GEO orbit under the influence of solar radiation pressure, gravitational perturbations, and aerodynamic drag. To maintain orbital station-keeping, the thrusters are fired every two hours for up to 30 minutes if the SO lies outside a 75km box

of the nominal GEO orbit. This is to provide a conventional thruster burn schedule for station-keeping maneuvers [88]. To obtain the expert trajectories, a hundred trajectories were simulated with Gaussian white noise. The cost function for these trajectories was determined using a quadratic cost function with constant weight matrices through time. Figure 23(a) shows a single expert trajectory in the GEO orbit with an initial perturbation from the nominal GEO orbit. From this figure, the eccentricity, inclination, and right ascension of the ascending node stay approximately zero through time due to stabilizing LQR control to the nominal GEO orbit. This figure explicitly shows the delta-v burns that occur every two hours when the SO is outside the 75km box around the nominal GEO orbit. A very small transient response in each element can be observed due to the burn itself. This control schedule is able to mitigate the variances due to the disturbances in the system. Figure 24(a) shows the true cost that is used for every expert trajectory. This LQR cost, which is what is considered to be unknown, is estimated using maximum causal entropy inverse RL.

Figure 23(b) shows a trajectory that uses the estimated LQR weights determined by maximum causal entropy. The trajectory is initialized with the same initial conditions as Figure 23(a). Similar to the expert trajectory, the SO is able to mitigate disturbances and maintain a GEO orbit with the same delta-v control schedule. The time history is similar, but Figure 23(b) has a less distinct transient response from the delta-v burns. The estimated cost function is shown in Figure 24(b). The estimated cost function is able to match the behavior of the expert trajectory. Specifically, the maximum error between the true and estimated cost function is 0.04 as shown in Figure 24(c). Even with these small differences, the estimated cost function captures the general trend of the station-keeping maneuver.



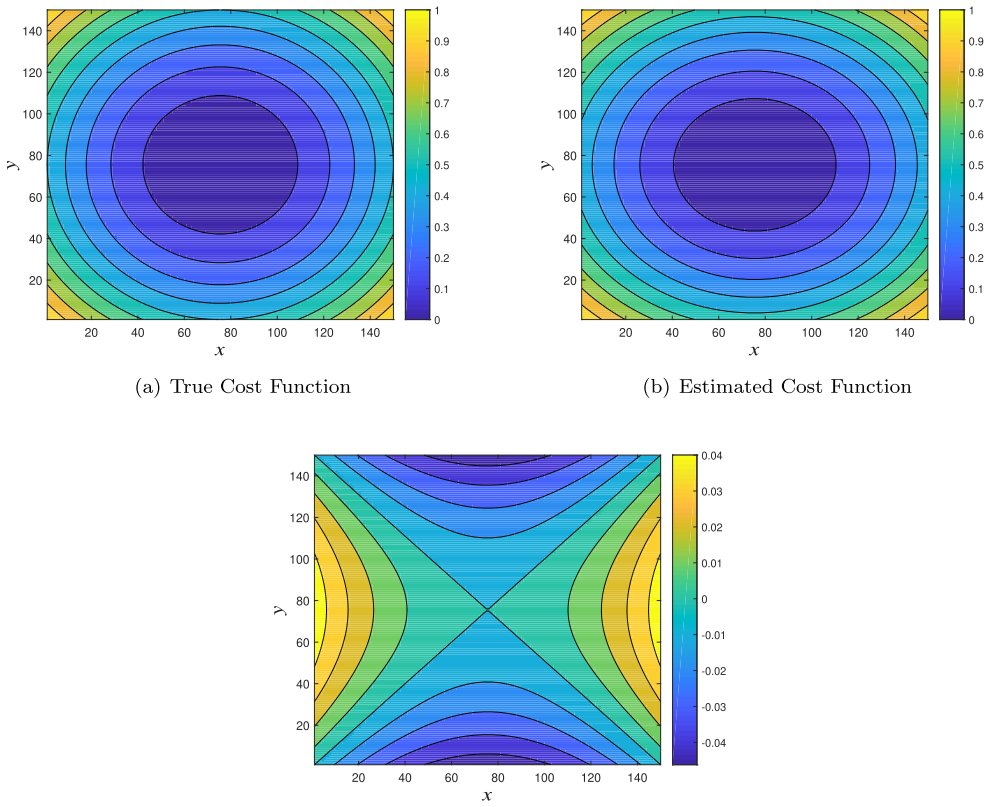
(a) Trajectory using True LQR weights

(b) Trajectory using Estimated LQR weights

Figure 23: SO Trajectory using True and Estimated LQR Weights for a GEO orbit

For the LEO scenario, the objective is for the SO to stay in LEO under external disturbances. Similar to the GEO case, the thrusters are fired every two hours for up to 30 minutes to maintain a nominal LEO orbit. One hundred expert trajectories were simulated with Gaussian white noise under constant LQR weight matrices. Figure 25(a) shows a single expert trajectory from an initial perturbation from an LEO orbit. From the figure, the eccentricity, right ascension of the ascending node, and inclination stay approximately constant at 0, 150°, and 50°, respectively. The oscillations are due to external disturbances on the SO which are not significant enough to perturb the SO outside a $75km$ box from the nominal LEO orbit in a 24-hour period. Thus, no thruster burns for SO station-keeping are observed. Figure 26(a) shows the true (and unknown) cost used for every expert trajectory. Since no control thrust is used, this contour does not depend on R. Figure 25(b) shows a trajectory using the estimated LQR weights found using maximum causal entropy. The estimated trajectory is initialized with the same initial conditions as Figure 25(a). The trajectory that is determined using the estimated weights has an almost identical response to the expert trajectory. Also, the SO trajectory obtained from the estimated weights did not use any thruster burns to maintain its nominal orbit. Thus, it follows the behavior of the expert. The estimated cost function is shown in Figure 26(b). The estimated almost matches the behavior of the expert trajectory. In Figure 26(c), the maximum error is 0.015 between the true and estimated cost functions. Since the trajectories obtained from the expert and estimated weights use no control within the 24-hour period, the behavior from the estimated weights captures the general trend of the LEO station-keeping maneuver.

In conclusion, by setting up expert SO trajectories using an LQR-based cost, the SO behavior can

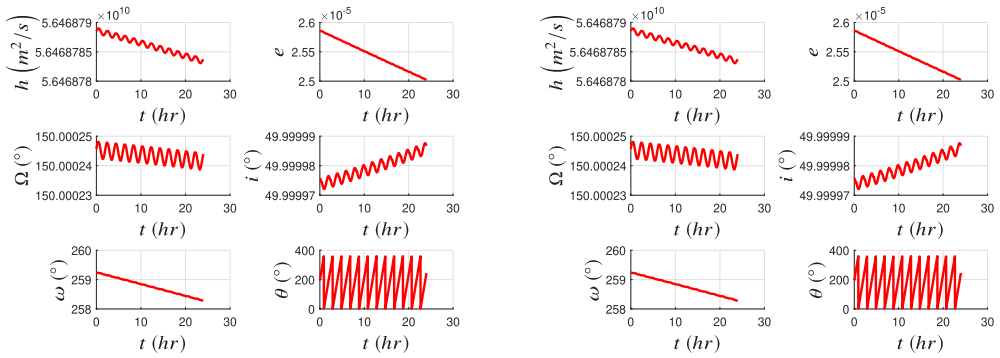


(a) True Cost Function

(b) Estimated Cost Function

(c) Estimated Cost Function Error

Figure 24: Estimated vs. True Cost Function for GEO Maneuvering SO

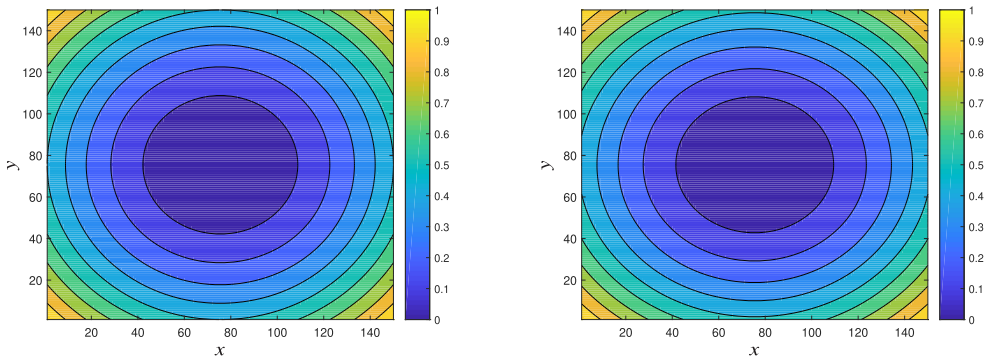


(a) Trajectory using True LQR weights

(b) Trajectory using Estimated LQR weights

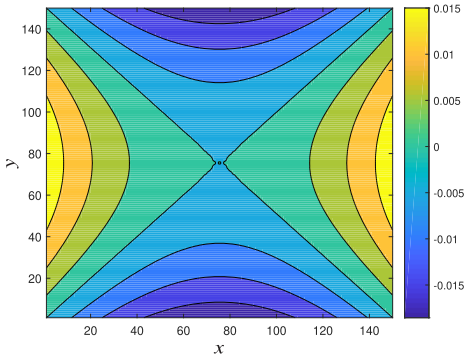
Figure 25: SO Trajectory using True and Estimated LQR Weights for an LEO Orbit

be learned using maximum causal entropy inverse RL. It is shown that the learned cost function in LEO or GEO is comparable to the observed expert SO's trajectory in the presence of orbital disturbances. Thus, maximum causal entropy is attractive in learning the reward function that produces the expert SO's maneuver and can be used to estimate the behavior of SOs.



(a) True Cost Function

(b) Estimated Cost Function



(c) Estimated Cost Function Error

Figure 26: Estimated vs. True Cost Function for LEO Maneuvering SO

2.5 Koopman Operator Theories for Space Application

A general framework for the analysis, estimation, and control of nonlinear systems remains an engineering grand challenge [89]. Accounting for nonlinear dynamics and measurement models has been an active area of research [90–92]. A promising direction for the analysis of nonlinear systems is to seek a transformation that embeds the nonlinear dynamics in a global linear representation [93]. The KO [94] provides such a framework and has been applied to many high-dimensional and nonlinear problems in engineering [95–100], as well as to the zonal harmonics problem in astrodynamics [101]. However, the use of the KO theory for SDA estimation problems has only been explored in limited contexts [102]. KO-based estimation methods have been developed to investigate ways to reformulate the strongly nonlinear estimation problems that arise in engineering into a linear framework. Furthermore, research has been performed to use the KO for improving maneuver detection and characterization approaches for cis-lunar satellites.

