TECHNOLOGY FOR LOW RESOLUTION SPACE BASED RSO DETECTION AND CHARACTERISATION

RYAN WILLIAM CLARK

A DISSERTATION SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

GRADUATE PROGRAMME IN EARTH AND SPACE SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO

DECEMBER 2022

Abstract

Space Situational Awareness (SSA) refers to all activities to detect, identify and track objects in Earth orbit. SSA is critical to all current and future space activities and protect space assets by providing access control, conjunction warnings, and monitoring status of active satellites. Currently SSA methods and infrastructure are not sufficient to account for the proliferations of space debris. In response to the need for better SSA there has been many different areas of research looking to improve SSA most of the requiring dedicated ground or space-based infrastructure. In this thesis, a novel approach for the characterisation of RSO's (Resident Space Objects) from passive low-resolution space-based sensors is presented with all the background work performed to enable this novel method. Low resolution space-based sensors are common on current satellites, with many of these sensors being in space using them passively to detect RSO's can greatly augment SSA with out expensive infrastructure or long lead times. One of the largest hurtles to overcome with research in the area has to do with the lack of publicly available labelled data to test and confirm results with. To overcome this hurtle a simulation software, ORBITALS, was created. To verify and validate the ORBITALS simulator it was compared with the Fast Auroral Imager images, which is one of the only publicly available low-resolution space-based images found with auxiliary data. During the development of the ORBITALS simulator it was found that the generation of these simulated images are computationally intensive when propagating the entire space catalog. To overcome this an upgrade of the currently used propagation method, Specialised General Perturbation Method 4th order (SGP4), was performed to allow the algorithm to run in parallel reducing the computational time required to propagate entire catalogs of RSO's. From the results it was found that the standard facet model with a particle swarm optimisation performed the best estimating an RSO's attitude with a 0.66 degree RMSE accuracy across a sequence, and ~1% MAPE accuracy for the optical properties. This accomplished this thesis goal of demonstrating the feasibility of low-resolution passive RSO characterisation from space-based platforms in a simulated environment.

Acknowledgements

There are many people who with out their support, big or small, I would not have been able to complete my thesis and become who I am today. In this acknowledgment section I would first like to thank my external partners; DRDC, Magellan Aerospace, and C-Core. Through there mentorship, expertise, and willingness to collaborate I was able to develop the skills required to compete this research. Secondly, I would like to thank supervisor Regina Lee for providing me the opportunity, tools, and guidance to perform this research. Next, I would like to thank my colleagues at the nanosatellite laboratory who have always been supportive through the years of research. Lastly, I would like to focus on my biggest supporter and role model I have had in my life, my dad. Ever since I can remember my dad's unwavering support in my interests has enabled me to foster the curiosity and courage to explore and grow as a person. Over the years as I decided I wanted to pursue research in the space sector my dad, while sceptical on if it was a sustainable career path, supported me every stage of the way. From late night phone calls explaining the troubles I was facing to celebrating every small victory along the way, my dad played an active role being there for support, advice, and just being a friend. With out my dad's support I can easily say I would not have been able to complete my thesis and I wouldn't be half the man I am today. I will never forget the quote he told me about being a father and a role model.

> *"Being a father/ (role model) is not about being a captain leading a ship to shore, rather it is about being a lighthouse. Always being there to light up the darkness on a stormy day, giving the vision to avoid the obstacles in your way, and being a familiar warmth on the horizon welcoming you home."*

Dad this thesis is dedicated to you, I hope that one day I can become even half as wise and caring as you are.

Table of Contents

Table of Tables

Table of Figures

Table 1: Table of Abbreviations

Abbreviations	
COTS	Commercial of the Shelf
CubeSat	Cube Satellite
RSO	Resident Space Object
SSN	Space Surveillance Network
FOV	Field of View
ECI	Earth Centered Inertial
ORF	Orbital Reference Frame
BFF	Body Fixed Frame
RAAN	Right Ascension of the Ascending Node
SGP4	Special Generalized Perturbation Model 4
SNR	Signal to Noise Ratio
EEA	Earth Exclusion Angle

SEA	Sun Exclusion Angle
MEA	Moon Exclusion Angle
TRL	Technology Readiness Level
GLARED	Graphically Learned Algorithm for RSO Enhanced Detection
STARED	Star Tracker Analytic RSO Enhanced Detector
HPOP	High Precision Orbital Propagator
SDP4	Simplified Deep Space Perturbations Model 4
FOM	Figure of Merit
RK4	Runge Kutta 4th Order
DP54	Dorman Prince 5(4)
LA	Look Ahead Algorithm
PD	Priority Dispatch Algorithm
EA	Evolutionary Algorithm
GRA	Greedy Algorithm
GA	Genetic Algorithm
HC	Hill Climbing Algorithm
SHC	Stochastic Hill Climbing Algorithm
SA	Simulated Annealing Algorithm
TA	Tabu Search Algorithm
HGA	Hybrid Genetic Algorithm
PMX	Partially Mapped Crossover
OX	Ordered Crossover
PBX	Position Based Crossover
OBX	Order Based Crossover

CX	Cyclic Crossover
SEX	Subsequence Exchange Crossover
TSP	Travelling Salesman Problem
JSP	Job Shop Problem
JBOX	Job Based Ordered Crossover
PSEX	Partial Schedule Exchange Crossover
SSEX	Substring Exchange Crossover
GTX	Giffler and Thompson Algorithm-Based Crossover
PDM	Phase Dispersion Minimization
SIMD	Single Instruction Multiple Data
GPU	Graphic Processing Unit
PSF	Point Spread Function

Table 2: Table of Common Variables

Common Variables	
X_e	X coordinate in the ECI reference frame
Y_e	Y coordinate in the ECI reference frame
Z_e	Z coordinate in the ECI reference frame
μ	Standard gravitational parameter
R	Distance from the center of Earth
h	Specific relative angular momentum
e	Eccentricity
θ	True Anomaly
a	Semi major axis

i	Inclination
ω	Argument of perigee
O	Right ascension of the ascending node
M	Mean anomaly
E	Eccentric anomaly
N_o	Mean motion
P_{J2x}	The perturbing force in the J2 propagation model in the x axis of ECI coordinate system.
P_{J2y}	The perturbing force in the J2 propagation model for the y axis of ECI.
P_{J2z}	The perturbing force in the J2 propagation model for the z axis of ECI.
J_2	The second unnormalized zonal harmonic of Earth's gravity.
R_E	The mean radius of the Earth.
SNR	The signal to noise ratio of a signal.
θ_{RSOn}	The angle between the RSO's position and the normal of the Star Tracker
θ_{FOV}	The half conic field of view of the star tracker.
θ_{EEA}	The Earth exclusion angle for a star tracker.
θ_{SEA}	The Sun exclusion angle for a star tracker.
e_s	The number of RSO signal photoelectrons.
δ_d	The dark current noise.
δ_r	The signal read noise.

δ_{ss}	The signal shot noise.
δ_{bs}	The background shot noise.
N_{DC}	The number of dark current photoelectrons per time.
t_{int}	The integration time of the star tracker.
e_{ss}	The number of signal shot noise photoelectrons.
e_{bs}	The number of background shot noise photoelectrons.
y_n	The known initial value of the dependent variable.
y_{n+1}	The state of the dependent variable being solved for.
x_n	The known initial value of the dependent variable.
t_n	The known initial value of the time variable.
k_i	The estimated slope of the at the intermediate point i.
$a_{i,j}$	The contribution of the estimated intermediate point slopes, represented by j, to calculate the next intermediate point i.
b_j	The contribution of the estimated intermediate point slopes, represented by j, to find the next state of the depend variable.
c_i	The coefficient of the step size that controls where the intermittent points are in the time domain.

b_j^*	The contribution of the estimated intermediate point slopes, represented by j, to find the next state of the depend variable. The star represents these coefficients being one order higher than b_j.
y_{n+1}^*	The state of the dependent variable being solved for. The star represents this estimate being one order higher than y_{n+1}.
O_{n+1}	The error between the estimates of different orders.
$atol$	The absolute tolerance used in computing the next size of the time step. Acts as an offset with the relative tolerance to create a linear function of the acceptable error.
$rtol$	The relative tolerance which acts as a coefficient of $\|y_{n+1}\|$ used to estimate the acceptable error as a linear function of the dependant variable.
TR_{n+1}	The tolerance ratio used to calculate the next step size for adaptive step size numerical methods.
h_{n+1}	The step size from point n to $n+1$ for adaptive numerical methods.
S	The safety factor used to calculate adaptive step sizes. It is implemented so that the step size does not get stuck in unstable equilibrium points.

$\tilde{y}$	The mean of the independent variable along for a subset of all y.
i	The counter for different iterative methods, calculated differently depending on what needs to be iterated.
f_{low}	The lower bound of frequency being searched over for light curve analysis methods, inputted by the user.
f_{high}	The upper bound of frequency being searched over for light curve analysis methods, inputted by the user.
f_{step}	The step size between frequencies being searched for light curve analysis methods, inputted by the user.
$f(i)$	The frequency calculated to be used to fold the light curve.
$F(t)$	The time series of the brightness representing the light curve.
σ_T^2	The total variance of a light curve.
σ_S^2	The total variance of the folded light curve.
σ_j^2	The variance of discrete point j.
Θ_{PDM}	The fit ratio for PDM, represented with a value from 0 to 1 with 0 representing a perfect fit.

Θ_{LSSA}	The fit ratio for LSSA, represented with a value from 0 to 1 with 0 representing a perfect fit.
$\hat{r}_{F(t),i}$	The residual value of the real light curve vs I which is a hold variable for what method is being compared vs real light curve. An example would be $\hat{r}_{F(t),LSSA}$ which represents the residual from LSSA method.
$LSSA(\omega_j, t_i)$	The estimates value from LSSA of the light curve where ω_j represents the j estimate of the frequency, and t_i represents the time of the control points.
Φ	The design matrix used for least squares analysis.
C_f^{-1}	The covariance matrix used for least squares analysis.
$\hat{x}_j$	The estimated parameters used in least squares analysis. For LSSA it represents the amplitudes of the sinusoids with frequency ω_j.
Φ_p	The phase angle between the sun, RSO and star tracker, also known as the solar phase angle.
ϕ_1	The latitude of the sun in the Satellite Orbital Frame.
ϕ_2	The latitude of the star tracker in the Satellite Orbital Frame.

θ_{lon}	The difference in longitude of the Sun and Star Tracker in the Satellite Orbital Frame.
Δ	Reflected fan of light set to be the diameter of the sun, 0.0093 rad at Earth.
$\boldsymbol{F}$	The matrix of the light curve containing all the brightness values.
A	Represents the design matrix used for light curve inversion
$\boldsymbol{s}$	Matrix of the effective area of the facets that make up the target RSO.
$\boldsymbol{N}_{i,j}$	The dot product between the normal vectors i and j.
EA_j	The effective area of the facet j on the target RSO.
χ_p^2	The minimal residual found from light curve inversion which corresponds to the best set of effective facet areas.
χ_F^2	The minimal residual found from the scaling of the best-found answer from light curve inversion. The residual corresponds to the best set of effective facet areas.
$\boldsymbol{F}_{LCI}$	The brightness estimates of the true light curve found from using the effective areas of the facets found from light curve inversion.

c_{scale}	The scale factor between the estimated light curve and true light curve.
α_j	The reflectivity in the visible spectrum of facet j.
$P\left(\Phi_p,\right)$	The phase function of the object calculated using the phase angle.
M_{obj}	The brightness of the object being examined, for RSO simulation this is the RSO in question.
M_{ref}	The brightness of the reference source of light, for RSO simulation this is the sun.
$d_{i,j}$	The distance from object i to get to object j.
M_{sun}	The brightness of the sun.
X_{Topo}	Represents the X axis in the topocentric coordinate system.
Y_{Topo}	Represents the Y axis in the topocentric coordinate system.
Z_{Topo}	Represents the Z axis in the topocentric coordinate system.
X_{Teme}	Represents the X axis in the TEME coordinate system.
Y_{Teme}	Represents the Y axis in the TEME coordinate system.
Z_{Teme}	Represents the Z axis in the TEME coordinate system.

MEA	Represents the Minimum Elevation Angle that a ground station can use to communicate with a Satellite.
ε	Represents the orbital energy of the RSO in question.

Table 3: Table of Constant Values

Constant Values		
Constant	**Value**	**Units**
μ	398600.8	km^3 s^{-2}
J_2	1.082616 x 10^3	None
R_E	6378.135	km
M_{sun}	-26.73	None

1 Introduction

Since the first satellite launch of Sputnik 1 in 1957 the barriers to access space have become easier to overcome. Currently there has been around 8950 active satellite put into orbit [1] with the launching of satellites becoming more and more common. The substantial increase in satellites has largely come in the last few decades from the leaps that have happened in space enabling technology, allowing for smaller and low-cost CubeSat™ (class of nanosatellites). Two of the more significant ways that CubeSat, and their larger subset of nanosatellites (spacecraft with less than 10 kg mass), differ from conventional satellites is their small size and weight allows them to act as secondary payloads on launch vehicles. This has the effect of reducing the cost of launch, as well as, increasing the flexibility of launch opportunities. The second significant difference between CubeSat's and conventional satellites is the large integration of Commercial of The Shelf (COTS) components. The integration of COTS parts gives CubeSat's the advantage of reducing the cost and lead time of special components, with these properties CubeSat's have been leveraged to enable them to be low cost, rapid-response, and suited to various applications [2] [3] [4]. In recent years the Nanosatellite industry has exploded with it estimated to be worth $566.2 million USD by 2022 [5] having companies now dedicated to making COTS products for CubeSat applications. With this growing field there are new promising trends such as mega constellations; constellations are groups of similar satellites that work together to accomplish a goal. Mega constellations refer to extremely large constellations of smaller satellites, one example is StarLink which is looking to launch over 10,000+ satellites to provide satellite-based internet services. With the proliferation of Resident Space Objects (RSO's) in Near Earth Orbit new problems have started arising, the most significant problem is the proliferation of RSO's, specifically space debris, which has caused an increased need for Space Situational Awareness (SSA) [6] [7]. An RSO is any object that orbits around the earth, both man made and naturally occurring. Space debris are any objects that are a non active satellite still orbiting around Earth. Some examples of space debris include old rocket bodies, dead satellites, nuts, bolts, and fragmentation debris from collisions. The growing space debris problem has been well documented in recent literature with current SSA assets being identified as inadequate for the rising need, this will only lead to the increase in the 3 C's of space; Congested, Contested, and Competitive [8]. Leveraging components on current and future space assets, such as star trackers and low-resolution imagers, can greatly augment current SSA efforts with out the need for expensive dedicated infrastructure. This thesis looks to present a novel method to leverage the large increase in CubeSat class satellites with COTS imagers to enable low resolution imagery for passive RSO detections. Passive RSO detections allow for low resolution images to perform their primary mission while contributing to SSA through observations of opportunity. Implementing this on CubeSat class satellites transforms the burden on SSA of the proliferation of CubeSat's to actively increasing SSA. In this chapter, we describe the background information and recent trends related to each topic in this thesis followed by the explanation of the Motivation for the current work on Technology for Low Resolution Space Based RSO Detection and Characterisation. The motivation is then followed by the goals of this thesis, the outline of this thesis, and this chapter is finished with a list of publications.

1.1 Nanosatellites, Mega Constellations, and the Proliferation of Space Debris

The rise in the use of CubeSat™ and nanosatellites has lowered the barrier for entry making space more accessible, with the first cube satellite launched in 2003. A satellite type is based on its weight, Table 4 [9] [10] shows the different classifications of satellites based on mass. CubeSat's™ are well defined structures that fit in the nanosatellite category. CubeSat's™ have fixed dimensions instead of weight, with the unofficial classifications being shown in Table 5 [11].

Table 4 Classification of Satellites Based on Weight [9]

Name of satellite type:	**Weight:**
Large	> 1000 Kg
Medium	500 Kg to 1000 Kg
Small	< 500 Kg
Subsets of Small Satellites:	**Weight:**
Mini	100 Kg to 500 Kg
Micro	10 Kg to 100 Kg
Nano	1 Kg to 10 Kg
Pico	100 g to 1 Kg
Femto	10 g to 100 g
Atto	1 g to 10 g
Zepto	0.1 g to 1 g

Table 5 Classification of Cube Satellites Based on Size [11]

Name of CubeSat*	**Length**	**Width**	**Height**
1U	11.35 cm	10 cm	10 cm
2U	22.70 cm	10 cm	10 cm
6U	34.05 cm	20 cm	10 cm
12U	34.05 cm	20 cm	20 cm

*Table 5 does not show all possible CubeSat configurations but rather some of the most common ones.

The reason there has been such a large increase of CubeSat comes from the low cost; enabling multiple satellites to be designed and launched for a similar cost to one larger satellite. This has led to the growth in the utility and feasibility of constellations. Many satellites collectively working together is called a constellation, a few examples of constellations today include the Iridium Constellation, IntelSat

Constellation, and the GPS constellation. Currently the largest constellation as of 2018 is the Flock constellation operated by Planet Lab. The Flock constellation has over 144 operational satellites, called Doves, used for imaging of Earth. Planet Lab's Doves also have the record for largest number of satellites launched at one time, launching 88 satellites on February 15th 2015 [12]. As Cube and Nanosatellites get more popular the size of the constellations will increase and allow for more satellites to be launched at once. Some benefits of constellations vs a single satellite include; better temporal resolution, mitigated risk due to the loss of one satellite, and quick replacement times for damaged satellites. Currently there have been a few companies which have looked to launch mega constellations of hundreds to thousands of satellites, a few examples are shown in Table 6 below [7].

Table 6 Proposed Mega Constellations

Name	**Number of Satellites***
OneWeb Constellation	720
SpaceX Constellation (LEO)	4425
SpaceX Constellation (VLEO)	7518
Telesat Constellation	117 +
Boeing Constellation	2956
Samsung Constellation	4600
Google Constellation	841-1218

*Table 3 the values for each constellation might change since it was initially made in 2019.

The large increase in space traffic in the same or similar orbital planes has led to a greater possibility of micrometeoroid impacts, and collisions from space debris. With the launching of these mega constellation better methods of RSO detection and avoidance will be vital for the survival of the constellations and to not trigger Kessler syndrome.

Space debris, in the context of Earth, are any uncontrolled artificial object that is orbiting Earth. A more encompassing category used to describe any satellites or orbiting object is Resident Space Object (RSO). The rising number of RSO's has become a growing problem over the last few years due to space being utilized more, leading to more RSO's of which a majority of the objects are space debris. Figure 1 shows the increase of RSO's and the source of the RSO's which can be primarily attributed to different types of space debris. As the number of RSO's will only increase with the increased use of space assets new technologies are required to avoid collisions and keep active satellites operational for their lifetime [13]. Currently the Space Surveillance Network (SSN) is a group of ground and space-based sensors dedicated to the tracking and detection of RSO's. Ground based sensors are the most common but come with the downside of having trouble being able to track debris greater than 10 cm in LEO [14]. There are currently 25 ground station participating in the SSN with the Space Fence, a US government project, to come online in 2019. The recently operational space fence looks to increase the number of RSO's detected by allowing for smaller debris to be seen in LEO, as it is able to see RSO's as small as 1 cm in LEO. With the ability to detect smaller debris the number of tracked RSO's is expected to jump to over 100,000 [15]. Even with the Space Fence operational there are still areas to improve on with the SSN, one of these areas is the temporal resolution of observations. Ground based sensors can only observe a RSO when it passes over them or into their FOV, and with optical sensors there are even more restrictions. To optically track an RSO the right Sun, Earth, and RSO geometry needs to happen while is nighttime for the observation station, these constraints give ground-based observations a poor temporal resolution [16]. Space based sensors offer many advantages over ground-based sensors including getting closer to target RSO's, not having to view from the atmosphere, and being able to take observations anytime of the day. Currently space based optical RSO tracking is a relatively new technology with only a few dedicated space-based sensors currently in orbit such as Sapphire, and NEOSSat [17]. As nanosatellite technology increase the cost, and lead time, to launching space-based sensors will be decreased. This allows space-based sensors to be a viable solution to help track the growing RSO problem around Earth. Figure 3 and Figure 2 below show more information of the size of the RSO population around Earth, showing number of objects per size, as well as the percentage of types of debris respectfully.

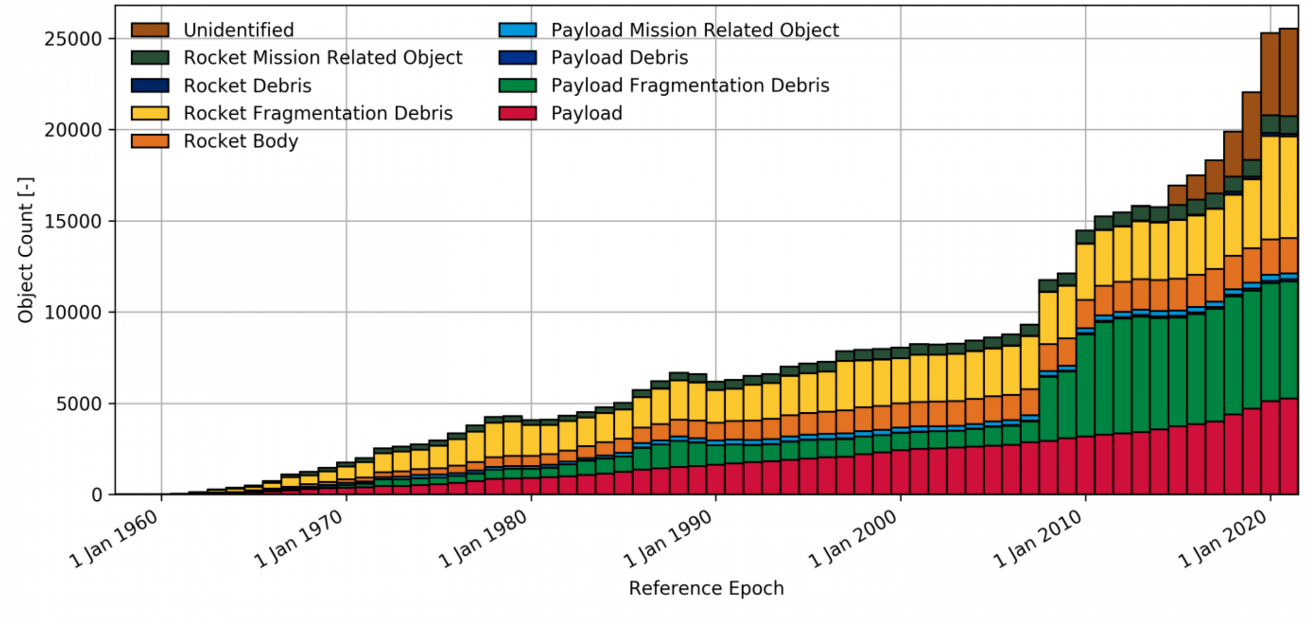


Figure 1 Space Debris in Orbit Over the Years [18]

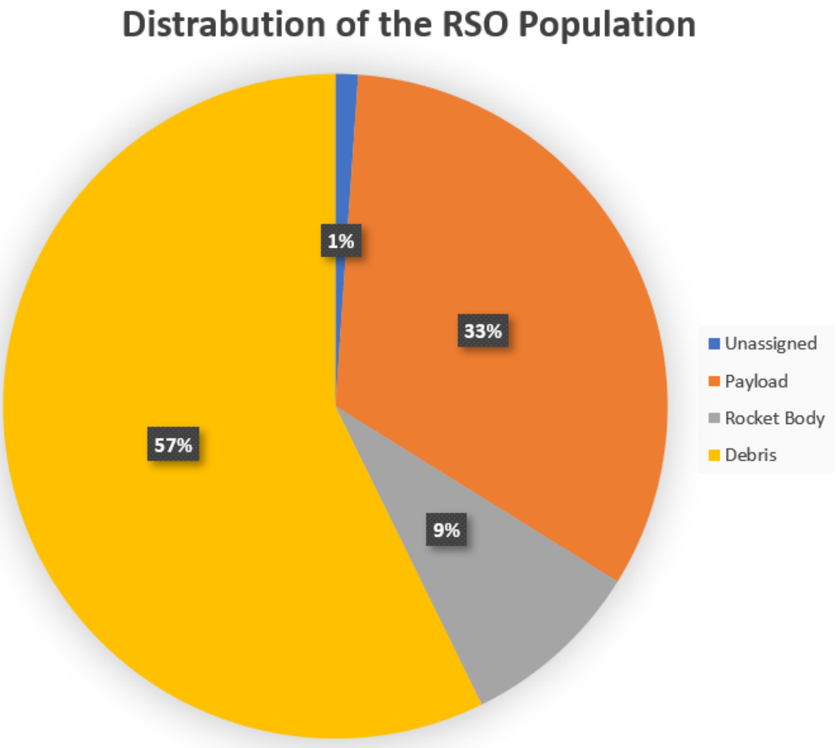


Figure 2: Percentage Distribution of the RSO population taken from Space-Track Box Score March 18th, 2022.

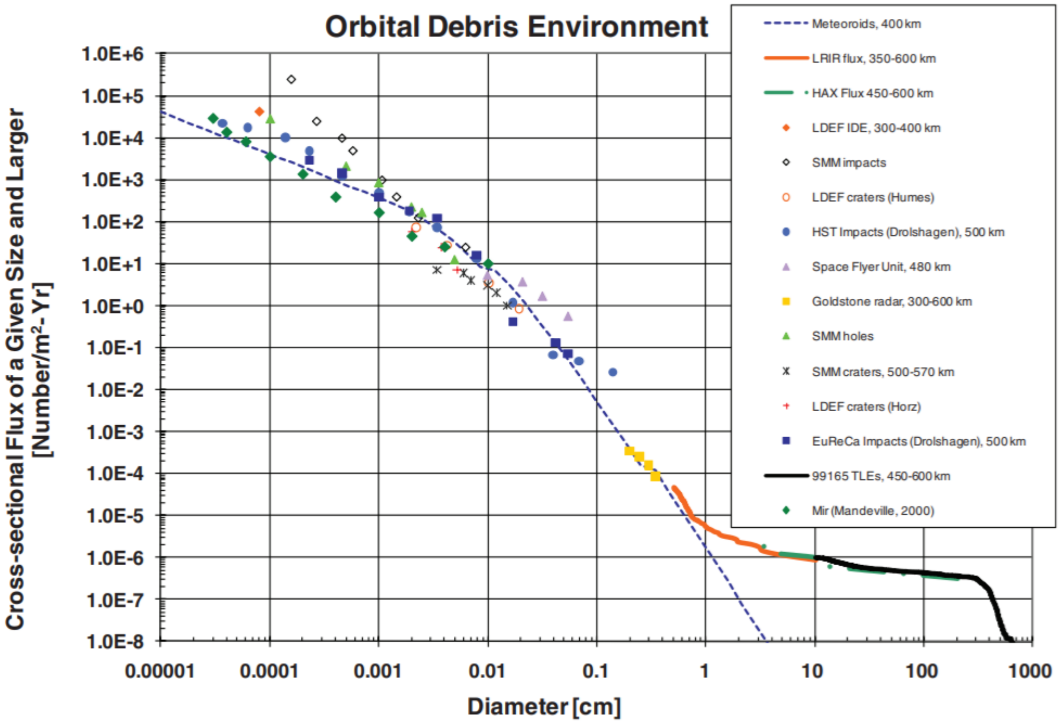


Figure 3: The trend of estimated number of space debris vs the debris size [19]

1.2 Parallel Processing for RSO Light Curve Analysis

In the last few decades the increase in computer technology has led to an exponential increase in available computational power with a decrease in the average cost. This decreased cost and more computational power has been leveraged by the use of parallel processing in many fields. Parallel processing is the method for a processor to perform multiple parts of the overall algorithm at once to reduce the over all computation time. There are many different forms of this, the simplest is Single Instruction, Multiple Data Parallel Processing (SIMD). SIMD is the method of performing one instruction with multiple data sets at once. Compared to traditional processing SIMD has the advantage of processing multiple uncorrelated data sets at once instead of the traditional sequential method. In the space sector this can be implemented to decrease propagation time of large constellation, faster image processing, and allows for on-orbit processors to be used more efficiently. When working with one data set and looking to perform multiple kinds of analysis, Multiple Instruction, Single Data (MISD) is used. This is where one data set has multiple different type of instructions to be performed on it, where these instructions have no correlation or relationship between the instructions allowing them to be performed in parallel. While MISD is one of the more uncommon implementations of parallel processing, it has large implication in the space sector with different flight control and on-board algorithms using MISD methodology. The most complex current implementation of parallel processing methodology is Multiple Instruction, Multiple Data (MIMD). This is where the computer uses multiple processors to perform multiple different sets of instructions on multiple data sets all at once. This level of parallel processing requires a high level of control and was originally implemented on super computers with most parallel processing computers after 2013 having MIMD capability. Parallel processing has many implementations with the context to this thesis which include but are not limited to; image processing, coordinate transformations, and RSO propagation. With the space catalogue expected to jump and order of magnitude the implementation of parallel processing methodology can drastically reduce required computation time. With the satellites being uncorrelated, instead of propagating 20,000+ RSO’s sequentially, the 20,000+ RSO’s can be propagated using either SIMD or MIMD methodology.

1.3 Optical Detections of RSO's

RSO detections are done primarily two ways, from radar, and optical sensors. Radar is the most common type being able to cover large area and see smaller objects then optical detections. Optical detections have the advantage of providing additional information when compared to radar detections. This information includes the shape, attitude, and optical properties in addition to the position and velocity. When an RSO is detected optically its brightness varies over the detection duration. There are many contributing factors to the change in brightness, with some of the more significant being the Sun, Earth, RSO geometry, the reflective surfaces and shape of the RSO, as well as the RSO relative motion to the sensor [20]. Plotting the brightness vs time generates the light curve of the RSO over that sequence of optical detections. Light curves are commonly used astronomy for a variety of purposes such as exoplanet finding, and rotation period estimates of celestial bodies [21] [22]. In comparison to an exoplanet or asteroid a RSO's the light curve is more complex to find due to the irregular shape and relative motion between the sensor and RSO. The technique of extracting information about an object from its light curve is called a light curve analysis and is currently a growing field, especially for RSO optical detections [23] [24] [25]. Recently in literature there have been different methods of light curve analysis using different techniques such as: phase folding, least squares spectral analysis, glint analysis, and machine learning algorithms [22] [26] [27]. Light curve inversion is an iterative light curve analysis methodology that extracts additional information such as shape, attitude, and optical properties. There is not one algorithm for light curve inversion; it rather references a general process to better characterize the target object. Both techniques have been used for RSO optical characterisation, depending on what information is required. below Figure 4 shows the difference in algorithm flow between light curve analysis and light curve inversion technique.

To meet the rising need for more and higher fidelity SSA optical detection of RSO's are becoming more common. With the rise in number of RSO optical detections the is a direct increase in the amount of computational power to analyse, extract, and estimate, SSA information such as Object ID, Position, Attitude etc. The integration of SIMD or MIMD, as mentioned above, has the potential to greatly increase the throughput of these algorithms. Specifically in the context of this thesis this is looking to be implemented in the simulation of RSO optical detections and extraction of information from real data. Simulated images of RSO detections have many uses including; estimating the suitability of future detections, confirming known motion in RSO's from deviations in the light curve, and enabling the finding of the optical and physical properties of RSO's [28]. Simulating images is common in other fields such as graphics and computer animation, the difficult part of preforming this in the context of space comes from the complex dynamic motion and systems that arise. Image simulation, at least in RSO context, involves tracing of the incoming light from all sources (active and reflective) that reach the RSO, then tracing all the light that reflects off of the RSO and enters the sensor. This methodology is called forward ray tracing and to preform this method good knowledge of all light sources are need for the system [29]. While the motion and brightness of larger celestial objects is well known such as the Sun, Earth, the Moon etc., the motion of smaller RSO's has large uncertainty. This causes a problem with space-based sensors as in space the large amounts of uncertainty and error in the position, size, and optical properties of objects directly impact the ability to simulate the proper brightness and motion of the object. Empirical models can preform well under these conditions, like the Anisotropic Phong model modified for RSO applications [30], the number of detections required to accurately estimate RSO parameters is large and generally not available in public datasets. With all objects in the sensors FOV, and some outside, effecting the image there is a lot of information to calculate and process, causing image simulation to be very computationally intensive. Implementation of parallel processing methodology in the image simulator will be preformed and compared to sequential methods to look at the advantage of parallel processing in image simulation. Simulated images will also then be used to help with light curve inversion and optical characterisation process extracting information on the RSO from the RSO's light curve.

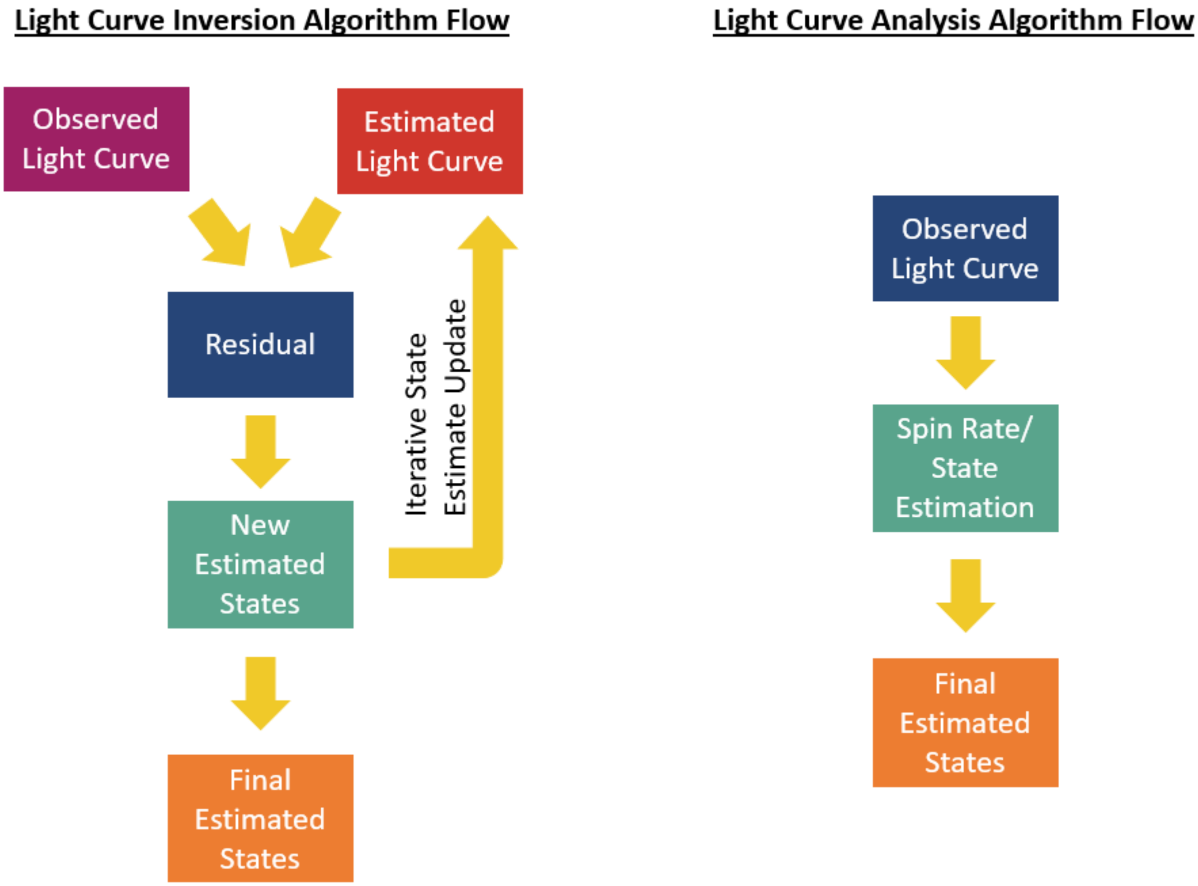


Figure 4 represents the algorithm flow of both Light Curve Inversion (left) and Light Curve Analysis (right)

1.4 RSO Optical Characterisation

RSO characterisation represents the estimation of RSO properties and orbital characteristics from one or more remote detections. In the context of this thesis RSO characterisation will only refer to using optical detections. RSO characterisation in the context of this thesis specifically refers to finding the size and shape, attitude, and optical properties of the satellite. Depending on the type of characterisation being performed; ex. Attitude vs Spin Rate Estimation, either light curve analysis or light curve inversion techniques are used, respectfully. This thesis looks to perform higher fidelity characterisation than spin rate analysis leading to light curve inversion techniques being implemented over light curve analysis. Different light curve inversion techniques have been proposed all requiring a priory estimate and optimiser [23] [31] [32].

Comparing a simulated (or estimated) light curve with estimated properties of the RSO to the true light curve the residual between the two can be calculated. After a residual is generated an optimisation algorithm is used to improve the estimate of parameters till the simulated light curve matches the true light curve. This provides an estimate of the RSO's true properties. Being an optimisation problem with a large multi-parameter design space it requires either heuristic or approximation algorithms to be implemented as the optimisation function. This creates a trade off between, accuracy, robustness, and speed. Priory estimates are used to reduce the search space, this limits the number of local solutions, and decrease the required time to converge to a global optimum. Cooperative targets, such as active satellites, have the possibility of having priory information available. When looking at space debris it is very unlikely to have priory information available, and if available is usually just the size and general shape with no attitude information. For a light curve inversion technique to be able to characterise all RSO's it needs to be robust enough to not require a priory estimate or require only a limited shape estimate. The increasing congestion of space and recent increase of on orbit event such as active debris removal, and on orbit servicing demonstrate a growing need for such RSO characterisation techniques [33] [34]

1.5 Motivation

The motivation for this research looks to augment current SSA capability by enabling passive space based optical sensors to contribute to SSA. The need for increased SSA has been well document but the use of space based optical sensors, specifically dual purpose/ passive sensors, is a new concept that shows advantages over traditional terrestrial sensors. The advantages include but are not limited to; avoiding atmospheric effects, different viewing geometries, and not requiring sundown to take observations. The advantages that space based sensor provide combined with the number of low-resolution optical sensors in space gives the perfect environment to leverage currently available space assets, while allowing for growth with future space assets. Low resolutions optical sensors, such as star trackers, are common on modern satellites, performing tasks such as attitude determination and remote sensing. These sensors generally have a large field of view making them ideal for observations of opportunity given from passive sensors. Passive sensing refers that the detections of RSO's will not be the primary goal of the sensor but when an RSO does pass the FOV of the detector the information can be extracted and used for SSA purposes. Currently most optical observations are tracked observations, tracked observations require priory knowledge of the targets position, as well as accurate control to track the target. For RSO's, specifically space debris, priory information like this is not always known making passive sensing ideal for large scale RSO identification and initial orbital determination. Observations of opportunity also enables RSO observations to be a secondary mission for optical payloads, one relevant example are star trackers images can be used for attitude determination, as well as RSO observations. Augmenting current low-resolution space based optical sensors with software updates to enable this can enable a new network of passive RSO observatories with out the cost and lead time of dedicated operations. To realise this goal of a passive space based RSO network 4 key technologies were identified to tackle current challenges. Namely, (1) parallel processing of RSO propagation, (2) RSO image simulation, (3) space based RSO detections (4) space based RSO characterisation. Below each of these technologies, the challenge they look to overcome, and how they technology enables passive space based RSO observations are mentioned.

With the proliferation of RSO's there is an increasing computational strain with more objects to be, propagated, simulated, and cataloged. With more RSO's there is an increased requirement for higher fidelity propagation models to for conjunction warnings and rendezvous, leading to an increase in computational resources required. In recent papers it has been noted that current methods do not have the speed or accuracy to keep up with current and future SSA demands requiring a large increase in efficiency to allow for space to keep growing in a sustainable matter [35]. Parallel processing looks to overcome this by reducing the computation time increase with the increase in objects propagated. Different methodologies: such as SIMD, and MIMD, have the potential to reduce the amount of computational resources required by making more efficent use of the computational resources available. Specifically in the context of this thesis parallel processing is looking to be implemented in 3 areas to improve throughput of data: RSO propagation, RSO image generation, and light curve extraction. This will start with a comparison of the accuracy and computation time of serial vs parallel propagation methodology. If parallel processing is shown to cause an improvement in computation time with out a decrease in accuracy, this methodology will be implemented in image simulation. Light curve extraction does not rely on propagation but does require the extraction of information from many similar images. The implementation of parallel processing will look to extract the light curve from sequence of images all at once, reducing the computation time. This key technology looks to enable and increase the impact of all other technologies through its implementation allowing for more efficent RSO image simulation, detection, and characterisation.