In the 1920s, with the development of quantum mechanics, there was general interest in reformulating classical mechanics using an operator theoretic perspective. In 1931 and 1932, Bernard Koopman [94] and Von Neumann [103] independently introduced and developed this operator theoretic perspective for classical mechanics. The key result was showing that there exists an infinite-dimensional linear operator, given by $\mathcal{K}$, that evolves all observation functions $g(\boldsymbol{x})$ of the state, $\boldsymbol{x}$, for any nonlinear system. The evolution of these observables and the KO is defined by the chain rule, which is a linear operator, giving KO its linear properties. The Koopman approach obtains an analytical solution of the system where the state is expressed as a linear combination of eigenfunctions. The linearity of the KO is very appealing, but this benefit is contrasted with the fact that it is infinite-dimensional. However, this issue can be overcome by capturing the evolution on a finite subspace spanned by a finite set of basis functions instead of capturing the evolution of all measurement functions in a Hilbert space [104]. In effect, this is a truncation of the KO to a finite subspace. A Koopman invariant subspace is spanned by a set of eigenfunctions of the Koopman operator. A Koopman eigenfunction, $\phi_i(\mathbf{x})$, corresponding to eigenvalue

λ_i, is invariant under the Koopman operator (but for a normalized constant). As such, the evolution of the Koopman eigenfunctions can be expressed as $\frac{d}{dt}\phi_i(\mathbf{x}) = \lambda_i\phi_i(\mathbf{x})$. The technique also provides an accurate spectral analysis of partial differential equations (PDEs) [105, 106].

A benefit of the proposed KO-based approach is that it makes use of linear spectral analysis for nonlinear systems to understand the behavior of these systems. With the KO approach, the time evolution of the state of the system is provided as a linear combination of a set of well-defined functions instead of performing a step-by-step propagation as done in numerical methods such as Runge-Kutta techniques. The goal of the proposed KO-based method is to reformulate nonlinear problems that arise in astrodynamics into a linear framework that can be solved with available linear techniques.

Using the Galerkin methodology [107], the proposed work seeks to develop a finite matrix representation of the KO. In the KO matrix representation, each entry of the Koopman matrix that defines the operator is computed analytically via the Galerkin methodology [107]. In Galerkin methodology, a set of basis functions (orthogonal polynomials are used in our work) is selected to describe the state-space of the dynamics and represent the eigenfunctions of the system. Our work uses Legendre polynomials due to their properties, which ease the evaluation of integrals over a well-selected domain or support [108]. Other approaches, based on numerical data, have been previously implemented to evaluate the KO matrix, such as the Extended Dynamic Mode Decomposition (EDMD). It has been shown that the computation of the eigenvalues and modes of the system through the Galerkin method is more precise than the data-driven based approach [109]. Previous work used the KO theoretic framework to model satellite motion in the proximity of libration points [110]. Additionally, the PI has developed proof-of-concept results for KO-based perturbation theory to model the motion of a satellite about an oblate planet, [111, 112] and to identify an approximate analytical solution to the zonal harmonics problem [108].

Accuracy, the most important performance index for analytical theories in astrodynamics, requires a precise representation of the dynamics, especially when considering perturbations [113]. KO-based analytical solutions of the J2 central-body problem have been developed by the PI [114] which enabled spectral analysis of the dynamics in terms of modes and eigenvalues. There are a large variety of different formulations [115–118] that have been investigated over the years to study the long-term evolution of a satellite subject to the perturbation produced by the oblateness of the central body [119]. Perturbation theory [120] appears to be a powerful methodology to connect the solution of the unperturbed problem to obtain a more complex analysis of the dynamics subject to small perturbing forces.

2.5.1 Solution of the Circular Restricted Three Body Problem (CRTBP)

The Koopman Operator methodology developed under this grant has been proven to provide an accurate transformation that embeds the nonlinear dynamics in a global linear representation. The selected application, the circular restricted three-body problem, is a highly nonlinear and highly chaotic system, whose periodic solution is hard to obtain. Nevertheless, the Koopman approximation provides an accurate analytical representation of the dynamics of the system, being able to accurately reproduce both Lyapunov and Halo orbits as a linear combination of well-selected eigenfunctions. As such, the dynamics of the system are transformed into a new framework, where each contribution has a linear dependence on the overall solution.

An analysis of the accuracy of the KO solution supports the validity of the newly proposed technique, especially for those initial conditions close to the equilibrium points, as shown by the study of the location of the minimum curves in the eigenfunctions. In addition, the accuracy of the newly proposed techniques has been assessed through a convergence analysis, where the position error of different families of Halo orbits has been compared to classical numerical integration techniques for the different orders of the KO. The results show that the behavior of the KO framework is similar to the perturbation theory since the robustness of the KO approximation decreases the further the initial conditions are selected from the equilibrium points.

Moreover, the KO mathematics has been proven helpful to provide the spectral behavior of the system. Particularly, a study of the frequencies and modes of the dynamics has been preformed. As such, periodic orbits are found by studying the eigenfunctions of the system directly and their curves of minima.

Furthermore, a comparison of different methodologies for the evaluation of the Koopman matrix has been carried out. The analytical approach proposed in our work is shown to better describe the spectral behavior of the dynamics of the system when compared to data-based methods such as the Extended Dynamic Mode Decomposition (EDMD).

Figure 27 reports a set of 30 different orbits calculated through a KO using basis functions of order

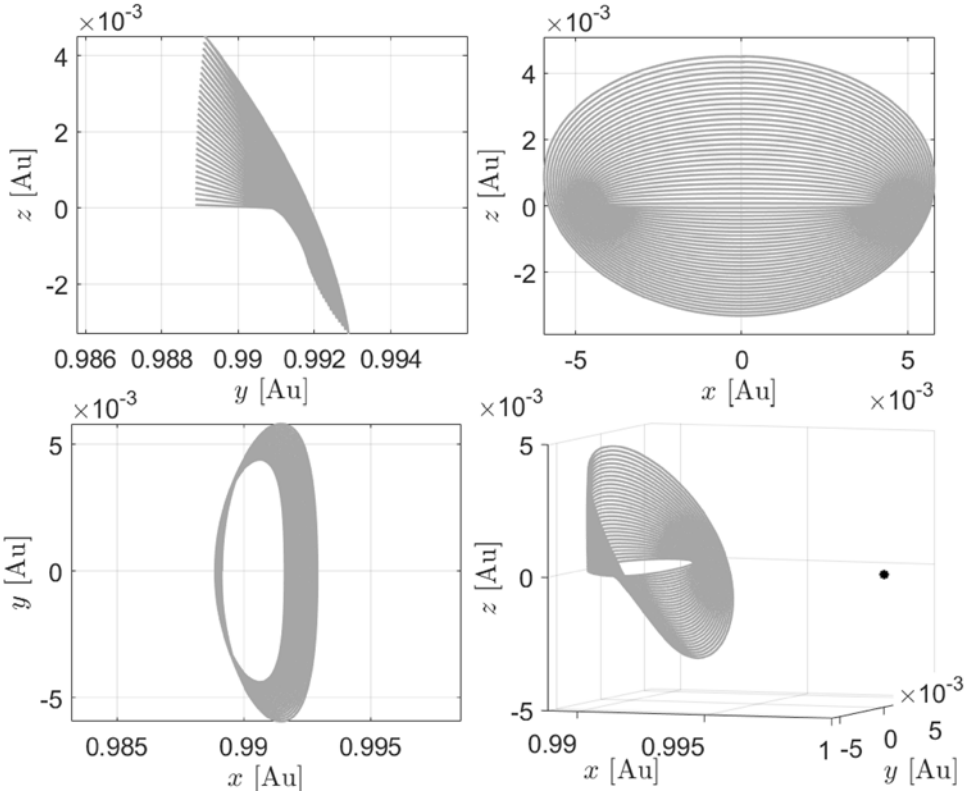


Figure 27: $\mathcal{L}_1$ Halo orbit family solved by a 5th Order Koopman Operator.

5. The figure includes the projection of the orbits in the three reference planes and an isometric three-dimensional view of the trajectories. The black point in the picture represents the secondary celestial body, which is the Earth in the selected Sun-Earth system. As the figure shows, the Koopman solution achieves a robust representation of the orbits. The accuracy of this solution will be later assessed. The periodicity is given by an accurate selection of the initial conditions, which have been found through the implementation of a differential corrector in such a way that the proposed solution describes the same orbits for multiple revolutions.

A similar pattern is reported in Figure 28, showing a family of Halo orbits at $\mathcal{L}_2$ solved using the 5th order KO. Once again, the black point in the figure represents the secondary body in the RTBP system. The KO is able to accurately represent orbits around both libration points. Since the value of γ is different when describing the dynamics around $\mathcal{L}_2$, Richardson's coefficients are different as well, and thus, the computed Koopman matrix is changed for the cases of motion about the $\mathcal{L}_1$ and $\mathcal{L}_2$ points.

2.5.2 Uncertainty Propagation, Filtering, and Control, in Astrodynamics

The KO has been applied to estimation and control problems. In particular, in the beginning, a station-keeping application is performed through a new filtering technique and a model predictive controller developed in the KO framework. The KO Particle Filter (KOPF) shows high accuracy levels for spacecraft state estimation, while the NMPC uses the KO solution to optimize the control feedback input for the satellite.

The station-keeping results using a KO with basis functions of order 3 are presented in Figure 29, where the reference orbit, the true position of the satellite, and its estimated position, are shown respectively as dashed, continuous, and point-dashed lines. As can be observed from the figure, the KOPF and NMPC are able to control the spacecraft and bring it to the desired pathway after just a few time steps. Particularly, the initial offset can be noted at the beginning of the simulation, and soon after, the spacecraft is already in the trajectory of the Halo orbit. In this example, the NMPC prediction horizon takes into account the following 10-time steps, which avoid the overshooting during the approaching to the desired orbit, and provides a smooth approach to the desired state. This means that after the quick transient, the three lines overlap for the remaining part of the trajectory, assessing the validity of the

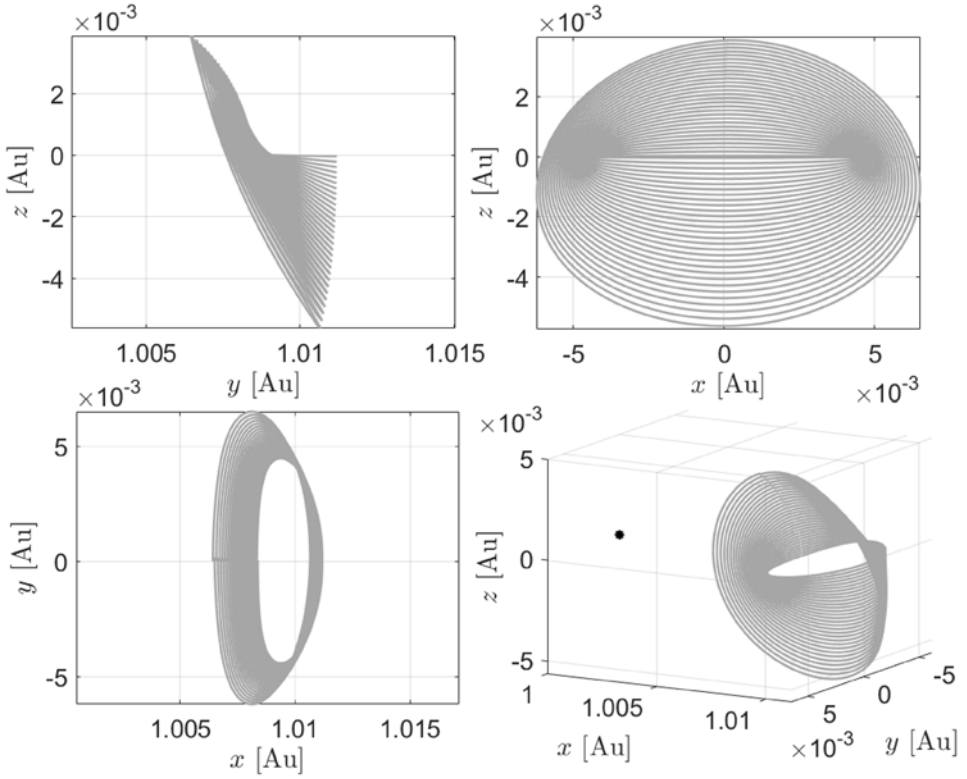


Figure 28: $\mathcal{L}_2$ Halo orbit family solved by a 5th Order Koopman Operator.

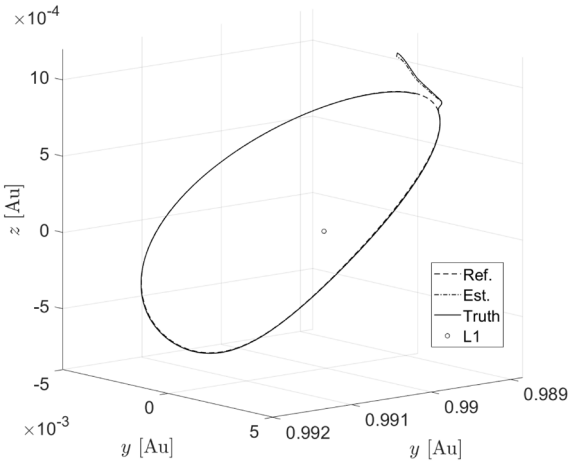


Figure 29: Station-keeping application of a spacecraft orbiting in a Halo orbit around $\mathcal{L}_1$.

proposed technique. The control horizon is set for 3 time steps.

In this regard, it is also possible to assess the tracking error of the spacecraft. This is done by evaluating the difference between the reference orbit and the true position. Analogously, the estimation error is calculated as the difference between the true position of the satellite and the estimated state from the filter. Figure 30 shows the tracking and the estimation errors for the presented application. After a quick transient where the system corrects the initial offset, the error levels settles down and it shows an accurate and robust estimate and control of the dynamics of the spacecraft.

The duration of the transient is directly connected to the maximum thrust available from the thrusters. Thus, in the first time steps, while compensating for the initial offset, the thrusters are saturated and reach the maximum available thrust. Having a more powerful thruster would reduce the transient time and achieve a quicker match between reference and target orbits. Figure 31 shows the

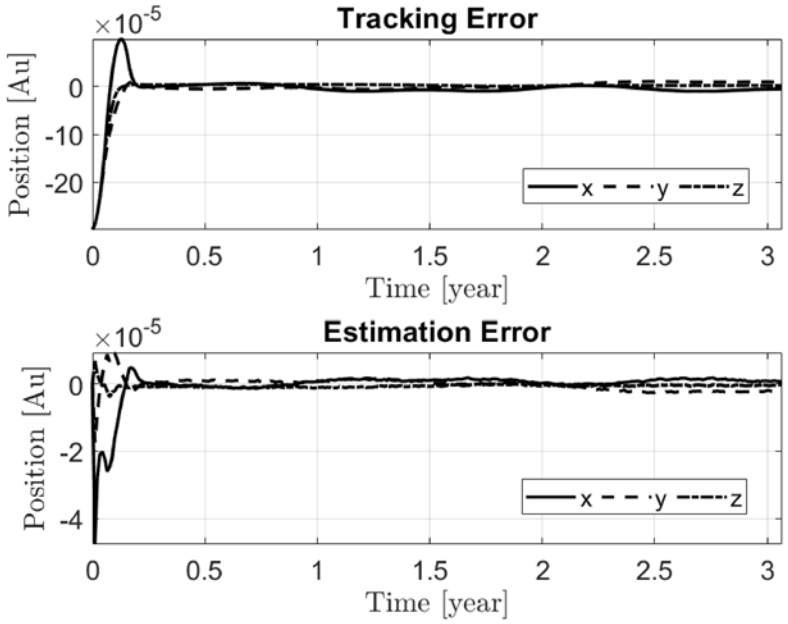


Figure 30: Tracking and estimation error of a spacecraft orbiting in a Halo orbit around $\mathcal{L}_1$.

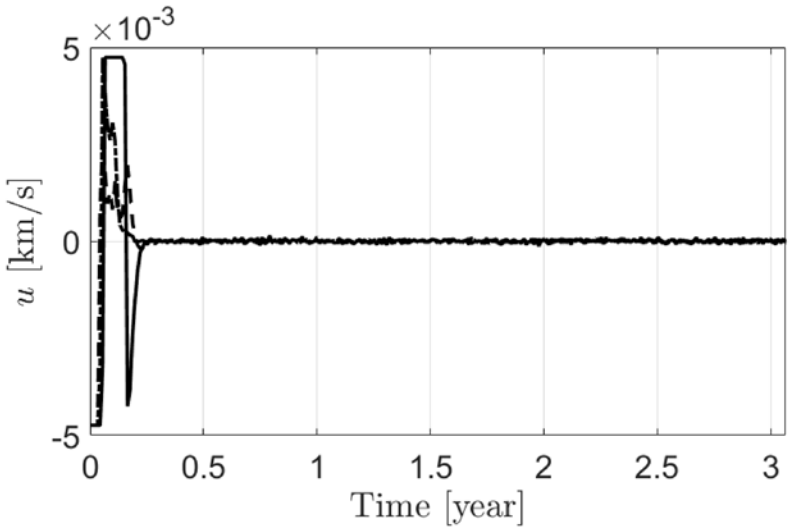


Figure 31: Feedback control input for a spacecraft orbiting in a Halo orbit around $\mathcal{L}_1$.

feedback control input, in terms of change in the spacecraft's velocity, during the complete simulation.

After the promising results of the new estimation approach, a new uncertainty quantification technique has been developed. The propagation of uncertainties under highly nonlinear dynamics has been performed in the KO framework, where the solution of the system is approximated using a set of well-defined orthogonal polynomials. Thanks to this representation, central moments of the state PDF can be predicted up to an arbitrary order. The new methodology has been paired with the Kalman measurement update to develop the Koopman Operator Filter (KOF): a nonlinear filtering algorithm where both the state and measurement distributions are represented using the KO eigenfunctions. Numerical applications have assessed the robustness and accuracy of the new theory. Uncertainty prediction in the CRTBP has shown that the new technique is able to track the exponential spread of the spacecraft position and velocity probability density function, providing estimates about the system's raw and central moments. Whenever information is provided in terms of external observations, the KOF is able to shrink the state uncertainties around the most current estimate. Monte Carlo analyses have assessed the accuracy, consistency, and robustness of the KOF, which has been proven to outperform other filtering

techniques reported for comparison purposes.

The newly developed technique has been applied to the CRTBP in the Earth-Moon system and results have been validated through a Monte Carlo analysis, where the predicted central moments of the state have been compared to the effective moments evaluated from all of the Monte Carlo runs. Figure 32 shows the time evolution of the uncertainties represented with the state raw moments. Indeed, the continuous lines represents the true evolution of the central moments of the system: the mean in black, the covariance, as $3\boldsymbol{\sigma}$ standard deviation, in blue, the skewness in cyan and the kurtosis in magenta. The values of skewness and kurtosis are evaluated similarly to the covariance of the system, with an analogy to the standard deviation. Therefore, for each component of the state, the high order central moment is evaluated by analysing all of the 10^4 simulations. On the contrary, the KO uncertainty propagation predicts the central moments of the state PDF every 0.2 years. The values of the predicted moments are reported, with the same colors used for the Monte Carlo, as points. The correct prediction of the mean and uncertainties of the system is asserted by the overlapping of the points over the continuous lines. The accuracy of the estimation of the moments decreases as the integrating time increases, meaning that, for extremely large propagation times, the prediction of the state PDF is unfeasible. The CRTBP is strongly diverging: thus, high order central moments after long time steps require an elevated amount of runs in order to have correct indicating values. Moreover, the figure shows the high level of accuracy in the first half of the simulation, for times shorter than 1 year, zoomed in in the figure. Station keeping applications, in filtering, usually receive measurements in the section where KO shows an extremely accurate prediction of moments. The KO technique can predict central moments up to any arbitrary high order. However, due to the fact that odd moments can be negative, the accuracy of the prediction of even moments is higher than their odd counterpart. Indeed, some values of the skewness, in the figure, are slightly off by the end of the simulation, where the state PDF largely expands without control. In the remaining sectors, the prediction of odd moments outperforms any other uncertainty prediction technique based on the Gaussian approximation of the state distribution (such as UT), since they provide a null value for any odd moment.