To be able to properly predict, schedule, and characterise RSO's from optical images, the images of the detection sequence need to be able to be replicated in a simulated environment. Currently one of the industry standard ways of generating this image come from Systems Tool Kit's (STK) EOIR Toolbox. STK EOIR allows for user to simulate binary and raw images from both ground and space-based sensors, but it does have its limitations. EOIR does not produced labeled images requiring the images to be labeled through a UI or 3rd party software, this requires large amount of human interaction to generate data sets. To overcome come this a star field and RSO optical image simulator was created in MATLAB. ORBITALS looks to overcome the downside from STK's EOIR by enabling the generation of simulated labeled images with limited human interaction in an efficent manner. ORBITALS looks to further enable SSA by providing labeled data sets for; RSO detection finding, RSO detection predictions, RSO identification, RSO characterisation, and training of machine learning algorithms. ORBITALS also further enables RSO characterisation by allowing for the replication of light curves with estimated parameters to be compared vs the observed light curve. This allows for higher fidelity characterisation of RSO's from there light curve in a more efficent manner then the currently available STK EOIR software.

Objects detected from ground-based observation can be seen down to a size of 10 cm, which poses a problem with a majority of space debris being below the 10 cm size [36]. As mentioned in the previous section space fence looks to aid in the detection of debris down to 1 cm size through radar observations which will cause a large increase in the number of tracked objects in the space catalog. This large increase has two major effects on SSA; an increased need for better RSO scheduling algorithms, as well as the ability to see under 10 cm sized objects from optical observations. Increasing the number of RSO observatories, both in optical and radar region, to account for the increase in the space catalog has the downside of being expensive and having a long lead time for observatories to become operational. Better scheduling algorithms will help augment current and future observatories allowing them to know optimal viewing conditions to detect the maximum number of RSO’s. To enable the development and testing of scheduling algorithms test scenarios with the possible detection geometries, detection times, and detection statistics are required. Normally this information is not well known with real world data, using ORBITALS to replicate the detections of real data in a simulated environment the detection time, detection geometry and detection statistics can be extracted. Scheduling observations will ideally allow for the increase in number of detection but does not improve the size of the objects that can be seen. To be able to see smaller than 10 cm objects better dynamic range and spatial resolution is required for current terrestrial optical sensors. To accomplish this larger aperture telescope with better lenses and sensors are required, increasing cost of building and operating these observatories. One way to over come this challenge is to augment terrestrial optical RSO detections with space based optical RSO detections. Space based RSO optical detection do not require atmospheric correction, are not limited to nighttime observations, and have the possibility of getting closer to the target object then terrestrial observatories. This allows for space-based observation to require less calibration, see smaller objects, and have a better duty cycle compared to terrestrial observations. Currently there are working examples of these types of sensors such as NEOSSat and SAPHIRE which are both Canadian space assets. These sensors act as dedicated space-based sensors, finding and tracking objects which allows for high resolution imagery of RSO’s. Like dedicated ground observatories these have large lead times and are expensive, which is not ideal to overcome the challenges caused by the proliferation of space debris. In recent literature passive low resolution RSO detection have been examined and demonstrated to be a low cost widely available solution for augmenting current SSA efforts [37]. RSO detections have been shown to happen from these platforms, such as from the Fast Auroral Imager (FAI) and star tracker images. However, these detections being lower resolution have not yet been shown to acquire the same amount of data as their terrestrial counterparts. Using ORBITALS in a simulated environment detection statistic, object determination, as well as optical characterisation can be performed to see what type of information can be determined from space vs terrestrial RSO observation. This looks to augment and improve current SSA efforts by allowing lower resolutions sensors; which are less expensive, have less lead time, and are commonly used on current active satellites, to contribute to SSA. The passive sensing component allows for these sensors to detect RSO's as a secondary mission not requiring tracking or dedicated attitude maneuvers. Ultimately this should enable all satellite with a low-resolution optical sensor, such as a star tracker, to act as a passive RSO observatory greatly augmenting the current number of observatories.

With the increasing reliance on space assets for everyday functionality just identifying an RSO's orbit is no longer sufficient for SSA. To get more information on the target object RSO characterisation is performed. RSO characterisation currently requires dedicated detections from a highly spatially accurate telescope. As mentioned above these types of observatories are expensive and have a large lead time. Currently space-based characterisation has been performed from dedicated sensors, such as NEOSSat and SAPHIRE, which leverages the advantages of space-based observatories. Demonstrating that RSO characterisation can be performed from low resolution space based RSO images will greatly increase the amount of SSA information available with out the required expenses and lead time as dedicated sensors.

Together these 4 technologies look to enable low resolution space based RSO detections to be used for RSO detection, tracking, and characterisation. Moving this technology to a more operational state looks to have a huge impact on augmenting current SSA efforts to meet the rising need. The short-term goals from this research look to enable RSO detections to take place from any space based optical instrument, including all current and future assets. Majority of satellites have either dedicated optical sensor, or star trackers which can be augmented to provide vital SSA contribution with out expensive infrastructure upgrades. The increase in number of detections will also lead to an increased need to process all of these detections and extract as much information on the RSO as possible. With parallel processing implemented in RSO characterisation solutions large number of data sets will be able to be processed in an efficient manner, lessening the amount of computational hardware required for SSA data processing and analysis. Long term goals for this research look to help train different machine learning algorithm for either in-situ or ground based data processing, as well as, to enable the creation of better image simulators through larger data sets. Improving RSO detection and characterisation technologies both for accuracy and computational speed will lead to improve most aspects of the space sector and Canada’s role in it. The augmentation of current SSA assets with the proposed technology will increase Canada’s current role in SSA and could lead to more ambitious project to solve the worlds need for better SSA. Canada’s large land mass and experience with space based SSA assets, like NEOSSat and SAPHIRE, put it in a position with the right terrestrial and space-based infrastructure to be the world leader in SSA. Accomplishing Canada’s goal of being Strong, Secure, and Engaged in the space sector.

1.6 Research Objectives

In this section each of the research objectives will be mentioned along with the technological gap that is causing the objective.

The first objective is improving the throughput of current propagation algorithms using parallel processing. This objective comes for the technological gap caused by the proliferation of RSO's requiring more efficent propagation algorithms to keep up with the growing catalog. More efficent propagation methods are affect the complete space sector and are used for tasks such as access determination, conjunction warnings, satellite scheduling, and mission planning. This provides a unique and novel propagation method to the space community which will increase the throughput multi-satellite propagation. This work performed is explored more in chapter 3 with a summary of the contributions of the improved throughput multi-satellite propagator at the end of chapter 3.

The second objective is to develop a RSO image simulation software that allows for the efficent generation of optical RSO detections from space-based platforms. This objective comes from the technological gap of the lack of efficent publicly available space based RSO simulation software's. Currently there are a few of these software's that are on the market but have their limitation including requiring expensive licences, not providing labeled images, and taking seconds to minutes to generate one image. This allows the ORBITALS image simulator to provide a unique roll allowing for the generation of large simulated image sets efficiently and accurately with out expensive licences. This research objective is explored more in chapter 4 with a summary of the contribution of the RSO image simulator at the end of chapter 4.

The third objective, and main focus of this thesis is to demonstrate the feasibility of performing RSO characterisation from space based optical observations. This objective come from the current technology gap in space based optical characterisation. There have been different techniques proposed for the characterisation of RSO's from high resolution terrestrial observations, but there has only recently have space-based observations been looked at in this context. Terrestrial is more common due to the availability of data for training and algorithm development. This is looked to be overcome by developing an image simulator that is able to generate labeled data sets similar to those present for terrestrial observations. This provides a novel method to allow for low resolution detection of RSO's to provide more that just orbital information. With the proliferation of RSO's this unique and novel method will allow for more information to be extracted from light curves of RSO’s to meet the rising need for better SSA. This research objective is explored more in chapter 5 with a summary of the contribution of the feasibility of performing RSO characterisation from space based optical observations to the research objectives at the end of chapter 5.

Together these three objectives look to improve the TRL level of low-resolution space based RSO detections and characterisation by demonstrating the feasibility and showcasing the accuracy of the proposed methods.

1.7 Thesis Outline

In this thesis the all the background information needed will be outlined in section 2, chapters 2 through 5 contain the published studies performed. Chapter 3 is the first technical chapter and focuses on the implementation of parallel processing in orbital propagation. The initial study, as well as the follow up study focusing on parallel orbital propagation for spacecraft maneuver detection. Both the studies set up, and results are discussed in this section including the implementations that these studies have on the other key technologies. Chapter 4 focuses on the ORBITALS simulator and comparing it to the industry standard, STK EOIR. To do this both ORBITALS and EOIR simulated images are compared against truth data, in the form of real space-based images from the FAI. Both the accuracy of the RSO position in image and the computation speed is compared between the two image simulators. Chapter 5 uses the ORBITALS simulator to test the feasibility of space based low resolution optical characterisation of RSO’s. To perform this study different optimisation and bidirectional reflectance distribution functions were compared in the ORBITALS optical environment. The optimal results from this study are then looked to be implemented in the real-world environment to verify the results. Chapter 6 looks to summarize the main findings of this thesis, as well as mention the future areas of work to be expanded upon.

1.8 Publications

The list below contains all publication made during the completion of the thesis, while not all publications are related to this thesis, they are all mentioned.

Journal Publications

Clark, Ryan, and Regina Lee. "Parallel processing for orbital maneuver detection." *Advances in Space Research* 66.2 (2020): 444-449

Bolshakov, Konstantin, et al. "Array-based digital sun-sensor design for CubeSat application." *Acta Astronautica* 178 (2021): 81-88

Under Review

Clark, Ryan, et al. “Space Situational Awareness (SSA) Data Simulation Using Parallel Processing”, Under Review, *Sensors*, 2021.

Clark, Ryan, et al. “RSO Attitude and Optical Property Estimation from Space-Based Light Curves”, Under Review, *Advances in Space Research*, 2021.

Conference Proceedings

Clark, Ryan, et al. “RSO Attitude and Optical Property Estimation from Space-Based Light Curves”, *43rd COSPAR Scientific Assembly*, Oral Presentation, January 31, 2021.

Dave, Siddharth, Clark, Ryan, & Lee, Regina “RSO position and velocity estimation using Convolutional Neural Networks from in-orbit star tracker images”. *43rd COSPAR Scientific Assembly*, 2021.

Clark, Ryan, et al. "Performance of Parameterization Algorithms for Resident Space Object (RSO) Attitude Estimates.", *Advanced Maui Optical and Space Surveillance Conference*, 2020.

Dave, Siddharth, et al. "Machine learning implementation for in-orbit RSO orbit estimation using star tracker cameras.", *Advanced Maui Optical and Space Surveillance Conference*, 2020.

Under Preparation

Sari, Naeimeh, et al. “SPIN STABILIZED SUN POINTING CUBESAT FOR SPACE GEOLOGY”, Under Preparation, 16th International Conference on Space Operations, 2021.

2 Background Information

2.1 RSO Propagation Method

To estimate the location of RSO’s a propagator and known position at a given time is required. Orbital propagators take an RSO’s position at a known time, then calculate the acceleration on the object using a force model. Once the acceleration is calculated the RSO’s motion can be approximated, allowing for the estimation of its position at some future point. This, for most propagators, creates a trade off between the accuracy of the system and the computational time, the more sophisticated the model the more computational time is required. All orbital propagators start by considering the two-body gravitational problem with one of the objects mass, in this case Earths, being much more massive than the RSO’s mass that is in orbit around it. This leads to equation 1-3 which is the acceleration of the RSO as a function of position around the larger object. These equations are a series of Ordinary Differential Equations (ODEs) used to solve for the position of the orbit of the RSO in ECI coordinates, this is called the state propagation method [38].

$$\ddot{X}_e = \frac{\mu X_e}{(X_e^2 + Y_e^2 + Z_e^2)^{\frac{3}{2}}} \tag{1}$$

$$\ddot{Y}_e = \frac{\mu Y_e}{(X_e^2 + Y_e^2 + Z_e^2)^{\frac{3}{2}}} \tag{2}$$

$$\ddot{Z}_e = \frac{\mu Z_e}{(X_e^2 + Y_e^2 + Z_e^2)^{\frac{3}{2}}} \tag{3}$$

In equations 1 to 3: μ represents the standard gravitational parameter for Earth. X_e, Y_e and Z_e represent the, X, Y, and Z, position of the satellite in the ECI reference frame respectfully. This is the base model for orbital propagation requiring numerical methods to solve for the coupled ODE equations which is shown in equation 4 below [39].

$$\ddot{\boldsymbol{R}} = -\frac{\mu}{|\boldsymbol{R}|^3} * \boldsymbol{R} \qquad 4$$

In equation 4: $\boldsymbol{R}$ represents the distance vector from the center of Earth to the satellite. μ represents the gravitational coefficient for earth. The two-body problem is a solvable problem with a closed form solution represented by equation 5.

$$\boldsymbol{R}(\theta) = \frac{h^2}{\mu}\frac{1}{\left(1 + ecos(\theta)\right)} \qquad 5$$

In equation 5: θ represents the true anomaly. e represents the eccentricity. μ represents the standard gravitational parameter. h represents the specific relative angular momentum.

An alternate form of propagation uses closed form solution of the two-body problem. Where the Keplerian elements are used to represent the RSO’s motion instead of a state vector. The Keplerian elements are 6 elements that perfectly describe the position and orbit of a satellite, the elements and their descriptions are listed in Table 7 and shown in Figure 5.

Table 7 The Different Keplerian Elements

Name	**Variable**	**Units**	**Description**
Semi Major Axis	a	Distance (commonly Km)	The largest line segment that runs through focus to each side of an ellipse. Has different definitions for parabola and hyperbola.
Inclination	i	Degrees or radians	The angle between the orbit plane and the equatorial plane of the earth.

Name	**Variable**	**Units**	**Description**
Eccentricity	e	Unitless	Gives the shape of the orbit with 0 representing circular, 0 to 1 representing an ellipse, 1 representing a parabola, and greater than 1 representing a hyperbola.
Argument of perigee (periapsis)	ω	Degrees or radians	The argument of perigee is an angle that passes from the ascending node to the perigee of the orbit.
Right Ascension of the Ascending Node (RAAN)	Ω	Degrees or radians	The angle measured from the vernal equinox reference point to the ascending node of the RSO.
True Anomaly	θ	Degrees or radians	The angle from the perigee to the target RSO about Earth.
Mean Anomaly*	M	Degrees or radians	The angle from the perigee to the target about Earth if the RSO was in a circular orbit.
Mean Motion*	N_o	Degrees or radians per second	The angular speed at which the RSO would move from an observer on Earth if the RSO was in a circular orbit.

* The mean motion and mean anomaly are not required Keplerian elements but are required when it comes to propagation of the RSO’s orbit.

There are only 6 required elements to find an RSO’s position which are the semimajor axis, inclination, eccentricity, argument of perigee, RAAN and one of the anomalies. Different types of anomalies are used in difference instances with the main two being the mean anomaly and true anomaly. The true anomaly gives the angle from the argument of perigee to the target RSO, about the Earth’s center. The mean anomaly gives the true anomaly of the RSO if it was in perfectly circular orbit. Due to Kepler's 2nd law the amount of area taken from an elliptical and circular orbit is the same over a time period. From this the mean and true anomaly can be changed between each other, the equations for transformation are shown in equations 6 and 7.

$$\tan\left(\frac{\theta}{2}\right) = \sqrt{\frac{1+e}{1-e}}\tan\left(\frac{E}{2}\right) \tag{6}$$

$$M = E - e\sin(E) \tag{7}$$

In equations 6 and 7: θ represents the true anomaly. e represents the eccentricity. E represents the eccentric anomaly. M represents the mean anomaly.

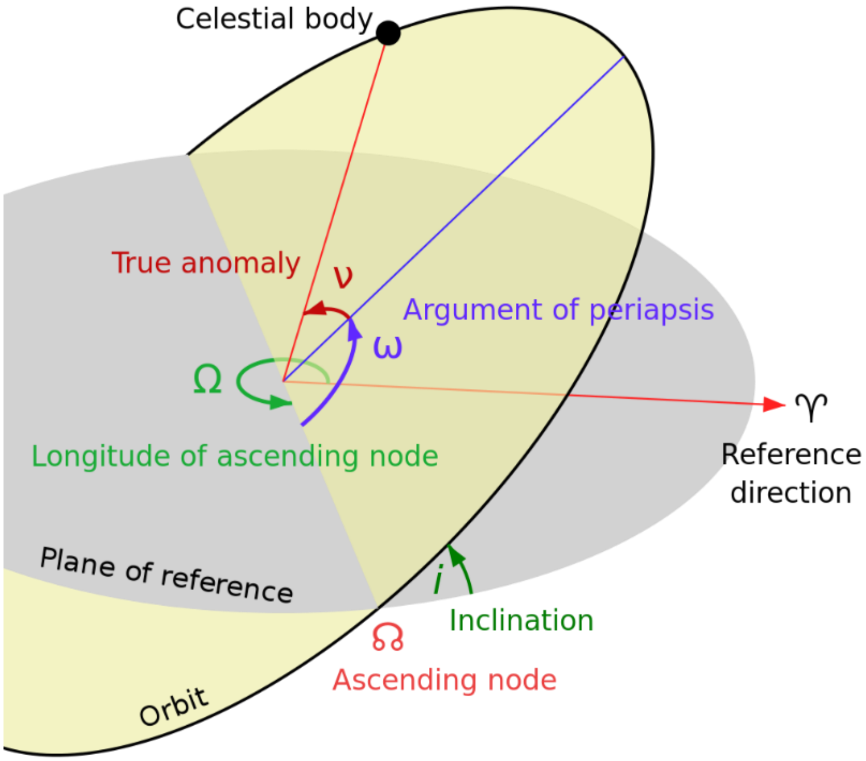


Figure 5 Diagram of the Keplerian Elements [40]

Both anomalies can tell where the satellite is in its orbit, with each anomaly having its specific use for propagation. The true anomaly allows for the position of an ecliptic orbit to be transformed into cartesian coordinates commonly ECI. The mean anomaly has the advantage having a constant rate of change (in an ideal two body system), this allows for the easy propagation of the mean anomaly. To propagate a satellite orbit using the Keplerian elements, and the closed form solution, the mean motion, and mean anomaly are used for the propagation. The other orbital elements describe the orientation of the orbital plane relative to Earth coordinate systems. The mean motion will represent the change in mean anomaly which represents how far a RSO will progress in its orbit over a given time step. Knowing the change in mean anomaly and the time, t, the updated mean anomaly can be calculated with equation 8. Unlike using the state propagation method using the mean motion and Keplerian elements does not require intermediate points to be considered. This allows the position to be calculated quicker than state propagation models, as well as the computation time not depending on propagation time. The downside to Keplerian propagation is that it does not include any perturbations making the propagator become inaccurate after a few orbits. Keplerian propagation does work well when requiring a RSO to be propagated over a small-time step and to be calculated quickly.

$$M_t = M_o + N_o t \tag{8}$$

In equation 8: M_t represent the mean anomaly at some future time t. M_o represents the mean anomaly at the starting point. N_o represents the mean motion of the RSO. t represents the amount of time passed from the initial to final mean anomaly.

Both the two-body state vector propagation and Keplerian propagation only account for Earth and the RSO to be the only objects in the system, in reality the system is much more complex. In reality effects such as atmospheric drag, solar and lunar gravitation, and solar radiation pressure change the orbit of the RSO over time causing both the two body and Keplerian propagator to be inaccurate for long period of propagation.

To over come this a perturbation term is included in equation 4 forming equation 9 where the perturbation term depends on the perturbation model used.

$$\ddot{\boldsymbol{R}} - \frac{\mu}{|\boldsymbol{R}|^3} * \boldsymbol{R} + \mathbf{P} \tag{9}$$

In equation 9: ***R*** represents the distance vector from the center of Earth to the orbiting object. ***P*** represents the vector of the perturbation's contribution to the acceleration of the orbiting object. Two propagation models that will be looked in this thesis are the J2 Perturbation model, which will be implemented as a feasibility study for parallel processing of orbital propagation. If the results are promising the more complex Special Generalized Perturbation Model 4 (SGP4) will be implemented and tested for common space applications such as orbital maneuver detection, and large catalog propagation. Both the theory behind these two different models, as well as, how each of the models were implemented for this thesis will be discussed in their respective sections below.

2.1.1 J2 Perturbations

The J2 perturbation model is a force model of the first gravitational harmonic caused by Earth being a spheroid. The J2 model is normally used when propagating satellites for days to weeks, when low accuracy position information is needed. Beyond a couple of days, the accuracy of the J2 model in not sufficient for accurate propagation. The J2 model is commonly used for academic or simple LEO propagation where the time steps are extremely small. The J2 model is the simplest of the state propagation methods that involve perturbations making it an ideal case to test different coding practises, and implementation of parallel processing for this thesis. The J2 force model is well known for state vectors and is shown in equation 10-12 which represent the X, Y, and Z axis in the ECI coordinates, respectfully [38] [41].

$$P_{J2x} = \frac{3J_2\mu R_E^2}{2R^7}(5Z_e^2 - X_e R^2) \tag{10}$$

$$P_{J2y} = \frac{3J_2\mu R_E^2}{2R^7}(5Z_e^2 - Y_eR^2) \tag{11}$$

$$P_{J2z} = \frac{3J_2\mu R_E^2}{2R^7}(5Z_e^2 - 3Z_eR^2) \tag{12}$$

In equations 8 to 10: J_2 represents the second gravitational harmonic of Earth. μ represents the standard gravitational parameter. R_E represents the mean radius of Earth. R represents the scalar distance to the RSO from the center of Earth. X_e, Y_e, and Z_e represent the RSO's coordinates in the ECI coordinate system. The J2 perturbation model when applied to equation 9 leads to equations 13-15 which represent the force in each ECI axis.

$$\ddot{X}_e = \frac{\mu X_e}{R^3} + \frac{3J_2\mu R_E^2}{2R^7}(5Z_e^2 - X_eR^2) = \frac{\mu}{R^3}\left(X_e + \frac{3J_2R_E^2}{2R^4}[5Z_e^2 - X_eR^2]\right) \tag{13}$$

$$\ddot{Y}_e = \frac{\mu Y_e}{R^3} + \frac{3J_2\mu R_E^2}{2R^7}(5Z_e^2 - Y_eR^2) = \frac{\mu}{R^3}\left(Y_e + \frac{3J_2R_E^2}{2R^4}[5Z_e^2 - Y_eR^2]\right) \tag{14}$$

$$\ddot{Z}_e = \frac{\mu Z_e}{R^3} + \frac{3J_2\mu R_E^2}{2R^7}(5Z_e^2 - 3Z_eR^2) = \frac{\mu}{R^3}\left(Z_e + \frac{3J_2R_E^2}{2R^4}[5Z_e^2 - 3Z_eR^2]\right) \tag{15}$$

The above equations are used with numerical methods to solve the ODE for the RSO's position, the different numerical methods used in this thesis are explained in section 2.6. If you are looking for more information on J2 perturbations some references are [38] [41] [42]. The results from the implementation of the J2 algorithm can be seen in section 3.1, Initial Feasibility Study.

2.1.2 Specialized General Perturbation Method 4

Specialized General Perturbations Method 4 (SGP4) and other variations of Keplerian propagators start with an initial point similar to state propagators. SGP4 takes the initial point from a Two-Line Element (TLE) instead of being a state vector. A TLE contains the Keplerian elements, mentioned above in Table 7, with auxiliary timing and drag information, for more information of TLE's please see [43]. Currently TLE's are the standard but are getting phased out due to the limited amount of space objects that can be recorded in the TLE format. The new space standard is looking to move to the Orbital Mean Elements Message (OMM). OMM contains the same information as a TLE in a different format with covariance matrices that are not given in TLE's, more information on OMM can be found at [44]. The standard two algorithms used are SGP4 and Special Deep Space Perturbation Method 4 (SDP4). These algorithms follow the same architecture with SGP4 focusing on perturbations that effect LEO RSO's, like atmospheric drag. SDP4 focuses on RSO's that are further away from Earth allowing the drag to be simplified but requires a more complex 3rd body perturbation model. Most RSO's that are currently of interest in this thesis are in the LEO region of space which is why the SGP4 was originally implemented. As the propagation of GEO and other non-LEO satellites became required for this thesis the original SGP4 method was upgraded to SDP4 to allow the propagation of all RSO's around Earth. The 6 Keplerian elements can be used for simple propagation using the mean anomaly and mean motion, shown above in equation 8. The SGP4 method was developed to reflect the perturbations on the RSO as changes in there Keplerian elements over time. This allows the algorithm to account for perturbations while providing a computationally efficient method of calculating the RSO position. These updated parameters are then used to calculate propagation constants such as drag and gravitational coefficients. After all the initial values have been calculated the propagation loop is entered where the perturbations are added, such as: Earth's obliquity, atmospheric drag, 3rd body perturbations, etc. To perform this the perturbations are split into sections and calculated for each propagation time. The sections are secular effects of atmospheric drag and gravitation, long-period periodics, and short-period periodics. Once all of perturbations have been included as changes in the Keplerian elements the state of the RSO is calculated from the updated elements, returning the position and velocity in the form of a state vector. The position and velocity returned by the SGP4 algorithm are given in True Equator Mean Equinox (TEME) coordinate system, this is because TLE's are made for the TEME coordinate system [45]. To transform this coordinate system into other commonly used space coordinate systems, such as J2000, see [46]. With propagation using SGP4 and SDP4 not requiring intermediate steps multiple propagation times can be performed at once in parallel. This does not reduce the accuracy of the results while increasing the throughput of the propagator. SGP4 and state vector propagation can both implement parallel propagation for each RSO, as they are independent of each other. When it comes to implementing parallel propagation for the different time steps only SGP4 can be implement with out sacrificing accuracy. This ability of SGP4 to use parallel processing for each RSO and time step allows SGP4 to be an ideal propagator in the context of this thesis for RSO access and detection finding. If you are looking for more information about the SGP4 algorithm, CelesTrack (https://celestrak.com) has good information, as well as, the original report for SGP4 [47]. Recently a new release of SGP4 called SGP4-X has been introduced. This updated algorithm follows the same flow while updating some of the perturbation estimates to improve accuracy, the updated algorithm and all of its changes can be found on the space track website (https://www.space-track.org/).

2.1.3 Parallel Specialized Perturbation Method 4 (PSG4)

PSGP4 is a version of the SGP4 propagator that is set up to propagate RSO's in parallel. Due to the propagation of RSO's following almost the same architecture and having no correlation between different RSO's, this allows for implementation of SIMD parallel processing. This is shown from the algorithm block diagram, Figure 6. While there are some sections where SIMD is not possible, such as the solving of Kepler's equation which is performed iteratively, it is implemented in a significant portion to see improvements in computation time with out a reduction in accuracy. This can be seen from the implementation of PSGP4 for maneuver detection in Parallel Propagation for Orbital Maneuver Detection [48]. Since the paper was published in 2020 the PSGP4 algorithm has been updated to include the Specialized Deep Space Perturbation Method 4 (SDP4). The difference between SGP4 and SDP4 is that SDP4 is used for objects with an orbital period above 225 minutes with SGP4 being for objects with periods below 225 minutes. The SDP4 algorithm follows the same architecture as the SGP4 algorithm with some added functions and different coefficients, this allows SDP4 to be integrated with a set of sub functions which are represented on the left side of Figure 6.

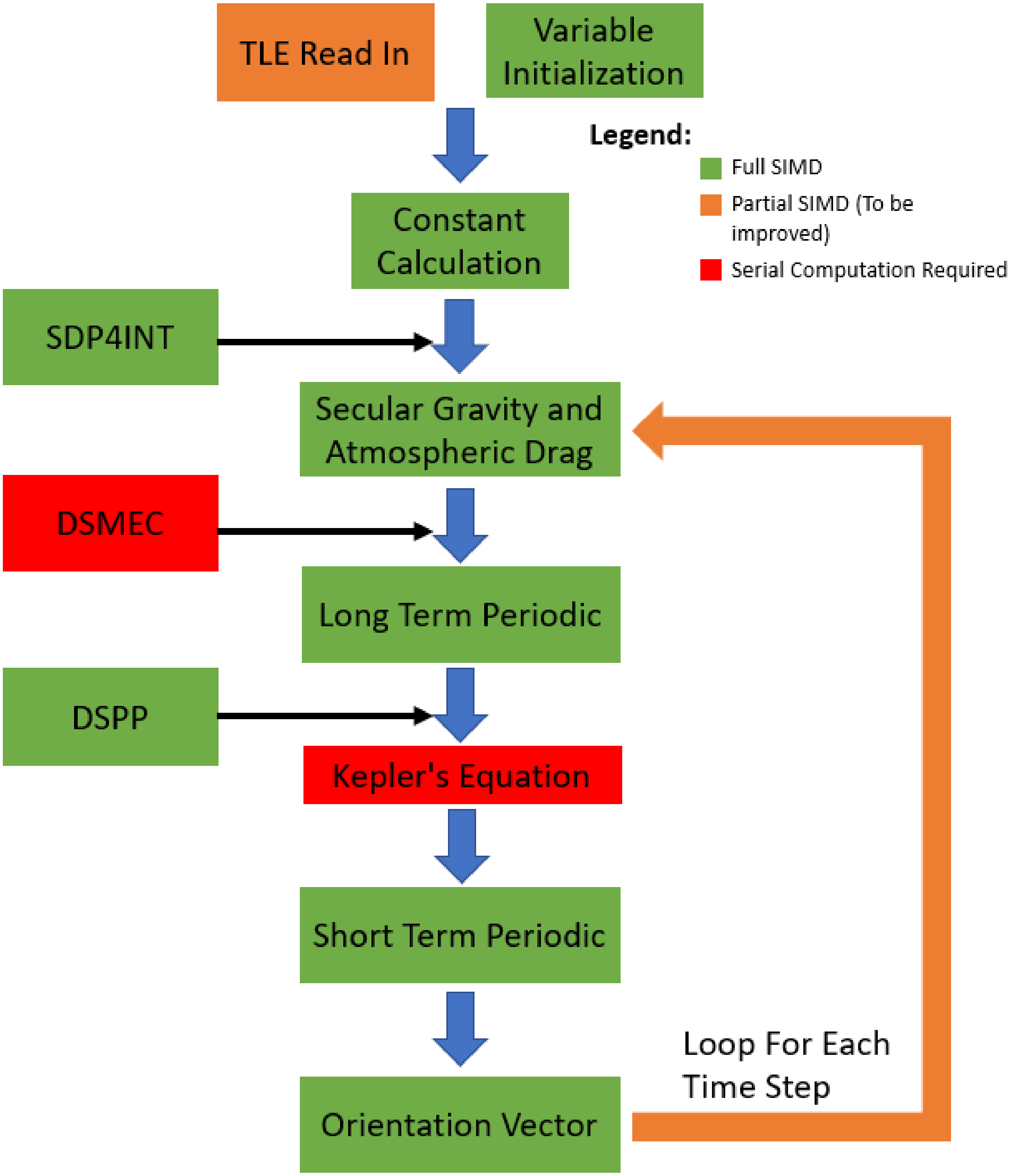


Figure 6: PSGP4 Algorithm Block Diagram, where the color represents the different implementations of SIMD processing. On the left of the main flow the additional PSDP4 functions are shown with the location of the implementation.

2.2 RSO Detections

To discuss RSO detection first the definition of what a detection means in this context is needed. In this thesis we use Samuel Clemens' definition of a detection and access [37]. A detection is when a RSO is in the Field of View (FOV) of the sensor and has a Signal to Noise Ratio (SNR) that is larger than the minimal detectible SNR. An access is defined as where an RSO is in the FOV of a sensor but does not have a large enough SNR for detection. The definitions are transformed into the basic equations 16 and 17. Equation 16 represents if a RSO is in access, while equation 17 represents if a RSO is detected.

$$Access = \begin{cases} 1\,, & \theta_{RSOn} < \theta_{FOV} \\ 0\,, & \theta_{RSOn} > \theta_{FOV} \end{cases} \tag{16}$$

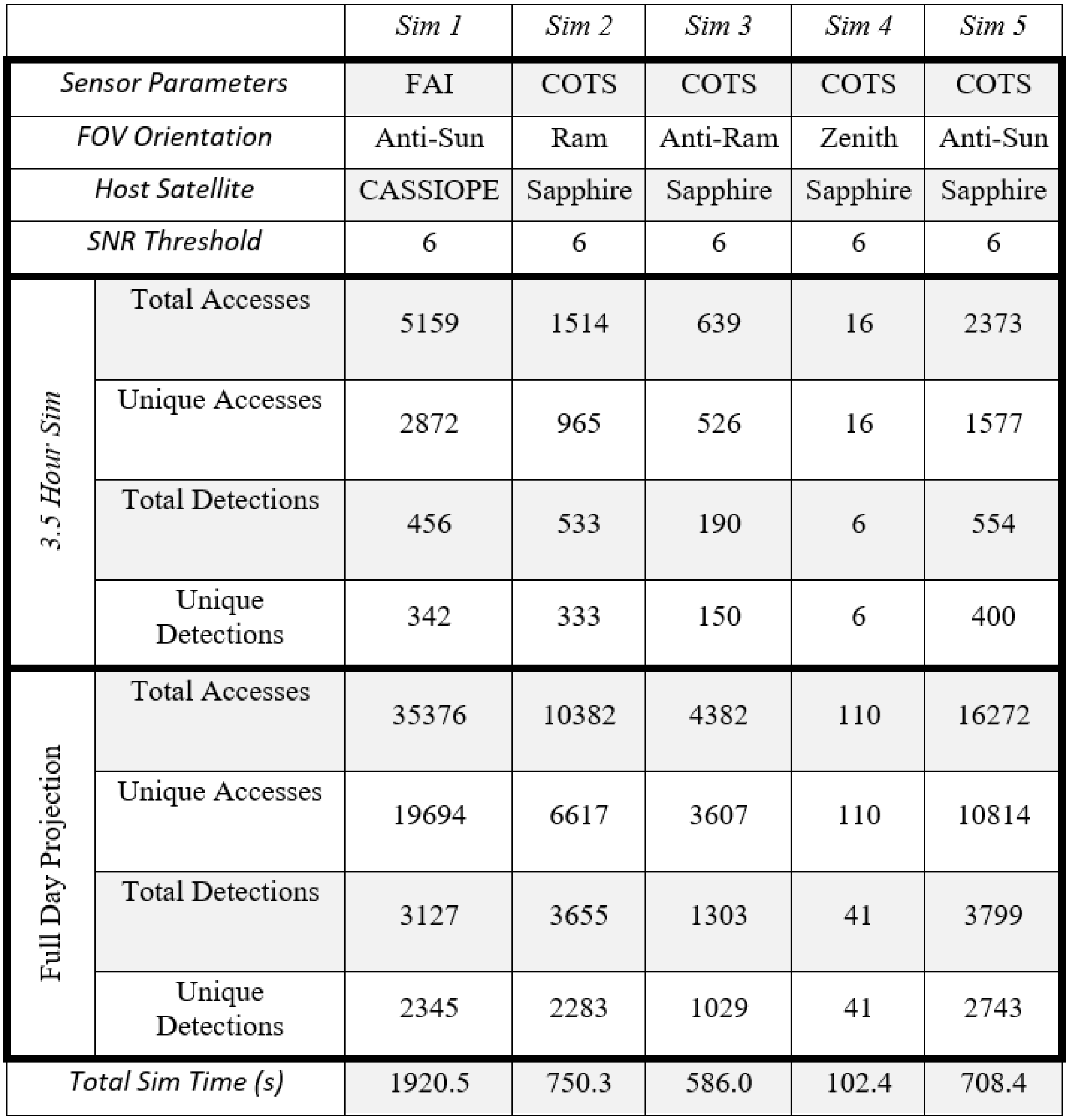
$$Detection = \begin{cases} 1, & \theta_{RSOn} < \theta_{FOV} \; and \; SNR_{RSO} > SNR_{Min} \\ 0\,, & \quad Every \; other \; case \end{cases} \tag{17}$$

In equations 16 and 17: θ_{RSOn} represents the angle from the boresight of the sensor to the RSO. θ_{FOV} represents the half angle FOV of the sensor. SNR_{RSO} represents the SNR of the RSO as detected from the sensor. SNR_{Min} represents the minimum SNR that the sensor can detect.

Equation 16 and 17 give the basic physical model of what is required for an access or detection, when operating with a real star tracker other constraints are considered. Other real-world constraints include the Earth exclusion angle, Sun exclusion angle, and eclipses, to name a few. These definitions of access and detection were used to generate different simulated results for COTS star trackers. In Table 8 the results from detection and access simulations are shown for different mounting geometries. From the results it can be seen that the number of detections and access heavily depend on what direction that the star tracker is facing, with the anti-sun having the largest number of detections and zenith having the fewest [37]. Over a full 24 hours of observations from the FAI had total detections of over 3127 detected, and 2345 unique objects. In Table 9 the pricing comparison between different RSO detection methods are shown.

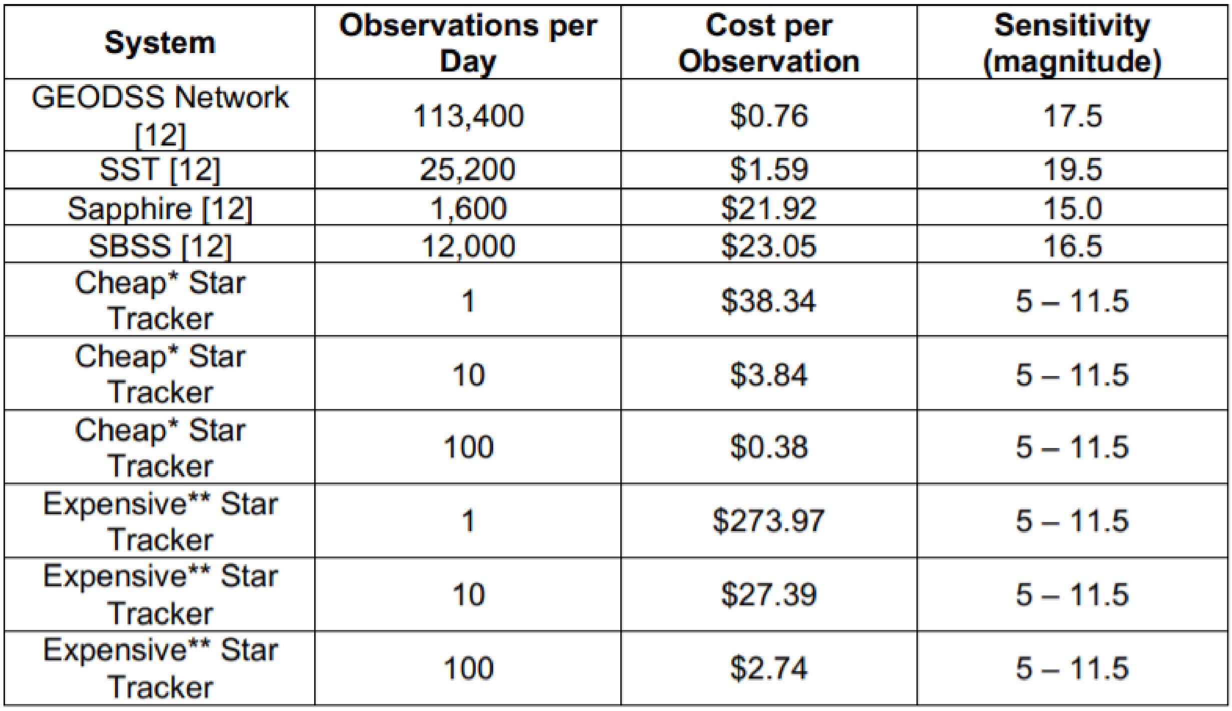
Comparing the price between terrestrial, active space sensors, and passive space sensors it can be seen that passive RSO tracking can become the most affordable option using COTS sensors. Currently passive sensors from space-based stations are not widely implemented due to the technology readiness level (TRL) being low in comparison with other more widely used methods. Increasing the TRL level through simulated environment tests, as well as, finding and correlating known RSO’s from COTS space-based sensor data looks to enable the use of passive space-based sensors on future and current missions. In Table 9 the results from detection and access simulations are shown for different mounting geometries. From the results it can be seen that the number of detections and access heavily depend on what direction that the star tracker is facing, with the anti-sun having the largest number of detections and zenith having the fewest [37].