The central moments estimation can be appreciated by reporting the uncertainties as around their mean. Indeed, while Figure 32 shows the correct prediction of the family of orbits, in terms of its overall possible trajectories, Figure 33 focuses on the spread around the mean. Therefore, Figure 33 displays similar results as Figure 32, but with a new prospective centered at the current mean. Figure 33 shows the chaotic nature of the problem, as the uncertainties, in terms of their σ value, increase exponentially and the spread of all the possible resolutions of the state creates a conical shape. The KO prediction is able to keep track of the central moments of the system, especially during the first half of the simulation. Both the state covariance and kurtosis expand without any boundaries. The star points show the correct estimation performed by the KO technique. The new methodology is keeping an accurate prediction of the state distribution even for a long period of time, even if slightly reducing the accuracy. The figure has a logarithmic scale for the values of moments, highlighting the diverging nature of the CRTBP dynamics. Thus, it is worth noticing the exponential increase of the spread of the state PDF, with standard deviations that are two order of magnitude bigger after merely three quarters of a full revolution.

This application assesses the correct prediction of uncertainties and opens the road to the KOF, a complete filtering algorithm, that completes the propagation of mean and covariance with the knowledge obtained externally though noisy observations.

While the previous problem focused on the corrected prediction of the moments of the state PDF under an highly nonlinear dynamics, a station keeping application is now proposed, where the system receives external information from a set of sensors, and the whole KOF algorithm is implemented. A Monte Carlo analysis is performed to assess the performance of the filter. The initial level of uncertainties is so large that the accuracy level at a steady state is two orders of magnitude smaller than the initial standard deviation. Therefore, in order to show convergence in a more pleasant manner, a logarithmic scale for the errors has been applied. Thus, Figure 34 shows, for each component of the state, the 500 runs of Monte Carlo analysis performed with the KOF on the selected Lyapunov orbit. Indeed, each gray line represents the errors connected to the j-th run of the Monte Carlo, for each i-th state component. The mean of the errors is portrayed with black lines, calculated among all the Monte Carlo runs. The black lines settle on the zero value, proving that the KOF is an unbiased filter, as expected from the Minimum Mean Square Error (MMSE) principle on which the update of the filter is based on. The Monte Carlo analysis provides also information about the spread of the state error and the level of uncertainties. Figure 34 has multiple pairs of dashed and continuous blue lines. Dashed blue lines represent the level of standard deviation, as $\pm 3\sigma_{eff}$, obtained from the Monte Carlo runs: they represent the *effective* level of

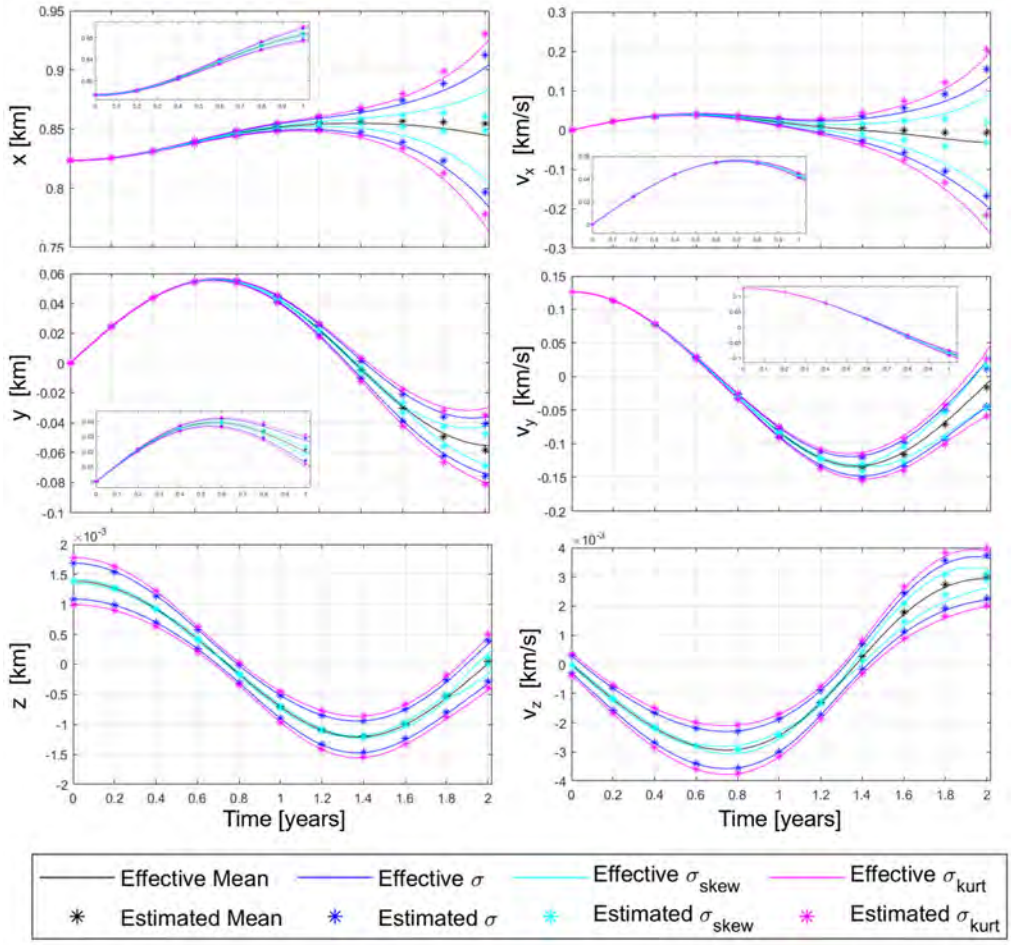


Figure 32: Prediction of raw moments of the state PDF for a Halo orbit orbiting around $\mathcal{L}_1$ in the earth-Moon system.

accuracy of the filter. These dashed lines are calculated, for each time step, by considering the values of the state errors among all the 500 simulations and they represent the actual behavior of the filter. On the contrary, the continuous blue lines report the *estimated* level of uncertainties, again as $\pm 3\sigma_{pred}$, predicted by the filter. Thus, for each time step, these lines are calculated extracting the error std directly from the update covariance of the filter. By looking at Figure 34, we can assess the consistency of the KOF, since the dashed and continuous blue lines overlap, meaning that the filter is able to correctly predict its own uncertainty. The KOF shows convergence and consistency, where steady-state accuracy levels are reached rapidly after a few updates. The large initial uncertainties are rapidly reduced around the most current estimate, and the updated covariance of the filter correctly represents the spread of the state error PDF.

The performance of the KOF can now be compared to other common filters: the EKF, the iterated extended Kalman filter (IKF), and the UKF have been selected as benchmark.

The comparison in accuracy among different estimators is performed by analyzing the updated level of uncertainties, with a consistency check obtained from the Monte Carlo. Therefore, the 500 runs Monte Carlo analysis has been repeated for the EKF, IKF and UKF. Standard deviations for position and velocity are calculated and compared separately. The predicted stds are obtained directly from the trace of the updated covariance matrix, while the effective stds levels, $\sigma_{pos,eff}$ and $\sigma_{vel,eff}$, are evaluated directly from the Monte Carlo runs. Thus, a consistent filtering algorithm has matching between σ_{eff} and σ_{pred}. Figure 35 reports the accuracy levels, in terms of error stds, of the four filters, displaying estimated values with continuous lines and the actual behaviour with dashed lines. The EKF is shown in red. The dashed red lines, both for position and velocity, are out of scale when compared to the continuous counterparts, which are overlapping with the green continuous lines. Indeed, the green lines

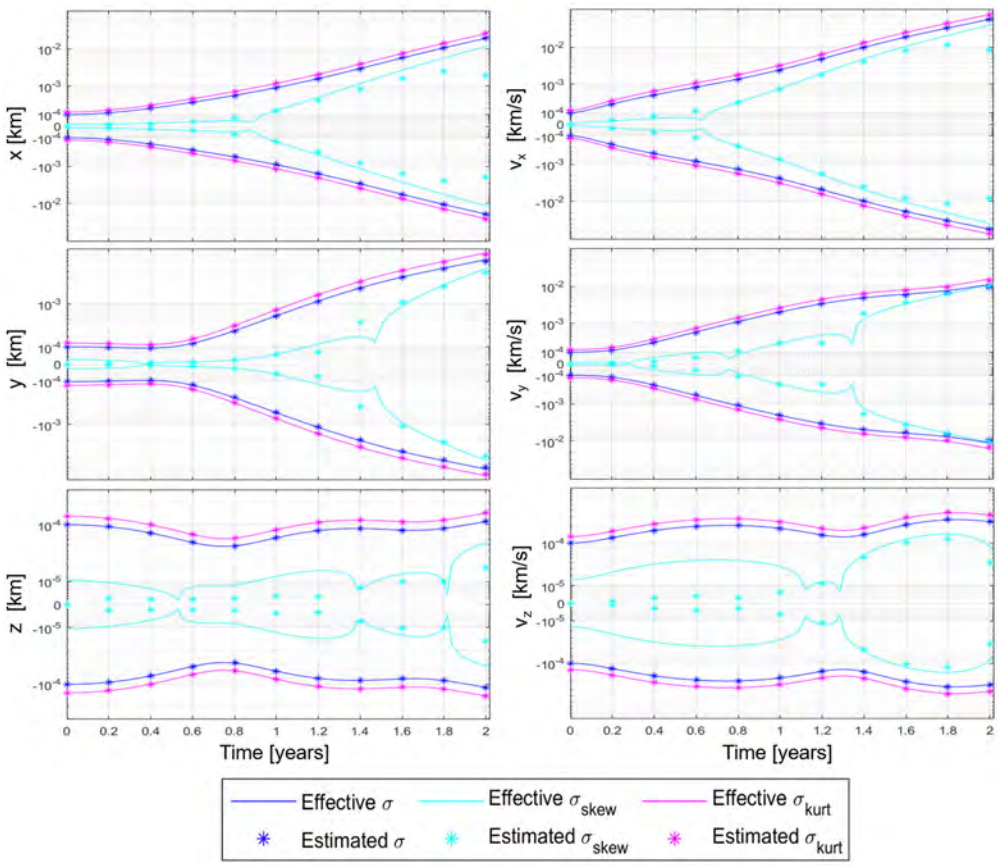


Figure 33: Prediction of central moments of the state PDF for a Halo orbit orbiting around $\mathcal{L}_1$ in the earth-Moon system.

are connected to the IKF. The EKF and the IKF share the same updated covariance matrix, while their estimate changes. Thus, the overlapping between the green and the red continuous lines matches with what is expected from theory: the two estimators approximate the posterior distribution of the state as Gaussian, with the same level of uncertainty (same stds), but different means. Indeed, while the EKF is a MMSE filter, the IKF is a filter based on the Maximum A Posteriori (MAP) principle, which outputs, as its estimate, the most likely state of the posterior distribution. However, the IFK and EKF share the same prediction step, based on the linearization of the dynamics, which is not sufficient to achieve an accurate approximation of the state prior distribution. Therefore, both filters are inconsistent and their dashed lines settle orders of magnitude above the continuous ones. The EKF and IKF believe that they are achieving higher accuracy than their actual results, and they are overconfident in their performance The difference between the green dashed lines and the red dashed lines is connected to the different update step performed by the two filtering techniques: a MMMSE one for the EKF, and a MAP one for the EKF. On the contrary, in blue, the UKF applies the unscented transformation in its prediction step to obtain a more accurate state prior distribution. Thus, the dashed blue line, which shows the effective accuracy level of the UKF, settles below the EKF and the IKF ones. However, the matching with the predicted uncertainty is missing and the filter is inconsistent as well. The UKF estimates a performance similar to the KOF, in black, shown by the overlapping between the black and the blue continuous lines. Therefore, the KOF is the only filter that converges with consistency. The KOF is able to correctly predict its own uncertainties and its estimate is the most accurate among the other filtering techniques. The dashed black lines are the lowest dashed lines for both position and velocity, and they overlap their continuous counterparts along the whole time-length of the simulation.

Lastly, the optimal control problem is being solved via the Koopman operator through the inverse method. The Koopman solution of the system is inverted to solve for the initial optimal control values that drive the state of the system following an energy optimal pathway. This technique has been applied in rendezvous applications, where a chaser spacecraft has been controlled to dock with its target in the

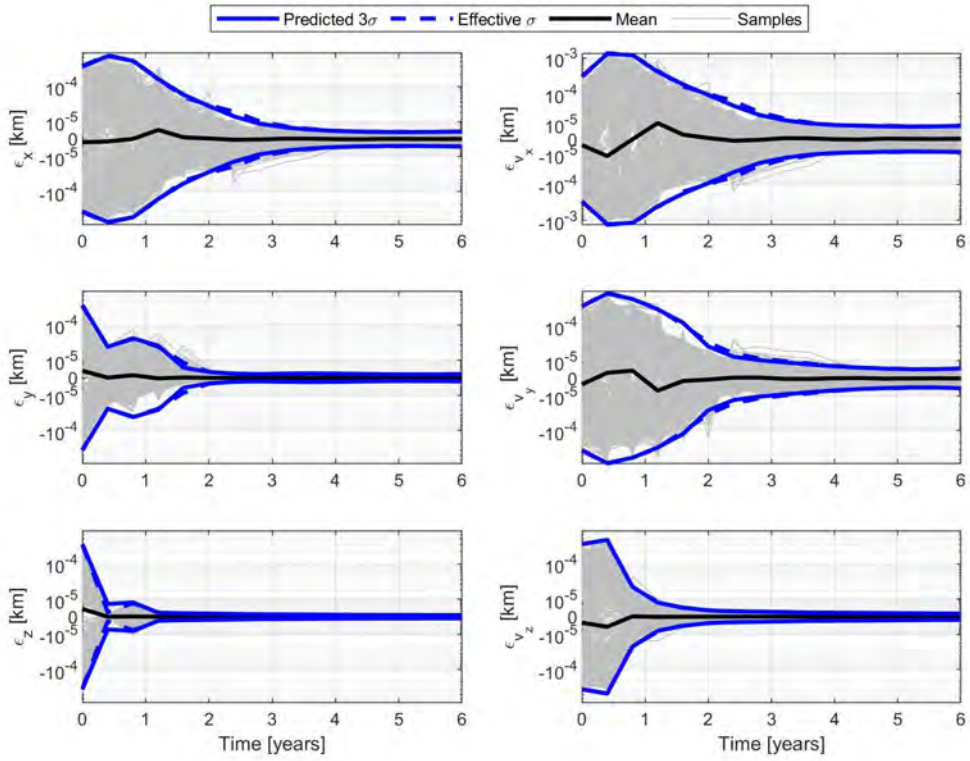


Figure 34: 500 runs Monte Carlo analysis of the KOF for a Lyapunov orbit around $\mathcal{L}_1$.

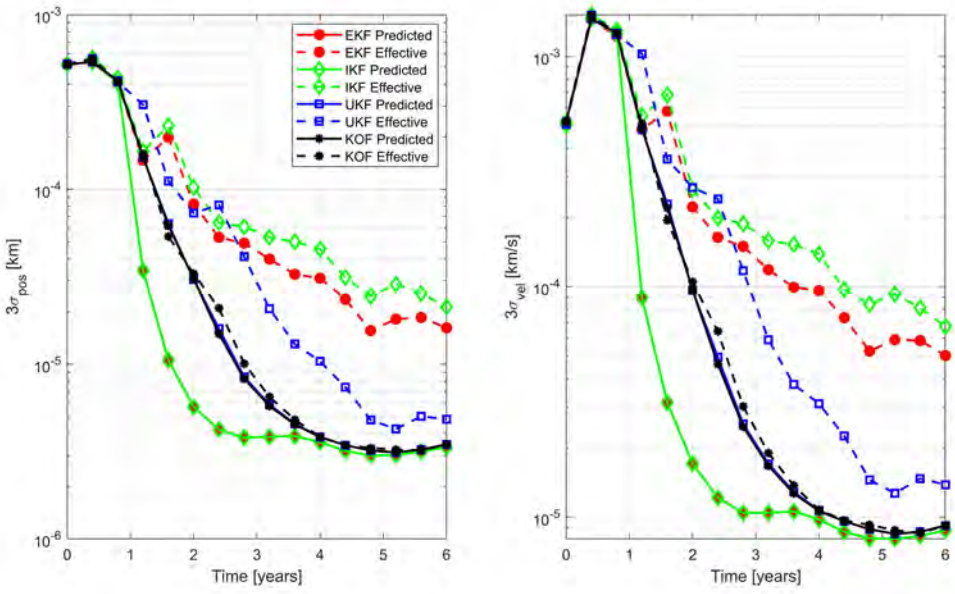


Figure 35: Position and Velocity standard deviation analysis among EKF, IKF, UKF, and KOF.

given amount of time.

3 Impacts

In this section, a brief summary of the impacts for each thrust is provided.

3.1 Large Scale Reinforcement Learning for Sensor Tasking in Space Situational Awareness

The presented work assessed the viability of utilizing deep reinforcement learning (DRL) techniques for the sensor tasking of ground-based and space-based sensors for space situational awareness. The quality of the tracked space object state estimates rapidly degrades with time due to the non-linear orbital dynamics, unmodeled dynamics, and imperfect knowledge of the non-conservative perturbation forces acting on the space object. Hence, frequent re-observation of the space objects in Earth's orbit is required in order to retain custody of the space objects.

With the ever-increasing number of space objects within Earth's orbit, space situational awareness (SSA) is becoming increasingly crucial to ensure the safety of space assets and space availability for future generations. Better sensor management algorithms are required to more efficiently task our limited resources to track the ever-growing catalog of resident space objects. Traditional sensor management algorithms tend to be myopic and only provide optimality for short-term benefits. A DRL-based sensor management algorithm is developed to capture high-level features while overcoming the curse of dimensionality inherent in the SSA sensor management problem. To this end, several custom SSA environments are constructed based on the OpenAI Gym framework with sgp4 for orbit propagation. The SSA environment uses real sensor characteristics based on the ZimSMART optical telescope and considers action slew time to achieve more realistic simulations. An Unscented Kalman Filter is used for state and covariance estimation. Various kernel density estimation models for different orbital regimes are developed to ensure consistent presence and a diverse population of RSOs in our field of regard throughout the observation window. These SSA environments can be easily adapted for different missions and objectives.

Our DRL agents were able to outperform the myopic policies in all performance criteria. We also demonstrated the robustness of the DRL agents to changes in the sensor location, sensor parameters, observation window length, number of targets to be tracked, and the orbital regimes of the targets. The performance and robustness of the DRL agents strongly motivate further research into utilizing DRL techniques for sensor tasking.

This work also provides the groundwork for the development of sensor tasking strategies for cis-lunar objects as well as enables multiple ground-based and space-based sensors to work cooperatively and efficiently for space situational awareness.

3.2 Space Object Classification via Light Curve Measurements Using Deep Learning

Developing a detailed understanding of the SO population is a fundamental goal of SSA. Such a simplified description limits the dynamic propagation model used for predicting the state of motion of SO to models that assume cannonball shapes and generic surface properties. The future SO catalog and SSA systems will have to be capable of building a detailed picture of SO characteristics. Traditional measurement sources for SO tracking, such as radar and optical, provide information on SO characteristics.

Generally, classifying SO is a challenging task. State-of-the-art methods rely on well established physical models that are embedded in an inversion scheme capable of processing the data and estimate the model parameters. We assessed the viability of a data-driven classification approach based on the Convolutional Neural Network (CNN) scheme for Space Object (SO) classification using light curve data. The classification approach determines the class from light curve observations of a SO. These classes are rocket bodies, payloads, and debris. A set of 1D-CNNs capable of ingesting light curves is separately trained on both simulated data (generated by a physics-based model) and real observed light curves and the performances are reported. It is shown that CNNs are highly accurate classifiers whenever trained on simulated data, yielding a 98% accuracy for the selected test set. However, whenever trained on simulated light curves, the CNN tends to lose the advantage over more conventional state-of-the-art machine learning methods (e.g., bagged trees, SVM). Nevertheless, CNNs significantly outperforms the other techniques whenever real data are considered. Indeed, on real light curve test sets, CNNs achieve 75% accuracy, whereas bagged trees report an accuracy of 62% and SVM an accuracy of 44%. These results are promising for the application of CNN-based methods to SSA applications.