Table 8: Number of Estimated Detection Possible in a Simulated Environment [37]



		Sim 1	*Sim 2*	*Sim 3*	*Sim 4*	*Sim 5*
Sensor Parameters		FAI	COTS	COTS	COTS	COTS
FOV Orientation		Anti-Sun	Ram	Anti-Ram	Zenith	Anti-Sun
Host Satellite		CASSIOPE	Sapphire	Sapphire	Sapphire	Sapphire
SNR Threshold		6	6	6	6	6
3.5 Hour Sim	Total Accesses	5159	1514	639	16	2373
	Unique Accesses	2872	965	526	16	1577
	Total Detections	456	533	190	6	554
	Unique Detections	342	333	150	6	400
Full Day Projection	Total Accesses	35376	10382	4382	110	16272
	Unique Accesses	19694	6617	3607	110	10814
	Total Detections	3127	3655	1303	41	3799
	Unique Detections	2345	2283	1029	41	2743
Total Sim Time (s)		1920.5	750.3	586.0	102.4	708.4

Table 9: Price Comparison of Different Types of RSO Detections [49]



System	Observations per Day	Cost per Observation	Sensitivity (magnitude)
GEODSS Network [12]	113,400	\$0.76	17.5
SST [12]	25,200	\$1.59	19.5
Sapphire [12]	1,600	\$21.92	15.0
SBSS [12]	12,000	\$23.05	16.5
Cheap* Star Tracker	1	\$38.34	5 – 11.5
Cheap* Star Tracker	10	\$3.84	5 – 11.5
Cheap* Star Tracker	100	\$0.38	5 – 11.5
Expensive** Star Tracker	1	\$273.97	5 – 11.5
Expensive** Star Tracker	10	\$27.39	5 – 11.5
Expensive** Star Tracker	100	\$2.74	5 – 11.5

2.2.1 Signal to Noise Ratio in Defining RSO Brightness

The SNR of an RSO represents the ratio of the received signal to the noise floor. High SNR for detections represents that the RSO’s intensity can easily be determined from the background noise. Low SNR detections are harder to find and extract the RSO’s intensity accurately, requiring more post processing to remove the noise present. Different optical sensors have different minimum SNR ratios that objects can be detected, for the FAI used in this thesis the SNR min used is 6. Being able to accurately simulate the SNR of RSO detections allows for the quality of detection to be determined for scheduling, correlation, and characterisation. To calculate the SNR of an RSO’s optical detection equation 18 [49], is used with the general equation being the first equation and the simplification in the context of this thesis shown below. This models most of the major noise sources present in a standard optical system. The noise sources have many sources with varying knowledge on accuracy. Better knowledge of these noises and the sources in the system leads to better extraction of the signal, allowing for lower SNR detection to be feasible.

$$\begin{gathered} SNR = \frac{e_s}{\sqrt{\delta_d^2 + \delta_r^2 + \delta_{ss}^2 + \delta_{bs}^2}} \\ SNR = \frac{e_s}{\sqrt{N_{DC} t_{int} + \delta_r^2 + e_{ss} + e_{bs}}} \end{gathered} \tag{18}$$

In equation 1817; e_s represents the signal received, in the case of optical detections it is received electrons. δ_d represents the thermal noise of the sensor, also referred to as dark current. δ_r represents the read noise of the sensor, which is caused from the pixel wells and ADC, in most cases read noise provides an absolute limit to the accuracy of the readings. δ_{ss} represents the shot noise of the sensor which follows a Poisson distribution and is due to the discrete nature of light. δ_{bg} in the noise contribution from all background sources of signal such as Stars, Moon, Zodiac, and more. N_{DC} represents the number of electrons received per second due to the dark noise, this is normally given for a CCD/CMOS, or it is found through calibration. t_{int} represents the integration time over which the signal was acquired. e_{ss} and e_{bs} represents the electrons received from the shot noise, and background noise, respectfully. In this thesis the signal to noise ratio will be used as a way to determine detections vs access. In the future the analysis and simulation of the SNR of detections can lead to; better scheduling of detections, better observatory location planning, and the training of machine learning algorithms for a wide variety of SSA applications.

2.2.2 Detection Algorithms for Low Resolution Space Based RSO Detections

With passive space-based sensors, such as star trackers, taking images regularly for there primary mission the images need to check to determine if there are RSO present in the image or not. To figure out what images contain RSO’s vs which ones don’t a detection algorithm is used to search the star tracker images. RSO detection algorithms have two main goals, to find out if an RSO is in the image, as well as find the location, size, and brightness of the RSO in the image. The location of the RSO in the image if given in an image sequence can be used for angle only orbital determination. The size of the RSO in the image represents the number of pixels that are lit from the object, this gives the quality of the detection. As most to all of the objects being detected are sub pixel size the spreading of the signal over multiple pixels comes from the Point Spread Function (PSF), with a smaller PSF representing a better-quality detection than larger found PSF. This will be covered in more detail in 2.5.3. The intensity of the RSO represents the amount of reflected flux received by the detector, this is used to generate a light curve allowing for further analysis. Determining if an image has an RSO detection will allow for filtering of the images in-situ, allowing only the ones with RSO's in them to be downlinked to the ground, or compressed and packaged for downlink. As downlinks from spacecrafts have a limited data budget having every image downloaded is not feasible for a passive system with out interrupting the main satellite mission. The filtering of the images in-situ aids with this limiting the number of images that need to be downloaded, but full quality images contain a large amount of data which would need to be downlinked for each detection. Due to the downlink data restrictions downloading images would make the system infeasible, a solution to this is to just downlink the pixels of interest, or the magnitude and location of the RSO. Exploiting the fact that a majority of the image can be disregarded a lit pixel list can be made which contains the row, column and intensity for each pixel associated with an RSO. Having a detection algorithm that finds the centroid of RSO's in the image allows for just the pixels for each RSO to be extracted, with the standard RSO size being 1 pixel with some spill over due to the PSF on average only 9 to 25 pixels need to be downlinked for each detection. With each pixel requiring 4 bytes to downlink; one byte containing the row, one byte containing the column and 2 bytes containing the DN value of the pixel. For an image with 1 RSO detection this reduces the required downlink size from megabytes of data to only 36 bytes of data, which makes the downlink of detection data feasible as a passive system. From which the RSO can be further analysed on the ground using the lit pixel list.

For this thesis developing a detection algorithm is outside the scope of the work, this is because detection algorithms are well defined for other applications with there being some made for RSO detections. Instead, the detection algorithms used for this thesis are made by Magellan Aerospace specifically for RSO detections from passive low-resolution space-based images. The two different detection algorithms used in the research are GLARED (Graphically Learned Algorithm for RSO Enhanced Detection) which is a machine learning algorithm and STARED (Star Tracker Analytic RSO Enhanced Detector) which is an analytical algorithm. A continuation of this is research currently being performed by the Nano Satellite laboratory at York testing different detection algorithms. More information on the different detection algorithms developed by Magellan Aerospace can be found here [49]. More information on York's work on detection algorithms can be found here [50]. This thesis will focus on the replication of detection from simulated images which will be gone over in chapter 4.

2.3 Light Curve Analysis

Light curve analysis, in the context of optical RSO detection, refers to the use of the light curve to determine physical characteristics of the RSO. Different types of methods from astronomy and planetary science are looked at for the purposes of RSO characterisation: phase dispersion minimization, Fourier analysis, least squares spectral analysis and glint analysis [51] [52] [53]. One method that is used for RSO characterisation but not always astronomy and planetary science is light curve inversion. Light curve inversion differs from the astronomical methods by giving the shape, attitude, and/or optical properties not just the spin period like the other mentioned methods. Light curve inversion is more computationally intensive and requires more background information then the other light curve analysis methods. All of these methods will be summarized below with which method is being selected for this research looked at in section 2.3.2.

2.3.1 Fourier Analysis

Fourier analysis of an objects light curve is done by taking the Fourier transform of the light curve to view the light curve in the spectral domain [54]. Fourier analysis has the advantage of being computationally efficient, as well as, easy to implement in most coding languages with fast Fourier transform function being common predefined functions. Fourier analysis provides the power spectrum of the light curve in the spectral domain, this includes the RSO spin rate, its harmonics and other distortion. The downside with Fourier analysis is that Fourier transforms require a constant sample rate to be used. This causes light curves that having missing measurements or varying sample rates not to be able to use Fourier analysis. A method that gets nearly the same result as Fourier analysis but can handle non consistent time series is the least squares spectral analysis method. For this reason, Fourier analysis is not considered to be a robust enough light curve analysis method for RSO light curves. For this reason, Fourier Analysis will not be tested in this thesis.

2.3.2 Overview of Light Curve Analysis Techniques

The above methods all provide a good selection for RSO characterisation with most methods being previously implemented. The light curve analysis methods give the spin rate of the object, but with current SSA needs more information is required. For this reason, light curve inversion is implemented to determine the attitude, shape, and optical properties of the RSO. This information is becoming more needed in the space environment to determine if space crafts are adhering to deorbit protocol, to see if active satellites are performing as they should, as well as, assisting in the initial attitude assessment for active debris removal [7].

2.3.3 Least Squares Spectral Analysis

Least Squares Spectral Analysis (LSSA) is a method that fits sinusoidal waves and generate a power spectral density of the function. LSSA is a very similar method to Fourier transform, both generating the power spectral density [55]. LSSA has the advantage of not requiring constant sampling rates and other restrictions of Fourier transforms. For this reasons LSSA is chosen in place of Fourier transforms to acquire the power spectral density of the light curve and use that for determination of characteristics of the light curve. The equations for LSSA is shown are below, with equation 19 representing the formula to calculate the residual, equation 20 represents the design matrix, equation 21 represents the estimated LSSA states, equation 22 represents the amplitude values for the frequency being looked at, and lastly equation 23 represents the fit of the LSSA which is 0 for a perfect fit.

$$r = F(t_i) - LSSA(\omega_j, t_i) = F(t_i) - \Phi\left(\Phi^T C_f^{-1} \Phi\right)^{-1} \Phi^{\mathrm{T}} C_f^{-1} F(t_i) \tag{19}$$

$$\Phi = [\cos(\omega_j, t_i), \sin(\omega_j, t_i)] \tag{20}$$

$$LSSA(\omega_j, t_i) = \hat{x}_1 \cos(\omega_j t_i) + \hat{x}_2 \sin(\omega_j t_i) \tag{21}$$

$$\hat{x} = \left(\Phi^T C_f^{-1} \Phi\right)^{-1} \Phi^{\mathrm{T}} C_f^{-1} F(t_i) \tag{22}$$

$$\Theta_{LSSA} = 1 - \frac{F(t_i)^T C_f^{-1} LSSA(\omega_j, t_i)}{F(t_i)^T C_f^{-1} F(t_i)} \tag{23}$$

In equations 19 to 23: r represents the residual from the found light curve to the LSSA fitted light curve. $F(t_i)$ is the true light curve brightness at point t_i. $LSSA(\omega_j, t_i)$ represents the Least Squares Spectral Estimate with a frequency of ω_j at point t_i. Φ represents the state matrix used in least squares analysis, this is represented by equation 20. C_f represents the covariance matrix for least squares analysis. $\hat{x}$ represents the estimated parameters used in least squares analysis, for LSSA it represents the amplitudes of the sinusoids with frequency ω_j. Θ_{LSSA} represents the fit of the sinusoids to the found light curve, this ranges from 0 to 1 with 0 representing a perfect fit.

2.3.4 Phase Dispersion Minimization

The Phase Dispersion Minimization (PDM), also called Phase Folding, method is very popular in the astronomical field, as it allows for the period of rotation to be extracted from non sinusoidal light curve patterns [56]. This method has the advantage of not requiring perfect sinusoidal periodics to determine the spin period. First a range of possible spin periods is estimated limiting the number of phases. From there the light curve is split into sections the length of the spin period being examined. These periods are then stacked on top of each other which is used to find a residual from the difference between the period chosen and the total light curve. The difference between the length of the total light curve and the small section being folded is called the phase. The stacked light curve sections are then binned to find the variance between each of the sections of the light curve. If the phase chosen is similar to the spin rate of the RSO the variance in the brightness should be minimal, if the period does not correspond with the spin rate there will be a high variance in the answers. The variance of each bin is then used to calculate a value between 0 and 1 (0 being the best fit and 1 being a poor fit), with the frequency corresponding to the lowest value being the estimated spin rate of the target RSO [57]. PDM has the advantages of allowing the underlying trends for light curves with large gaps or low sample rates to be reconducted from the PDM [58]. Equation 24 and 25 shows the range of frequencies chosen to search over, equation 26 shows how the phase is calculated for each point, and equation 27 to 29 showing the calculation of the variance [56]. The required input parameters to PDM are shown in Table 10.

Table 10 Input Parameters for PDM

Parameter	**Variable**
The lowest frequency to be tested.	f_{low}
The highest frequency to be tested.	f_{high}
The step size of the frequency, used to control the accuracy of the solution.	f_{step}
The number of bins that the phase gets grouped into, this is also used to control the fidelity of the system.	N_{bin}

$$i = \frac{f_{high} - f_{low}}{f_{step}} \tag{24}$$

$$f(i) = f_{low} + f_{step} * i \tag{25}$$

$$rem(F(T), f(i)) = Phase \tag{26}$$

$$\sigma_S^2 = \frac{\sum (y_i - \tilde{y})^2}{N_S - 1} \tag{27}$$

$$\sigma_T^2 = \frac{\sum (N_S - 1)\sigma_S^2}{\sum N_S - N_{bin}} \tag{28}$$

$$\Theta_{\mathrm{PDM}} = \frac{\sigma_S^2}{\sigma_T^2} \tag{29}$$

In equation 24 to 2529: f_{low} represents the lower frequency bound set by the user. f_{high} represents the higher frequency bound set by the user. f_{step} represents the frequency step size set by the user. i represents the number of iterations, which represent different folds, to be performed. $f(i)$ represents the frequency selected for folding on i^{th} iteration. $F(T)$ represents the total frequency of the observed signal. $Phase$ represents the phase shift, this needs to be considered as not all frequencies will create a perfect multiple of the total frequency. Assigning each point, a phase, for in respect to the examined frequency, allows for the grouping of the phases not to lose any data to non perfect multiple of the signal frequency. σ_S^2 represents the variance in one bin, bin S, of the folded light curve. y_i represents the different points that fit in the S bin. $\tilde{y}$ represents the mean of the different points that fit in the S bin. N_S represents the number of points in bin S. σ_T^2 represents the total variance of all the folded sections. N_{bin} represents the number of bins used for the phase folding. Θ_{PDM} represents the fit of the chosen frequency ranging between 0 and 1 with 0 being a perfect fit.

2.3.5 Glint Analysis

Compared to the other methods mentioned above glint is very different. Glint analysis looks at the periodicity between the peaks in the light curve instead of trying to take a look at the complete curve. A glint is a specular (mirror like) reflection off of a surface instead of diffuse reflection which is normally seen off of RSO’s, for more information see section 2.5. Specular reflection happens when specific viewing geometry between the light source, object, and detector is achieved. This method has been used since the 70’s with papers being published containing phase functions of common satellite shapes called facets [20] [59].

An example of a light curve that is good for glint analysis is shown in Figure 7. From Figure 7 it can be seen that the brightness has periodical peaks that correspond with the spin rate of the satellite, or one of its harmonics. One of the challenges, when it comes to glint analysis, is differentiating between periodic spikes and the true spin rate. The peaks in Figure 7 show a spin period of around 12.6 seconds, taking only the wider peaks gives a spin period of approximately 25.3 seconds. The two different peaks could correspond to different solar panels or reflective surfaces. With out prior knowledge of the satellites shape it is hard to determine which spin period is the satellites true spin rate. In the case of KMS 3-2 the satellite has one large polar panel facet made up of deployable solar arrays and is reported to be tumbling since its launch [60]. Glint analysis has the advantage of being very computationally efficent looking at only the peaks in the curve not the whole curve, also it does not need to do repetitive calculations unlike PDM and LSSA.

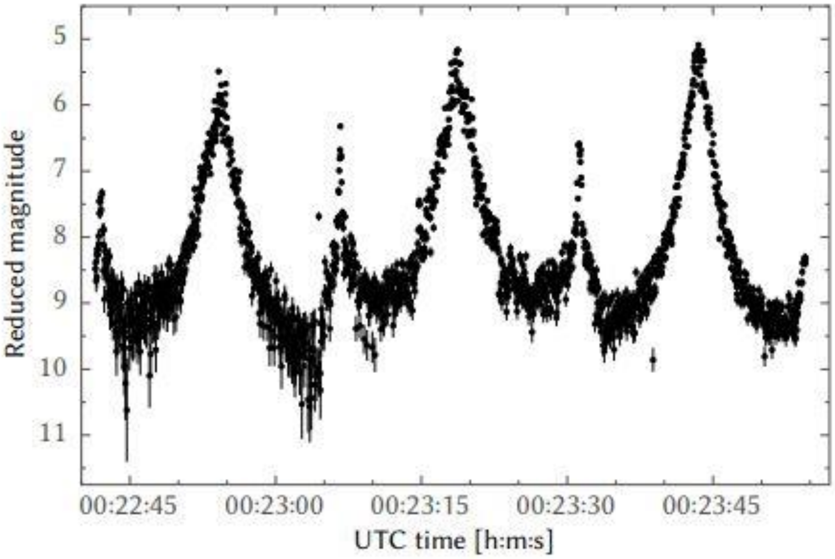


Figure 7 KMS 3-2 light curve taken on Feb 2, 2016. [60]

2.3.6 Light Curve Inversion

Light curve inversion is not one single method, but rather an umbrella term that cover different methods that determine more information about the object than standard light curve analysis methods. Standard light curve analysis methods find the spin rate with light curve inversion finding other properties such as shape, size, and optical properties. This main way of performing light curve inversion is through a facet analysis of the RSO. Facets are surfaces with a well know Bidirectional Reflection Distribution Function (BRDF). To create an accurate 3D model of the target RSO different facets are put together represent the shape of the satellite, I the form of reflective surfaces. Equation 30 through 35 shows the implemented example, in the context of this thesis, of the calculations required for light curve inversion, with a description of the variables and procedure below [31].

$$\boldsymbol{F} = A\boldsymbol{s} \tag{30}$$

$$A_{in} = BRDF\left(\boldsymbol{N}_{n,sen}, \boldsymbol{N}_{n,sun}, \Phi_{p_i}\right)\alpha_n \tag{31}$$

$$\boldsymbol{S_n} = e^{EA_n} \tag{32}$$

$$\chi_p^2 = arg\min \left|\sum \frac{\boldsymbol{F}^k - A^k\boldsymbol{s}}{\bar{\boldsymbol{F}}^k}\right| \tag{33}$$

$$\chi_f^2 = arg\min \left|\boldsymbol{F} - c_{scale}\boldsymbol{F}_{LCI}\right| \tag{34}$$

$$c_{scale} = \frac{\sum \boldsymbol{F}^k \boldsymbol{F}_{LCI}^k}{\sum {\boldsymbol{F}^k}^2} \tag{35}$$

In equations 30 to 35: $\boldsymbol{F}$ represents the matrix of brightness measurements. A represents the design matrix which in this case represents the bidirectional reflectance distribution function. $\boldsymbol{s}$ represents the matrix of the facet sizes and shape, described in more detail in section 2.5. $\boldsymbol{N}_{n,sen}$ represents the normalized vector in the facet frame n to the sensor. $\boldsymbol{N}_{n,sun}$ represents the normalized vector in the facet frame n to the Sun. Φ_{p_i} represents the phase angle in the facets frame. α_n represents the reflectivity of facet n. EA_n represents the effective area of facet n. χ_p^2 represents the preliminary set of parameters found to represent the best estimated states. χ_f^2 represents the final set of parameters found to represent the best estimated states. c_{scale} represents the scaling factor of the light curve.

2.4 Maneuver Detection

Maneuver detection is done to monitor active satellites and space debris, look for space events, and ensure that satellites adhere to the deorbiting standards. There are two main types of orbital maneuvers: in-plane and out-of-plane. In-plane orbital maneuvers affect the energy of the orbit, changing its eccentricity and semi-major axis. Out-of-plane orbital maneuvers affect the inclination and right ascension of the ascending node (RAAN), usually manifesting as abrupt changes [61]. There are two main methods used for RSO maneuver detection, the first using previous TLE data to predict future TLE parameters. The second method uses state propagators to estimate the RSO's location, both methods are described in sections 2.4.1 and 2.4.2, respectively. Section 2.4.3 describes how the two approaches are fused into a hybrid maneuver detection method.

2.4.1 TLE Analysis for Maneuver Detection

TLE analysis for maneuver detection uses historical data in the form of previously taken TLEs. These TLEs are used together to create a trend of the objects' Keplerian orbital elements versus time, which is used to estimate the Keplerian orbital elements for a TLE at a new time. New TLEs are compared to the estimated value from the trend. If the new TLE is significantly different, it is flagged as indicating a possible orbital maneuver. Using historical TLE data for maneuver analysis has the advantage of being less computationally intensive than the state propagation method, because it does not require the use of an orbital propagator [62]. Existing TLE analysis detection uses various fitting techniques, threshold values, and historical data sizes [61] [62] [63]. Some of the disadvantages of the TLE analysis method come from the uncertainty in TLE measurements, as well as the dependency on historical TLE data [61]. The uncertainty of the TLE measurements adds error into the system, and because the TLE generation method is not known, error correction can be challenging. The method implemented in this research is a version of the method proposed by Kelecy et al. [61].

In this method, semi-major axis, eccentricity, and orbital energy calculated with equation 36 [62], are examined over time with a moving window. In equation 36, ε is the orbital energy and n_o is the mean motion. The moving window uses the parameters shown in Table 11 Parameters for Moving Filter, which represents a simple case of using the previous 14 TLEs with equal weighting. For more information on moving windows, see [64], and for specific SSA implementations, see Kelecy et al. [61]; Lemmens and Krag [63].

$$\varepsilon = -0.5\left[\frac{\pi N_o \mu}{43{,}200}\right]^{2/3} \qquad 36$$

Table 11 Parameters for Moving Filter

Parameter	Value
Size of the moving window	14
Weight of incoming point	1
New data point included	1

The standard deviation and mean are calculated for each iteration and each parameter of the moving window. Once the mean and standard deviation are found, a new point is introduced to the window, which is the point being looked at for an orbital maneuver. The new point is used to update the mean and standard deviation, and then tested against the mean using an n-σ threshold method. To estimate the trend, a line of best fit is used to estimate the variables within the moving window. Each point that is over the n-σ threshold generates a flag for possible maneuver, and the flags over the entire time series are then revisited afterward to determine whether the TLE is just an outlier or in fact due to an orbital maneuver.

2.4.2 State Propagation for Maneuver Detection

State propagators propagate from one TLE to the next to identify whether any unexpected deviation occurs. This method is computationally more intensive than TLE methods but has the advantage of not requiring historical TLE data, which is not always available for all space objects. Just like TLE analysis, state propagation is very sensitive to TLE errors, which can cause false detection of orbital maneuvers. The TLE propagation is done via a parallel implementation of the SGP4 propagator to increase the computational speed and then compared to the nonparallel SGP4 propagator. The error associated with the error in TLE measurements is shown in Equation 37.

$$|R_i| = \sqrt{\left(X_{TEME,i} - \hat{X}_{TEME,i}\right)^2 + \left(Y_{TEME,i} - \hat{Y}_{TEME,i}\right)^2 + \left(Z_{TEME,i} - \hat{Z}_{TEME,i}\right)^2} \tag{37}$$

In Equation 37, $X_{TEME,i}$, $Y_{TEME,i}$, and $Z_{TEME,i}$represent the *x*, *y*, and *z* components of the estimated position in the TEME (true equator mean equinox) coordinate system, respectively, and, $\hat{X}_{TEME,i}$, $\hat{Y}_{TEME,i}$, and $\hat{Z}_{TEME,i}$ represent the *X*, *Y*, and *Z* components of the true TLE position in the TEME coordinate system, respectively. For information on the TEME coordinate system, see Seago and Vallado [65].

2.4.3 Hybrid Detection Method

This section describes the hybrid detection method that fuses the TLE method with the state propagation method. First, the hybrid detection method uses TLE analysis to see which RSOs have the possibility of orbital maneuvers. Then, the orbits of RSOs flagged for possible maneuver are propagated, with the state propagation method used to confirm and characterize the maneuver. The flow of this structure is shown in Figure 3. Here, the colors represent the different stages. Blue is the initialization stage and TLE analysis, yellow is the state propagation stage, and green is the update stage. There are two paths that an RSO can take after TLE analysis if there is no historical data or if a possible maneuver is flagged. RSOs follow Route 1 if a maneuver flag is generated or there is insufficient historical data to allow for TLE analysis. RSOs follow Route 2 only if the TLE analysis was successful and no maneuver was detected.

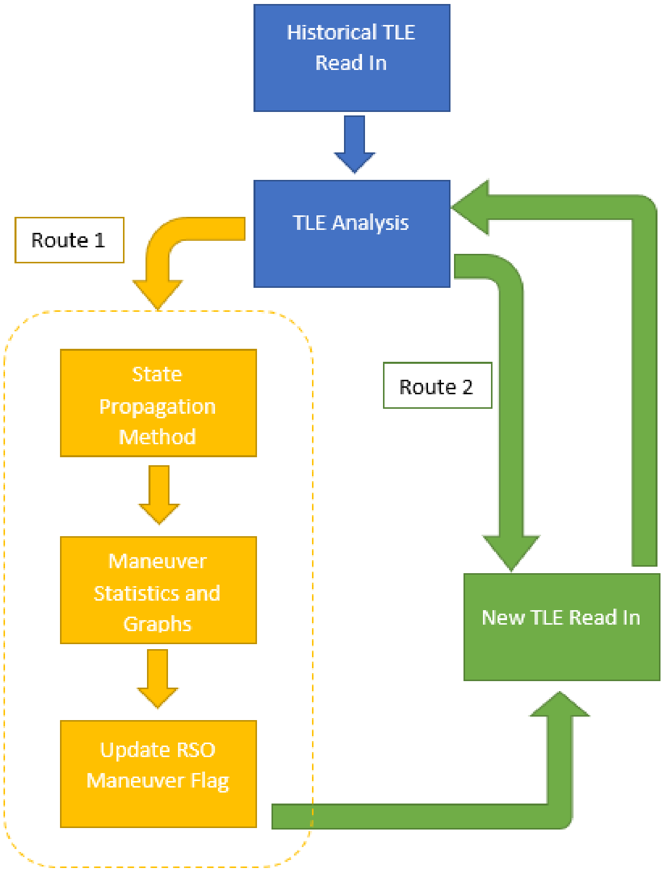


Figure 8 Flow Diagram Representing the flow of the Orbital Maneuver Hybrid Detection Method

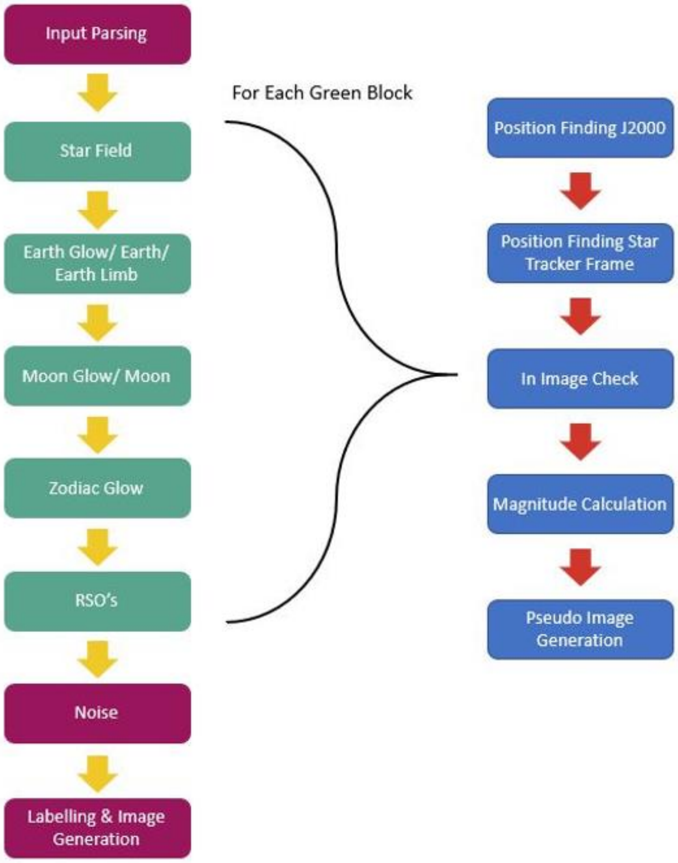
2.5 RSO Image Simulation (ORBITALS)

RSO image simulation is a form of image reconstruction using the known or estimated values of all light sources and objects in the sensors FOV. Different methods are available to reconstruct an image, such as forward and backward ray tracing models. The difference between these two models what information they use to recreate an image, forward ray tracking using information about the light sources and objects to recreate the image starting from when the light leaves the light source. Backward ray tracing does the opposite of forward raytracing and starts with the light that enters the sensor and traces it back to the light source. Both methods are optimal in different situations, for RSO image simulation the light sources are well known leading to a forward ray tracing model to be used.

ORIBTALS was originally designed to examine the feasibility of RSO detection in space. In order to understand the characteristics of on-orbit images with RSOs, we simulated various scenarios based on commercial grade star trackers currently available in the market. For the purpose of simulation, we considered several star trackers, including AD-1 Star Tracker by Mars Bureau [66], and BOKZ-MF Star tracker by IKI RAN [67]. All models are great candidates for star imaging on a small satellite platform for attitude de-termination. However, for the purpose of this study, we focused on Fast Auroral Imager (FAI) parameters as the baseline imager. The FAI sensor is on board the CASSIOPE satellite as part of the ePOP (Enhance Polar Outflow Probe) science mission. The purpose of FAI is to measure large-scale auroral emissions. It measures auroral emissions with the near-infrared wavelengths (650 nm – 1100 nm) and a monochromatic wavelength of 630 nm. The parameters for FAI are shown in Table 1. Images from FAI are publicly available on the ePOP website (https://epop.phys.ucalgary.ca/) because the images are easily accessible opposed to images from COTS star trackers that are not readily available. FAI images were used to validate different portions of the image simulator.

Table 12: FAI Parameters

Parameter	Value
Pixels	256 x 256
Quantum Efficiency, QE	0.66
Optical Transmittance Loss, τ	0.9
Aperture Diameter	1.7 cm
Pixel Size	26 μm
Effective Focal Length	1.38 cm*
Integration Time	0.1 seconds



*Noise sources include read-out noise, shot noise, and the point spread function; **pseudo image refers to the simulated image that contains the single light source, namely Earth glow, moon, zodiac glow and RSO

Figure 9. space-based optical image simulator (SBIOS) system architecture (left), image generation sequence for steps 2 through 6 (right)

ORBITALS assumes that the RSOs are not active sources of flux but rather reflectors of so-lar radiation, which is common in current RSO models [68], [29] , and [69].) For the Bidirectional Reflectance Distribution Function (BRDF) modeling, we assume that satellites consist of several well-defined surfaces: sphere, flat surface and cylindrical. More details of the BRDF and facet modeling of ORBITALS is described in [70]. ORBITALS also simulates the various noise and ambient light, such as Earth glow, Moon glow, Zodiac glow, read-out noise, shot noise, and the point spread function. There are several types of active and inactive RSO's which contain different types of position data, from GPS accurate data to basic

TLE information. Due to this ORBITALS robustness includes Ephemeris, State, and TLE input options for both the RSOs and host satellite. The goal of having such a robust simulator allows for the generating large data sets for machine learning application.

2.5.1 Brightness Magnitude of Simulated Objects

Brightness is a relative measure of incoming flux that compares the flux incoming to the flux of known object with an assigned brightness value. Brightness is a negative logarithmic value with the difference between -2.5 brightness value corresponding to 10 times difference in flux. All objects are assumed spheres, except for RSO's, and their brightness is calculated with equation 38 and 39 representing the combination of specular and diffuse reflections phase functions. The phase function for spheres and common facets are shown in Table 13. B_{diff} and B_{spec} are weighting values between the specular and diffuse reflection and are related to each other via equation 40 [29].

$$M_{obj} = M_{ref} - 2.5 \log\left(\frac{d_{obj}^2}{d_{ref}^2} \alpha P(\Phi_p)\right) \tag{38}$$

$$P(\Phi_p) = B_{diff} P_{diff}(\Phi_p) * B_{spec} P_{spec}(\Phi_p) \tag{39}$$

$$1 = B_{diff} + B_{spec} \tag{40}$$

Where in equations 38 : M_{obj} represents the brightness of the target object, M_{ref} represents the reference brightness, d_{obj} represents the distance from the sensor to the object, d_{ref} represents the distance from the sensor to the reference light source, α represents the reflectivity of the object, and lastly $P(\Phi_p)$ represents the phase function which is made up of the specular and diffuse components and there weighting shown by equation 39 and 40.

The specular and diffuse components represents the two different type of reflections which together produce the reflection seen off of the target object. Specular reflection is best thought of like mirror like reflection, where the angle of incidence of the incoming light beam is the angle that the outcoming light beam leaves. Specular reflection does not disperse as much as diffuse reflection leading it to have brighter reflections than diffuse reflections, but it does have stricter geometric conditions on the objects facet, reference light source, and sensor. Below contain sub sections that cover the main light sources considered in ORBITALS.

2.5.1.1 RSO Brightness Defined Model

The modeling of RSO's is difficult to do with out any priori information, as well as this needs to be done for each RSO individually. Due to this reason, it is common for RSO's to be simulated or thought of as a perfectly spherical object with .175 reflectivity, though recent work shows that .2 reflectivity is a better approximation [37]. While this works well for simulating RSO detections and the location of the RSO in the image when it comes to accurately representing the light curve modeling RSO as spheres is not sufficient. Other methods have been suggested to allow for the RSO's shape, attitude, and optical properties to be considered and have had promising results. The problem with these methods is they require priori information about the RSOs shape, attitude, and optical properties, if these are not known accurately, they can be estimated for but will contain some error. Taking into account the shape of the RSO is done by considering the RSO to be made up of a defined number of facets. Each facet has the properties of size, orientation, and optical properties, each facet also has a unique phase function $P(\Phi_p)$. The phase function for each of the facets is dependent on the star tracker, RSO, sun geometry, as well as, the attitude of the facet, and the shape of the facet. There are two main ways of calculating the phase function for different facets, one method uses phase function that are well defined for common shapes like spheres, flat surfaces, and cylinders. These phase functions are shown in Table 13. It can be seen that for all facets, but spheres and diffuse cylindrical, the phase functions have geometric conditions to provide nonzero answers. The viewing geometry can be seen in Figure 10.

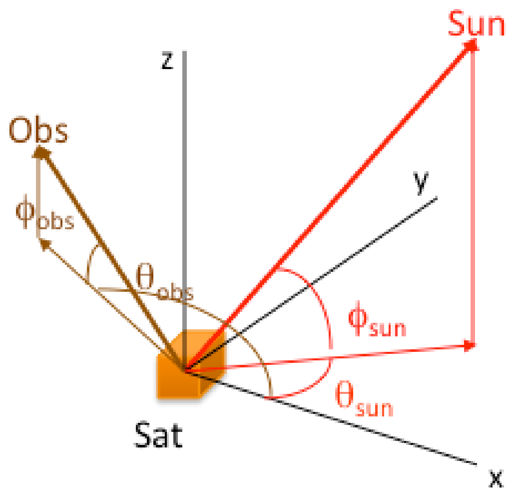


Figure 10: Defined Model Illumination Geometry [71]

Table 13 RSO Facet Dependent Phase Functions

Facet Type (Degrees of Curvature)	**Sphere (2)**	**Flat Surface (0)**	**Cylinder (1)**
Diffuse Phase Condition	None	ϕ_{obs} and ϕ_{sun} are the same sign	None
Diffuse Phase Equation F_{diff}	$\frac{2}{3\pi^2}[(\pi-\Phi)\cos(\Phi)+\sin(\Phi)]$	$\frac{1}{\pi}\sin(\phi_{obs})\sin(\phi_{sun})$	$\frac{\cos(\phi_{obs})\cos(\phi_{sun})}{4\pi}[(\pi-\theta)\cos(\theta)+\sin(\theta)]$
Specular Phase Condition	None	$\lvert\phi_{obs}-\phi_{sun}\rvert \leq \frac{\Delta}{2}$ & $\lvert\theta_{sun}-\theta_{obs}\rvert = \pi$	$\lvert\phi_{obs}-\phi_{sun}\rvert \leq \frac{\Delta}{2}$
Specular Phase Equation F_{spec}	$\frac{1}{4\pi^2}$	$\frac{4\cos(\Phi/2)}{\pi\Delta^2}$	$\frac{\cos(\Phi/2)\ \alpha(t)}{4\Delta}$

In Table 13: Φ represents the phase angle between the light source and host about the facets normal. θ_{obs} represents the longitude angle from the facet’s X axis (or a defined axis) to the host sensor about the facet normal. θ_{sun} represents the longitude angle from the facet's X axis (or a defined axis) to the Sun (or active light source) about the facet normal. ϕ_{obs} represents the elevation angle of the host sensor from the facet plane. ϕ_{sun} represents the elevation angle of the Sun from the facet plane. Δ represents the angular size of the Sun at Earth's distance from the sun. $\alpha(t)$ is a time dependent function that describes that one sees the integrated light across a chord of the sun's disk, and for this code it is assumed to be a maximum value of 1.2723. The equation for spectral reflection for a cylindrical facet is the minimum flash, the reason for this is that the time dependency of $\alpha(t)$ requires the known start and stop time of the flash which are not always available.

The other methods of calculating the phase functions does not assume the facet is a typical shape and uses the normal and area to calculate the phase function with the BRDF that best describes the curvature of the facet.

2.5.1.2 RSO Brightness Anisotropic Phong model

Since the initial facet model for satellites were first developed in the 70's, different BRDF models have been examined and implemented to represent the satellites in orbit. An example that has been widely accepted in recent years is the Anisotropic Phong model [29] [27]. This model uses empirically found values, n_u and n_v, which accounts for the reflection across the different axis of the facet. This model is proposed for RSO's image simulation because it has the possibility of reducing the possible state estimations due to the reducing ambiguity in the facet orientation. The Phong model is a much more complex example, with a simplification for flat facets shown in equations 41 through 43. Flat facets sufficiently represent active satellites with large flat surfaces, such as solar panels and reflective facts. For this current study, we have applied flat facet model to represent the RADARSAT Constellation Mission (RCM) satellites. Figure 11 shows the observation geometry of the Phong model.

$$F_{spec} = \frac{\sqrt{(n_u+1)(n_v+1)}}{8\pi} \frac{(\boldsymbol{u}_n \cdot \boldsymbol{u}_h)^{n_u(\boldsymbol{u}_h \cdot \boldsymbol{u}_u)^2 + n_v(\boldsymbol{u}_h \cdot \boldsymbol{u}_v)^2}}{\boldsymbol{u}_n \cdot \boldsymbol{u}_{sun} + \boldsymbol{u}_n \cdot \boldsymbol{u}_{obs} - (\boldsymbol{u}_n \cdot \boldsymbol{u}_{sun})(\boldsymbol{u}_n \cdot \boldsymbol{u}_{obs})} F_{ref} \qquad 41$$

$$F_{ref} = B_{spec} + \left(1 - B_{spec}\right)(1 - \boldsymbol{u}_{sun} \cdot \boldsymbol{u}_h)^5 \qquad 42$$

$$F_{diff} = \frac{28B_{diff}}{23\pi}\left(1 - B_{spec}\right)\left(1 - \left(\frac{\boldsymbol{u}_n \cdot \boldsymbol{u}_{sun}}{2}\right)^5\right)\left(1 - \left(\frac{\boldsymbol{u}_n \cdot \boldsymbol{u}_{obs}}{2}\right)^5\right) \qquad 43$$

$$F = F_{spec} + F_{diff}$$

In equations 41 to 43: F_{spec} represents the specular reflection. F_{diff} is the diffuse reflection. $F_{reflect}$ is the Fresnel reflectance. B_{spec} and B_{diff} represent the Hejduk Weighting Coefficients; n_u and n_v represent the directional distribution of the specular reflection. $\boldsymbol{u}_n$ represents the facet normal vector normalized, represented in Figure 11 as **n**. $\boldsymbol{u}_h$ represents the normalized half- vector between the light source and observer, represented in Figure 11 as **h.** $\boldsymbol{u}_u$ represents the local horizontal normalized vector, represented in Figure 11 as **u**. u_v represent the local vertical normalized vector, represented in Figure 11 as **v.** $\boldsymbol{u}_{sun}$ represent the normalized sun vector, represented in Figure 11 as $\mathbf{k_1}$. and $\boldsymbol{u}_{obs}$ represents the normalized observer vector, represented in Figure 11 as $\mathbf{k_2}$.

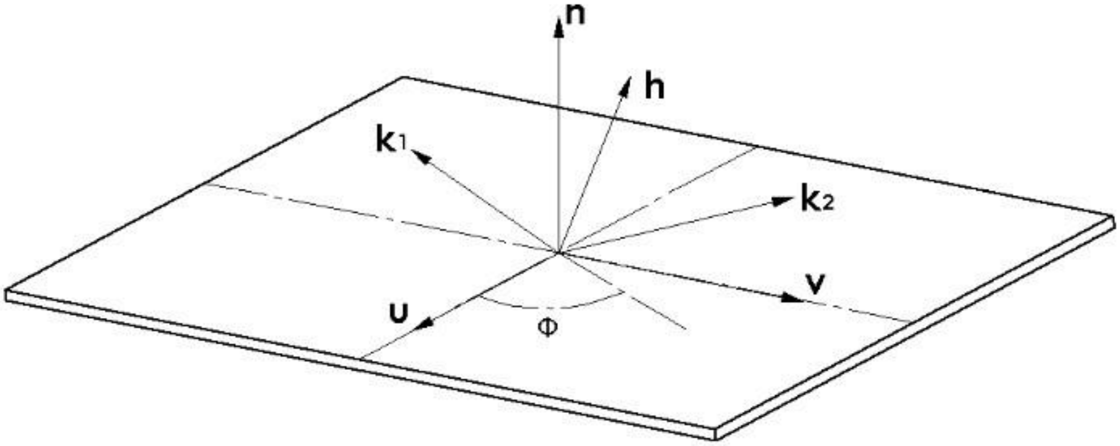


Figure 11 Observation Geometry for Phong Facet Model

2.5.1.3 Stars

The stars that were used for image simulation are from the bright star catalog [72], with the option. The reason that this catalog was used as typical star trackers can see objects with visual magnitude below 6. The position of these star is rotated from the given J2000 frame to the star tracker frame, then the ones in the FOV are included in the image.

Earth

In these images Earth is normally excluded but sometimes Earth's limb and Earth can be included. Modeling of the Earth itself is complex and does not affect the RSO. RSOs can often be seen in Earths limb, its modeling needs to be accurate enough to extract signal about RSO's if detection in Earth's limb is to be used. As the highest number of RSO detections are performed when the sensor is faced away from Earth, as seen in Table 8. Because of this high-fidelity model of the Earth are out of scope of this thesis, so Earth and its limb is modeled as a homogenous brightness sphere. To perform this homogenous model Earth and Earth's limb brightness is calculated from measurements of the emission spectrum integrated over the star tracker effective band to get brightness per arcsecond.

2.5.1.4 Moon

The moon is usually excluded through a moon exclusion angle, like the Sun, but in rare cases it can be seen in images. As it does not affect the RSO the moon will be modeled as a homogeneously bright sphere similar to the Earth, mentioned in the subsection above.

2.5.1.5 Sun

The suns brightness does not need to be included in the image due to the fact that imaging the sun would burn most of the electronic equipment. The magnitude of the sun is an important quantity that is used as a reference magnitude for RSO magnitude calculations. The reason for this is that the flux and flux density of the sun is well known, as well as the majority of the light that reflects off of the RSO is from the sun. In this thesis the brightness of the Sun is set to be -26.73 represented by M_{sun}.

Zodiac Brightness

Zodiac brightness represents the brightness from the background of space. There are quite a few different ways to model this brightness including implemented methods such as [73] [74]. For the context of this thesis the zodiac brightness was modeled as a constant function of the spectral band of the sensor.

2.5.2 Brightness Magnitude to Digital Number Value

The brightness magnitude of an object gives the flux given off of the object, but not the amount of flux received by the sensor and its digital number value. Digital number value (DN) is the amount of flux received from the sensor after all losses converted from an analog to a digital number which acts as the raw data that is received for each pixel from the sensor. Converting from the found magnitude of the object to the DN value uses the parameters of the sensor used. To account for the active sources of flux, they are considered either as point sources or area sources with different calculation performed for each. Point sources include the Sun, and stars that are calculated from their known magnitude. The equations to calculate the given magnitude to the Digital Number (DN) values are shown in equations 44 to 45 [75].

$$E_{obj} = E_{zero} \times 10^{-0.4m_{obj}} \tag{44}$$

$$DN_{x,y} = E_{obj} \times eDN \times QE \times \tau \times A_{app} \times t_{int} \times PSF\left(x, y, x_{obj}, y_{obj}\right) \tag{45}$$

In equations 44 and 45: E_{obj} represents the photon irradiance from the objects in photons per second per meter squared $ph/s/m^2$. E_{zero} represents the zero-point photo irradiance found from integrating the solar spectrum over the effective band, in $ph/s/m^2$. m_{obj} represents the absolute magnitude of the object. $DN_{x,y}$ represents the Direct Number (DN) value at pixel x and y on the sensor. eDN represents ratio of photons received to the electrons given by the sensor properties, the units are electrons per DN, e/DN. QE represents the quantum efficiency of the sensor system in the band used. τ represents the optical feed loss of the sensor. A_{app} represents the area of the aperture given in m^2. t_{int} represents the integration time of the sensor in seconds. $PSF\left(x, y, x_{obj}, y_{obj}\right)$ represents the point spread function on pixel x, y for a source that is located at x_{obj}, y_{obj}. Note that the Sun is never included in the image due to the Sun exclusion angle.

For area sources such as Earth, Earth's limb, the Moon, Moon glow, and zodiac glow, are calculated from there surface brightness. The equations to transform the given magnitude to the DN values are shown in equations 46 to 48.

$$L_{obj} = E_{zero} \times 10^{-.04 m_{asc,obj}} \left(\frac{180}{\pi}\right)^2 3600^2 \tag{46}$$

$$E_{obj} = L_{obj} \times IFOV^2 \tag{47}$$

$$DN_{pp} = E_{obj} \times eDN \times QE \times \tau \times A_{app} \times t_{int} \tag{48}$$

In equations 46 to 48: L_{obj} represents the photon radiance give in units of photons per second per meter squared per steradian $ph/sec/m^2/sr$. E_{zero} represents the zero-point photo irradiance found from integrating the solar spectrum over the effective band, in units of $ph/s/m^2$. $m_{asc,obj}$ represents the magnitude per square arc-second of the surface object given in magnitude per arc-second squared m_{obj}/arcsec^2. E_{obj} represents the photon irradiance from the objects in photons per second per meter squared, $ph/s/m^2$. $IFOV$ represents the instantaneous field of view which is found from dividing the pixel size by the focal length. DN_{pp} represents the DN values per pixel across the sensor. eDN represents photo electron the DN ratio given in electrons per DN, e/DN. QE represents the quantum efficiency of the sensor system in the band being used. t represents the optical feed loss of the sensor. A_{app} represents the area of the aperture given in units meters squared m^2. t_{int} represents the integration time of the sensor in seconds. Note that in equation 3, 3600^2 represents conversion from arcseconds squared to degrees squared, with $(180/\pi)^2$representing the conversion from degrees squared to steradians.

To calculate the brightness of the RSO's in the image, a BRDF, like the ones mentioned in the section above, is used with the incoming solar flux. To give a simple example of the RSO magnitude to DN value conversion RSO’s are assumed to be perfectly spherical objects with a 0.2 reflectance value. Equations 49 to 54 shows the calculation of DN values from the spherical facet model of RSO’s.

$$\rho AF = \rho A(B_{diff}F_{diff} + B_{spec}F_{spec}) \tag{49}$$

$$F_{diff,Spherical} = \frac{2}{3\pi^2}\left((\pi - \Phi)\cos(\Phi) + \sin(\Phi)\right) \tag{50}$$

$$F_{spec,Spherical} = \frac{1}{4\pi^2} \tag{51}$$

$$m_{obj} = m_{sun} - 2.5\log_{10}\left(\frac{\rho AF}{a^2}\right) + 5\log_{10}\left(\frac{d_{rso,sun} \times d_{rso,sensor}}{a^2}\right) \tag{52}$$

$$E_{obj} = E_{zero} \times 10^{-0.4m_{obj}} \tag{53}$$

$$DN_{x,y} = E_{obj} \times eDN \times QE \times \tau \times A_{app} \times t_{int} \times PSF(x, y, x_{obj}, y_{obj}) \tag{54}$$

In equations 49 to 54, ρAF represents the reflective area phase function which is an intermittent step on calculating the magnitude of the RSO. ρ represents the reflectivity of the object. A represents the effective area of the RSO in meters squared. B_{diff} and B_{spec} represent the diffuse and specular Hejduk mixing coefficients, respectfully [29]. F_{diff} and F_{spec} represent the diffuse and specular phase function respectfully. Φ represents the solar phase angle of the Sun, RSO, and host sensor. m_{obj} represents the magnitude of the RSO. msun represents the magnitude of the sun, which is constant at -26.73. a represents one astronomical unit, the distance from the Sun to Earth. $d_{x,y}$ represents the magnitude distance from object x to object y. The rest of the variable are the same as in equation 44 and 45 which are described above.

2.5.3 Simulated Noise

Including noise in the simulated images allows for them to be more similar to true images and allow for machine learning algorithms to be trained off of the images. Noise also allows for the quality of the real detection to be estimated instead of an ideal detection that would be given from no additional noise. The different types of noise that are included in ORBITALS are described below. All noise modeling was taken from Samuel Clemens work and can be found in On-Orbit Resident Space Object (RSO) Detection Using Commercial Grade Star Trackers [37].

2.5.3.1 Shot Noise

Shot noise is from wave particle duality of radiation that hits the sensors and excited the pixel. Due to the discrete nature of photons there is random variation in the photos received over same integration times. This is important as shot noise can cause false positives for low SNR RSO detection so including it in simulated images would allow machine learning algorithms to be more effectively trained to avoid false positive detections.

2.5.3.2 Dark Current

The dark current is the background bias that the camera has, while ideally it is the same over all of the pixels in the camera, in reality each one can have a different dark current. To correct for this, images with a cover over the camera can be taken and used to calculate the average dark current. Most sensors provide the average dark current for a given temperature in the data sheet allowing for easy implementation in the simulator.

2.5.3.3 Hot Pixels

Hot pixels are caused by difunctional pixels that are always fully or partially triggered if something is in that pixel or not. These appear as either fully bright or dim pixels in images that do not move across a sequence of images. To correct for this a sequence of images can be used to remove the hot pixels as they can be mistaken for RSO’s.

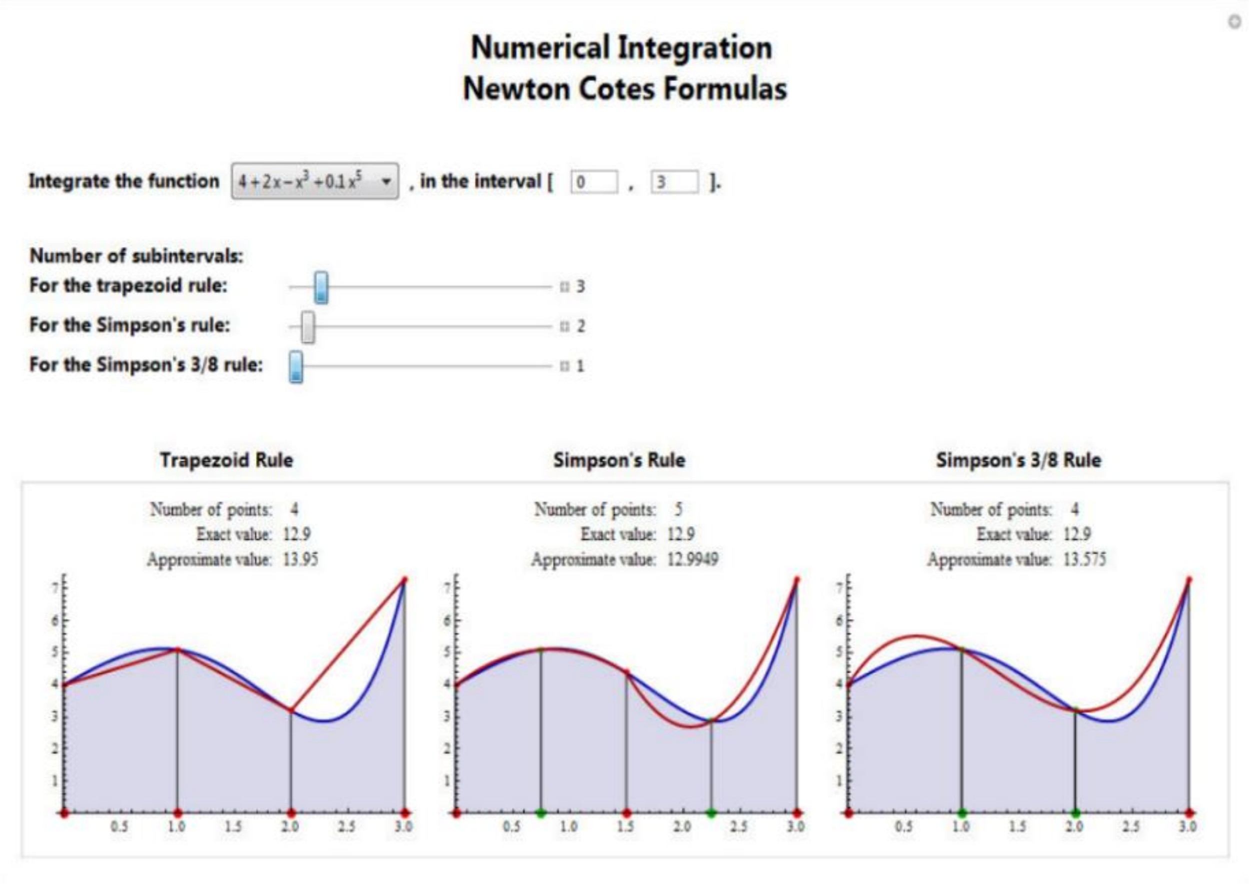
2.5.3.4 Point Spread Function

The point spread function (PSF) is the overflow of values from one pixel to another this happens from the optics being non ideal. The PSF is applied to all incoming signals and has to be corrected for the get scientifically correct values for the flux off of an object. Different methods of point spread function deconvolutions are possible most of which require a non blurry or distorted image to find the values used for deconvolution. In the context of this thesis a 2D gaussian PSF is used with the two dimensions (X and Y axis pixels) being independent from each other. More complex modeling of the PSF is required for higher resolution images, due to this thesis focusing on low resolution images this is considered out of scope.

2.6 Numerical Methods

Numerical integration allows computers to do integration and derivations of formulas as they require analytical methods unlike how humans normally do numerical integration. The focuses on numerical integration as a numerical method and the effects it has on the outcome. The easiest way to think of numerical integration as having a graph with some curve as represented in Figure 12, the numerical integration of the function is summing all of the area under the curve as shown by Figure 12 [76]. This is easily accomplished by humans when the function is well known and described but when the function is not known it can become a problem, numerical integrator used to solve integration by computers. Numerical integrators work by finding a start and end point on the curve and approximating the area underneath by different methods, the more sophisticated the higher accuracy the approximation in most cases. The simplest method is estimating a linear slope in between the points with more complex methods fitting higher order polynomials and calculating the area underneath.

Problems like the two body and J2000 equation are solved by numerical methods integrating the well known $\ddot{X}$ to become the position represented by X. This causes numerical integrators to become important when considering the accuracy and computation time of propagation. Simple numerical integrators use linear or low order equations to estimate the function curve, while this is method is able to run quickly it does not approximate the integral accurately over large periods of nonlinear change. Higher level methods have the advantage of being able to approximate the function well but are more computationally intensive than the lower order methods. The choice of state propagator used has a large effect on the computation time and accuracy of the resulting propagation. More sophisticated state models require higher order numerical methods to keep their accuracy. For example, when propagating an RSO to know its rough position a second from the starting time a simple two body problem or J2 force model can be used with a low order numerical integrator. Yet if we were looking to propagate an RSO for a day or longer more sophisticated higher order models are required to get accurate answers. For this reason, in this thesis higher order numerical integrators are used.



Numerical Integration
Newton Cotes Formulas

Integrate the function $4+2x-x^3+0.1x^5$, in the interval [0 , 3].

Number of subintervals:
For the trapezoid rule: 3
For the Simpson's rule: 2
For the Simpson's 3/8 rule: 1

Trapezoid Rule
Number of points: 4
Exact value: 12.9
Approximate value: 13.95

Simpson's Rule
Number of points: 5
Exact value: 12.9
Approximate value: 12.9949

Simpson's 3/8 Rule
Number of points: 4
Exact value: 12.9
Approximate value: 13.575

Figure 12 An Example of Different Numerical Methods [76]

Looking at the types of numerical methods we can split numerical methods into two main methods, implicit vs explicit methods. Explicit methods are defined as when the dependent variables can be calculated directly from known quantities, implicit methods are for when the dependent variables are given via coupled equations that need to be solved with a matrix or iterative process. While both methods can be used for satellite propagation typically explicit method are used due to their ease to implement, as well as low computational intensity compared to implicit methods. Implicit methods require multiple calculation for one value making them more computationally intensive but have the advantage of having better system stability [77]. Currently in the field of propagation more and more implicit methods are being used with the combination of parallel processing to increase their computational intensity some example are present in [78] [79] [42]. As numerical methods are not the focus of the research but rather a tool to get the correct answer, the simpler version of both methods will be used.

With explicit numerical methods there are two subcategorize adaptive vs fixed step size. Fixed step size methods use constant steps in the independent variable, adaptive step size methods change the size of the step size based on control values. Many control values can be used, for example: approaching a critical point or the n^{th} order error, which is implements in section 2.6.2 Dorman Prince 5(4) below. In the preliminary research both an adaptive step size and fixed step are used to test the difference between the two, the two methods used are Runge Kutta 4^{th} Order (RK4) and Dorman Prince 5(4) (DP54). While these are relatively simple numerical methods for orbital propagation, with higher order methods such as Runge Kutta 7 (8) and Gauss Jackson being commonly used. The point of this is to provide a preliminary test on the differences between methods to help decide what type of numerical method should be used in the ORBITALS simulator, and the effect of introducing parallel processing techniques.

2.6.1 Runge Kutta 4th Order

Runge Kutta 4th order method (RK4) is a 4th order numerical method that uses fixed step sizes to solve ODE's. RK4 method uses the weighted average slope of the function at multiple intermittent points to estimate the point after a specified step size. The general equation for explicit methods is described in equation 55 and 56 below [80].

$$k_i = hf\left(t_i + c_i h, x_n + a_{i1}k_1 \dots + a_{i,i-1}k_{i-1}\right) \tag{55}$$

$$y_{n+1} = y_n + \sum_{i=1}^{j} b_i k_i \tag{56}$$

In equations 55 and 56: y_n is a known initial value of the dependent variable. y_{n+1} is the value of the dependent variable being solved for. x_n and t_i are known initial values of the independent variables. k_i represent the estimated slopes at intermediate points. $a_{i,j}$, b_j, and c_i represent coefficients for the RK4 method and are summarized in a Butcher tableau below.

Table 14 The Butcher Tableau for Runge Kutta 4th Order

j:	1	2	3	4	i:
0					1
1/2	1/2				2
1/2	0	1/2			3
1	0	0	1		4
	1/6	1/3	1/3	1/6	

The Butcher tableau shows what coefficients need to be used for explicit and implicit numerical methods. Butcher tableau represents 3 coefficients with two independent indices; “i” represents the intermediate point being looked at and are displayed vertically above the horizontal black line. “j” indices represent the contribution from intermediate values, which is shown to the right of the vertical back line. The 3 different coefficients represented are $a_{i,j}$, b_j , c_i which are separated by the black solid lines. $a_{i,j}$ are the values above the horizontal line and to the right of the vertical line, b_j are the values below the horizontal solid line and c_i to the left of the vertical solid line.

2.6.2 Dorman Prince 5(4)

The Dorman-Prince 5(4) method (DP) is a Runge-Kutta method that uses a fourth order method with a fifth order error correction for adaptive step size. This method is a special case of Runge-Kutta methods that allows for convenient and fast adaptive step size. What makes this method unique is that it uses first same as last property (FSAL) allowing seven stages with only six function evaluations. The Butcher tableau for the Dorman Prince 5(4) method can be found below in Table 15 [80].

Table 15 Dorman Prince 5(4) Butcher Tableau

j:	1	2	3	4	5	6	i:
0							1
1/5	1/5						2
3/10	3/40	9/40					3
4/5	44/45	-56/15	32/9				4
8/9	19372/6561	-25360/2187	64448/6561	-212/729			5
1	9017/3168	-355/33	46732/5247	49/176	-5103/18656		6
1	35/384	0	500/113	125/192	-2186/6784	11/84	7
	35/384	0	500/113	125/192	-2186/6784	11/84	0
	5179/57600	0	7571/16695	393/640	-92097/339200	187/2100	1/40

The Butcher tableau for DP has the same variables as the Butcher tableau for RK4 with one extra variable: b_j^*. b_j^* represents the 5th order Dorman Prince coefficients being located below b_j which represents the 4th order Dorman Prince coefficients. Using the 4th order coefficients equation 57 and 58 below are used to estimate the error at that point.

$$y_{n+1}^* = y_n + \sum_{i=1}^{j} b_i^* k_i \qquad 57$$

$$O_{n+1} = |y_{n+1} - y_{n+1}^*| \qquad 58$$

The error at the given point is then used to find a tolerance ratio using equation 59 below.

$$TR_{n+1} = \frac{O_{n+1}}{atol + |y_n| rtol} \qquad 59$$

In equations 58 and 59: TR_{n+1} represents the tolerance ratio for the updated point. O_{n+1} represents the error between the estimates of different orders, in this case it is between the 4th and 5th order. $atol$ represents the absolute tolerance of the system, this is a constant value used to help update the step size. $rtol$ represents the relative tolerance of the system, like the absolute tolerance it is used to update the step size and represents how the error scales with the increase in y_n. Calculating the tolerance ratio allows for the step size to change depending on if the error is with in, or outside, the error tolerance. If the tolerance ratio is greater than one the error is above the allowed tolerance and the step size needs to decrease to increase accuracy. If the tolerance ratio is less than one the error is below the allowed tolerance and the step size is increased to decrease computation time. Lastly, if the tolerance ratio is one the step size does not change, this can be seen in Equation 60 below which calculates the next iterations step size.

$$h_{n+1} = Sh_n \sqrt{\frac{1}{TR_{n+1}}} \qquad 60$$

In equation 60: h_n represents the initial step size/ the step size used the last iteration. h_{n+1} represents the updated step size for the current iteration. S represents the safety factor which is elaborated upon below. TR_{n+1} represents the tolerance ratio for the current iteration. The safety factor (S) is there to change the step size each time slightly so the step size does not get stuck in unstable equilibrium points. The safety factor chosen for this research is 0.9. This is how the step size for all iterations is calculated starting with an initial value for the step size, in this study the initial step size was chosen to be 60 seconds which is a commons step size for orbital propagation. As a safety feature to make sure that the step size does not approach 0 or infinity there is a minimum and maximum step size; 1 and 300 seconds respectfully. If the step size is outside of these bound the step size gets reset to the initial value which is 60 seconds.

2.7 Optimisation Algorithms

Optimisation algorithms are used to perform optimisation on complex problems that either can not be solved through brute force methods or are too computationally intensive to solve in a required time period. Optimisation algorithms search the domain of inputted variables to find the best answer given by a set of criteria. In the context of this thesis the optimisation problem that is looking to be solved is the determination of the best possible RSO parameters to replicate the RSO's light curve. These parameters include the shape, optical properties, and attitude of the target RSO. To determine these properties the parameters are estimated then used to simulate light curves which are then compared to the real light curve of the object. Thousands to millions of simulated light curves are generated then used to find the brightness residual for the estimated parameters. The brightness residual and the corresponding RSO parameters are fed into an optimisation algorithm, with the goal of generating better RSO parameter estimates that reduce the brightness residual. Light curves are made of many individual measurements with some of the RSO parameters being static across the sequence, such as shape, and optical properties. Other RSO parameters vary across the light curve, such as attitude, which requires the estimation of different attitudes for each point in the light curve. Requiring estimation of the attitude at each point causes this problem to be solved with factorial time $O(n!)$, acting a type of traveling salesman problem [81]. The optimisation function when focusing on attitude sequences tries to reduce the brightness residual in one image, as well as, across the time series provided by the light curve. This has been split into two different portions of the research. This was done so that the best optimisation method for single image best attitude determination and best attitude sequence determination can be studies and modularly changed. For the image specific attitude determination only the brightness residual is being minimized in the optimisation problem. The response surface of brightness as a function of attitude is a complex function with no well-defined gradients, requiring pseudo gradients or non gradient optimisation functions. For the best attitude sequence determination, the resultant best attitudes for each point in time are compared to find the most probable attitude sequence. This uses not only the brightness residual but the change in yaw, pitch, and roll across the sequence to find an optimised estimate. Using the physical dynamics and limits of a satellites attitude the number of results can be reduced through throwing out large jumps and discontinuities that would not be possible. This is an example of a physical constrained optimiser similar to those implemented in [23]. Below different optimisation methods chosen to be studied are mentioned, as well as a definition of the objective function that is used with each of the optimisation algorithms.

2.7.1 Objective Function

The objective function is one the most important parts of optimisation as it determines what the optimiser is looking to optimise. In this thesis the optimisation function that is used for RSO characterisation from light curve inversion is defined as a weighting between the RMSE of the estimated light curve compared to the true light curve, as well as the RMSE of the change in estimated light curve compared to the true light curve. The equation for the objective function is shown in 61. The objective function in this context is a minimum function which means the optimiser is used to find the global minimum. Below in section

2.7.2 to 2.7.6 describe the different type of optimisation algorithms which are used with the objective function described in 61.

$$F_{obj} = W_1\sqrt{\frac{\sum\left(M_{i,true} - M_{i,sim}\right)^2}{n}} + W_2\sqrt{\frac{\sum\left(\dot{M}_{i,true} - \dot{M}_{i,sim}\right)^2}{n-1}} \qquad 61$$

$$1 = W_1 + W_2 \qquad 62$$

2.7.2 Hill Climbing and Stochastic Hill Climbing

The hill climbing algorithm (HC) is a global optimization algorithm that is commonly used. The HC has two stages to it, the global search section and local search section. HC behaves like recursive random search procedure, starting with the global search phase, the starting points are chosen randomly or intelligently. The goal of the global search phase is to cover as much ground as possible in the design space so that global optimum does not get missed. After the global points are chosen, the local search phase begins, this searches around the chosen point to develop a gradient. The gradient is then used in the neighborhood of the global point to try and find a better solution, the local solution. After the best solutions are found in the neighborhood of the initial global points, new global points are generated, and the process repeats for a number of iterations or till continuous convergence on an answer. HC results can be very different from each other depending on what method is used for the global point selection, local gradient calculation, and number of data points used. Simpler HCs are fully random on there selection of global optimum which leads to a chance of finding a sub-optimal solution, to combat these simpler algorithms need more data points taken to make sure the design space is fully covered. More complex HC have intelligent point selections and more complex formula for approximating the gradient of the local search area. Thus, leads to less points being required to converge on a global optimum, but requiring more computation time per point. As a result, both simple and complex HC are used depending on the situation.

Stochastic Hill Climbing (SHC) varies from HC by producing child optimal estimates from already selected parent optimal estimates. The SHC starts the same way with a number of optimal estimates being chosen in the design space called the parent optimal estimates. After the local optimization section, the SHC algorithm uses mutations and the gradient to create new optimal estimates, called child optimal estimates, from the parent optimal estimates. The child optimal estimates are then compared with the parent, if the mutation makes the child have a better objective function score, the child replaces the parent for the next round of global search selection [82]. This allows an addition of randomness to be added to HC allowing for more efficent searches objective functions with jumps or discontinuities.

One downfall of the HC and SHC is that they can get caught in local maxima and have trouble dealing with plateaus in the objective function. Different variants of the stochastic hill climbing algorithms are possible depending on how the global points are picked after the first random distribution. The results from multiple sources show that intelligent global point selection preforms better than random point selection [83]. In the worst-case HC and SHC have exponential computational time [84] which makes it less computationally intensive than SA, and GA, mentioned below. Currently both HC and SHC methods have been used in [85].

2.7.3 Simulated Annealing

Simulated annealing (SA) is a process that is similar to SHC with one major variation. There is a probability that estimates that provide less optimum solutions are chosen based on the difference between the objective functions of both the parent and child estimates. Parent estimates are similar to global estimates being the first estimate chosen before local optimisation. Child estimates are found from the local gradient around the parent estimate and mutations. Once a child estimate is found from the parent equation 63 is used to determine if the parent or child estimate will go to the next round of optimisation [86].

$$p = \begin{cases} e^{-\frac{F_p - F_c}{c}}, & F_p > F_c \\ 1 \quad , & F_p \le F_c \end{cases} \tag{63}$$

In equation 63; p represents the probability of a sub optimal step being chosen. F_p represents the fit of the parent solution given by the objective function. F_c represents the fit of the child solution given by the objective function. c represents the cooling factor which is mentioned in more details below.

The cooling factor is used to control the probability of a non ideal step. The cooling factor is chosen based on the optimisation being performed, commonly starting with a low value and increasing as the number of iterations increase. The "cooling" of the system refers to the decrease in random steps as the number of iterations increases. There are many different types of cooling schemes such as linear, exponential, and geometric [87]. In the context of this thesis a geometric cooling scheme was chosen due to its ease of implementation and rapid fall off. A way of visualizing the SA is well put by Dan Weld's CSE 473 slides [88], where it is said that the SA algorithm is similar to allowing a ball to roll down a hill, which by itself would represent HC for minimization of a function. The difference between the two is in SA, the surface is shaking while the ball rolls adding randomness into the system, with the shaking of the surface being reduced over time [88]. Figure 13(a) shows an example of how the SA algorithm can outperform SHC and HC algorithms by avoiding the local minimum of the first trough and having the ball make its way down to the global minimum shown by Figure 13 (b) [88].

Figure 13: An example of simulated annealing and when it would outperform hill climbing.

Figure 13(a) on the left represents the initial problem. Figure 13(b) on the right represents how simulated annealing would overcome the first local minimum and find the global minimum. As the SA is a modified form of SHC and HC it has the same worse case exponential computation time. Generally SA takes slightly longer to run each iteration than SHC and HC due to the added complexity of probability calculations. The advantage of SA is having the non optimal step increase the search space allowing for more optimal answer to be found by not getting trapped in local optimum.

2.7.4 Genetic Algorithm

The genetic algorithm (GA) is based off of natural evolution, it uses lists of solutions called chromosomes to compare different answers. The GA algorithm uses hundred thousand to millions of chromosomes depending on the design space in order to find an optimal solution. The GA has 4 stages, 3 of which loop till completion, and these four stages are: initial population, sort, recombination, and mutation. The initial population stage involves the initial solutions represented the chromosomes, being randomly or sudo-randomly generated. This stage is only performed once for initialisation and after is where the 3 iterative stages take over to supply an optimal solution. The first iterative stage is the sorting stage, in this stage the objective function is calculated for each of the chromosomes (parameter estimates) and all the chromosomes are ranked. The second stage is the recombination stage, in this stage chromosomes, called parent chromosomes, are chosen from the optimal estimates with the top objective function scores and combined with other chromosomes. The new chromosome produced from this combination is called the child chromosome. Moreover, how the optimal estimates are combined is by using cross over operation which are used to determine what aspects of the parent chromosomes are transferred over to the child chromosome. In the context of this thesis position based cross over method was implemented with more methods and their implementation being available in Appendix C. The goal of the cross over operation is to create new and better optimal estimates that are slightly different from the original parent chromosomes. This does not always happen with the changes to the chromosome potentially lowering the score of the objective function. To correct this, the lowest scoring optimal estimates from both the initial and new population are disregarded until the number of chromosomes left equals the number that was in the initial population. While this is not an optimal way to find a better optimal solution by itself, using a large initial population and many iterations it will slowly converge to local maximum in the design space. Converging on only local optimal solutions provide a problem not just for GA, but for most other optimization algorithms. To avoid this, the mutation step is introduced to include randomness in the design space. The third step in the iterative stages is the mutation step which operates very differently than the other stages. The mutation stage depends on the conditions for when a mutation operation can happen and what mutant operator is being used. One example would be if an operator chooses one of any two optimal estimates with the same objective function to mutate in order to provide a different solution. This in turn keeps randomness in the design space which prevents converging on a sub optimal solution. Different mutation operators and their conditions are possible for this thesis the displacement mutation operator was implemented with more information being available in Appendix B. After the mutation stage is finished, the iterative stages: sort, recombination and mutate are iterated over until the solution converges or a maximum number of iterations have taken place [89] [90].

GA is a type of Evolutionary algorithm, which allows GA to be good at handling functions where the assumptions on the output of the objective functions is not well known. This is a huge benefit when using GA for RSO parameter estimation, as not only are these problems complex with little known about the objective function output, but it makes the GA algorithm flexible for different satellites and missions that would have varying objective functions. The drawback to the GA includes that it is very computationally intensive requiring the creation of thousands to millions of optimal estimates in order to be able to find an optimal solution [90].

2.7.5 Hybrid Genetic Algorithms

Hybrid Genetic Algorithm (HGA) uses the complementary properties of genetic algorithms with other heuristic algorithms. Most commonly for traveling salesman problems a greedy algorithm is used, but recently more complex HGA have been examined for when the objective function output is not well known, an example of this is for satellite scheduling applications, and RSO parameter estimation [91]. There are many ways that heuristic or local optimization algorithms can be integrated with the genetic algorithm such as: incorporating heuristic algorithms to generate an initial population, involve heuristics into the GA objective function, and incorporate local optimization around child chromosomes to improve the child before going to their evaluation. While the computational intensity of the HGA is large, it makes up for it by often outperforming EA and Greedy algorithms when operating on complex problems [92].

2.7.6 Particle Swarm Optimization

Particle swarm optimization (PSO) is an optimisation method that is gradient independent optimisation method that acts similar to SHC or SA. PSO places thousands of particles randomly throughout the response surface with each of these particles having an initial ‘Velocity’. Velocity in the context of PSO is how much the particle changes the parameters it is looking to optimise each iteration of the optimiser. Normally to find the change in parameters, velocity in PSO’s case, for gradient optimisers, such as SHC and SA, the gradient is calculated. PSO being gradient independent optimiser instead uses the distance to different local and global optimums to find the change in velocity each iteration. PSO breaks the response surface into local grids, the size of these grids is determined by the user and the optimisation problem. In each of these grids the local optimum is recorded with the global optimum being found as the best of the local optimum. A particle updates its velocity from a weighted average of the previous velocity, distance to the local optimum of the grid the particle is in, distance to the global optimum, and finally a random component. For each of the optimum contribution to velocity the direction of the velocity change is found from the direction to the optimum with the size of the velocity change being related to the distance to optimum and difference in the objective function between the particle and optimum. The equation for calculating the velocity of the particles used in thesis is shown below in 64. This method has the advantage of not requiring computationally expensive gradient calculations, which make a large difference when populating the response surface with tens of thousands of particles. The particles act as a swarm slowly converging to the global optimum allowing large portion of the response surface to be covered while still converging to an optimum.

$$\boldsymbol{V}_{i+1} = W_{Pre}\boldsymbol{V}_i + W_{Loc}(M_{Loc} - M_P)(\boldsymbol{LOC} - \boldsymbol{P}) + W_{Glo}(M_{Glo} - M_P)(\boldsymbol{GLO} - \boldsymbol{P}) + \boldsymbol{Rc} \tag{64}$$

In equation 62: $\boldsymbol{V_i}$ represents the velocity vector of the particle for the ith iteration, W_{Pre} represents the weighting of the previous velocity on the next iterations velocity, W_{Loc} represents the weighting of the local optimum on the next iterations velocity, M_{Loc} represents the objective function score of the local optimum, M_P represents the objective function score of the particle, $\boldsymbol{LOC}$ represents the location vector in the response surface of the local optimum, $\boldsymbol{P}$ represents the location vector in the response surface of the particle, W_{Glo} represents the weighting of the global optimum on the next iterations velocity, M_{Glo} represents the objective function score of the global optimum, $\boldsymbol{GLO}$ represents the location vector in the response surface of the global optimum, $\boldsymbol{Rc}$ represents the random component of the velocity vector of the next iteration taken by using an uniform distributed random function bound at 25% of the maximum step size.

2.8 Optimisation Algorithm Tuning

Tuning of an optimisation algorithm is performed by evaluating the algorithm with different optimisation parameters to see which values of the parameters allow the optimisation algorithm work the best. Tuning optimisation algorithms is a well-defined topic in academia and industry, due to this tuning was not the focus of this thesis with a simple grid search being performed to tune all of the algorithms in Table 16. The simple grid search tried every possible discrete combination of tunning parameters to create a response surface from which the tuned parameters were chosen. For each of the algorithm's papers have been provided which go over more complex tunning and the application.

Table 16: Tunning Parameter for the Optimisation Algorithms Used

Optimisation Function	**Tuning Parameter/s**	**Range [Min, Max, Step]**	**Found Optimal Value**	**Tuning References**
Gradient Descent	Magnitude to Degrees*	[5, 90, 5]	30	[93] [94] [95]
	Degree Step Max*	[10, 30, 5]	15	
	Degree Step Min*	[0.1, 1, .1]	0.1	
	Total Iterations*	[50, 450, 50]	100	
	Number of Points*	[10,000, 100,000, 1,000]	20,000	
Stochastic Hill Climbing	Magnitude to Degrees*	[5, 90, 5]	30	[96] [93]
	Degree Step Max*	[10, 30, 5]	15	
	Degree Step Min*	[0.1, 1, .1]	0.1	
	Total Iterations*	[50, 450, 50]	100	
	Number of Points*	[10,000, 100,000, 1,000]	20,000	
Simulated Annealing	Magnitude to Degrees*	[5, 90, 5]	30	[97] [98] [99]
	Beta	[.85, 1, 0.01]	0.99	
	Cooling Factor	[1, 10, 1]	5	
	Degree Step Max*	[10, 30, 5]	15	

	Degree Step Min*	[0.1, 1, .1]	0.1	
	Total Iterations*	[50, 450, 50]	100	
	Number of Points*	[10,000, 100,000, 1,000]	20,000	
Genetic Algorithm	Cross Over Parameters	[.3, .7, .01]	0.5	[100] [101]
	Mutation Chance	0.0001, 0.0005, 0.001, 0.005 **	0.001	
	Total Iterations*	[50, 450, 50]	100	
	Number of Points*	[10,000, 100,000, 1,000]	20,000	
Particle Swarm Optimisation	Max Velocity*	[5, 90, 5]	30	[102] [103]
	Current Velocity Weight	[.5, 1.5, .01]	.9	
	Local Optimum Velocity Weight	[.5, 1.5, 0.l]	1.2	
	Global Optimum Velocity Weight	[.5, 1.5, 0.1]	1.2	
	Total Iterations*	[50, 450, 50]	100	
	Number of Points*	[10,000, 100,000, 1,000]	20,000	

* Represents values that were average over all of the optimisation methods for comparison study shown in section 5.1.