3.3 Space Constellation for Increased Orbital Capacity

Shell-wise orbital slotting in Low Earth Orbit (LEO) can improve space safety, simplify space traffic coordination and management, and optimize orbital capacity. We looked into two different methods to generate 2D Lattice Flower Constellations (2D-LFCs) that are defined with respect to either an arbitrary degree or an arbitrary degree and order Earth geopotential. By generating shells that are quasi-periodic and frozen with respect to the Earth geopotential, it is possible to safely stack shells with vertical separation distances smaller than the osculating variation in the semi-major axis of each shell or a corresponding Keplerian 2D-LFC propagated under aspherical geopotential. This helps mitigate the single inclination per shell requirement in prior work by admitting more shells for a given orbital volume while retaining self-safe phasing in each shell. These methods exploit previous work on the Time Distribution Constellation formulation and designs of closed 2D-LFCs under arbitrary Earth geopotentials using RGT orbits.

We also identified the factors that influence the widths and shapes of these frozen shells and presented simplified formulas for estimating shell geometry and thickness. Both capacity per shell and shell thickness influence aggregate orbital capacity. Nevertheless, denser shell stacking will likely be the more critical driver as demand in LEO continues to increase. Previous work has demonstrated methods for estimating shell capacity and that large shell capacities are possible at reasonable orbital separation distances. It is shown that sequencing shells to group similar or ascending inclinations improves capacity versus arbitrary inclination ordering.

As orbit demand increases, conversations between planned orbital neighbors will become increasingly important to ensure efficient and compatible orbits. The use of common constellation-agnostic methods and schema for discussing slotting and orbit coordination will be helpful to enable interoperability and prevent unsafe states. Such analysis would require information beyond that disclosed in most regulatory filings, including mean eccentricity and eccentricity vector information, intended station-keeping control box sizes (as opposed to general and overly expansive orbital tolerances), and greater precision in distinguishing between nominal, mean, and osculating numbers in orbit disclosures. If operators or regulators wish to adopt these or similar techniques, such sharing and explicit coordination can improve the efficiency with which operators make use of orbital volume.

3.4 Deep Reinforcement Learning Techniques for SO Intent Identification and Anomaly Detection

The presented work assessed the viability of using inverse optimal control (IOC) theory and inverse reinforcement learning (RL) techniques to determine the behavior of space objects (SOs). This is achieved by estimating the reward function that an SO is using for control. The proposed approach can be used to analyze the maneuvering of SOs from observational data. The inverse RL problem is solved using maximum causal entropy. This approach determines the optimal reward function that a SO is using while maneuvering with random disturbances by assuming that the observed trajectories are optimal with respect to the SO's own reward function.

The IOC problem is solved using Pontryagin's minimum principle which results in a convex optimization problem. Three simulation test cases were studied; the first two cases study the behavior estimation problem using satellite relative motion equations, while the third case studies the behavior estimation problem for a continuous thrust orbit raising maneuver. Additionally, a general approach for modeling the SO reward function using Extreme Learning Machines (ELM) is presented. The nonlinear basis functions used for this work included polynomial terms and extreme learning machines. Good performance was shown for the first two cases, and it was observed that the accuracy of the IOC solution improves with the number of observed trajectories used. For the third and final case, the estimated control objective function approximated the true control objective function well, and good performance was shown for the proposed approach.

Two different case studies involving Low Earth Orbit (LEO) station-keeping and Geostationary Orbit (GEO) station-keeping are used to study the viability of the inverse RL technique for maneuver detection. The SO trajectories are first designed using an LQR-based cost. The developed algorithm is then used to learn the SO behavior via a maximum causal entropy inverse RL. It is shown that the learned cost function in LEO or GEO is comparable to the observed expert SO's trajectory in the presence of orbital disturbances. Thus, maximum causal entropy is attractive in learning the reward function that produces the expert SO's maneuver and can be used to estimate the behavior of SOs.

3.5 Koopman Operator Theories for Space Applications

The presented work assessed the benefits of the Koopman operator for space applications. The results, as shown in the previous chapter, are promising. The Koopman approach offers a new methodology to analyze and study autonomous systems, and, therefore, it can be applied to a multitude of different applications with various aims. Indeed, after having solved the circular restricted three body problem in the Koopman framework, the next step was to introduce uncertainties in the system, and we developed a reliable uncertainty quantification technique. Therefore, we aim to exploit the Koopman mathematics to cover a large variety of problems. The PI has already started collaborations to develop novel control techniques with the KO: in particular, work has started to achieve rendezvous between a chaser and a target satellite in an energy optimal manner via indirect control, and for bang-bang orbit transfer in the solution of the Lambert problem.

Consequently, the proposed technique opens to an interdisciplinary scenario, where the Koopman operator can be applied in robotics and mechanical engineering. Indeed, the analysis of the spectral behaviour provides the modes of the system, and the newly developing estimation and control techniques fit the application in the robotics field.

The mathematics presented in the previous pages constitutes one of the first times that the Koopman operator is derived analytically. As such, the novelty of the work guarantees the possibility of upcoming ideas, projects and collaborations, providing opportunities for research and teaching in astrodynamics. The impact of the technique is therefore multilayered, since it requires effort form the the PI, professors, researchers, postdocs and students.

In particular, the Koopman operator has a strong impact on teaching and educational experiences. The PI noticed that the scientific community is interested in the technique, in its possibilities and results, and that there was a lack of material to share the new knowledge. As such, a tutorial has been written and openly shared, comprehensive of computer code [121]. This tutorial is being shared daily world-wide to teach the Koopman technique here developed to students and everyone that is willing to approach the new methodology.

4 Personnel

The following is a list of individuals who have worked on research supported in whole or in part by the Air Force Office of Scientific Research under Grant FA9550-18-1-0115:

Prof. Richard Linares, Department of Aeronautics and Astronautics, MIT

Prof. Demoz Gebre-Egziabher, Aerospace Engineering and Mechanics, University of Minnesota

Dr. David Arnas, post-doctoral associate, MIT (Currently working as an assistant professor at the School of Aeronautics and Astronautics at Purdue University)

Dr. Simone Servadio, post-doctoral associate, MIT

Dr. Peng Mun Siew, post-doctoral associate, MIT (was previously a graduate student working on this project and completed his Ph.D. and transitioned to a post-doctoral associate position at MIT)

Dr. Bryce Doerr, post-doctoral associate, MIT (was previously a graduate student working on this project and completed his Ph.D. and transitioned to a post-doctoral associate position at MIT)

Miles Lifson, graduate student, MIT

5 Publication

The publications listed below represent papers, reports, and theses supported in whole or in part by the Air Force Office of Scientific Research under Grant FA9550-18-1-0115:

[1] B. Doerr, R. Linares, and R. Furfaro "Space objects maneuvering prediction via maximum causal entropy inverse reinforcement learning," proceedings in AIAA Scitech 2020 Forum, 2020.

[2] D. Arnas and R. Linares "Koopman operator theory applied to the J2 perturbed motion around an oblate planet," proceedings in 31st AAS/AIAA Space Flight Mechanics Meeting, 2021.

[3] D. Arnas and R. Linares, "Approximate Analytical Solution to the Zonal Harmonics Problem Using Koopman Operator Theory," Journal of Guidance, Control, and Dynamics 44.11: 1909-1923, 2021.

[4] D. Arnas, R. Linares, "A set of orbital elements to fully represent the zonal harmonics around an oblate celestial body," Monthly Notices of the Royal Astronomical Society, Volume 502, Issue 3, April 2021, Pages 4247–4261

[5] D. Arnas, R. Linares, and K. Alfriend "An Analysis of Koopman-based Perturbation Theory Applied to the Motion About an Oblate Planet," proceedings in 31st AAS/AIAA Space Flight Mechanics Meeting, 2021.

[6] D. Jang, P.M. Siew, D. Gondelach, and R. Linares "Space Situational Awareness Tasking For Narrow Field Of View Sensors: A Deep Reinforcement Learning Approach," proceedings in 2020 International Astronautical Congress, 2020.

[7] M. Lifson, D. Arnas, M. Avendaño, R. Linares "A Method for Generating Closely Packed Orbital Shells and the Implication on Orbital Capacity," arXiv preprint arXiv:2203.13354. 2022 Mar 24.

[8] P.M. Siew "Multiple Target Tracking Using Random Finite Sets," Ph.D. thesis, January 2021.

[9] P.M. Siew, D. Jang, and R. Linares "Sensor Tasking for Space Situational Awareness Using Deep Reinforcement Learning," proceedings in 2021 AAS/AIAA Astrodynamics Specialist Conference, 2021.

[10] P.M. Siew, D. Jang, T.G. Roberts, and R. Linares "Space-based Sensor Tasking using Deep Reinforcement Learning," under peer review for Journal of Astronautical Sciences, 2022.

[11] R. Furfaro, R. Linares, and V. Reddy "Space objects classification and characterization via deep learning and light curves: Applications to space traffic management," proceedings in 2019 Space Traffic Management Conference, 2019.

[12] R. Furfaro, R. Linares, and V. Reddy "Space objects classification via light curve measurements using deep convolutional neural networks," Journal of the Astronautical Sciences 67.3: 1063-1091, 2020.

[13] R. Furfaro, R. Linares, and V. Reddy "Space objects classification via light curve measurements: deep convolutional neural networks and model-based transfer learning," proceedings in 2018 AMOS Technologies Conference, Maui Economic Development Board, 2018.

[14] R. Linares, and JB. Raquepas "Physically-constrained inverse optimal control for satellite maneuver detection," proceedings in 2018 AAS/AIAA astrodynamics specialist conference, 2018.

[15] S. Servadio, D. Arnas, and R. Linares, "Dynamics Near the Three-Body Libration Points via Koopman Operator Theory," under peer review for Journal of Guidance, Control, and Dynamics, 2021.

[16] S. Servadio, R. Armellin, and R. Linares, "A Koopman-Operator Control Optimization for Relative Motion in Space," proceedings in 2023 AIAA SciTech Conference, 2023.

[17] S. Servadio, W. Parker, and R. Linares, "Uncertainty Propagation and Filtering via the Koopman Operator in Astrodynamics," proceedings in 2022 AAS/AIAA Astrodynamics Specialist Conference, 2022.

[18] S. Servadio, W. Parker, and R. Linares, "Uncertainty Propagation and Filtering via the Koopman Operator in Astrodynamics," under peer review for Journal of Guidance, Control, and Dynamics, 2022.