** These were the 4 values tested for mutation chance instead of the range and step formula performed for the other parameters

3 Parallel Propagation of Multi Satellite Systems

In this section an initial feasibility study and the results for parallel propagation are shown in section 3.1. In section 3.2, the results from the initial feasibility study were implemented for orbital maneuver detection to test serial vs parallel processing methodology. This study was published in the Advances in Space Research journal [48].

3.1 Initial Feasibility Study

This study examines the parallel propagation computational methods relative to sequential propagation computational methods for multi satellite propagation using various numerical methods. Runge-Kutta, Dorman-Prince, and Systems Toolkit (STK) Automation methods are examined over 3 design variables for computational efficiency for multi satellite propagation application rather than qualitative characterization. Results indicate that computational time is reduced up to 80% using parallel computation when a large number of satellites are propagated, or propagation is over a long period.

3.1.1 Methodology

3.1.1.1 Access Time

To find access time in this paper there are two conditions, the range to the satellite and the minimum elevation angle (MEA). These constraints are related to the position of the satellite and ground station by Equation 1 and 2 below.

$$sin(MEA) \leq \frac{z}{r(x,y,z)} \qquad 65$$

$$Range_j \geq \mathrm{r(x,y,z)} \qquad 66$$

In the above equations x, y and z represents the x, y, and z component of the satellite position in topocentric coordinates respectfully, r represents the distance from the ground station to the satellite, MEA represents the minimum elevation angle that the satellite needs to communicate with a ground station, and Range_j

represents the maximum distance the satellite can be away from the j^{th} target area or ground station to communicate with it. In this paper, ground stations and target area act the same; therefore, will be referred to as ground stations from now on. If a satellite and ground station satisfy these two conditions, the satellite is declared in access. At all other times the satellite is declared of out access. To find a satellites position at some time, t, it is required to know the position of the satellite at some previous point, t_0, as well as the force model on the satellite. The position of the satellite at the start of propagation period will be given to each numerical method, with all the different methods using the same force model. The force model being used is J2 perturbation model which is less complex when compared to the common SGP4 and other more sophisticated force models. The J2 perturbation model was chosen for its easy of implementation and to first validate that parallel methods do reduce computation time on simple models before moving to more sophisticated models. More information on force models for satellites and the J2 force models equations can be found in section 2.1.1 J2 Perturbations above.

3.1.1.2 Simulation Environment

In order to test this computational efficiency, satellite orbits will be simulated to find the access times for a number of satellites to a number of ground stations. The simulation will randomize the satellite Keplerian orbital elements as shown in Table 18 as well as, the ground station locations before passing the information to five different methods of propagation shown in Table 17. Three different variables will be changed during the simulation to see how each of the methods react to the changes. The three variables are: The period of propagation, the number of satellites, and the number of ground stations. Each variable will have a range that is being tested and a default value where the two independent variables not being tested will be held at. Table 17 shows all 3 tests that will be preformed are listed with the independent variables used and the number of points taken. The number of points were chosen to balance the time taken of each simulation while still getting the trend result.

Table 17 Simulation Test Parameters

	Period Test	**Satellite Test**	**GS_TA Test**
Period of Propagation	1 to 180 Days	5 Days	5 Days
Number of Satellites	100	10 to 5,000	100
Ground Stations and Target Areas	4	4	1 to 10
Number of Points	50	60	44
Point Repetition	10	10	10

The simulations will start by using MATLAB's "rand" function to randomly make the satellites Keplerian elements and ground station position. The ranges for all the Keplerian elements are specified by the user. The range of values randomized for all the Keplerian elements can be specified by the user, however, in this paper the values used for simulations are given in Table 18.

Table 18 Range of RSO's Keplerian Elements

Name of Element	**Variable**	**Range**
Semi Major Axis	a	7000 km to 42,164 km
Eccentricity	e	.00 to .99
Inclination	i	-90^o to 90^o
Argument of Perigee	ω	0^o to 360^o
Longitude of The Ascending Node	O	0^o to 360^o
True Anomaly	v	0^o to 360^o

The ground stations are randomly placed anywhere on Earth's surface. The Keplerian elements and position of the ground stations are passed to each method which computes the access time. The computational time is defined as the time from when the Keplerian elements and position of the ground stations are passed to the method until the access times are calculated from the method.

3.1.2 Results

The results of the simulation were taken and linearly fit using least squares analysis to see which functions act linearly with time. Below are the results of the simulations which are plotted with a linear line of best fit generated from least squares analysis.

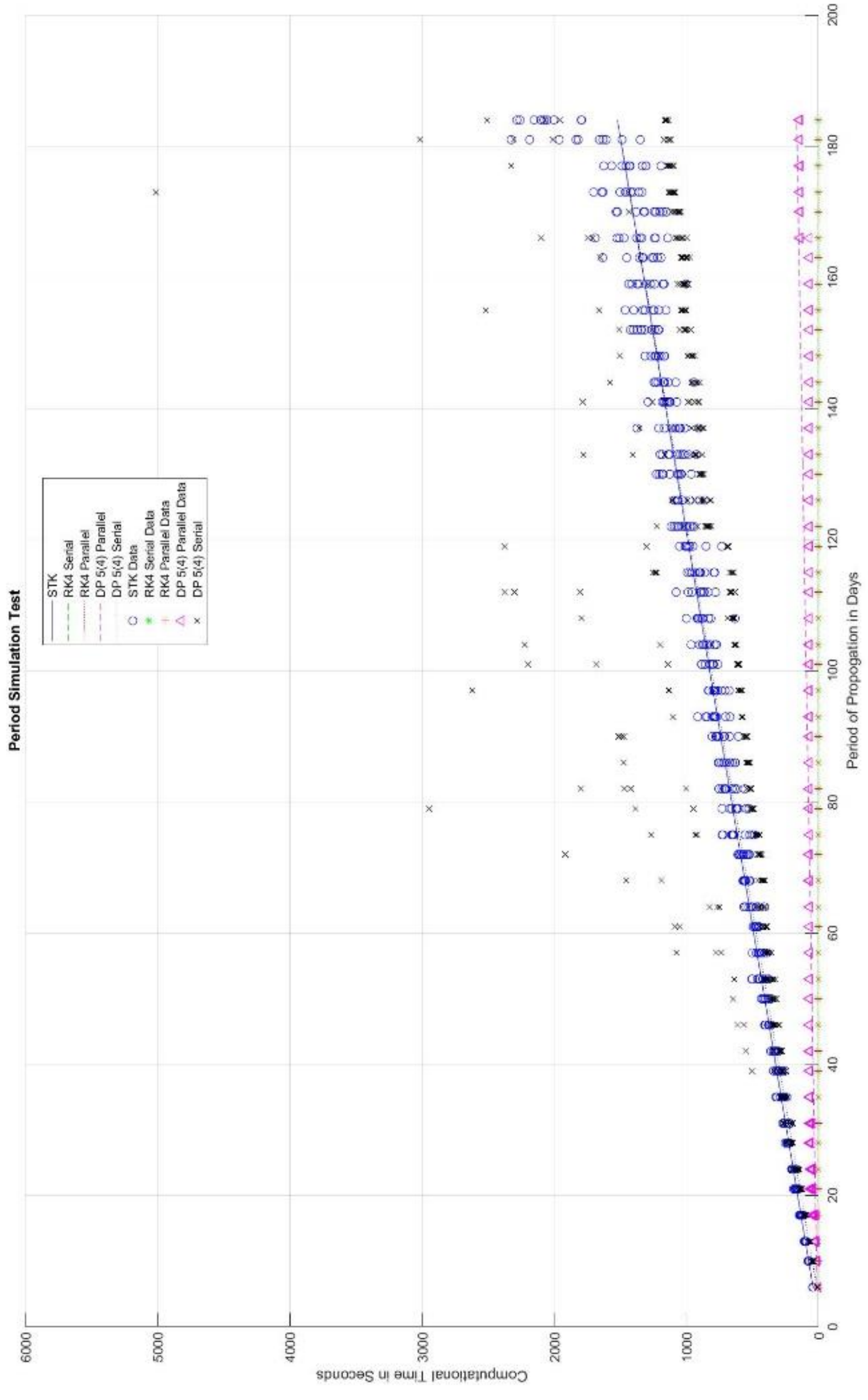


Figure 14 Period of Propagation vs Computation Time

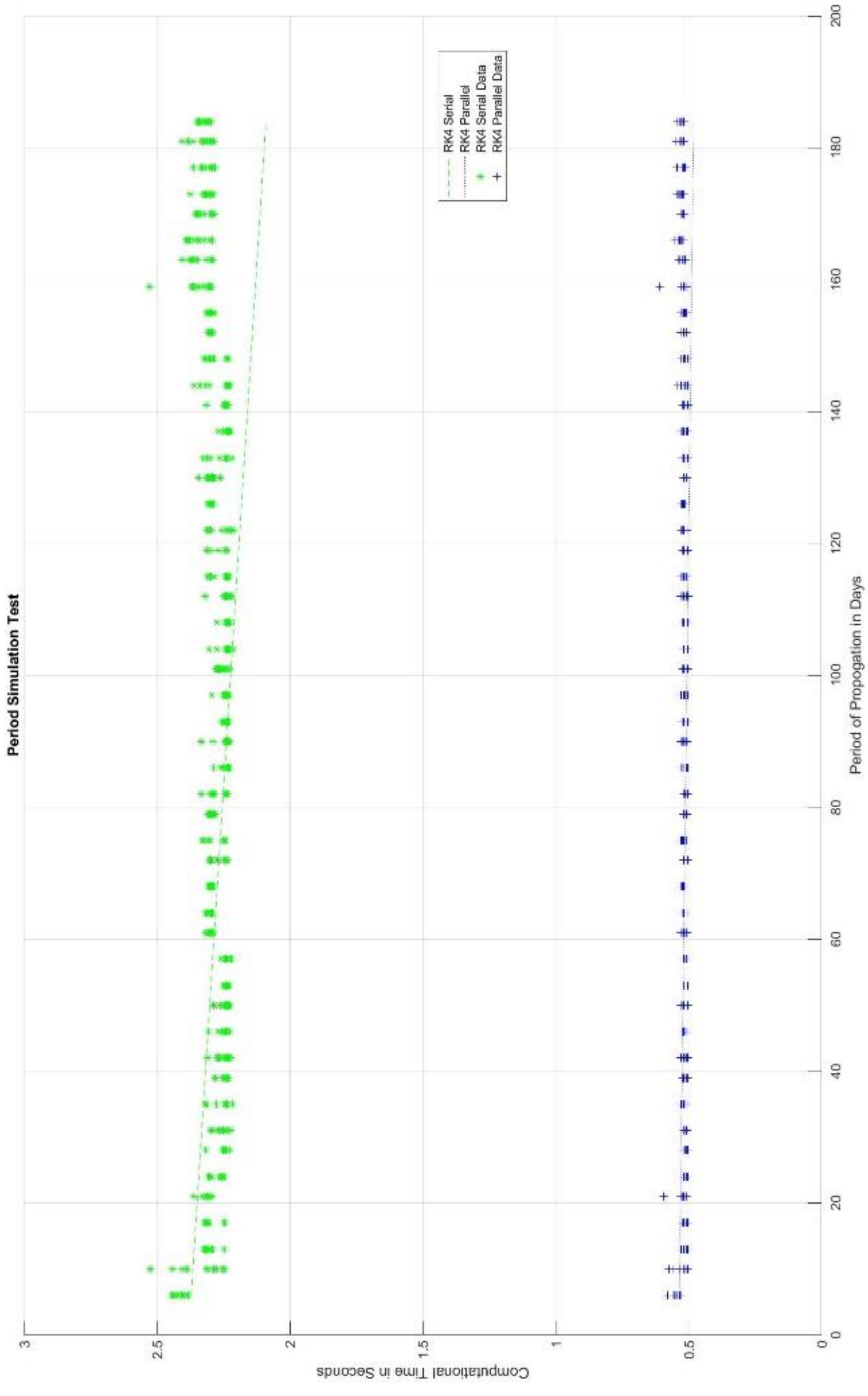


Figure 15 Runge Kutta Period comparison for Parallel vs Serial methodologies

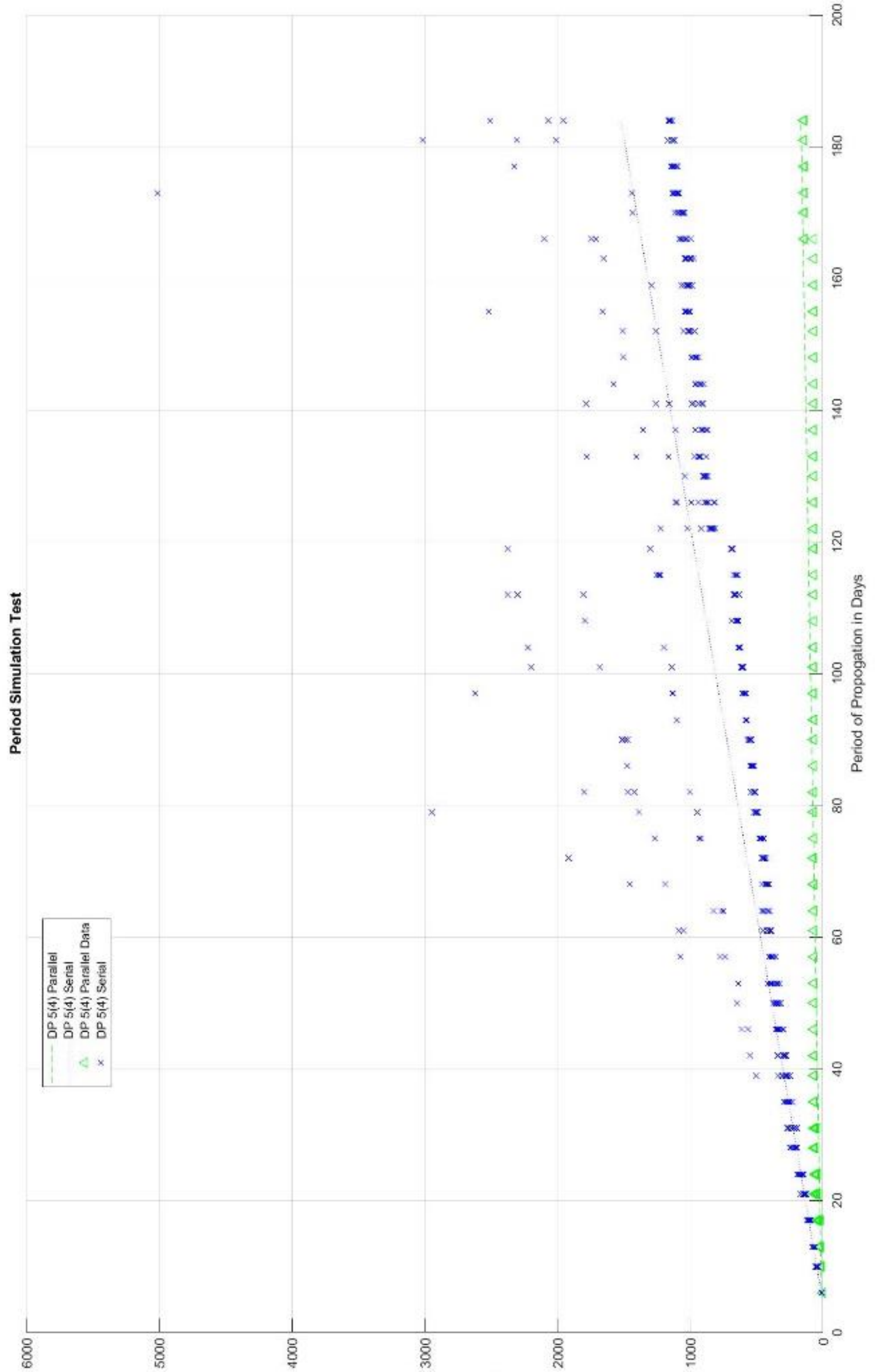


Figure 16 Dorman Prince 5(4) Period comparison for Parallel vs Serial methodologies

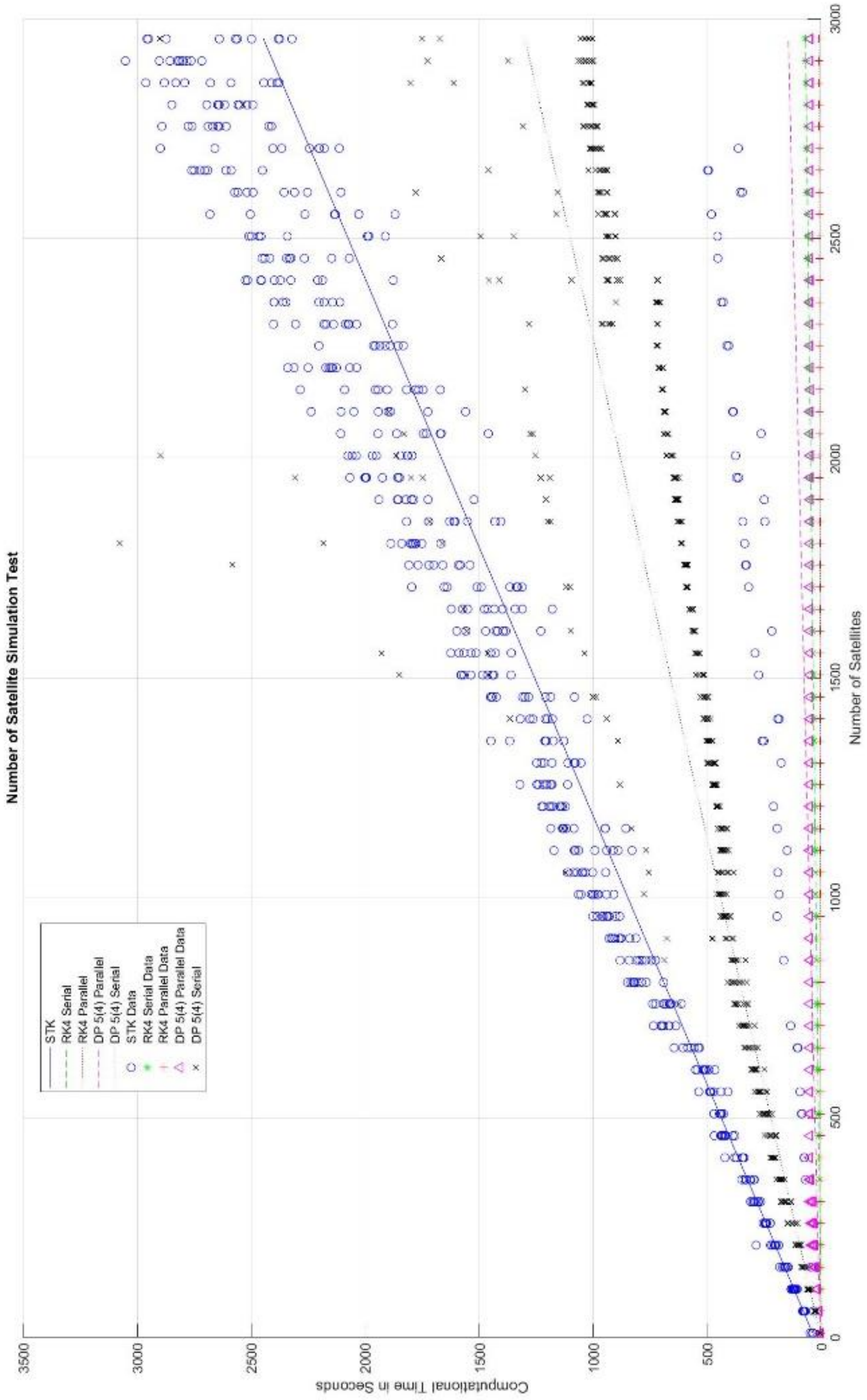


Figure 17 Number of Satellites vs Computation Time

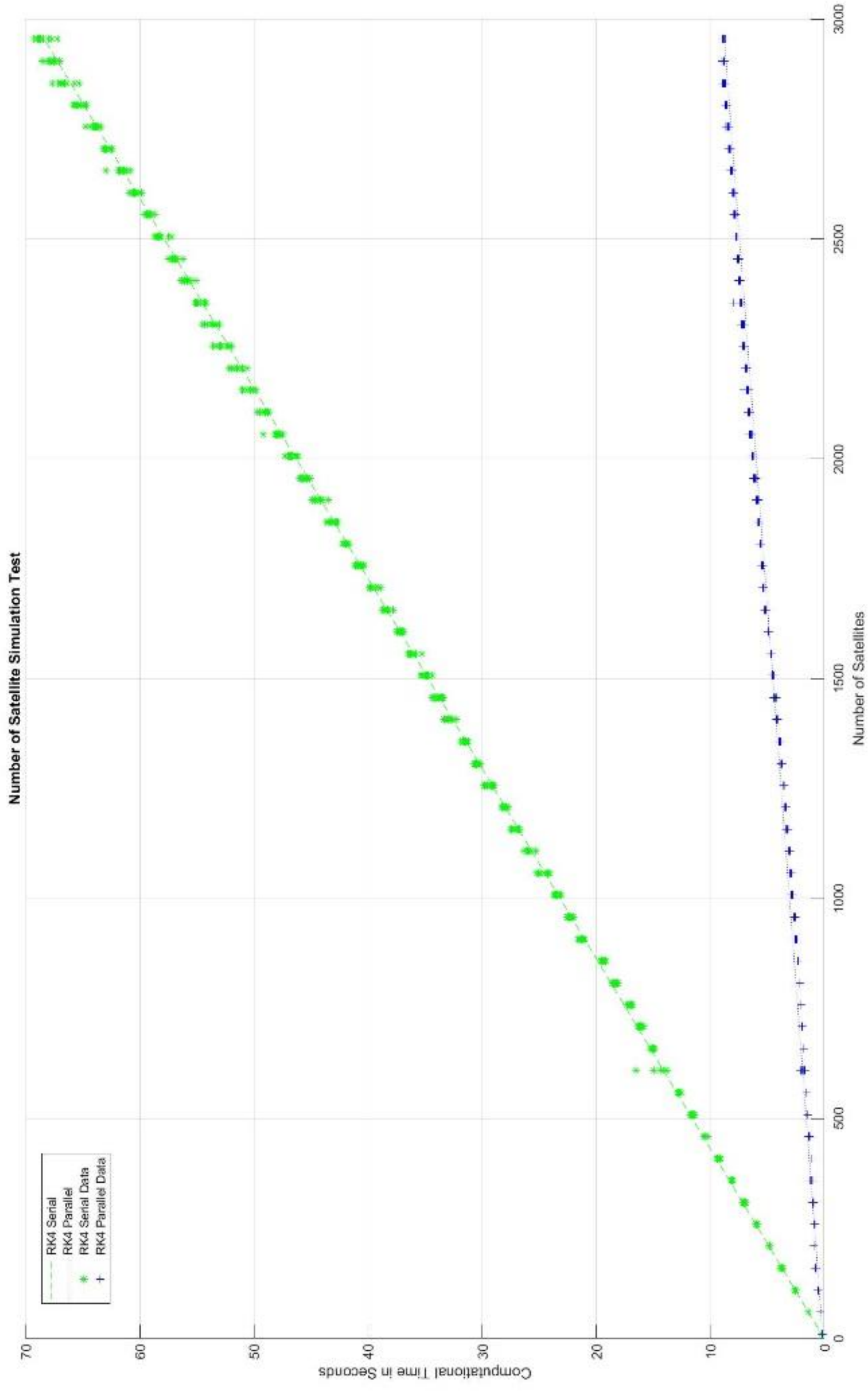


Figure 18 Runge Kutta Satellite comparison for Parallel vs Serial methodologies

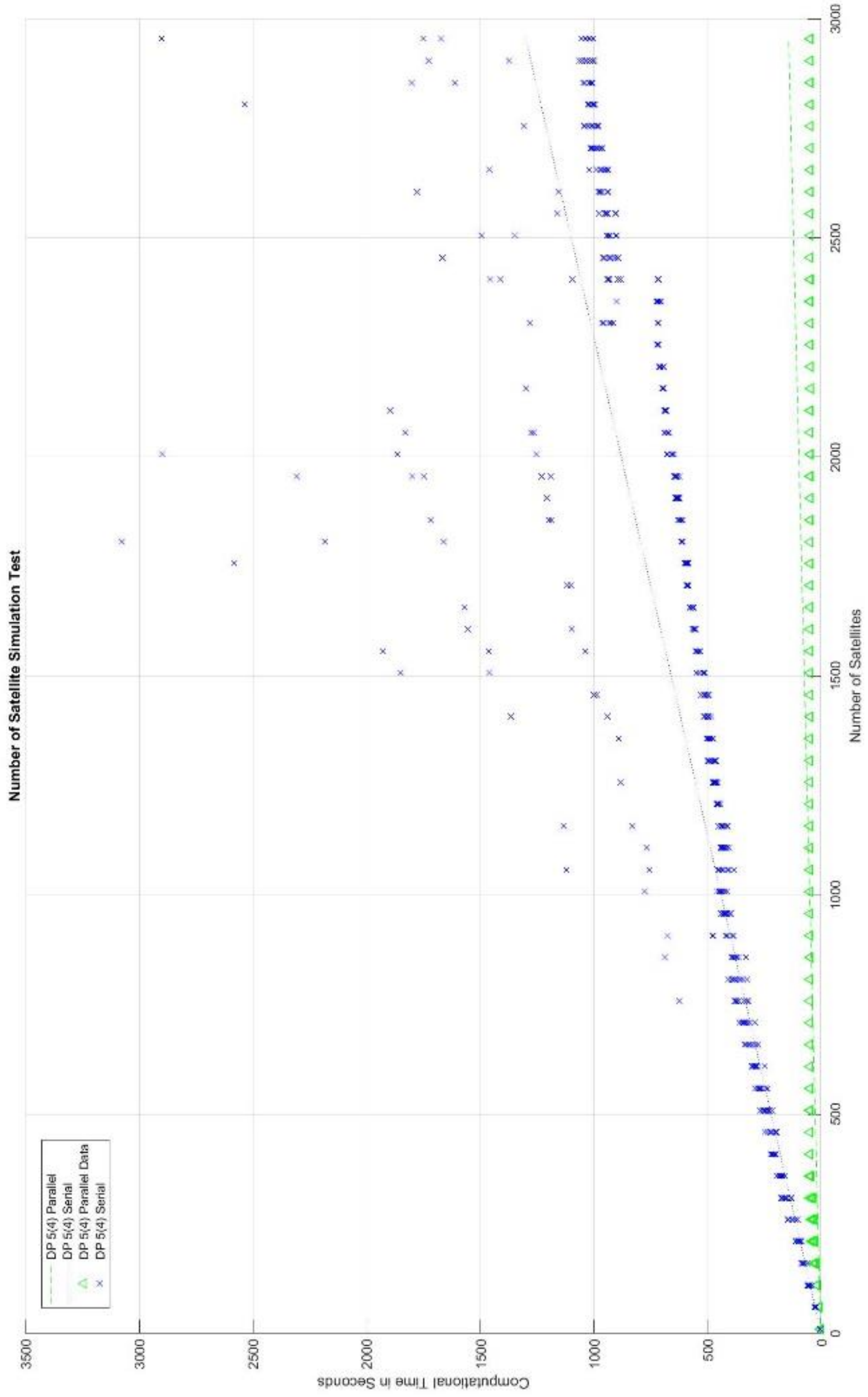


Figure 19 Dorman Prince 5(4) Satellite comparison for Parallel vs Serial methodologies

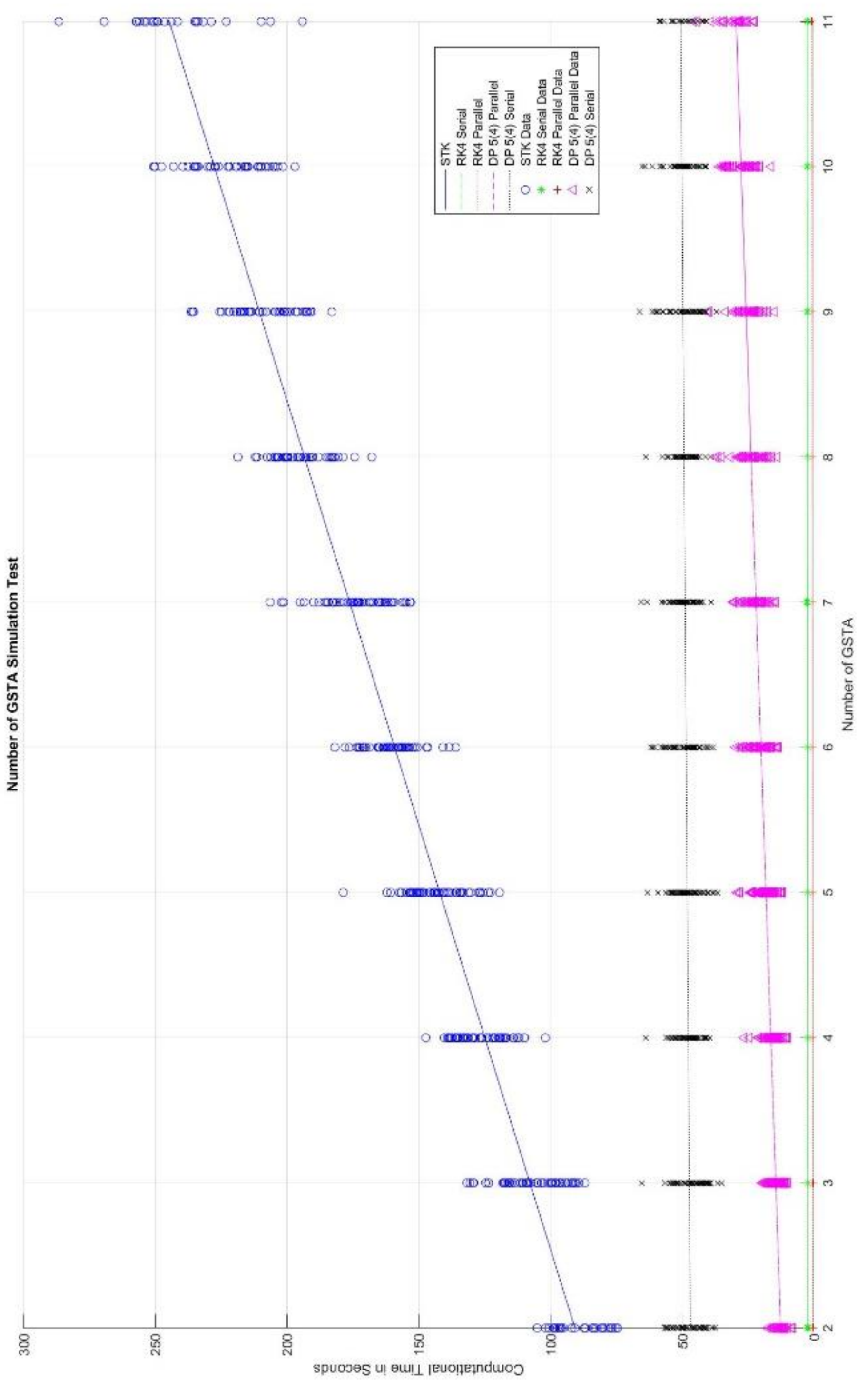


Figure 20 Number of Ground Station vs Computation time

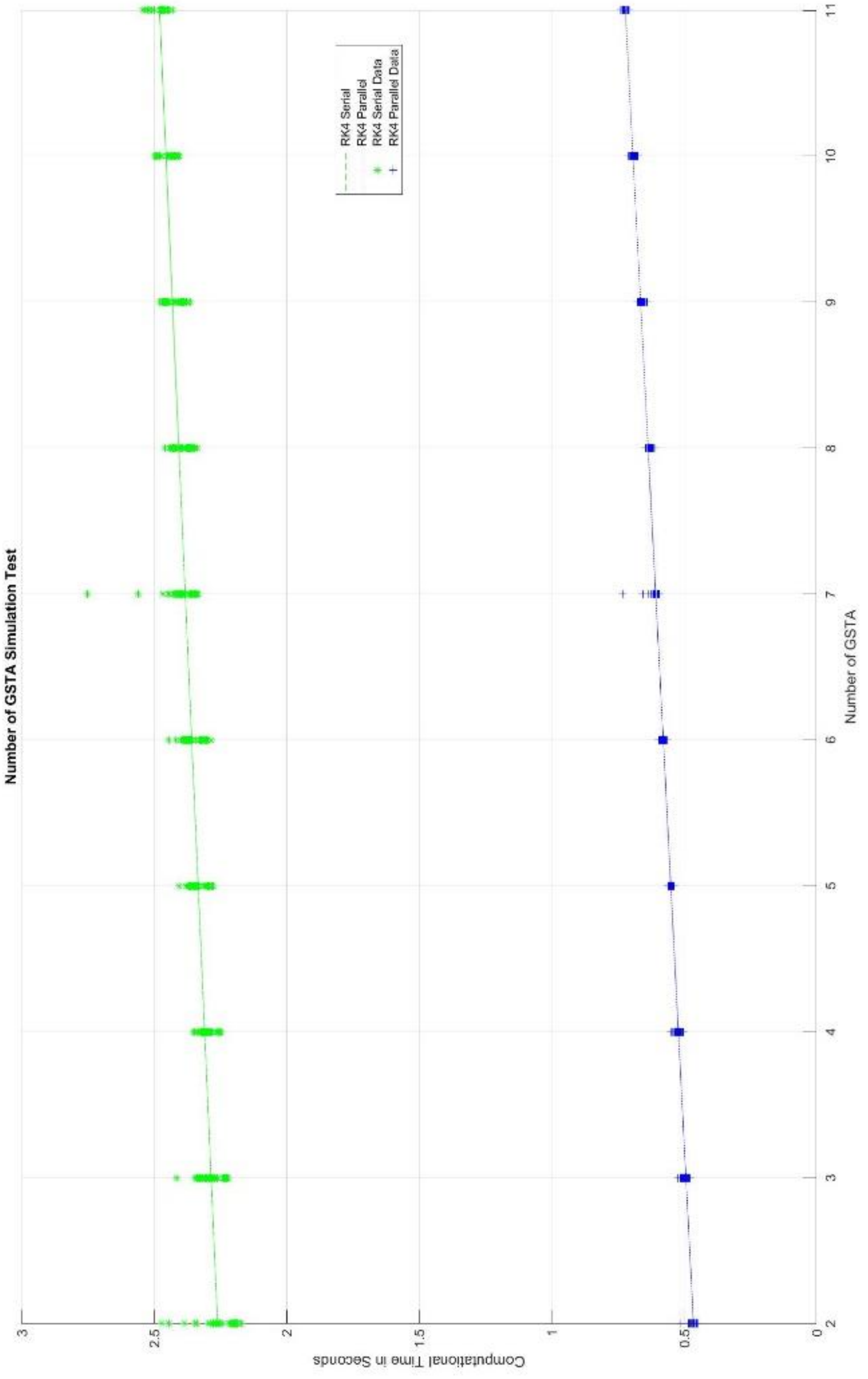


Figure 21 Runge Kutta Ground Station comparison for Parallel vs Serial methodologies

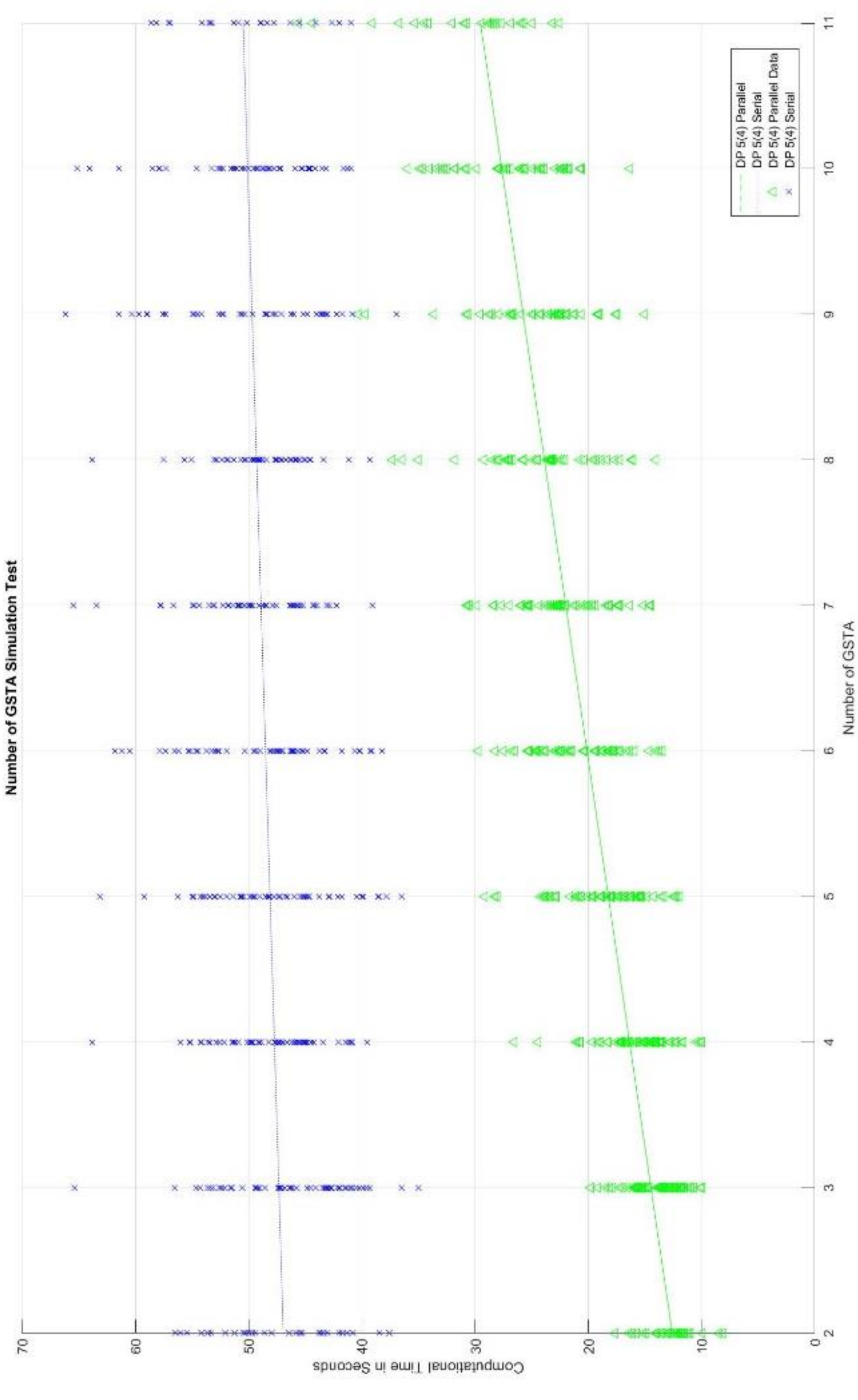


Figure 22 Dorman Prince 5(4) Ground Station comparison for Parallel vs Serial methodologies