[19] S. Servadio, D. Arnas, and R. Linares, "Dynamics Near the Three-Body Libration Points via Koopman Operator Theory," proceedings in 31st AAS/AIAA Space Flight Mechanics Meeting, 2021.

[20] T.G. Roberts, P.M. Siew, D. Jang, and R. Linares "A Deep Reinforcement Learning Application to Space-based Sensor Tasking for Space Situational Awareness," proceedings in 2021 AMOS Technologies Conference, Maui Economic Development Board, 2021.

6 Transition

We are working with engineers at The Aerospace Corporation on the transition of our space situational awareness (SSA) simulation environment and trained deep reinforcement learning agents for sensor management of SSA sensors. The Aerospace Corporation's point of contact is Richard Y. Yau (richard.y.yau@aero.org)

References

[1] W. House, "National space policy of the united states of america," *Retreived from https://www.whitehouse.gov/sites/default/files/national_space_policy_6–28–10.pdf*, 2010.

[2] D. F. Woods, R. Shah, J. A. Johnson, A. Szabo, E. C. Pearce, R. L. Lambour, and W. J. Faccenda, "Space surveillance telescope: focus and alignment of a three mirror telescope," *Optical Engineering*, vol. 52, no. 5, p. 053604, 2013.

[3] M. Hart, S. Jefferies, D. Hope, J. Nagy, O. Durney, R. Codona, and S. Williams, "Quantitative measurements of daytime near infrared sky brightness at the aeos 3.6 m telescope," AIR FORCE RESEARCH LAB KIHEI MAUI HI DETACHMENT 15, Tech. Rep., 2014.

[4] D. Hall, B. Calef, K. Knox, M. Bolden, and P. Kervin, "Separating attitude and shape effects for non-resolved objects," in *The 2007 AMOS Technical Conference Proceedings*. Maui Economic Development Board, Inc. Kihei, Maui, HI, 2007, pp. 464–475.

[5] M. Jah and R. A. Madler, "Satellite characterization: angles and light curve data fusion for spacecraft state and parameter estimation," in *Proceedings of the advanced Maui optical and space surveillance technologies conference*, vol. 49, 2007.

[6] R. Linares, M. A. Shoemaker, A. C. Walker, P. M. Mehta, D. Palmer, D. C. Thompson, J. Koller, and J. L. Crassidis, "Photometric data from non-resolved objects for space object characterization and improved atmospheric modeling," Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 2013.

[7] R. Linares, M. K. Jah, J. L. Crassidis, F. A. Leve, and T. Kelecy, "Astrometric and photometric data fusion for inactive space object mass and area estimation," *Acta Astronautica*, vol. 99, pp. 1–15, 2014.

[8] M. J. Holzinger, K. T. Alfriend, C. J. Wetterer, K. K. Luu, C. Sabol, and K. Hamada, "Photometric attitude estimation for agile space objects with shape uncertainty," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 3, pp. 921–932, 2014.

[9] R. Linares and J. L. Crassidis, "Resident space object shape inversion via adaptive hamiltonian markov chain monte carlo," in *AAS/AIAA Space Flight Mechanics Meeting*, 2016, pp. 2016–514.

[10] C. J. Wetterer, R. Linares, J. L. Crassidis, T. M. Kelecy, M. K. Ziebart, M. K. Jah, and P. J. Cefola, "Refining space object radiation pressure modeling with bidirectional reflectance distribution functions," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 1, pp. 185–196, 2014.

[11] R. Linares, M. K. Jah, J. L. Crassidis, and C. K. Nebelecky, "Space object shape characterization and tracking using light curve and angles data," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 1, pp. 13–25, 2014.

[12] C. M. Kreucher, K. D. Kastella, and A. O. Hero III, "Information-based sensor management for multitarget tracking," in *Signal and Data Processing of Small Targets 2003*, vol. 5204. International Society for Optics and Photonics, 2004, pp. 480–489.

[13] C. Kwok and D. Fox, "Reinforcement learning for sensing strategies," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 4. IEEE, 2004, pp. 3158–3163.

[14] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal processing magazine*, vol. 19, no. 2, pp. 61–72, 2002.

[15] S. Crary and Y. Jeong, "Bayesian optimal design of experiments for sensor calibration," in *Proceedings of the International Solid-State Sensors and Actuators Conference-TRANSDUCERS'95*, vol. 2. IEEE, 1995, pp. 48–51.

[16] V. Gupta, T. H. Chung, B. Hassibi, and R. M. Murray, "On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage," *Automatica*, vol. 42, no. 2, pp. 251–260, 2006.

[17] Y. Oshman, "Optimal sensor selection strategy for discrete-time state estimators," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, no. 2, pp. 307–314, 1994.

[18] B. Grocholsky, "Information-theoretic control of multiple sensor platforms," 2002.

[19] C. Stachniss, "Exploration and mapping with mobile robots," Ph.D. dissertation, Citeseer, 2006.

[20] T. Alpcan and I. Shames, "An information-based learning approach to dual control," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 11, pp. 2736–2748, 2015.

[21] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[22] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[23] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Advances in neural information processing systems*, 2009, pp. 1096–1104.

[24] M. Jaderberg, V. Dalibard, S. Osindero, W. M. Czarnecki, J. Donahue, A. Razavi, O. Vinyals, T. Green, I. Dunning, K. Simonyan *et al.*, "Population based training of neural networks," *arXiv preprint arXiv:1711.09846*, 2017.

[25] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6765–6816, 2017.

[26] S. Falkner, A. Klein, and F. Hutter, "Bohb: Robust and efficient hyperparameter optimization at scale," *arXiv preprint arXiv:1807.01774*, 2018.

[27] L. Li, K. Jamieson, A. Rostamizadeh, E. Gonina, M. Hardt, B. Recht, and A. Talwalkar, "Massively parallel hyperparameter tuning," *arXiv preprint arXiv:1810.05934*, 2018.

[28] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[29] T. Zahavy, M. Haroush, N. Merlis, D. J. Mankowitz, and S. Mannor, "Learn what not to learn: Action elimination with deep reinforcement learning," in *Advances in Neural Information Processing Systems*, 2018, pp. 3562–3573.

[30] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[31] R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez, and I. Stoica, "Tune: A research platform for distributed model selection and training," *arXiv preprint arXiv:1807.05118*, 2018.

[32] J. Herzog, "Cataloguing of objects on high and intermediate altitude orbits," Ph.D. dissertation, Universität Bern, 2013.

[33] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[34] S. S. Blackman, "Multiple-target tracking with radar applications," *Dedham*, 1986.

[35] R. Linares, M. K. Jah, and J. L. Crassidis, "Inactive space object shape estimation via astrometric and photometric data fusion," *Adv. Astronaut. Sci.*, vol. 143, pp. 217–232, 2012.

[36] ——, "Space object area-to-mass ratio estimation using multiple model approaches," *Advances in the Astronautical Sciences*, vol. 144, pp. 55–72, 2012.

[37] J. C. Hinks, R. Linares, and J. L. Crassidis, "Attitude observability from light curve measurements," in *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2013, p. 5005.

[38] D. Gaylor and J. Anderson, "Use of hierarchical mixtures of experts to detect resident space object attitude," in *Advanced Maui Optical and Space Surveillance Technologies Conference*, vol. 1, 2014, p. 70.

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[40] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in neural information processing systems*, 2014, pp. 487–495.

[41] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[42] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*. Springer, 2000, pp. 1–15.

[43] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[44] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[45] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[46] M. Ashikhmin and P. Shirley, "An anisotropic phong brdf model," *Journal of graphics tools*, vol. 5, no. 2, pp. 25–32, 2000.

[47] G. Beskin, S. Karpov, A. Biryukov, S. Bondar, E. Ivanov, E. Katkova, N. Orekhova, A. Perkov, and V. Sasyuk, "Wide-field optical monitoring with mini-megatortora (mmt-9) multichannel high temporal resolution telescope," *Astrophysical Bulletin*, vol. 72, no. 1, pp. 81–92, 2017.

[48] S. Alfano, D. L. Oltrogge, and R. Shepperd, "Leo constellation encounter and collision rate estimation: an update," in *2nd IAA Conference on Space Situational Awareness (ICSSA), Washington DC, IAA-ICSSA-20-0021*, vol. 15. ICSSA, 2020.

[49] E. Cutting, J. Frautnick, and G. Born, "Orbit analysis for seasat-a," *Journal of the Astronautical Sciences*, vol. 26, pp. 315–342, 1978.

[50] M. Rosengren, "The orbit control of ers-1 and ers-2 for a very accurate tandem configuration," *RBCM*, vol. 21, pp. 72–78, 1999.

[51] R. B. Frauenholz, R. S. Bhat, B. E. Shapiro, and R. K. Leavitt, "Analysis of the topex/poseidon operational orbit: Observed variations and why," *Journal of spacecraft and rockets*, vol. 35, no. 2, pp. 212–224, 1998.

[52] D. A. Vallado, *Fundamentals of astrodynamics and applications*. Springer Science & Business Media, 2001, vol. 12.

[53] C. Bombardelli, G. Falco, D. Amato, and A. J. Rosengren, "Space occupancy in low-earth orbit," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 4, pp. 684–700, 2021.

[54] N. Reiland, A. J. Rosengren, R. Malhotra, and C. Bombardelli, "Assessing and minimizing collisions in satellite mega-constellations," *Advances in Space Research*, vol. 67, no. 11, pp. 3755–3774, 2021.

[55] M. E. Avendaño, J. J. Davis, and D. Mortari, "The 2-d lattice theory of flower constellations," *Celestial Mechanics and Dynamical Astronomy*, vol. 116, no. 4, pp. 325–337, 2013.

[56] D. Arnas, M. Lifson, R. Linares, and M. E. Avendaño, "Definition of low earth orbit slotting architectures using 2d lattice flower constellations," *Advances in Space Research*, vol. 67, no. 11, pp. 3696–3711, 2021.

[57] D. Arnas and R. Linares, "Non-self-intersecting trajectories and their applications to satellite constellation design and orbital capacity," *arXiv preprint arXiv:2110.07823*, 2021.

[58] M. Avendaño, D. Arnas, R. Linares, and M. Lifson, "Efficient search of optimal flower constellations," *Acta Astronautica*, vol. 179, pp. 290–295, 2021.

[59] D. Arnas and R. Linares, "Uniform satellite constellation reconfiguration," *Journal of Guidance, Control, and Dynamics*, pp. 1–14, 2022.

[60] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 1.

[61] N. D. Ratliff, J. A. Bagnell, and M. A. Zinkevich, "Maximum margin planning," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 729–736.

[62] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning." in *Aaai*, vol. 8. Chicago, IL, USA, 2008, pp. 1433–1438.

[63] B. D. Ziebart, J. A. Bagnell, and A. K. Dey, "Modeling interaction via the principle of maximum causal entropy," 2010.

[64] B. Ziebart, A. Dey, and J. A. Bagnell, "Probabilistic pointing target prediction via inverse optimal control," in *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*. ACM, 2012, pp. 1–10.

[65] M. Monfort, A. Liu, and B. Ziebart, "Intent prediction and trajectory forecasting via predictive inverse linear-quadratic regulation," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[66] S. Levine, Z. Popovic, and V. Koltun, "Nonlinear inverse reinforcement learning with gaussian processes," in *Advances in Neural Information Processing Systems*, 2011, pp. 19–27.

[67] M. Wulfmeier, P. Ondruska, and I. Posner, "Maximum entropy deep inverse reinforcement learning," *arXiv preprint arXiv:1507.04888*, 2015.

[68] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Advances in Neural Information Processing Systems*, 2016, pp. 4565–4573.

[69] R. E. Kalman, "When is a linear control system optimal?" *Journal of Basic Engineering*, vol. 86, no. 1, pp. 51–60, 1964.

[70] M. Masak, "An inverse problem on decoupling optimal control systems," *IEEE Transactions on Automatic Control*, vol. 13, no. 1, pp. 109–110, 1968.

[71] P. Moylan and B. Anderson, "Nonlinear regulator theory and an inverse optimal control problem," *IEEE Transactions on Automatic Control*, vol. 18, no. 5, pp. 460–465, 1973.

[72] A. Y. Ng, S. J. Russell *et al.*, "Algorithms for inverse reinforcement learning." in *Icml*, vol. 1, 2000, p. 2.

[73] C. Finn, S. Levine, and P. Abbeel, "Guided cost learning: Deep inverse optimal control via policy optimization," in *International conference on machine learning*, 2016, pp. 49–58.

[74] M. Kuderer, S. Gulati, and W. Burgard, "Learning driving styles for autonomous vehicles from demonstration," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 2641–2646.

[75] J. Mainprice, R. Hayne, and D. Berenson, "Goal set inverse optimal control and iterative replanning for predicting human reaching motions in shared workspaces," *IEEE Transactions on Robotics*, vol. 32, no. 4, pp. 897–908, 2016.

[76] M. Krstic and P. Tsiotras, "Inverse optimal stabilization of a rigid spacecraft," *IEEE Transactions on Automatic Control*, vol. 44, no. 5, pp. 1042–1049, 1999.

[77] A. Keshavarz, Y. Wang, and S. Boyd, "Imputing a convex objective function," in *2011 IEEE International Symposium on Intelligent Control*. IEEE, 2011, pp. 613–619.

[78] A.-S. Puydupin-Jamin, M. Johnson, and T. Bretl, "A convex approach to inverse optimal control and its application to modeling human locomotion," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 531–536.

[79] N. Aghasadeghi, A. Long, and T. Bretl, "Inverse optimal control for a hybrid dynamical system with impacts," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 4962–4967.

[80] A. V. Terekhov, Y. B. Pesin, X. Niu, M. L. Latash, and V. M. Zatsiorsky, "An analytical approach to the problem of inverse optimization with additive objective functions: an application to human prehension," *Journal of mathematical biology*, vol. 61, no. 3, pp. 423–453, 2010.

[81] M. Johnson, N. Aghasadeghi, and T. Bretl, "Inverse optimal control for deterministic continuous-time nonlinear systems," in *52nd IEEE Conference on Decision and Control*. IEEE, 2013, pp. 2906–2913.

[82] K. Mombaur, A. Truong, and J.-P. Laumond, "From human to humanoid locomotion—an inverse optimal control approach," *Autonomous robots*, vol. 28, no. 3, pp. 369–383, 2010.

[83] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[84] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006.

[85] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *Neural Networks, IEEE Transactions on*, vol. 17, no. 4, pp. 879–892, 2006.

[86] E. T. Jaynes, "Information theory and statistical mechanics," *Physical review*, vol. 106, no. 4, p. 620, 1957.

[87] G. Kramer, *Directed information for channels with feedback*. Hartung-Gorre, 1998.

[88] F. J. De Bruijn, S. Theil, D. Choukroun, and E. Gill, "Geostationary satellite station-keeping using convex optimization," *Journal of Guidance, Control, and Dynamics*, no. null, pp. 605–616, 2015.

[89] J. L. Junkins and P. Singla, "How nonlinear is it? a tutorial on nonlinearity of orbit and attitude dynamics," 2004.

[90] G. Terejanu, P. Singla, T. Singh, and P. D. Scott, "Uncertainty propagation for nonlinear dynamic systems using gaussian mixture models," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 6, pp. 1623–1633, 2008.

[91] J. T. Horwood, N. D. Aragon, and A. B. Poore, "Gaussian sum filters for space surveillance: theory and simulations," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 6, pp. 1839–1851, 2011.

[92] R. Madankan, P. Singla, T. Singh, and P. D. Scott, "Polynomial-chaos-based bayesian approach for state and parameter estimations," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 4, pp. 1058–1074, 2013.

[93] J. E. Hurtado and A. J. Sinclair, "Using carleman embedding to discover a system's motion constants," *Advances in the Astronautical Sciences*, vol. 147, no. 5, pp. 475–482, 2012.

[94] B. O. Koopman, "Hamiltonian systems and transformation in hilbert space," *Proceedings of the National Academy of Sciences*, vol. 17, no. 5, pp. 315–318, 1931.

[95] I. Mezić, "Analysis of fluid flows via spectral properties of the koopman operator," *Annual Review of Fluid Mechanics*, vol. 45, pp. 357–378, 2013.

[96] I. Mezić and A. Banaszuk, "Comparison of systems with complex behavior," *Physica D: Nonlinear Phenomena*, vol. 197, no. 1-2, pp. 101–133, 2004.

[97] I. Mezic, "Koopman operator, geometry, and learning," *arXiv preprint arXiv:2010.05377*, 2020.

[98] M. Budišić, R. Mohr, and I. Mezić, "Applied koopmanism," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 22, no. 4, p. 047510, 2012.

[99] I. Mezić, "Spectral properties of dynamical systems, model reduction and decompositions," *Nonlinear Dynamics*, vol. 41, no. 1-3, pp. 309–325, 2005.

[100] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data–driven approximation of the koopman operator: Extending dynamic mode decomposition," *Journal of Nonlinear Science*, vol. 25, no. 6, pp. 1307–1346, 2015.

[101] D. Arnas and R. Linares, "Approximate analytical solution to the zonal harmonics problem using koopman operator theory," *Journal of Guidance, Control, and Dynamics*, 2021, accepted for publication.

[102] A. Surana, "Koopman operator based observer synthesis for control-affine nonlinear systems," in *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE, 2016, pp. 6492–6499.

[103] J. v. Neumann, "Zur operatorenmethode in der klassischen mechanik," *Annals of Mathematics*, pp. 587–642, 1932.

[104] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz, "Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control," *PloS one*, vol. 11, no. 2, p. e0150171, 2016.

[105] H. Nakao and I. Mezić, "Spectral analysis of the koopman operator for partial differential equations," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 30, no. 11, p. 113131, 2020.

[106] J. N. Kutz, J. L. Proctor, and S. L. Brunton, "Koopman theory for partial differential equations," *arXiv preprint arXiv:1607.07076*, 2016.

[107] R. Linares, "Koopman operator theory applied to the motion of satellites," *Advances in the Astronautical Sciences*, vol. 171, 2019.

[108] D. Arnas and R. Linares, "Approximate analytical solution to the zonal harmonics problem using koopman operator theory," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 11, pp. 1909–1923, 2021.

[109] S. Servadio, D. Arnas, and R. Linares, "Dynamics near the three-body libration points via koopman operator theory," *arXiv preprint arXiv:2110.12119*, 2021.

[110] R. Linares, "Koopman operator theory applied to the motion of satellites," in *Advances in the Astronautical Sciences*, vol. 171. AAS/AIAA, 2019, pp. AAS 19–821.

[111] D. Arnas, R. Linares, and K. Alfriend, "An analysis of koopman-based perturbation theory applied to the motion about an oblate planet," in *31st AAS/AIAA Space Flight Mechanics Meeting*, 2021.

[112] D. Arnas and R. Linares, "Koopman operator theory applied to the j2 perturbed motion around an oblate planet," in *31st AAS/AIAA Space Flight Mechanics Meeting, AAS/AIAA*, 2021.

[113] H. Schaub and K. T. Alfriend, "J 2 invariant relative orbits for spacecraft formations," *Celestial Mechanics and Dynamical Astronomy*, vol. 79, no. 2, pp. 77–95, 2001.

[114] D. Arnas and R. Linares, "A set of orbital elements to fully represent the zonal harmonics around an oblate celestial body," *Monthly Notices of the Royal Astronomical Society*, vol. 502, no. 3, pp. 4247–4261, 2021.

[115] D. Brouwer, "Solution of the problem of artificial satellite theory without drag," YALE UNIV NEW HAVEN CT NEW HAVEN United States, Tech. Rep., 1959.

[116] Y. Kozai, "The motion of a close earth satellite," *The Astronomical Journal*, vol. 64, p. 367, 1959.

[117] A. Deprit, "Canonical transformations depending on a small parameter," *Celestial mechanics*, vol. 1, no. 1, pp. 12–30, 1969.

[118] J. Liu, "Satellite motion about an oblate earth," *AIAA Journal*, vol. 12, no. 11, pp. 1511–1516, 1974.

[119] M. J. Holzinger, D. J. Scheeres, and R. S. Erwin, "On-orbit operational range computation using gauss's variational equations with j 2 perturbations," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 2, pp. 608–622, 2014.

[120] B. Bamieh, "A tutorial on matrix perturbation theory (using compact matrix notation)," *arXiv preprint arXiv:2002.05001*, 2020.

[121] S. Servadio, D. Arnas, and R. Linares, "A koopman operator tutorial with othogonal polynomials," *arXiv preprint arXiv:2111.07485*, 2021.