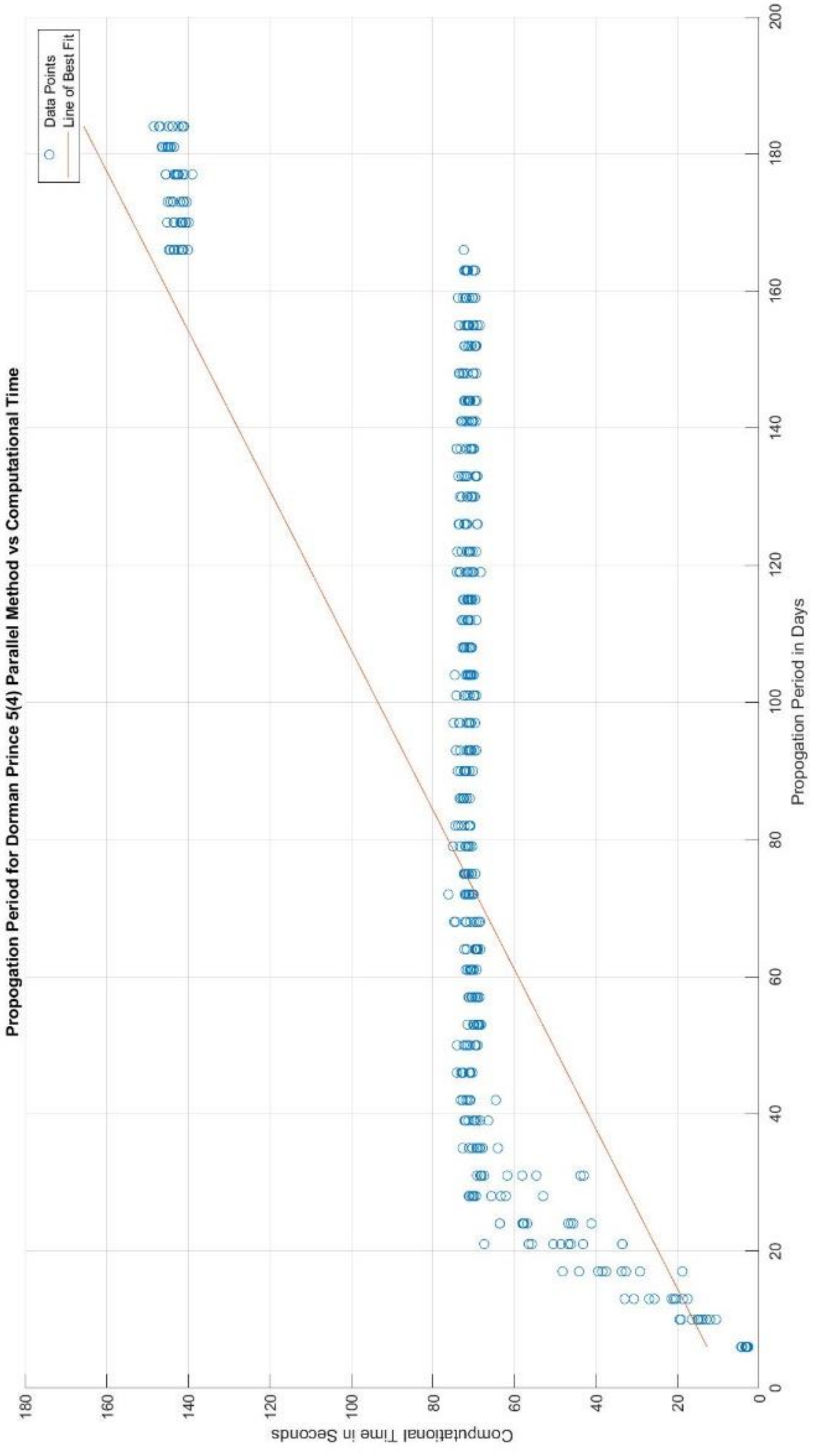


Figure 23 Dorman Prince 5(4) Parallel method results for Period Testing

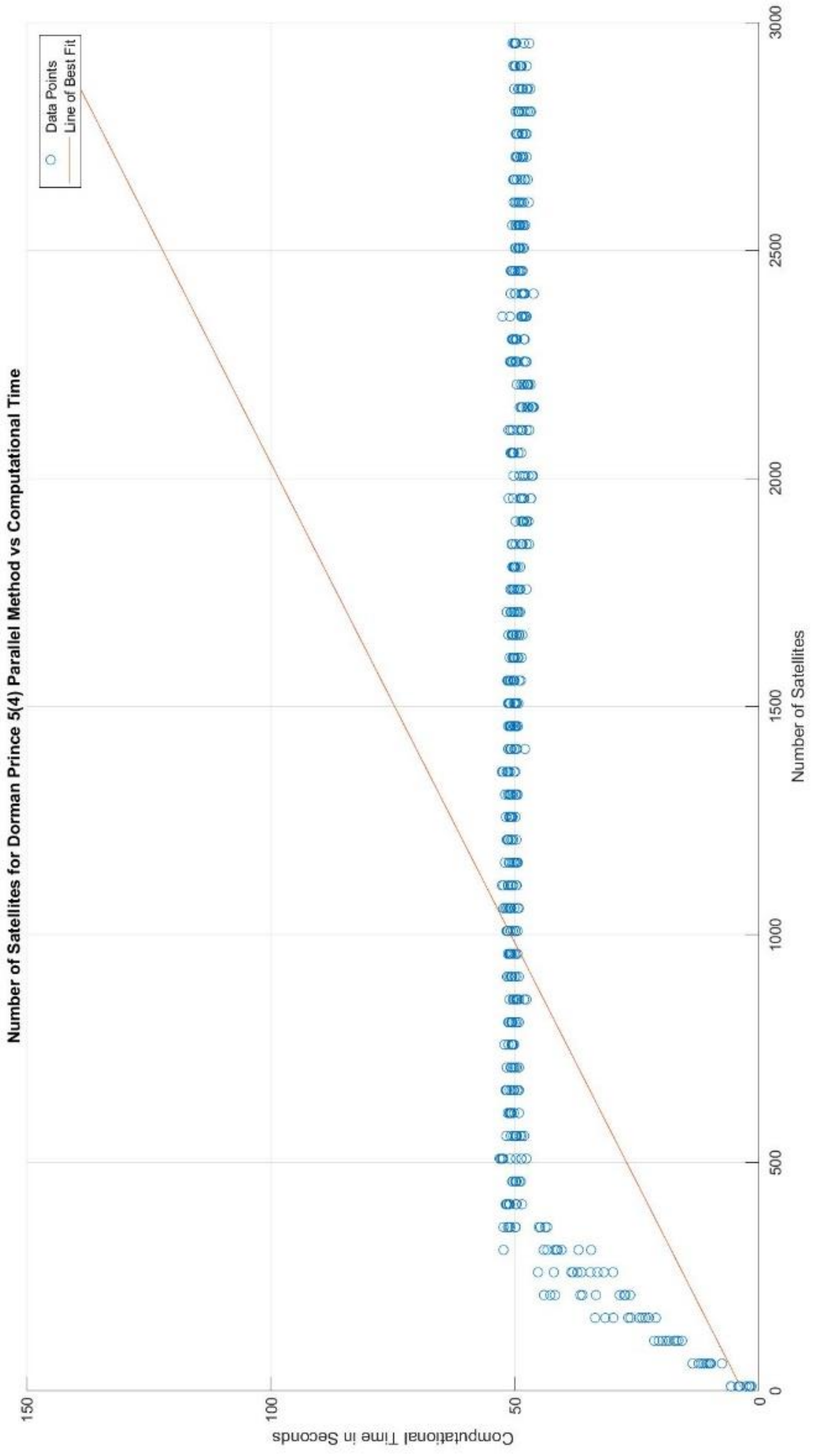


Figure 24 Dorman Prince 5(4) Parallel method results for number of satellite testing

3.1.3 Discussion and Future Work

3.1.3.1 Period Testing

The results from the period test illustrate how the different numerical methods are affected by changing the period of propagation. It was expected that all methods increase in computation time as the period of propagation increases. In Figure 15 there is a noticeable difference between the parallel and series methods of RK, looking at the overall computational time, as expected, the parallel method is significantly less than series method only requiring around 20% of the computation time. For both the parallel and series methodologies the trend is almost constant with a slight downwards slope found from the least squares linear analysis. These downwards slope indicates that a longer characterization period should be used for RK method due to their computation time not varying significantly over the observed propagation periods. A larger range of propagation periods will be looked over for the RK method to see if the trend is linearly upwards as expected, or the slight downwards slope continues which then would require more testing to explain the cause behind downward trend.

Looking at how the DP methodologies deal with change in propagation we notice some similar trends to the RK methods. The first and most significant similarity between the two is that the parallel method takes notably less computational time than the series method as expected. The DP methods trends are different from RK, as the period of propagation increases the more computationally efficient the parallel methodology is. With a small propagation period the serial and parallel methods are around the same computation time. The real change takes place when the propagation period increases. Once the propagation period reaches 60 days the parallel method takes around 20% of the computation time of the serial method and decreases further as propagation period increases. Where with the RK method the difference in computational intensity stayed relatively constant for the entire duration. As seen in Figure 23 where unlike the other methods, the parallel method acts like a step function instead of linearly. Possible causes of this behavior of DP parallel stems may be attributed from some of the orbits being unstable, which, causes all the satellites to continually be propagated. The DP methodology has adaptive step size which normally decreases the time it takes to integrate non-strict problems; however, unstable orbits lead to strict problems with large error. The strict problem causes the step size to get smaller and smaller than reset back to the initial step size value. It is coded to propagate the satellites till they all reach the propagation period, so when one might reach it in N iterations if not all satellites have reached the full propagation period, the satellite will keep iterating. These extra iterations are wasted and could be eliminated to increase the computational efficiency in the future. Looking at the DP series method we see the computational time becoming quite sporadic having most data points near linear with some taking much larger computational time. The randomly selected orbits between the sporadic and linear data will be compared to see if a cause for this is unstable orbits.

Comparing both the RK and DP series methods to STK we can see that they are clearly much more computationally efficient then STK, with DP parallel method being the closest to STK for computational efficiency. From the trends, the DP parallel method will become less efficient than STK over longer periods which will be confirmed once more data points are taken.

3.1.3.2 Satellite Testing

The results from the satellite test show similar results to the period test with the parallel methods preforming better than serial methods. Looking firstly at the RK trends, the parallel method out preforms the serial method, unlike the period tests, where the increase of satellites cause an upward linear trend. This shows that the RK methods computation time is affected more by the increase of satellite compared to an increase in period of propagation. The computation time of the parallel method relative to the series method decreases as the number of satellites increase. When the number of satellites is near 500, which is around the number of satellites in proposed mega constellations the parallel method requires less than 20% of the computational time. With the parallel method getting more computation- ally efficient as the number of satellites increase, it would be useful for large catalogs like NORAD’s space catalog, which has more that 40,000 tracked objects.

Looking at the DP methods, the trends are similar to the period testing, with the serial method having a near linear trend with sporadic outliers, currently the reason for these outliers is unknown and will be looked into further. The parallel method is found to be more efficient than the series method, at 500 satellites taking around 20% of the computational time of the series method. With the satellite range tested the step that was found in period testing was not found with the satellite testing. To confirm that the satellite testing follows the same step a larger range of number of satellites will need to be tested. The DP takes more computation time when compared to RK methods; however, DP methods get more computationally efficient when comparing the parallel to series method as the number of satellites increase. One should note that the DP parallel method requires less computation time than RK series as the satellite number increases, this shows how effective parallel methods are for multi-satellite propagation.

When comparing the DP and RK methods to STK, the RK methods take significantly less computation time, as well as the DP parallel method. However, the DP series method computational time is similar to that of STK. This demonstrates how inefficient serial methods are when dealing with large number of satellites. The US space fence is looking to come online this year allowing the tracking of more space debris to be possible, this is expected to have the Space Catalog jump to over 100,000 satellites and space debris.

3.1.3.3 Ground Station Testing

The ground station testing presents a very different trend than the other two tests. According to the ground station test, it is clear that while the addition of ground stations does increase the computational time; however, it does not affect the computational time as much as increase in propagation period and number of satellites. This was as expected as the period of propagation and the number of satellites effects the propagator which takes the most computational time. The position calculation for ground stations does not require a propagator, as the motion is well-known and can be modeled as a function of time. Looking at Figure 22 the trend of increasing the ground stations has a similar effect on both the parallel and series methods. The slight upwards trend from the two methods is as expected as the additional ground station doesn’t affect the propagation time, which take up the most computation time, but rather effects the post processing after the propagator. The computational time of the ground station test is similar to the period test which shows that with RK methods the computation time depends mostly on the number of satellites being propagated.

The DP methods have a similar trend as seen in the RK methods but being an order of magnitude greater in computational time. The trend confirms what was mentioned above which is with the number of ground stations not effecting the propagator it has a much smaller effect on the computation time compared to how the DP method handles increases in satellites, as well as propagation period. The parallel method is more efficient than the series method as expected, but the parallel method has a steeper upward slope than the series method alluding to series DP methods are more efficient for the addition of ground stations than parallel methods are. To be able to verify this statement more data points with larger amount of ground stations needs to be taken. Looking at how DP methods compare to RK methods we can see similar trends to what were seen in the past two tests. The DP have more computational time than the RK methods and both follow very similar trends, this is expected with both methods having the same post processing which is what the number of ground stations effects. Both DP and RK methods are significantly below STK's results, which as expected take longer and have a steeper upward trend when compared to DP and RK. This is due to STK having information and a graphical component to compute. It also adds the additional computational time of the graphics, rather than just adding an extra calculation step when the number of ground stations increase.

3.1.3.4 Conclusion and Overall Summary

The trends from all 3 tests bring interesting qualitative results that will be described below which look to be expanded upon and made into qualitative results in the future with more data points taken. First looking at the similarity between the tests the period and satellite test are the most similar with the ground station test looking much different. This as described in the ground station section is because the number of ground stations effects the post processing after the propagator while period and satellite number effect the propagator. One of the most important and similar trends between all the tests and numerical methods is that parallel methods out preformed serial methods in most case shown. While there are some cases where serial methods out preform parallel methods with multi satellite propagation parallel methods have a clear advantage in computational time. The goal of this paper was to prove that parallel processing can save 50% of the computational time when compared to series methods, which was proven in to be true under certain parameters. While the computational time and the amount saved depends on the inputs, it was shown that as the number of satellites, and period of propagation increase so does parallel methods efficiency when compared with series methods. The next step towards introducing parallel methods into multi-satellite propagation is to upgrade the propagation model to SGP4 and use TLE data to test how accurate the methods are and compare them to STK. As well as characterize their computational efficiency more accurately with more data points over a longer range, which will be done in a follow up paper. Looking toward the future as the space domain becomes busier more efficient methods will not only be needed by for ground calculations, but with the autonomy of satellites ever-increasing allowing for in- situ calculations of orbits will be extremely valuable for space situational awareness, navigation, and collision avoidance.

3.2 Parallel Processing of TLE’s for Orbital Maneuver Detection

The below publication was originally submitted and published by the Advances in Space Research journal [48]. The background information was move to chapter 2 of this thesis with the resulting methodology and results being contained in the section below.

Space situational awareness (SSA) is an important and growing field in the space sector. With space becoming filled with active satellites and space debris, the Big Sky Theory [104] is becoming less and less applicable. There are currently 20,000 tracked objects from NORAD, called resident space objects (RSOs). RSOs are any active or inactive satellites that orbit Earth. The number of tracked RSOs is estimated to jump an order of magnitude with the space fence coming online [105]. With more objects to account for, more efficient tracking methods are required.

The parallel processing of orbital propagation first occurred in the early 1990s [106] [107]. Since then, the advances in parallel processing have allowed for more efficient calculations of complex algorithms. The parallel processing discussed in this paper is single instruction multiple data (SIMD). SIMD parallel processing takes a single instruction, such as adding two numbers, and applies the instruction to multiple data points, making it the easiest form of parallel processing to implement. Integrating SIMD parallel processing into current methods of TLE analysis and state propagation should decrease the computation speed on large datasets.

In recent years, there have been different implementations of SIMD parallel processing, such as implementing these algorithms on processing units specially designed for parallel processing, with the graphics processing unit (GPU) being a notable example [108]. With the implementation of SIMD parallel processing, computer resources are being used more efficiently, allowing for an increase in computational speed without the downside of loss of accuracy. With recent advances in parallel processing, new algorithms for multi-satellite orbital propagation have arisen, as described by San-Juan et al. [109] and Koblick et al. [110]. Multi-satellite propagation has applications in several areas of the space sector, such as access finding, satellite scheduling, and constellation design.

This study focuses on determining the effectiveness of parallel processing at increasing the computational efficiency of orbit determination through the use of both simulated and truth data. In addition, a novel method of orbital detection, which combines TLE analysis with state propagation, is proposed.

3.2.1 Methodology

To evaluate the TLE analysis, state propagation, and hybrid detection methods, computation speed and detection accuracy are used as comparison metrics. Computation speed is determined for each method not including the TLE read in. The reason that the TLE read-in is not taken into account is because it is completely independent of the two different maneuver detection algorithms and can be optimize separately. Two different figures of merit that are being looked at are the number of true detections and the number of false detections. The number of true detections refers to TLEs that have been flagged as having maneuvers that correspond with the truth maneuver data given by the ENVISAT catalog. The number of false detections refers to the TLEs that have been flagged as having maneuvers that do not correspond with the truth maneuver data. The 3 different methodologies for the maneuver detection techniques, are outline in the background section 2.4 Maneuver Detection.

The parameters were varied over a range of values to find the optimal results from the different methods in the design space. The accuracy of these methods has been shown in Kelecy et al. [61] and Lemmens and Krag [63], hence the focus on the computational speed, with the appropriate tuning of parameters to be performed in the future.

To test the methods with truth data, ENVISAT's TLEs and maneuver history from 2002 to 2012 have been used. Both the maneuver and TLE history are available at: https://ilrs.cddis.eosdis.nasa.gov/data_and_products/predictions/maneuver.html, and contain 11,816 TLEs and 177 maneuvers, with a total of 220 burns. Both sequential and parallel methods are shown, with the sequential method representing the nonparallel methods.

3.2.2 Results

The results for true detections and false detections for each method are shown in Table 19.

Table 19 : True and false detections from each method

Hybrid method	n-σ:	2.6	True detections:	49
	$PTol$:	0.85	False detections:	45
	$VTol$:	0.01	Ratio of false to true detections:	0.9184
Parallel TLE method	n-σ:	3	True detections:	71
			False detections:	66

		Ratio of false to true detections: 0.9296
Parallel state method	$PTol$: 0.85	True detections: 106
	$VTol$: 0.01	False detections: 192
		Ratio of false to true detections: 1.8113
Sequential TLE method	n-σ: 3	True detections: 71
		False detections: 66
		Ratio of false to true detections: 0.9296
Sequential state method	$PTol$: 0.85	True detections: 106
	$VTol$: 0.01	False detections: 192
		Ratio of false to true detections: 1.8113

From Table 19 it can be seen that the performance of the parallel and series implementations of both the TLE and state methods does not affect the accuracy of the results. In addition, the hybrid method outperforms the other methods at distinguishing true detections from false detections. While the hybrid method does not have the largest number of true detections, it does minimize false detections of maneuvers. The results from all the methods show a few of the true maneuvers detected and a large number of false detections. This is assumed to be a result of low-order interpolation and improper tuning of threshold values. While linear interpolation was used to estimate the variance in expected parameters, other studies have used cubic splines or higher-order functions [62], which will improve accuracy. The tuning of the maneuver parameters is method- and RSO-specific, due to the focus on computational speed of the different methods in this paper; in addition, the design space for the tuning of the parameters was small and all parameters are assumed to be suboptimal [61].

The effect of parallel processing can be seen in Figure 25 and Figure 26. Figure 25 and Figure 26 show how the computational time changes with the n-σ threshold value for each method. The reason that the computational time is plotted versus n-σ is that, in the hybrid method the number of maneuvers that are detected by the TLE analysis method is then given to the state propagation method. Larger values of n-σ will result in fewer maneuvers being detected and fewer RSOs moving onto state propagation, which should result in a change in computation time due to the fact that fewer calculations are required.

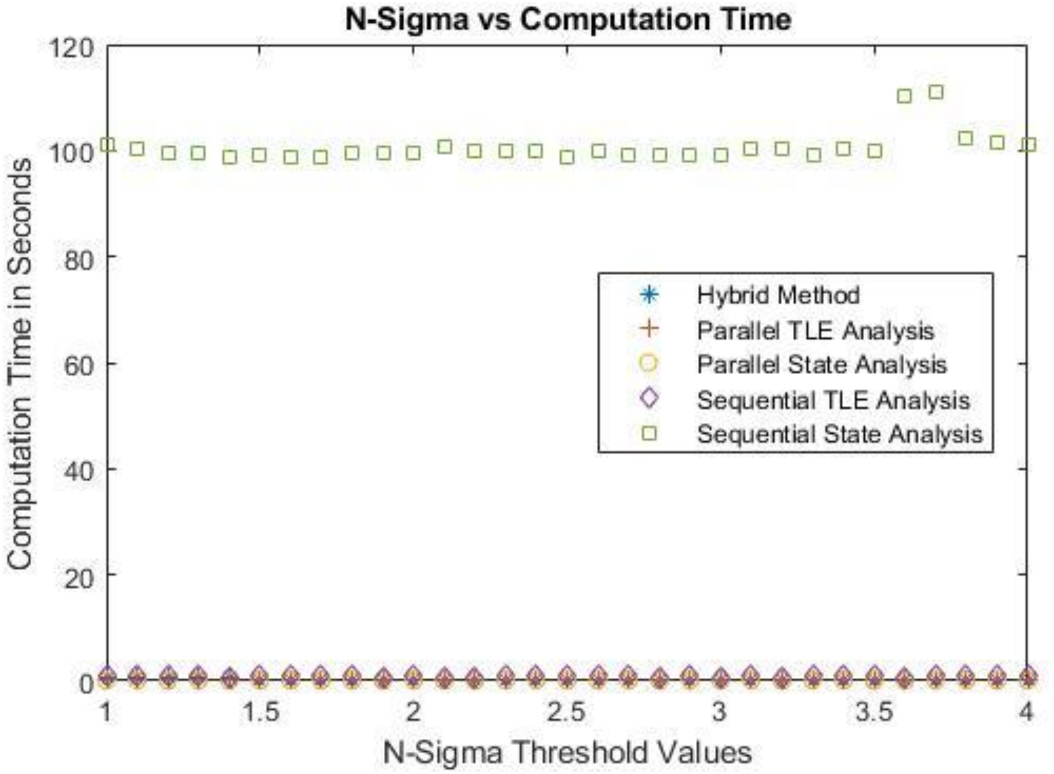


Figure 25 The computation time versus the n-σ threshold values for each method.

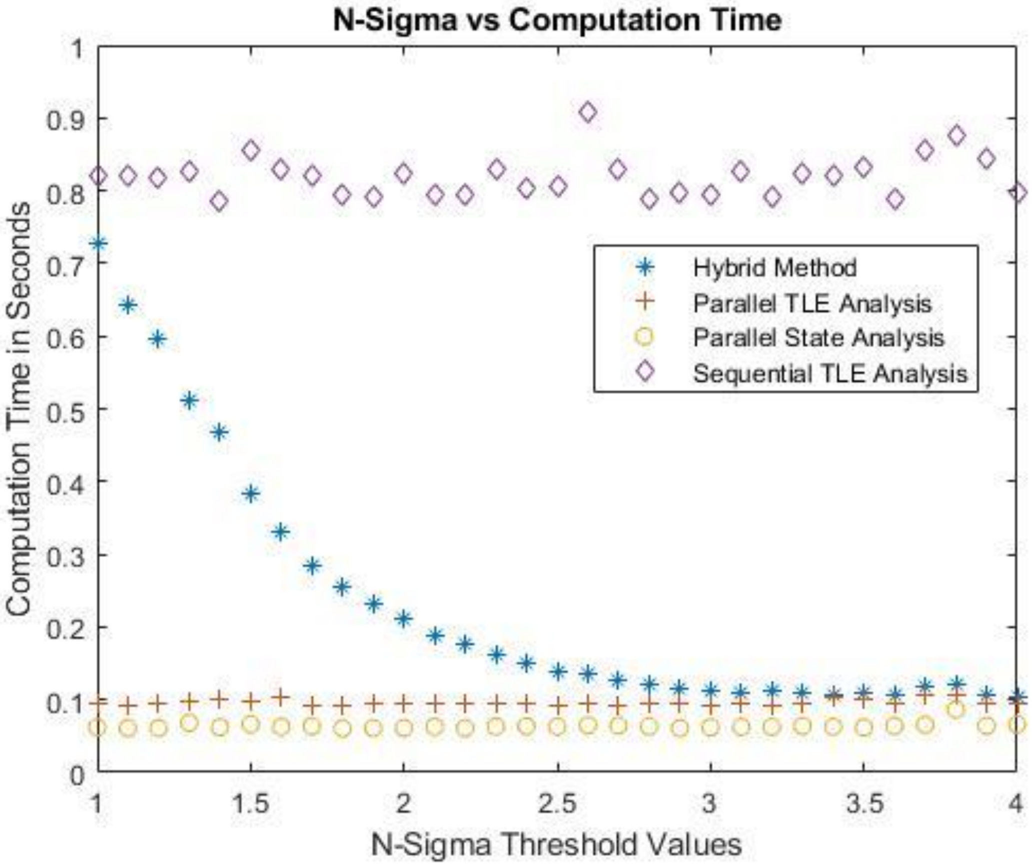


Figure 26 The zoomed-in version of Fig. 2 excluding sequential state analysis.

The results show that parallel processing improves computational speed by a different amount for the different methods. For TLE propagation, parallel processing improves the computational speed by approximately eight times when taking the average over the n-σ range. The difference between the parallel and sequential computational speed of state propagation is approximately four orders of magnitude when taking the average over the n-σ range. This large difference between the parallel and sequential is due to the parallel processing of the orbital propagation, which takes up the majority of the computation time for state propagation methods. Previously, TLE analysis had the advantage of quicker computational speed; however, with the implementation of parallel processing, state propagation has achieved nearly the same computational speed. It was found that the computational time of sequential TLE analysis is on average approximately 105 times less than sequential state propagation. In contrast, computation time of parallel state propagation is on average approximately 1.6 times less than parallel

TLE analysis. This gives an advantage to state propagator methods in most parallel instances because they generally provide more accurate results than TLE analysis.

When comparing the parallel implementation of TLE analysis and state propagation, Figure 25 shows that the hybrid method is worse in terms of computational speed, although it varies based on the n-σ value. As expected, when the n-σ value increases, the number of detections decreases, which decreases the number of RSOs going to state propagation. This, in turn, decreases the computational speed. While the hybrid method was designed to minimize the amount of required propagation in order to save computational time, the computational speed of the parallel processing of propagation shows that it is not required. In the future, hybrid methods that combine TLE analysis and state propagation should not be used to increase computational speed, but rather to reduce the false detection rate, and possibly improve the accuracy of results.

3.2.3 Conclusion

The purpose of this paper was to show that parallel methods reduce the computational time required for RSO detection methods, which was accomplished. In the worst case, parallel processing demonstrated an eight-fold decrease in computational time. State propagation benefited the most from this implementation, which dropped the computation speed by four orders of magnitude, thereby bringing its computation speed to a level similar to that of the TLE analysis method. With 11,816 TLEs propagated from ENVISAT, the numbers are similar to performing orbital maneuver detections for the public TLE catalog, which contains approximately 20,000 tracked objects. This shows the feasibility of large-scale RSO maneuver detection from publicly available TLEs, which matches the results of others [108]; [111].

The results from the hybrid method showed worse performance than all other parallel methods in computational speed and improved performance in the ratio of false to true detections. These results suggest that the hybrid method can be implemented for improving detection accuracy rather than computational speed. Future hybrid method architecture should run both TLE and state propagation in parallel and compare their results to reduce false positives and improve the accuracy on the orbital maneuvers.

Further improvements consist of refining the hybrid method, with a focus on improving accuracy in order to detect and characterize orbital maneuvers. The characterization of maneuvers is done by available information to estimate the direction and ΔV of the maneuver, which can be calculated accurately using TLE analysis and state propagation methods. Using information from both TLE analysis and state propagation leads to fewer false detections and can lead to better characterization of the maneuver. Using different methods of detection, such as changes in the semi-major axis, predicted states, and orbital energy, it could be possible to achieve better estimates of the type of maneuver and reduce the error on the estimated ΔV.

3.3 Summary

The work performed on parallel processing for multi-sat propagation has shown a significant reduction in computation time compared to serial methods. This has major implications in reducing the amount of resources needed to large scale multi-satellite propagation which are needed for analysis such as constellation scheduling and access time, conjunction warnings and analysis, as well as, in maneuver detection. In section 3.2 maneuver detection was used as an example to see the improvement of the implementation of parallel propagation in contrast to serial propagation which showed orders of magnitude reduction of computation time. By integrating parallel propagation into the RSO image simulator it looks to significantly reduce the amount of propagation time as satellite propagation is one of the most computationally intensive steps in RSO image simulation. From work performed in chapter 3 the PSGP4 code was verified and validated allowing for it to be easily integrated into the RSO image simulator. This demonstrates another area in which parallel propagation has a positive contribution to the space community, as well as aids in the accomplishment of research objective two and three.

4 Simulation of RSO Images for Space Situation Awareness (SSA) Using Parallel Processing

The below publication of Simulation of RSO Images for Space Situation Awareness (SSA) Using Parallel Processing is submitted and currently in review for publication in the Sensors journal. The background information was move to chapter 2 of this thesis with the introduction, methodology, and results being contained in the section below.

4.1 Introduction

Space Situational Awareness (SSA) is becoming an increasingly important issue around the world as the number of resident space objects (RSOs) in space is continually increasing. SSA requires innovative, robust and reliable solutions to identify, track and characterize RSOs. Previously, we demonstrated the feasibility of using low-resolution on-orbit images (such as star tracker images) for RSO detection. In [49], we examined the number and frequency of RSOs that are detectable (given the physical and optical parameters of the objects and imager such as distance, brightness, motion) when a commercial grade star tracker is used instead of dedicated high-resolution imagers. Using the simulation study, we concluded that hundreds of objects in low earth orbit can be observed using a star tracker in a day. The starfield simulator named ORBITALS developed for this feasibility study was modeled after FAI imager onboard CASSIOPE mission. Similar simulators: such as the ones outlined in [112] and [113], use a propagation and ray tracing algorithm where ORBITALS performs all calculations in MATLAB with no additional software required. The simulation represents low-resolution star-tracker-like images in IR range. More information on the preliminary star field image simulator is also provided in [49].

ORBITALS also serves as the training tool to enable an artificial intelligence (AI) algorithm design to identify and characterize the RSOs from low-resolution images. AI algorithms have been shown in recent years to perform accurately and efficiently after training, such as [37] for RSO detection and [50] for RSO characterization. Research such as [114] and [115] highlight the need for high quality and quantity of training data pertaining to star field images and corresponding labels. A labelled star field database, as produced by SBOIS, enables novel model development for resolving imagery for classification purposes. As detailed in [116] and [50], machine learning algorithms adapted to classification and characterization are able to leverage large dataset variations which is only achievable via tools like SBOIS. Therefore, from the perspective of image processing algorithms, the key value metrics of an image simulator like SBOIS are labelled image generation accuracy, variability in simulation and image generation time.

While the tool provides powerful means to generate and simulate large number of realistic star field images, the simulator's performance was limited in its computation efficiency, as well as the versatility in providing images from multiple sources. In this study, we present the next generation starfield simulator design that features object centroid estimation; implementation of parallel processing for optimal computation; and robustness in simulation parameters. To validate the accuracy of the simulator, we compare the resulting simulation of the starfield images to the actual images obtained from FAI as well as the simulated images from the Systems Tool Kit (formerly Satellite Tool Kit, STK) a commercially available application.

Accuracy and efficiency of the simulation in starfield images is a critical step in developing SSA algorithms. In [23], the importance of 'high-fidelity and innovative simulation architecture' is described to engage SSA mission design. PROXOR™/RT-PROXOR™ by Bell Aerospace is one of the few commercially available starfield simulators using multi-thread architecture. STK offers an alternative tool to visualize and analyze stars and RSOs in orbit. For the purpose of this study, we compared the proposed simulator design to STK with electro-optical and infrared sensors (EOIR) toolkit. Originally developed by the Space Dynamics Laboratory for missile defense, STK-EOIR supports radiometric sensor modeling of optical sensors. Combined with STK's various mission design capabilities, STK-EOIR is a unique platform where optical RSO tracking mission can be planned, simulated and studied with graphical user interface for convenience; more details of STK-EOIR features are also described in [117] [118] presents a study on space object identification using deep neural networks using STK-EOIR where the photometric observations are generated from STK-EOIR with the GEODSS sensor. While STK-EOIR proves to be a versa-tile platform to simulate starfield images for mission planning and proof-of-concept demonstration, it still lacks flexibility and efficiency we seek for the current study. We recognize that a commercial product like STK (by Analytical Graphics, Inc.) offers a unique capability to perform complex analyses as well as training and research opportunities. AGI reports that STK has more than 50,000 installations at more than 800 global organizations. While limited STK functionalities are available for free, advanced functions still require expensive licensing. Furthermore, for the purpose of the current study, flexibility in low-level implementation is required to accommodate the large set of data and multiple sensor platform. As such, a custom starfield simulator specifically designed for RSO identification study was designed and based on the simulator we had developed for the feasibility study. In developing ORBITALS, three key parameters were considered to enable RSO tracking algorithms: (1) RSO centroid estimation; (2) implementation of parallel processing; and (3) versatility in simulation parameters.

4.2 Simulator Performance Metrics

4.2.1 RSO Pixel Position Comparison

To validate the accuracy of the ORBITALS to a readily available commercial product such as STK-EOIR, a comparison study using the identical orbital parameters in both simulators was conducted. Images taken by FAI on June 15th, 2019, 23:35:50 to 23:36:15 were simulated. This time period was chosen to capture the first sighting of the RADARSAT Constellation Mission (RCM) spacecrafts after they were launched on June 12, 2019. Their distinct formation of three spacecraft, combined with other reference stars within FAI’s field of view help not only in determining the pixel locations, but also in identifying and correcting for potential lens distortions since it forms a geometrical shape rather than a single data point. Three stars from the Bright Star V5 catalogue #6111, #6158, and #6195 were used for as references to prevent errors caused by discrepancies in the simulation of the attitude of the observer spacecraft CASSIOPE. They were chosen for their unique in-line formation that are almost parallel to the RCM formation during the observation period used in this study. Figure 27 illustrates a set of images from FAI (left), ORBITALS (center) and STK-EOIR (right). Highlighted in the red ellipses are the RCM constellation imaged on June 15th. Yellow ellipses depict the locations of three reference stars (Bright Star V5 #6111, #6158, #6195). The RCM, three-satellite constellation providing daily revisits of Canadian land and ocean, was launched three days prior to the study period; thus, not fully deployed and still retaining their distinctive formation. The formation remains visible within FAI's field of view for 32 seconds, with a total of 31 images taken. Table 20 illustrates sample data with three satellite locations with corresponding errors in pixels defined as the difference between FAI images and simulated images from ORBITALS and STK-EOIR, respectively. More examples of the pixel locations and errors are listed in the Appendixes.

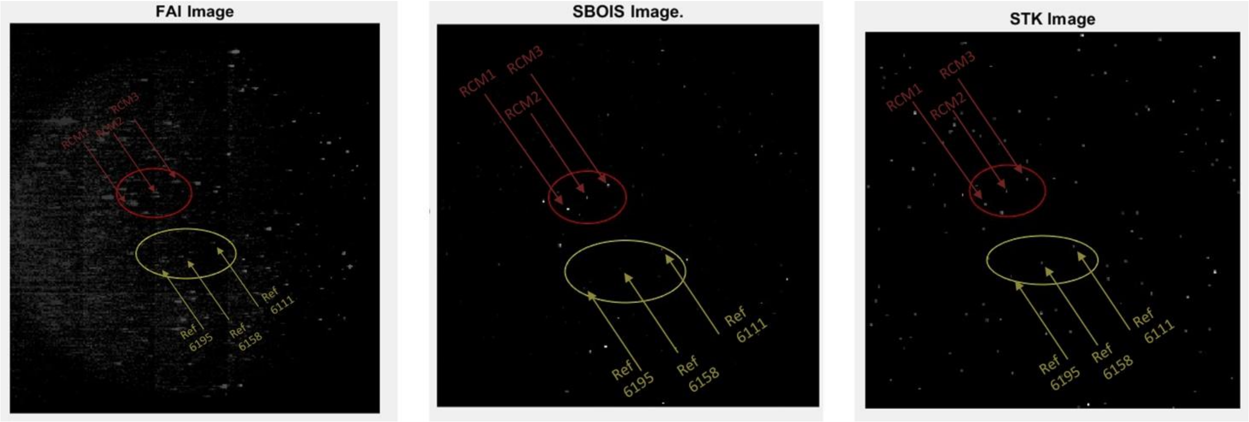


Figure 27 Sample Images from: ***(a)*** *FAI on-orbit observation (left);* ***(b)*** *Simulated using ORBITALS (center);* ***(c)*** *Simulated using STK-EOIR (right)*

The comparison results are very promising. Pixel accuracy for RCM satellites is mostly within 5 pixels for ORBITALS. This error can be attributed to slight position and attitude discrepancies between the real host satellite and the host within the simulator. Observing satellite (CASSIOPE satellite that hosts FAI) ephemeris, for example, plays a critical role in presenting RSO orbit in simulated images. The error of RSO position, as seen from a CCD, is minimal compared to the effect of host satellite position.

While shape and size of the objects effect the brightness of the object, the position of the RSO in the starfield is mostly defined by the accuracy of the RSO position, observer satellite position, and attitude of the observer satellite. RSO attitude and facets (surfaces or shape of the RSO) contribute to the light curve in the form of brightness variation; however, they do not affect the position of the RSO on the simulated images. For the purpose of this study, it was assumed that all space objects are 10-m diameter with a perfect sphere for visibility on the simulated image. The spheres are estimated to have 0.2-reflectance across FAI's effective spectral band. Statistics, such as lights reflectance of materials used, physical size of the satellite, solar panel placement and orientation could also affect the centroid of the reflected light perceived by another sensor, but only on a subpixel level. The comparison study between ORBITALS and STK-EOIR did not consider the above listed physical parameters. Instead, all objects were assumed to be of the same size for comparison and to eliminate potential errors which may occur from the continuous attitude changes. STK's SGP4 propagator was used for the RSO positioning to keep the propagation method the same between ORBITALS and STK.

Table 20: Summary of Simulator Accuracy

Objects	**ORBITALS**		**STK-EOIR**		**ORBITALS Error**(X,Y)*	**STK Error** (X,Y)*
	Row(X)*	Col(Y)*	Row(X)*	Col(Y)*		
			2019-06-15 23:35:50			
Ref 6111	143.0	151.2	143.6	151.0	(1.1, -8.6)	(1.7, -8.8)
Ref 6158	121.6	161.3	120.6	160.7	(-4.4, -5)	(-5.4, -5.6)
Ref 6195	101.1	170.2	100	170.0	(-1.5, -4.8)	(-2.6, -5)
RCM-1	22.6	155.0	13.7	152.6	(3.8, 5.1)	(-5.1, 2.7)
RCM-2	35.5	147.8	30.0	144.6	(1.3, 4.8)	(-4.2, 1.6)
RCM-3	49.9	139.6	45.4	136.6	(-0.3, 4)	(-4.8, 1)

* Values are given in pixels.

The pixel locations of RCM satellites and reference stars were compared by extracting their pixel location from both simulators; then, by calculating differences in the X and Y-axes. The data from ORBITALS are extracted directly from the simulator, as there is a built-in function to output pixel location. STK, on the other hand, has no such functions to export a file with pixel locations of objects within the field of view of a sensor. It does have a details window which can be viewed via EOIR synthetic scene window, listing the information within each pixel as the user selects it. The information includes X-and Y-coordinates, objects within pixel, and object distance from sensor. The lack of batch-export ability in STK significantly hinder any future analysis if the user does not know the general location of the target at the time period, and likely have to cross-reference with the 3D scenario window in STK to determine the pixels the objects are in before analyzing them. The reference stars are compared in a similar manner. Since both simulators use the same star catalogue (Bright Star V5), significant differences between the two simulators outputs were not expected nor observed. It is worth noting that on rare occasions at 23:36:15, for example the scale of the star formation seems to be altered slightly (pixel differences between first and last star is different on both simulators), with STK showing a formation that is slightly larger than that of ORBITALS. Also, noteworthy that in the last 2 images, as the satellites move towards the edge of the image, lens distortion effect of FAI sensor caused image to warp, decreasing the accuracy of both simulators compared to the real image, as seen in the comparison shown in Appendixes. As stated in the RCM analysis, ORBITALS images depict stars with 6.6 and RSO's with 5.1-pixel accuracy, comparable to the images generated from STK (mean difference of 6.5 compared to original FAI image. Note that we reported accuracy of 9.4 pixels using the original simulator without sub-pixel feature [119]. As the simulator was used only to examine the feasibility of RSO identification using low-resolution, the reported pixel accuracy was considered acceptable. With the improvement made since then, we now represent on-orbit images with better than 5.1-pixel accuracy, making simulator pertinent for various applications. In the current study, we focus on the validity of the simulator in creating large dataset for AI algorithm training where accuracy and variance of the simulated data is a key characteristic. Regardless of the approach we take in developing AI for this purpose, simulated data needs to represent the real dataset with as much accuracy and flexibility as we can afford. It is also noted that in comparison with ORBITALS, STK-EOIR does not have a built-in sub-pixel output option; if one needs to identify the exact location of an object, it can only be done through a third-party algorithm that calculates the centroid of a multi-pixel shape. This may lead to some albeit small, but frequent inaccuracies in future studies.

4.2.2 Computational Comparison

Currently, there are over 20,000 in-orbit objects such as satellites, debris, rockets, revolving around Earth in various orbits, and this number is expected to rise by about an order in magnitude with the newly constructed US space fence coming operational [105]. To account for the large scale of propagation required a Parallel SGP4 method (PSGP4) as outline in [48] was implemented. In addition, the Bright Star catalog adds around 10,000 objects in the form of stars into the scenario. Combining all the objects in any simulation is a challenging task, as well as the synthetic scene generation required to simulate the field of view and images of a sensor installed on a spacecraft. To successfully complete given tasks, a simulator needs to be capable of simulating the propagation of a great amount of RSOs, while taking the Bright Star Catalog, or any chosen catalog, into account when generating a simulated scene for a sensor; this should be done in a way that does not require significant computation time. This requires the simulator to be optimized, robust, flexible, and customizable. We compare ORBITALS and STK-EOIR in the above listed aspect, with emphasis on computation speed since ORBITALS utilizes such methods in its computation. It is worth noting that STK has the advantage of having a graphical user-interface (GUI) and a visual 3D scenario, but these can take valuable computational resource away from the actual simulation; this can be a burden if the user does not require the visualization nor the GUI. The comparison study was designed with both simulators generating 200 and 1000 images, then comparing their processing time. The results from ORBITALS were collected via an output function built into the simulator akin to other auxiliary data, which gives the time to generate the specified time period. The initialization time for ORBITALS is between 0.13 to 0.15 seconds insignificant for this comparison. Results from STK were taken by subtracting the image creation time of the 1st image from the last image generated by checking the option “auto generate scene bitmap”, which gives the in-between time of the two images. It is worth noting that the size of the EOIR window affects STK calculation speed but closing it will also stop the generation of images. Therefore, for the following comparisons, all windows were adjusted to the smallest dimension possible to achieve the best results. Table 21 below compares the computational speed of STK12.0.1 and its compatible EOIR to ORBITALS in terms of total time to generate specified number of images for each comparison category. From the comparison, it is clear that ORBITALS is superior to STK in processing time required under the circumstances that are tested. Apart from total time needed, average time taken for each image was also compared. Time needed for ORBITALS increases alongside the number of images within a set. the average time per image for a 200-image sequence was 0.34 seconds and decreased to 0.28 seconds when the number of images was increased to 1000. STK takes significantly longer than ORBITALS to produce the images in both cases with the images produced not being automatically labeled. This implies that ORBITALS is more computationally efficient for generating large, labeled datasets. As mentioned earlier, the 3D visualization capability and the GUI of STK helps the user greatly in many scenarios but can slow down the program down significantly if large number of objects are active at the same time. To test the limitations of STK, the entire GPS constellation and some independent satellites were imported into a scenario. STK crashed several times during the import process after there are over 60 objects in the scenario. Moreover, the import process can be time consuming for the user if TLE or other satellite files are not readily available. On average, it takes 1.5 to 2 seconds to import an object from the AGI database via GUI. Importing all objects within ORBITALS into an STK scenario can be extremely time consuming. Although object import time can be reduced significantly when considering the best-case scenario when the user has satellite orbit files ready, the possibility of STK crashing during the import is still present.

Table 21: Computation Time Comparison

	Processing time with 200 image sequence	**Average time per image**	**Processing time with 1000-image sequence**	**Average time per image**
ORBITALS	67 Sec	0.34 Sec	280 Sec	0.28 Sec
STK	681 Sec	3.4 Sec	3322 Sec	3.3 Sec

4.3 RSO Detection Algorithms using Simulated Images

Detection algorithms are out of the scope of this paper, with the goal just to provide simulated data for the training and verification of these algorithms. Here some of the different types of RSO detection algorithms are mentioned and how simulated images are used with them. Firstly, RSO detection algorithms are used to determine if an RSO is in the image, as well as distinguish between stars and RSO's. The main way of determining the difference between stars and RSO's is from the motion that they produce in the image, with stars streaking due to motion of the camera and host platform. The streaking of the RSO's is due to the motion of the camera, host platform, and RSO's motion. When accounting for the sensor and platform motion the only objects that should be moving are the RSO, allowing for determination between the stars and RSO's. For more information on different detection and centroiding methods, such as machine learning and analytical algorithms see [70], [119], [120]. For the training of machine learning detection algorithms hundreds to thousands of images and image sequences are required, which are not always available for space-based platforms. To make up for the lack of data simulated images can be used to train these algorithms. One example of detection algorithms being trained of ORBITALS simulated images is presented in Mr. Dave's AMOS technical presentation [50].

4.4 Conclusion

In this paper, we described the architecture of the custom design star field simulator, ORBITALS and compared its performance with the real images as well as the simulated images generated using a commercial product, STK-EOIR. In comparing FAI images taken on June 15th, 2019, 23:35:50 to 23:36:15 where a distinct RADARSAT Constellation Mission (RCM) formation was observed, an average of approximately 5-pixel accuracy was observed in both simulators, demonstrating very similar performance in generating realistic starfield images. ORBITALS, however, features an annotation function that each image generated has labeled data for the simulated objects (both RSOs and stars) with position, velocity, pixel centroid, and SNR. In comparing the processing time, ORBITALS out-performs for 200 and 1000-image cases with less than 1/3 of the processing time to generate the images. As part of future algorithm development, we are extracting light curve information of RSOs to estimate target range and improve the accuracy of the temporal classifier with further characterization parameters. This will be similar to the light curve inversion methods outlined in [42], [60], [23] to estimate an unresolved RSO's shape, attitude and optical properties. An additional area of research is point spread function modelling of sensors using convolution filters from the proposed classification algorithm. The algorithm proposed for PSF modeling would improve the accuracy of the simulator in recreating images and reduce centroiding and classification for future sensor simulations. Performing a similar analysis on varying range of image datasets from ground and space, from varying sensor types and resolutions could also widen the application of the simulator.

4.5 Summary

In this chapter it was shown the SBOIS was able to generate simulated images similar to COTS products such as STK EOIR. This is a huge step in enabling future research to be performed for space based RSO detection, correlation, and characterisation. SBOIS allows for labeled image data sets to be generate that allow for machine learning algorithm to be trained to allow for detection, correlation, and characterisation. This can be seen with recent research that has been enabled from the SBOIS simulator including the training of machine learning algorithms including automatic detection and centroiding of RSO in images performed in Dave et al. [50]. This allows for the training of more advanced detection algorithms which was the main contribution to RSO optical detections as the development of these codes was found to be outside the scope of work for this project. The ability to simulate RSO images shown against the currently available commercial option, STK EOIR, shows that the ORBITALS simulator provides accurate and quick data set enabling the training and validation of detections algorithms with large, simulated data sets. The ability to generate large data sets of optical detections of RSO's also aided in the development and testing of satellite correlation code. Specifically, code was developed by C-CORE under an IDEaS grant to enable the correlation of unknown RSO to its most likely candidate in the NORAD Satellite Catalog, which was used with ORBTIALS as it provided labelled images to be used as truth data for testing the algorithm. Other uses of the SBOIS include enabling mission preparation for space based SSA missions, which has been used to aid in the development of the RSOnar Balloon Mission out of York University. These few examples show the far-reaching positive impacts in the space community that the SBOIS contributes to with research objective 2. SBOIS has the benefit of allowing for replication of detections which allows for higher fidelity RSO characterisation to be performed. This is used in chapter 5 to enable the RSO characterisation from light curve inversion which would not be possible without an image simulator that is able to simulate thousands of images in a short period of time. Overall, the SBOIS provides a novel contribution to the space research community enabling machine learning, mission planning, and RSO correlation to be performed with large, simulated data sets where real data is scares or not available.

5 RSO Simulated Light Curve Analysis

In this section a comparison between different optimisation algorithms for RSO characterisation and the results are shown in. The results from this study were published as technical proceeding from the Advance Maui Optical and Space Surveillance Technologies Conference (AMOS) [28]. In section, the results from the optimisation comparison were implemented in a simulated environment to test different BRDF models, as well as the performance of characterising different RSO parameters. This study is currently under review for publication at the Advances in Space Research journal. For both studies the background information has been moved to the Background Information section, as well as, been formatted to meet thesis standards.

5.1 Performance of Parameterization Algorithms for Resident Space Object (RSO) Attitude Estimates

5.1.1 Introduction

The Resident Space Object (RSO) population is rapidly growing in numbers, which has become a significant concern worldwide. Today, according to NASA, more than 22,000 objects that are larger than 10-cm in diameters are in Earth orbit; many more objects that are smaller than 10-cm diameters are estimated to be in orbit. It is still not clear how many hundreds of thousands of uncatalogued debris pose threats to space assets around the world. While there are continuing collaborative efforts among various space agencies and research communities to monitor the space objects from the ground and on-orbit, a large number of uncertainties in RSO numbers, trajectories and identifies still remain. In this paper, we present a novel method to estimate RSO shape, attitude and optical properties using their light curve characteristics to enhance tracking and identification of RSOs, namely Space Situational Awareness (SSA). SSA is an important aspect of establishing space resiliency as an effective means to recognize, view, track, and forecast potential threats; much of the SSA comes from RSO’s. In future missions, it is anticipated that the further study of optical imaging of RSOs will lead to the investigation of precise orbit determination, shape, and attitude of the RSOs by closely examining light curves.

Light curve refers to a time series of the RSO’s brightness versus time as received by a sensor. Characterization of RSO properties using a light curve is a well-known approach in both astrophysics and RSO studies. In [121], for example, the authors present a new technique to interpret the light curves of debris objects. In [122], a complex modeling technique to simulate space-based surveillance images considers photometric light curves. In Section 2 of this paper, we provide an overview of recent studies on RSO characterization using light curve. The majority of the optical detection and light curve analysis are based on ground observations; limited studies reported on space-based observations.

Ground-observations of RSO’s are often managed by national space agencies. In the United States, GEODSS network provides the ‘backbone’ of the SSA effort. In [14], a major weakness of the GEODSS architecture is described as “*not provide[ing] world-wide coverage; the present network has a wide gap in coverage over Western Europe and a narrow gap in coverage over East Asia. As the great 40-inch refractor at the Yerkes Observatory, Williams Bay, Wisconsin, is still in operation after more than 115 years, there is no reason the GEODSS network cannot remain effective for many more decades*.”

Comparatively, the European Space Agency’s (ESA) Space Surveillance and Tracking (SST) activities are undertaken by the SSA program with Space Weather and Near-Earth Object tracking. In [123], an overview of SSA trends is presented; various sensors are listed, including optical, radar, RF and laser ranging for SST. In Canada, DRDC reports several observation sites with SST capabilities; for example, [124] describes the DRDC Ottawa Space Surveillance Observatory; another example is showcased in [125] where it demonstrates ground-based observation of the light curve of a small satellite.

Although ground-observation provides the baseline of SST and there is continuing effort to advance algorithms, automation, and operationalization of RSO tracking, challenges pertaining to the robustness and resilience of these systems still remain. Some disadvantages that come from ground-based observation include atmospheric scattering, temporal restrictions, possible weather interference, and viewing angle restrictions.

In comparison, space-based observations ignore atmospheric and weather effects, is not limited to nighttime observations and can view RSO's at multiple viewing angles. Currently, the largest well-known space-based SST mission is Space Based Space Surveillance by the United States DOD. Canada has already invested in building capabilities to track RSOs, such as the Sapphire and NEOSSAT missions [17]. Unfortunately, to date, there are no other currently planned on-orbit Canadian SSA missions to replace aging satellites; both Sapphire and NEOSSat have now exceeded the original mission design lifetimes. The missions described here are dedicated SST spacecrafts capable of selecting and tracking target RSOs. Disadvantages that accompany dedicated sensors are long lead times, high cost, and smaller coverage of regions of space.

We proposed an alternative to the space-based dedicated SST mission to rely on passive observation using low-resolution imagers with observation-of-opportunity for RSO tracking and identification. Star trackers have been proven feasible for RSO tracking from pervious works, such as in [49]. Fig 1 below (left) illustrates a sample FAI image where three RCM is visible.

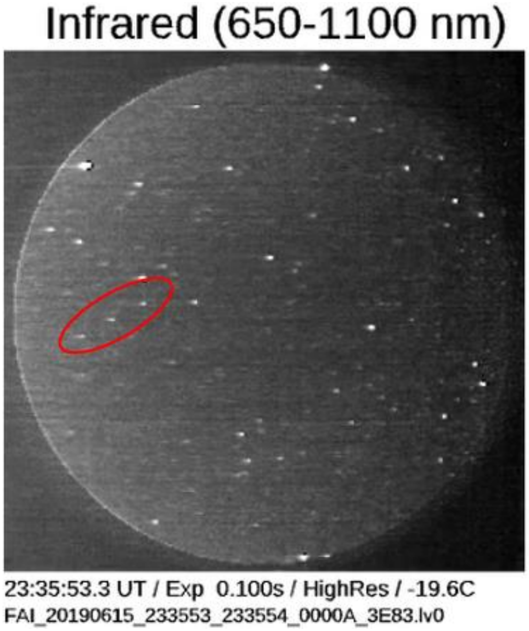


Figure 28: FAI Image with RADARSAT Constellation Mission in FOV circled in red. [126]

In the current study, we focused on light curve inversion using low-resolution images to take advantage of the observation-of-opportunities. Light curve 'inversion' refers to the process of modeling the surfaces (facets or shape) of an object from their brightness variations; in this case, reflected light. There are several commonly used techniques in light curve inversion, most of which follow the same underlying method:

- Using the shape, optical properties, and attitude to estimate the brightness of the RSO
- Comparing the brightness to the observed brightness of the object (represented by the light curve)
- Updating the shape, optical properties, and attitude estimate to reduce the error between the true and estimated brightness

In the remaining paper, we describe the overview of recent studies on RSO light curve analysis, followed by a brief review on optimization methods suitable for RSO cauterization. In Section 3, we describe the methods and algorithms we have implemented for RSO attitude estimation. We outline the results from the simulation study in Section 4, with a comparison of optimization algorithms under investigation. The last section of the paper summarizes the conclusion and suggests future tasks for further analysis of light curve.

5.1.2 Methodology

To perform the comparison of parameterisation algorithms, a simulated environment in MATLAB was used. The ORBITALS was used to simulate low resolution images taken from a space-based station. This replicates detections that would be similar to an RSO passing in the FOV of a star tracker camera. In the simulated environment, one sequence was simulated with 20 different attitude profiles for 4 unique satellite shapes to act as the simulated observed data. The unique shapes are shown in Figure 29 to Figure 32, and represent 1U, 3U, Box-wing, and Iridium First Generation satellites. Each parameterisation algorithm ran a maximum of 30 iterations with the convergence rate and final accuracy being used as a figure of merit to compare the performance of the parameterisation algorithm. Table 22 contains the input parameters for each of the parameterisation algorithms:

Table 22 Parameterisation Algorithm Parameter Inputs

Algorithm	**Parameter**	**Value**
Gradient Descent	Magnitude to Degrees	30
	Degree Step Max/Min	15/0.1°
	Bottom Random Percentage	50%
Stochastic Hill Climbing	Magnitude to Degrees	30
	Degree Step Max/Min	15/0.1°
Simulated Annealing	Magnitude to Degrees	30
	Degree Step Max/Min	15/0.1°
	Beta	0.99
	Cooling Factor	5
Particle Swarm Optimisation	Max Velocity	30°
	Current Velocity Weight	0.9
	Local Optimum Velocity Weight	1.2

Algorithm	Parameter	Value
	Global Optimum Velocity Weight	1.2
Genetic Algorithm	Cross Over Parameter	0.5
	Mutation Chance	0.01

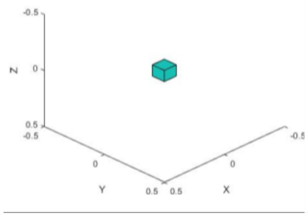


Figure 30 1U Cube Satellite Geometric Model

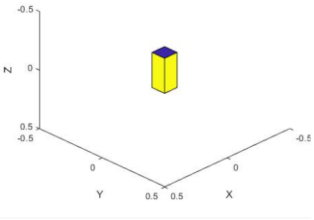


Figure 32 3U Cube Satellite Geometric Model

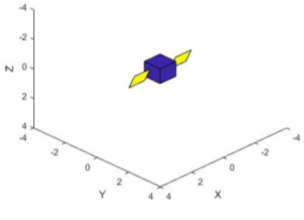


Figure 31 Box-Wing Satellite Geometric Model

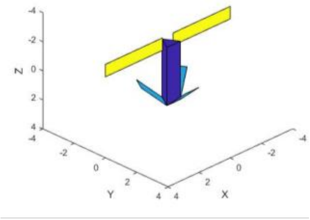


Figure 29 Iridium First Generation Satellite Geometric

5.1.3 Results and Discussion

The results for the two figures of merit, accuracy and convergence rate, for each algorithm and each satellite model is shown in Table 23. The average RMSE of the 1000 best estimated light curves for each algorithm is shown in Figure 33 to Figure 36 for each satellite model. The attitude profile residual for 1 of the 20 true sequences is shown for each algorithm and each satellite model in Figure 38 to Figure 39. How each of the 2 figures of merit are calculated is described below.

Two different types of accuracy are looked at in the results: the brightness accuracy and the attitude profile accuracy. The brightness accuracy represents the RMSE of the 1000 best light curve brightness residuals, this given over the iterations of the parameterisation algorithms give insight into the convergence of each. Attitude profile accuracy is the RMSE in the best Yaw, Pitch, and Roll parameters estimated, compared to the true Yaw, Pitch, and Roll parameters for a given sequence. The attitude profile RMSE is then averaged over different true sequences to get an average RMSE for the model and parameterisation algorithm.

The convergence rate is compared both by iteration and computation time, which is calculated from each iteration being timed. The iteration convergence rate is defined as the number of iterations it takes for the Brightness Average RMSE to drop by one half life. The computation convergence rate is the average time it takes to converge one half life.

Table 23 Parameterisation Algorithm Results

Algorithm	**Geometric Model**	**Brightness Average RMSE** (Magnitude)	**Attitude Profile RMSE** (º)	**Iteration Convergence Rate (**Iterations)	**Computation Convergence Rate** (seconds)
Gradient Descent	1U	0.0085	4.6	6.8	16.6
	3U	0.0102	4.7	6.7	16.6
	Box Wing	0.0154	5.5	6.8	20.9
	Iridium F.G.	0.0187	5.4	6.9	26.1
Stochastic Hill Climbing	1U	0.0066	12.1	6.3	13.7
	3U	0.0085	11.5	6.3	13.9
	Box Wing	0.0111	10.4	6.2	16.8
	Iridium F.G.	0.0146	10.3	6.5	21.6

Algorithm	**Geometric Model**	**Brightness Average RMSE** (Magnitude)	**Attitude Profile RMSE** (°)	**Iteration Convergence Rate (**Iterations)	**Computation Convergence Rate** (seconds)
Simulated Annealing	1U	0.0062	9.3	6.3	29.1
	3U	0.0081	8.8	6.3	29.0
	Box Wing	0.0103	9.2	6.1	35.0
	Iridium F.G.	0.0137	8.9	6.4	46.9
Particle Swarm Optimisation	1U	0.0035	2.4	5.3	49.6
	3U	0.0049	1.8	5.4	51.0
	Box Wing	0.0056	1.7	5.2	60.1
	Iridium F.G.	0.0080	2.0	5.4	76.2
Genetic Algorithm	1U	0.0074	2.6	6.6	93.8
	3U	0.0093	2.6	6.5	94.1
	Box Wing	0.0126	2.6	6.5	110.6
	Iridium F.G.	0.0171	2.7	6.8	110.0

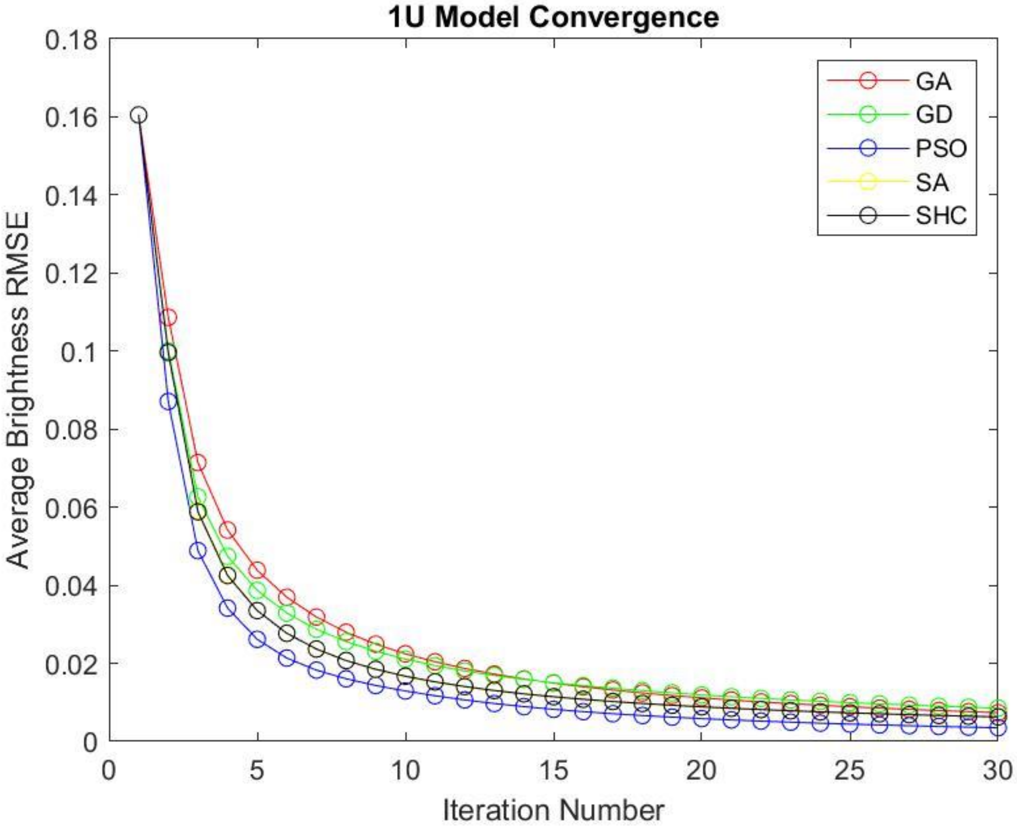


Figure 33 1U Model Algorithm Average RMSE

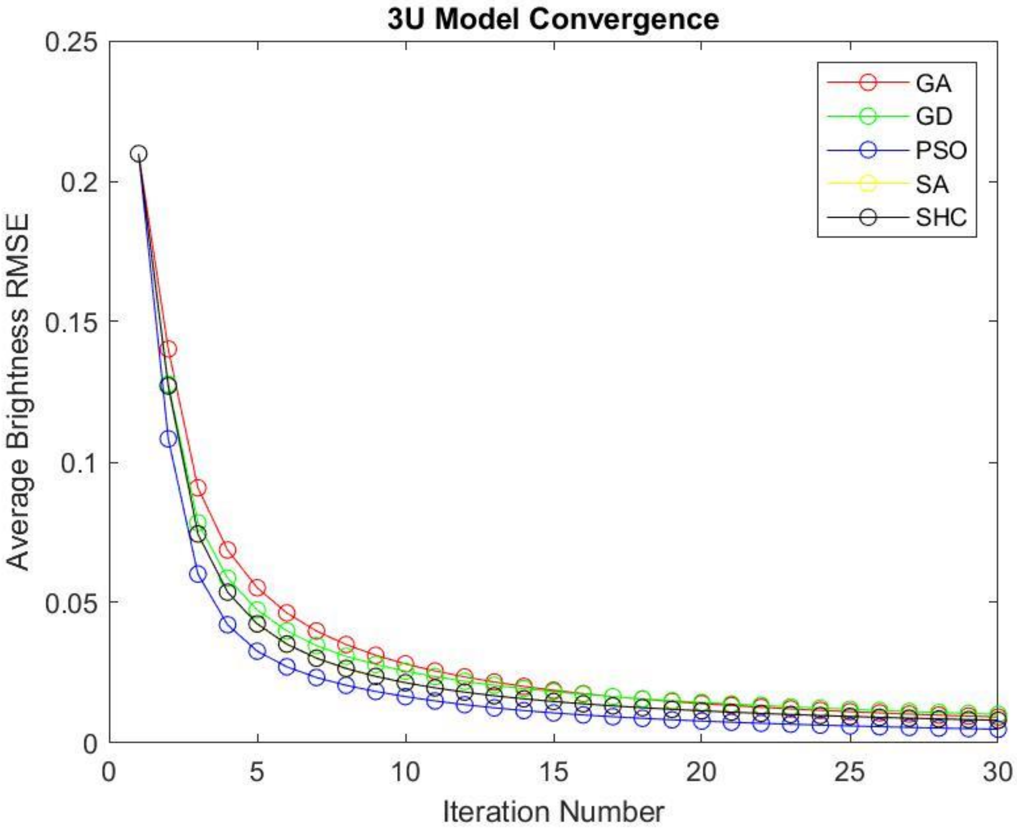


Figure 34 3U Model Algorithm Average RMSE

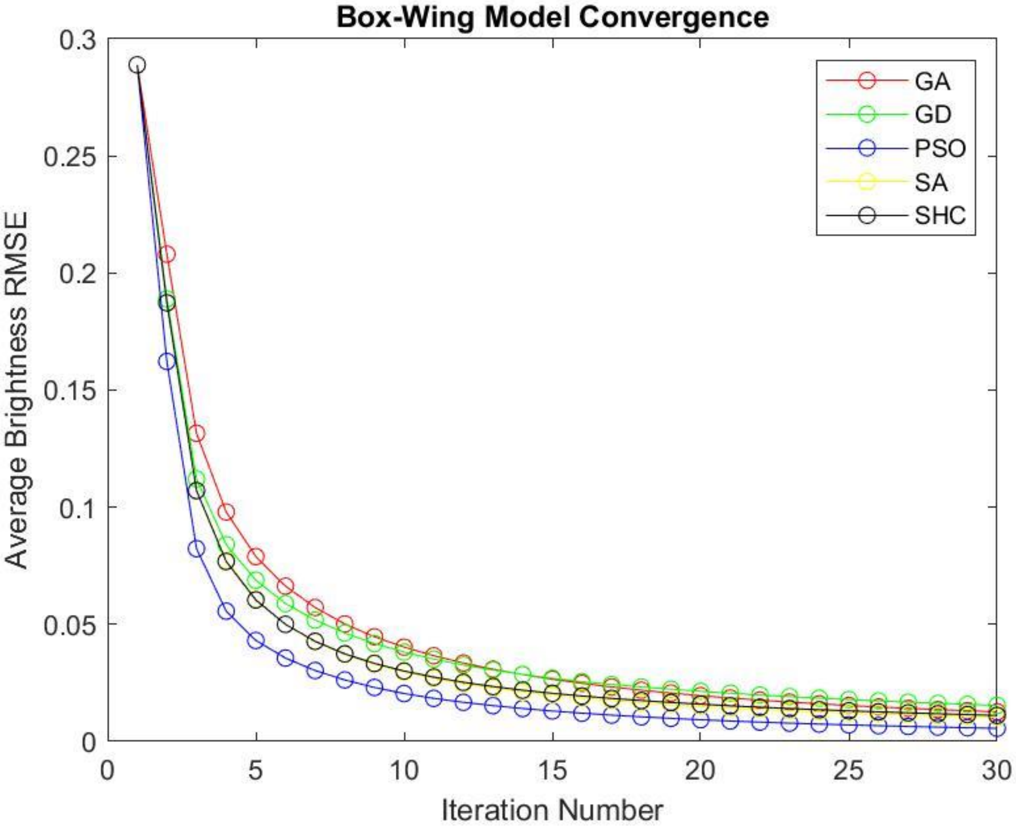


Figure 35 Box Wing Model Algorithm Average RMSE

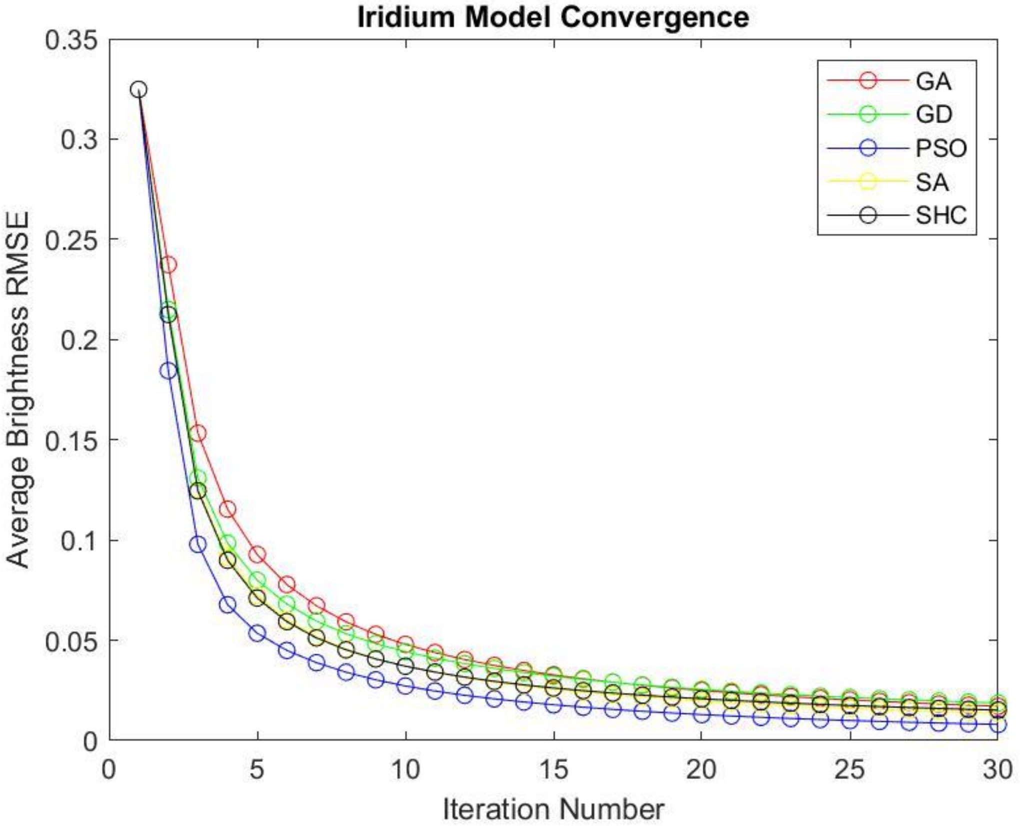


Figure 36 Iridium First Generation Model Algorithm Average RMSE

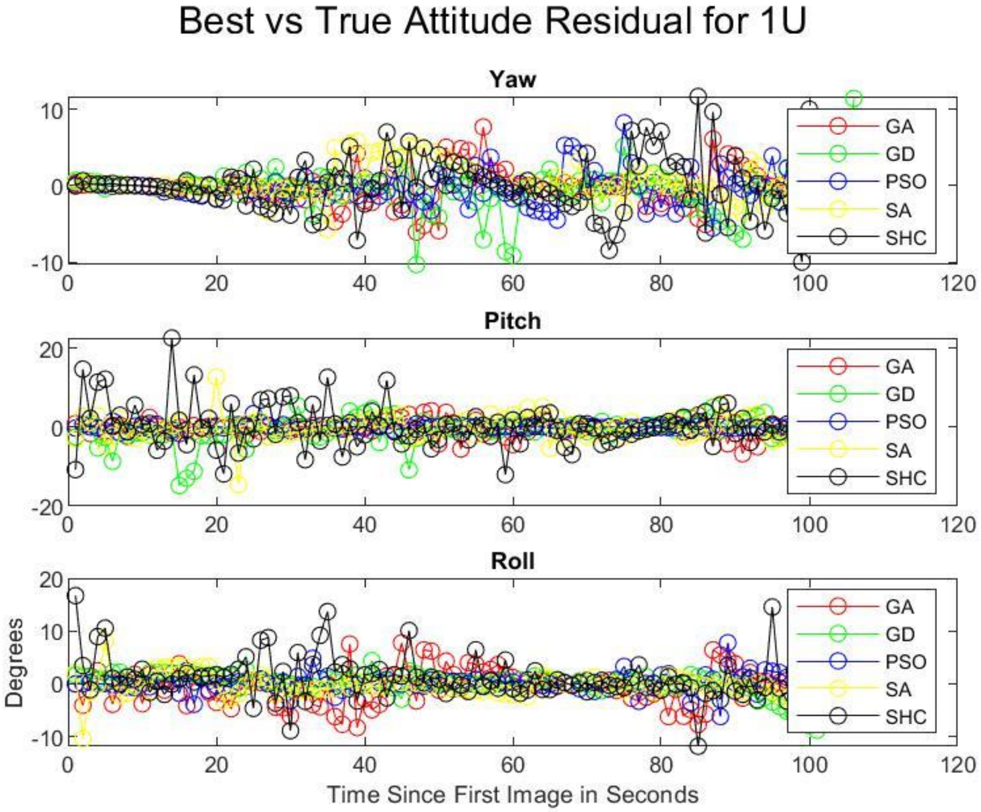


Figure 37 1U Model Attitude Residual

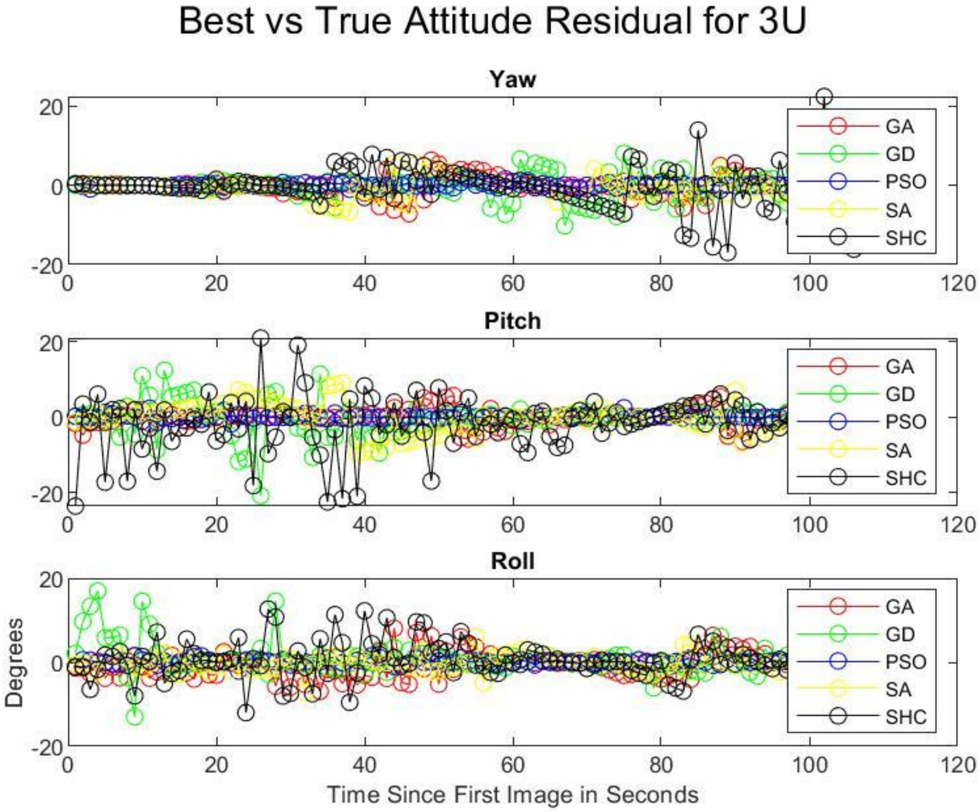


Figure 38 3U Model Attitude Residual

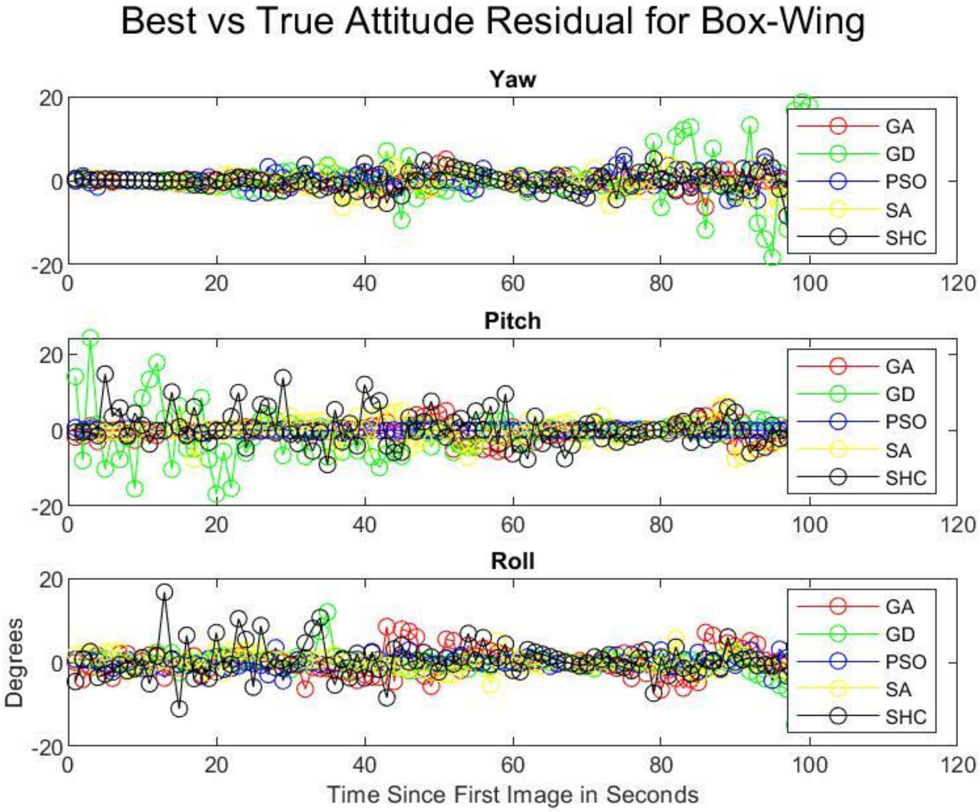


Figure 39 Box Wing Model Attitude Residual

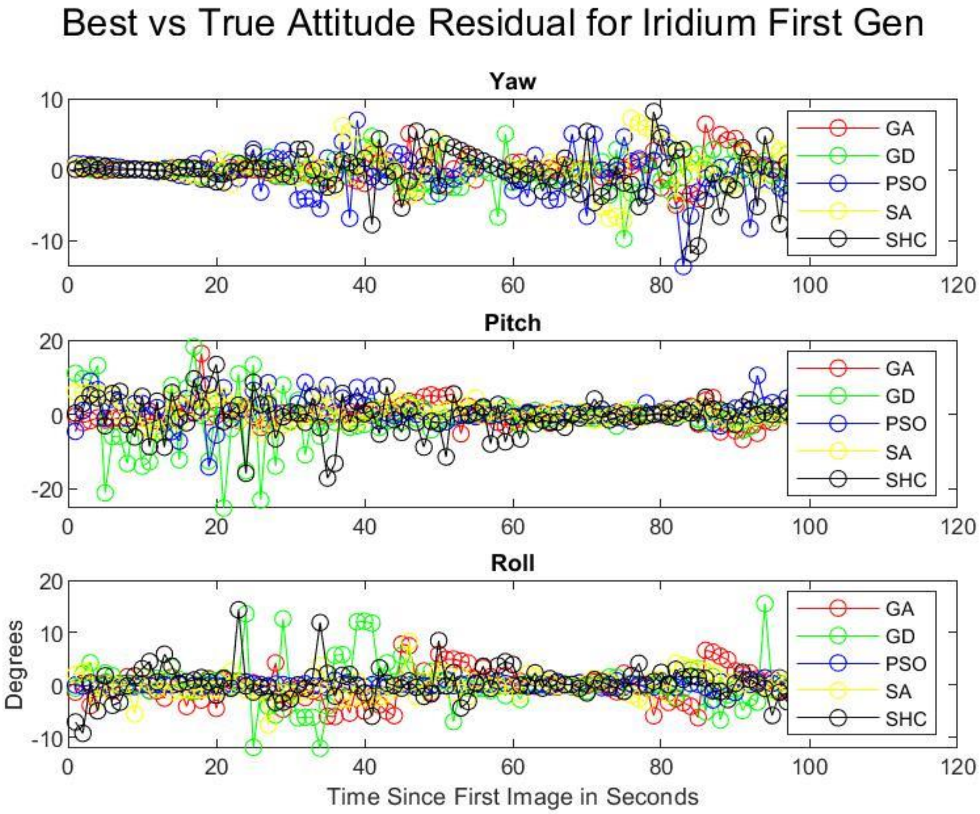


Figure 40 Iridium First Generation Attitude Residual

It can be seen in Table 23 that all the algorithms follow a similar convergence trend, with all algorithms reaching similar average RMSE at 30 iterations. PSO shows the best average RMSE for all models, with SHC and SA having similar average RMSE and performing the second best. This is followed by GA and GD, which performed the worst for average RMSE. Estimating a RSO's attitude produces a complex design space with many local optimums; GD does not have any way of escaping local optimum, which is hypothesised to contribute to its poor performance. GA has mutations and cross over operations which should allow the algorithm to escape local optimum. The mutation and cross over operation can widely change a child from its original parent; this contributes to a larger solution space being searched with less accurate local optimum evaluation. To allow for better local optimum finding with GA, different intelligent cross over and mutation operators should be tested. Hypothetically, well performing intelligent operators for this solution space would be able to consider the chromosome's relative gene differences, as well as local and global optimum. SA and SHC perform the second best as they both have a random probability of making non-optimal steps to escape the local optimum. In SA, the cooling scheme was used to calculate probability of taking a non-optimal step. A geometric cooling scheme was used, however other cooling schemes such as linear, exponential, and logarithmic should be tested and compared. In SHC, the non-optimal step is randomly chosen in this implementation. Implementing more intelligent, non-optimal step probability and direction is likely to improve the performance. It should be noted that implementing intelligent over random methods only improves performance if the intelligent methods accurately represent the solution space. PSO performed very well in contrast to the other algorithms in the solution space; this is attributed in considering a local and global optimum for the updated point calculation. The tuning of the PSO weighting coefficients and velocity calculation should lead to a better convergence for the average RMSE. The brightness average RMSE is a good indication of the performance of the parameterisation algorithms, however the attitude profile RMSE is the value that is looking to be minimized.

It can be seen from the different algorithm attitude profile RMSE that the brightness average RMSE does not fully corelate to the performance when it comes to attitude estimation. SHC and SA were the worst performing algorithms in this category, with SA performing slightly better than SHC. While looking at the top 1000 results for SA and SHC over the iterations, it can be seen that these algorithms estimates are clumped together around the local optimum, not searching the full solution space. To overcome this problem, possible solutions include better tuning of the parameter and the introduction of a tabu list to force solutions away from known local optimum. Tabu lists, commonly used in Tabu search algorithms, force the solution away from previous calculated points to keep searching for a new local and global optimum [127]. In their current implementation, with a performance of approximately 11 and 9 degrees for SHC and SA respectfully, these parameterisation algorithms are not ideal for use for attitude estimation of low-resolution optical detections. It is important to note that a different implementation of SHC and SA in the future could improve the feasibility of these algorithms for attitude estimation of low-resolution optical detections. GD is the next best thing, having an attitude profile RMSE of 4 to 5 degrees depending on the RSO shape model. The current implementation of GD takes the poorest performing attitudes and randomises them to improve the area searched in the solution space. Throughout the various iterations, this causes the 1000 best ones to change and not be as clumped as was seen with SHC and SA. As GD is very likely to get stuck in the local optimum, tabu lists is likely to increase the area searched in solution space, which possibly leads to better attitude optima estimations. GD does not perform the best in attitude profile RMSE but with its quick computation speed; it is feasible for certain implementation of attitude estimation where getting speed is prioritised over accuracy. The best two performing algorithms with similar performance are PSO and GA: PSO performing the best with 1.7-to-2.4-degree RMSE verses GA's 2.6 to 2.7 RMSE across all models. GA, having such a consistent RMSE across all models, is consistent with the cross over operator searching a large solution space but not accurately converge. Implementing more intelligent operators is likely to improve the attitude profile RMSE by allowing better convergence on local optima. PSO's ability to search a complex solution space efficiently considering local and global optima makes it ideal for attitude estimation of low-resolution optical detections. Tuning of GA and PSO will likely lead to more accurate attitude RMSE profile, as well as the addition of higher fidelity local optima searching. While comparing the different models, it can be seen that there is a similar attitude performance across all models with different algorithms performing differently for each one, GD performs better with the less complex models (such as 1U and 3U) compared to Box-Wing and Iridium First Gen. SHC is reversed where it performs better in the more complex models and worse on the less complex models. PSO and SA perform marginally better in 3U and Box-Wing than in 1U and Iridium First Gen. GA performs approximately the same across all models. While it is not fully known why algorithms perform differently across the models, it hypothesised that the symmetry and complexity of the models are expected to influence the outcome. Having less symmetric RSO should lead to fewer well-performing local optima; though, it would normally increase the complexity of the model in doing that. Increasing the complexity increases the possibility of error to be introduced in the system; this can be seen by the brightness average RMSE increasing as the model complexity increases. Additionally, more complex models require more computation time, making it a trade-off between accurate modeling and computation speed. More comparison will need to be performed on how the model of the RSO's affect the attitude estimate capability of each algorithm.

5.1.4 Future Work and Conclusion

There are many areas that require further research in order to get from an academic to a functional framework. The different areas include testing and tuning more advanced parameterisation algorithms; testing and tuning different filtering methods for the top 1000 attitudes; updating the BRDF models to be more accurate; and performing the analysis in a real verse simulated environment.

By looking at testing more advanced parameterisation algorithms, we can see the implementation of MMEA, unscented Kalman filters, and different machine learning being proven to work for RSO attitude determination. Comparing these methods with PSO and GA will give a better understanding of the algorithms' performance relative to each other, as well as in which situations each algorithm outperforms the other. One example is comparing an uncontrolled (or slow) rotating RSO with a controlled RSO that is performing a slewing maneuver. Identifying which algorithm performs better in each case can also help with the determination between controlled and uncontrolled RSO's. While one implementation of each algorithm was tested in this study, different combinations of these algorithms are also possible; for example, a geometric cooling scheme was used for simulated annealing, regardless of there being many different cooling schemes that could be implemented that would change the algorithm's performance. Currently, the algorithms optimise the attitude for each image separately and then use the derivative information to limit those results. Including the derivative analysis and filtering in the parameterisation algorithms could possibly lead to a better convergence accuracy and rate and is currently being implemented and tested for all the algorithms mentioned in this paper.

Different filtering methods on the Yaw, Pitch and Roll of the best light curve residuals can be implemented to reduce the number of possible attitude sequences and more accurately give the best estimate for attitude. For this paper, a first-order derivative analysis was used to limit the number of possible sequences. This was chosen as it was easy to implement, in the future testing different filtering methods such as second order derivative analysis, moving window filters, and whether to perform the filtering after the parameterisation algorithm or include it in the parameterisation algorithm. When comparing different types of RSO's, such as controlled verses uncontrolled, the filtering algorithms will also perform differently. While first-order filtering is sufficient for an uncontrolled satellite, it does not handle slewing maneuvers and large spin rate changes effectively and thus requires the implementation of more robust filters. Currently, testing is being performed to see how the different types of filters impact the best estimated sequence, as well as identifying if performing the filtering in the parameterisation algorithm leads to a more efficient convergence rate and accuracy.

The BRDF used for the light curve generation is a significant portion of the objective function. Updating the BRDF to a more accurate or computationally efficient methods will have a significant impact on the results of the optimisation. There are many different known methods for artificial satellite BRDF: the methods outlined in Linares, R. et al. (2014) [27]; Fan, S. & Frueh C. (2019) [128]; and Subbarao, K. & Henderson, L (2019) [129]. Different methods like the Phong model implementation include facet directional reflectance represented by Nu and Nv, which has a possibility of reducing the number of false possible sequences. Different BRDF models have also been shown to represent different RSO shapes and material properties with different accuracy [129] [128]. To properly test different BRDF methods, the accuracy and computation speed should be compared on a range of possible satellite shapes to see which method is the most robust and which method performs best for common RSO shapes. Currently, research is being completed to compare different light curve models against the current defined facet model implementation, such as the methods outlined in [128] [27].

Training and comparing algorithms in a simulated environment are effective ways to demonstrate feasibility, however, are useless without the real-world implementations. Moving from a simulated to a real environment is the next big stepping in moving low resolution optical characterisation of RSO's from an academic to an operational idea. One of the biggest challenges in taking this step is having well-labelled detections, from sensor with known parameters, where the target RSO true attitude and shape is known. Currently, thanks to the Defence Research and Development Canada (DRDC), attitude information and ground-based detections of the Radar Constellation Mission (RCM) have been provided to York University to allow for the real-world testing. By using detections of RCM from the Fast Auroral Imager, as seen in Figure 28, as well as low resolution terrestrial detections, this method feasibility and accuracy will be evaluated for terrestrial and space-based observations.

In conclusion, this paper demonstrates the feasibility of using different parameterisation algorithms for light curve inversion in a simulated environment. While all algorithms showed the ability to converge to different attitude sequences, PSO and GA showed the best results for attitude residual accuracy with both having under 3 degrees RMSE for all models. SHC and SA provided a good convergence rate but will need to be tuned to search a larger solution space.

Another finding from this paper was the different areas that still needs improvement to move from academic to operational model. Below is a summary of the areas identified to enable operational model of low-resolution optical characterisation:

- Implementation on real world data
- Improving filtering methods to handle both controlled and uncontrolled RSO's
- Tuning of individual parameterisation algorithm and implementation of meta-heuristics
- Characterise BRDF models performance with different RSO shapes and detection angles
- Generation of standard data sets with known parameters for algorithm performance testing

- Determination of minimum detection statistics required for accurate attitude estimation

5.2 RSO Attitude and Optical Property Estimation from Space Based Light Curves

5.2.1 Introduction

Damage to space systems has a significant and immediate impact on technologies we rely on every day, including navigation, communication, resource management and weather forecasting. The 2009 collision of Iridium 33 and Kosmos 2251 destroyed both satellites and created thousands of debris, which resulted in a loss of service [130]. As such, it is imperative to enhance technologies and further develop capabilities that identify both active satellites and inactive resident space objects (RSO’s) such as debris. According to NASA, there are more than 500,000 > 1cm objects in Earth’s orbit. It is still unclear how many hundreds of thousands of uncatalogued objects pose threats to space assets, such as the International Space Station. While there are continuing and collaborative efforts among various space agencies and research communities to monitor the resident space objects from the ground and on-orbit, there still remains many uncertainties in RSO numbers and characteristics.

In [37], we demonstrated RSO identification from space-based observations using low-resolution imager spacecraft such as the Fast Auroral Imager (FAI) onboard the CASSIOPE satellite. A sample FAI image is illustrated in Figure 41, with the corresponding simulated image in Figure 42. In order to interpret the low-resolution spaceborne images further study of the light curve is required to infer the parameters of the RSO. In the context of this paper the parameters include optical properties, shape, and attitude of the RSO. Knowledge of these parameters increases SSA, which aids in active debris removal, RSO correlation, and confirming that satellite ahead to current international regulations. Recent research has been focused on techniques that allow the interpretation of light curves from the images of debris objects, such as in [60] [131] [132] [133]. Other research in this field focuses on complex modeling techniques to simulate space-based surveillance images and photometric light curves [69] [22]. These algorithms provide a baseline for the current study for image simulation of the space-based images for RSO identification, tracking, and characterization. The majority of other image processing research focuses on ground-based observations with limited applicability for space-based images. Space-based research is still in its early stage; the design and implementation processes require further optimization. In this paper, we address the rising need for better SSA by comparing two different bidirectional reflectance distribution functions (BRDF) in a simulated environment. The with the goal of developing a suitable algorithm to estimate parameters from light curves extracted from low-resolution space-based images. This is accomplished through the generation of simulated light curves with known parameters then comparing the generated light curve to the found light curve of the object.

Figure 41: An image from the Fast Auroral Imager

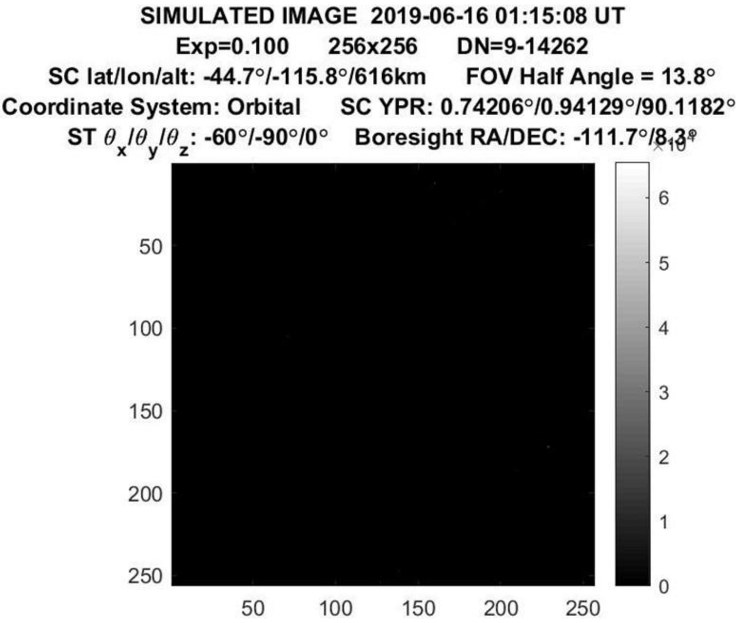


Figure 42: A simulated image from the Fast Auroral Imager

A light curve is a representation of the brightness of an object as a function of time. Light curves are used to interpret various physical parameters of the object under investigation, mentioned previously. In the case of RSO characterization, the temporal optical analysis of light curve is often used to estimate the rate of rotation (also referred to as spin-rate) of the object. In [134] (and many articles with similar analysis), the authors present an overview of how light curve analysis is used for RSO characterisation using low-resolution images. More generally, light curve inversion is a mathematical technique used to estimate different states of an objects from its brightness and brightness variation over a series of timed measurements, which provides more information than spin rate analysis. Light curve inversion is frequently used for the detection of exoplanets and the shape and spin estimation of asteroids, such as in [135] and [28]. To invert the light curve to determine more information than just the temporal spin rate of the object, simulated environments are commonly used. They determine how the variation of brightness corresponds to a change in parameters which are used to generate the simulated light curve. To perform a light curve inversion an optical image simulator, in the case of this paper ORBITALS is used to replicate the brightness of the RSO with estimated states that consists of shape, attitude and optical properties. ORBITALS is further described in above in the section 2.5 of this thesis. The complexity of the light curve inversion problem stems from the brightness giving non-unique solutions for state estimation, where hundreds to thousands of solutions are plausible for one brightness measurement. To overcome the challenge of non unique answers a time series of brightness (the light curve) is used. The time series allows for higher derivatives of data to be used to reduce the number of potential solutions, or local optimum, present. The higher derivative data limits the solution space by ignoring solutions with large discrepancies in state parameters that would not be physically possible. One example of this is large attitude jumps that can sometime be larger than 30 degrees a second. It is unlikely to leave only the ideal solution, even after higher derivative fitting, requiring optimization to find better estimated states. It was demonstrated in previous research that Particle Swarm Optimization performs effectively for attitude estimation; thus, it is implemented in the current study to calculate the optimal state estimates [75].It is worthwhile to note that any optimization algorithm in theory can be implemented to retrieve an optimal state estimate. In this paper, the methodology for generation of simulated images, as well as simulated environment used is described.

Followed by the results and discussion of the simulation tests and concluded with a summary of the results and areas of future research mentioned.

5.2.2 Simulation Parameters

For the optimization algorithm, the same implementation of particle swarm optimization was used as in [28], with the number of iterations and parameters in Table 24: Particle Swarm Optimization Parameters. There are three unique tests being performed on the simulated data a) Attitude only, b) Optical Property only, c) Attitude and optical property estimates. In each unique test all the other information is known with only the parameter being analyzed being unknown. From the results of the three unique tests, a comparison was performed on the convergence rate to an optimal brightness estimates as well as the accuracy of the estimated states.

Figure 6: Sample Images of RCM constellation from FAI on-orbit observation (left), image credit Cowan and E (2008), simulated images of the same RCM constellation using SBIOS (Right) take on June15, 2019, 23:36:05.

Table 24: Particle Swarm Optimization Parameters

Particle Swarm Optimisation Parameters	
Total Iterations	100
Number of Points	20,000
Max Velocity	30
Current Velocity Weight	0.9
Local Optimum Velocity Weight	1.2
Global Optimum Velocity Weight	1.2
Beta	0.99
Cooling Factor	5

Table 25: Fast Auroral Imager Parameters

Fast Auroral Imager Parameters	
FOV	26 degrees
Aperture Diameter	1.7 cm
Focal Length	6.89 cm
Integration Time	0.1s
Spectral Band	650 – 1100 nm
Pixels	256 x 256
Pixel Size	26 µm
Quantum Efficiency	0.66

5.2.3 Results

The results from the (a) Attitude, (b) Optical, and (c) Optical and Attitude estimate tests are shown in Table 26, Table 27, and Table 28, respectively.

Figure 43 and Figure 44 illustrate only the convergence rate for optical estimation. Each of the figure of merits used in the comparison of the models are also discussed below.

The first figure of merit used in all three tests is the Average Best Brightness RMSE (ABB RMSE), defined as the root mean squared error between the best estimated light curve and the true light curve. ABB is then averaged across each of the true values and detections. Secondly, computation time is used as a second figure of merit, which includes the average computation time, the iteration convergence rate, and the convergence rate. The average computation time is the average time taken to perform on iteration of parameter estimation for the given model and test. The iteration convergence rate refers to the number of iterations it takes for the ABB RMSE to drop by one half-life (drop by 50%). The convergence Rate is defined as the average time it takes to converge one half-life. The third figure of merit, Average Best Attitude RMSE (ABA RMSE), is used to compare the different estimated attitudes. ABA RMSE is the RMSE of the yaw, pitch, roll residual which is then averaged across all true values and detections. For the optical figure of merit, due to the different scales and parameter used for Phong and Defined model, a direct comparison is not possible. Instead, a mean absolute percentage error (MAPE) is produced for each test to give a comparable figure of merit. MAPE is a standard technique in statistical analysis and regression models Guo et al. (2019). This is calculated by taking the absolute present error of the optical parameters then averaging it across all true values and detections.

Table 26: Simulated Attitude Estimation

Parameter	**Phong Model**	**Defined Model**
Average Best Brightness RMSE (Relative Magnitude)	$3.6\text{x}10^{-7}$	$2.2\text{x}10^{-6}$
Average Best Attitude RMSE (Degrees)	1.6	1.2
Average Computation Time (Seconds per Iteration)	11	8.2
Iteration Convergence Rate (#Iterations per Half Life)	10	10
Convergence Rate (Seconds per Half Life)	108	81

The results from the attitude only simulated test shows that the Phong model outperformed the Defined model in ABB RMSE, but not in ABA RMSE. These FOM indicates that the Phong model’s complex design space leads to the particle swarm optimization algorithm to converge better on local optimum but not on the global optimum which is represented by the true attitude. The Phong and Defined models have approximately the same iteration convergence rate the Phong model has a longer average computation time compared to the Defined model. In summary, the Defined model yields more accurate attitude estimates in less time; thus, it is a preferred modeling technique over the Phong model with particle swarm optimization. Just like the Defined model, the Phong model performance changes based on the optimization algorithm, indicating that implementing other optimization algorithms may lead to improved performance both in accuracy and computation speed.

Table 27: Simulated Optical Parameters Estimation

Parameter	**Phong Model**	**Defined Model**
Average Best Brightness RMSE (Relative Magnitude)	2.2×10^{-5}	1.7×10^{-3}
Specular Weighting RMSE (Reflectivity)	7.1×10^{-3}	9.6×10^{-3}
Diffuse Weighting RMSE (Reflectivity)	7.1×10^{-3}	9.6×10^{-3}
Reflectance RMSE (Reflectivity)	N/A	5.4×10^{-3}
n_u RMSE	1.1×10^{3}	N/A
n_vRMSE	1.2×10^{3}	N/A
Optical MAPE (% error)	23	2.1
Average Computation Time (Seconds per Iteration)	3.7	6.7
Iteration Convergence Rate (#Iterations per Half Life)	8.5	1.5

Parameter	**Phong Model**	**Defined Model**
Convergence Rate (Seconds per Half Life)	32	10

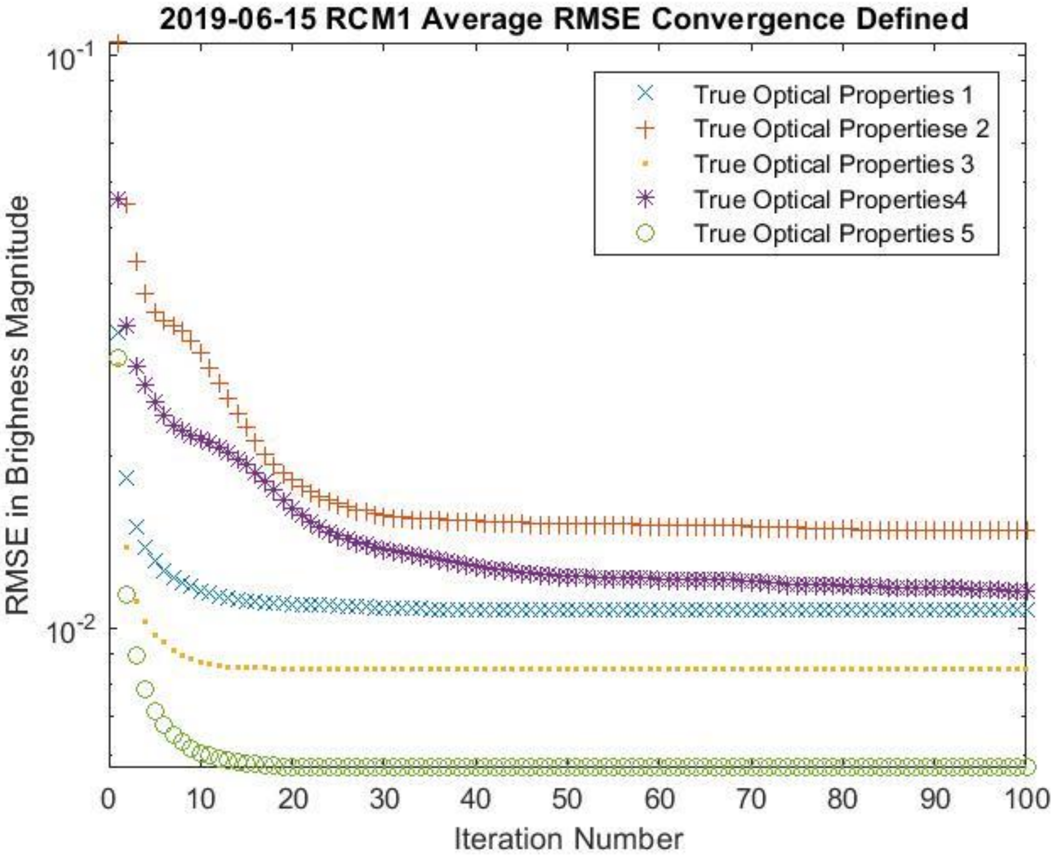


Figure 43: Convergence Rate of Optical Parameters for the Defined Model.

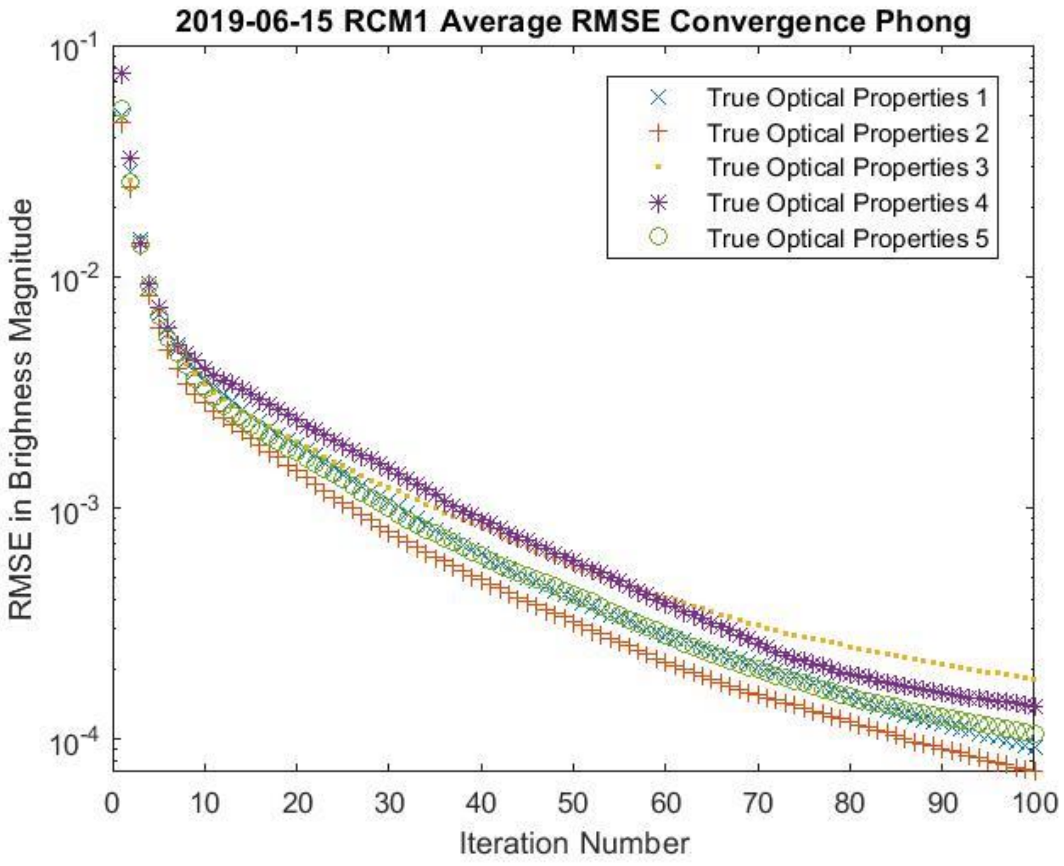


Figure 44: Convergence Rate of Optical Parameters for the Phong Model.

The results from the optical parameter estimates show that the Phong model had better convergence in ABB magnitude, but not as good Optical MAPE as the defined model. This is similar to the attitude only tests, the results showed greater convergence to local minimum, but had problems converging to the global optimum. By comparing the optical parameter results, the same trend can be seen with the Defined and Phong model. The MAPE for all optical parameters is 2.1% for the Defined model, compared to the Phong model, which has a 23% MAPE. The Phong model requiring 5 inputs versus the Defined models 4 leads to a larger response surface that will contain more local optimum. The effect of the large response surface is not just reflected in the accuracy, but also in the iteration convergence rate between the two models, with the Defined model converging about 5.5 times faster. It is observed from the convergence rates (

Figure 43 and Figure 44) that the Defined model converges before 20 iterations, whereas it takes more than 80 iterations for the Phong model. The results from the attitude and optical parameter estimation test (Table 27) show that increased number of parameters greatly reduces the ABB RMSE convergence for both models. The Defined model performed better than the Phong model in ABA RMSE; however, the two methods have different convergence rates compared to the attitude estimation test. The difference in convergence rates is attributed to the defined model converging quicker and having a slightly faster computation time. Comparing the attitude to attitude and optical parameter test shows that the Defined model had a decrease in ABA RMSE, while the Phong model had an increase of ABA RMSE. The increase in ABA RMSE was predictable in the Phong model as the response surface greatly increased in size. In comparison, the ABA RMSE in the Defined model decreased with the increase in response surface. It is theorized with the Defined model converging quicker than the Phong model, the increase in parameters allows for more sensitivity in the solution and a larger number of sub-optimal solutions being rejected in the optimization process. If the Phong model was run through more iteration a similar trend should emerge.

Table 28: Simulated RSO Parameter Estimation Test Results

Parameter	**Phong Model**	**Defined Model**
Average Best Brightness RMSE	0.51	0.27
Average Best Attitude RMSE (degrees)	5.3	0.66
Specular Weighting RMSE (Reflectivity)	8.8×10^{-4}	5.0×10^{-3}

Parameter	**Phong Model**	**Defined Model**
Diffuse Weighting RMSE (Reflectivity)	$8.8\text{x}10^{-4}$	$5.0\text{x}10^{-3}$
Reflectance RMSE (Reflectivity)	N/A	$2.2\text{x}10^{-2}$
n_u RMSE	$4.5\text{x}10^{3}$	N/A
n_v RMSE	$3.3\text{x}10^{3}$	N/A
Optical MAPE (% error)	8.5	0.9
Average Computation Time (Seconds per Iteration)	34	27
Iteration Convergence Rate (#Iterations per Half Life)	277	160
Convergence Rate (Minutes per Half Life)	159	73

Both methods performed well in determining the optical properties and attitude; the Defined model determining the ABA with less than one-degree RMSE across the full sequence of images. The comparison of the optical parameter MAPE has 0.9% accuracy for the Defined model and 8.5% for the Phong model. The Defined model outperformed the Phong model, which matches the results from the optical property and attitude only test. The largest change in the parameter test is observed when looking at the convergence rate, which increased over an order of magnitude compared to the attitude and optical parameters only tests. The increase in parameterization accounts for a majority of the increased iteration convergence rate, which is largely attributed to the local optimum distribution becoming more complex with the more parameters. Another factor in larger parameterization is that more information needs to be calculated and updated each time, leading to (at best) approximately 2 times increase in the computation time for both models. The Defined model outperformed the Phong model in both the convergence rate and computation time, allowing it to converge around twice as fast as the Phong model. With better accuracy and computation speed, the Defined model was found to perform better than the Phong model for parameter estimation of RSO's in the current implementation for light curves from low resolution space-based imagers.

Comparing the proposed RSO attitude and optical property estimation technique to other techniques the strengths of the currently implemented models and areas to further research can be seen. The [136] implementation of light curve inversion follows a similar methodology, as well as the results [137]. In this publication, the attitude is determined from the orthographic projection of the target with an iterative attitude estimation. This algorithm was implemented in a lab set up with a known target and a distinct and non-homogeneous circular pattern; the results for single image attitude estimation shows that the attitude of the object can be found to within 1 degree. These results are similar to the ones found in the paper with two large differences between the methodologies. First, the iterative method is used for singular images, not requiring image sequences. The driving factor behind the ability to only use one image for attitude estimation is the second major difference between these two methodologies. The second difference is that the target object is a significant portion of the FOV, unlike with low resolution RSO detections which are commonly subpixel or few pixel-sized objects. The smaller size of RSO's in low resolution space-based images limits the applicability that this method has in this paper's context. Having both a relatively high spatial resolution and low spatial resolutions sensor for attitude determination of RSO's can lead to leveraging the strength of both types of sensors in current SSA applications, leading to better estimated attitude states.

Comparing more similar methodologies for light curve inversion techniques we look at results such as presented in [137] and [138]. A similar methodology was followed to methodology presented in this paper, but with the implementation of an unscented Kalman filter (UKF) as an optimization algorithm. In [137] a simplified Cook and Torrance reflectance model is used in contrast with the Defined model used in this paper. The target objects were rocket bodies, which give easy shapes for simulated environment tests. The simulated environment tests give preliminary results that the UKF and light curve model was able to estimate both the attitude and optical properties within reasonable accuracy. When moving to real data, it was found that the simplified reflection model would not be sufficient for the more complex RSO shapes. The Defined model implemented in this paper has shown the ability in a simulated environment to overcome the higher fidelity model requirements. The implementation of the UKF shows impressive results with more complex methods, such as Multiplicative Extended Kalman Filter (MEKF) being implemented in [30]. The power of the UKF and other Kalman filtering methods, highlighted in [137] and [138]), should be compared to the currently implemented optimization algorithm (PSO) to see the accuracy and computation speed differences between the two.

In a recent study performed by [138], a comparison of different filter techniques for state estimation of uncooperative objects was performed on MEKF and different minimum energy filter models. The results showed similar performance on the estimation of targets attitude from both MEKF and a second order minimum energy filter. The MEKF was able to estimate attitude with a transient error RMSE of 1.21 degrees and steady state error RMSE of 0.35 degrees, with a given priory and inertial matrix of the target. With no priory estimate and inertial matrix, it was mentioned that the second order minimum energy filter does perform better, giving an error rate of less than one degree after convergence. Comparing the results found from this paper shows that the method for attitude estimation from light curve inversion and PSO give similar accuracy as other currently used techniques. From [138], it can also be seen that the choice of state estimator greatly effects the accuracy of the attitude estimate depending on the scenario and information available to the filter. The results of different state estimators means that it is highly unlikely there will be one optimum RSO attitude estimation algorithm for all situations. Instead, it is much more probable that the optimization algorithms will be chosen based on the spatial resolution, sensitivity, and temporal resolution available, much like with orbital propagators. Identifying which algorithms perform optimally in different scenarios will allow for better attitude estimation and high-fidelity attitude estimation. Some of those scenarios include terrestrial verses space-based detections, low vs high spatial resolution cameras, and tracking vs stare mode.

5.2.4 Conclusion and Future Work

We have compared the Defined model with the Phong model of an RSO for three types of parameter estimation using low-resolution images in a simulated environment. The results indicate that, with particle swarm optimization, the Defined model outperforms the Phong model in almost all metrics, accuracy, computation time, and convergence rate. The Phong model was examined due to the directionality of incoming light effecting the specular and diffuse reflections off the facet, which is not considered in the Defined model. It was expected to increase the accuracy of the estimated states due to the additional parameters (n_u, n_v), reducing the number of geometries that provide the same brightness. Instead, the results indicate that the increase in number of parameters exponentially increased the size of the response surface, making convergence to the local optimum require more iterations. Furthermore, the increase in response surface leads to more local optima present when compared to the Defined model. The increase in number of local optima leads to the Phong model requiring more iterations and data points, making it significantly more computation- ally intensive to achieve a similar accuracy. The implementation of different, more intelligent optimization algorithms in the future could lead to an increase in performance of the Phong model.

In conclusion, the Defined model form the foundation in characterization of unidentified RSO attitude and optical properties. In the future this technology will lead to better remote sensing of RSO’s health and operation status. Future work looks to take this method from being performed in a simulated environment to using real data. To perform a real-world feasibility, test the light curve analysis for attitude estimation and optical characterization will be performed using on-orbit images of operational spacecraft with known health status. Currently this is being performed on NEOSSAT images from their Early Orbit Phase (EOP) taken from a ground-based telescope provided by DRDC Canada. While ideally the images would be from a space-based sensor, due to the lack of labeled space-based images with auxiliary information ground-based observations were chosen. By comparing the results from the proposed estimation technique to on-orbit NEOSSAT data (truth data), we can validate the accuracy of the proposed light curve inversion methodology. The current methodology still has area of improvement that will be further enhanced such as: including intelligent optimization, improved BRDF modeling, and efficient parameterization.

5.3 Summary

RSO characterisation from low resolution space-based platforms was demonstrated as feasible in a simulated environment providing a novel solution for determining characteristics of RSO's from these types of detections. The current novel method of RSO characterisation is different from RSO characterisation method by allowing more details to be determined that standard glint or period analysis. These analyses do not require any priory information to provide the body spin rate of the object but are not able to give more information look at in the dissertation such as size, shape, and optical properties. When performing glint and spin rate analysis the light curve of the RSO needs to be longer than the spin period, if it is not the full spin rate period is near impossible to find. The novel method proposed allows for any length light curve to be analysed with longer light curves providing more accurate results. Some RSO characterisation method do exist that allow for the determination of size, shape, and optical properties but these methods either require lots of data, as seen in [24], or require priory information, such as [31], on the target RSO which is not always available. The novel RSO characterisation method allows for these properties to be estimated with out priory information, but priory information can be introduced to limit the search space and converge to an answer quicker. The last large contribution to the space community that this novel method provides is its robustness to simulate multiple types of detections. This novel solution with SBOIS allows for both ground and space based light curves from optical sensors of any type to be used for characterisation, though only space based light curves were focused on this thesis. This will allow SBOIS and RSO characterisation to be used by a wide range of institutions and in a wide range of situation fill in a large gap currently present in RSO image simulation, without expensive licences. Overall, the novel method of RSO characterisation to accomplish research objective 3 is the largest contribution to the space research community. This novel solution incorporates the work done to accomplish research objective 1 and 2 providing use cases and results for SBOIS and the PSGP4. This novel solution fills a unique gap in currently available RSO characterisation methods opening the door for future research opportunities, as well as the capability to use this method to solve current issues with the proliferation of RSO's in space.

6 Summary, Contribution, and Future Work

6.1 Summary

In the current space environment star trackers are commonly used for attitude determination and are becoming more common on Cube-Sat missions, and very common in larger scale missions such as Radar Constellation Mission. Currently star trackers and other low resolutions space-based imagers are not commonly used for RSO detection to contribute to SSA. There are many factors required to accomplish this including knowing when an RSO is in the FOV, having the detected RSO have a high enough SNR to be identified in the image, and confirming that RSO's shape and orbit information can be determined from the detection. This leaves a large section of currently available space assets that are not be utilised to their full potential and have the capability to augment the current SSA networks to combat the problem tracking the proliferation of RSO's. From the research performed in this thesis the main technological gaps needed to enable passive low-resolution space based RSO were identified and addressed. These technological gaps included parallel processing for orbital propagation, the simulation of the low-resolution passive optical RSO detections, and the characterisation of RSO's from low resolution passive optical RSO detections. The end goal of this research looks to be the first step in the augmentation of star trackers, and similar imagers, to be able to passively detect RSO's to contribute to SSA. The first area of research explored in this thesis was the implementation of parallel processing to orbital propagation and image simulation, with the goal of reducing the required computation time with no reduction in accuracy.

Parallel processing has become a large field of study in space and other terrestrial sectors because of its ability to leverage the rapid improvement of computation power in processors and associated hardware. The implementation of parallel processing for orbital propagation was does in two phases, first focusing on the state propagators. The point of the research on state propagators was to first confirm that parallel processing will reduce the required time to perform multi-satellite and catalog propagation. This technological gap has become of increasing importance in the space sector with the estimated number of

RSO’s in the space catalog estimated to jump an order of magnitude in the next few years. The initial hypothesis was that parallel processing will reduce the computation time by leveraging the increase in hardware capabilities using SIMD and the fact that satellites are independent. From the initial study performed in section 3.1 the results show that parallel processing was able to greatly decrease the computational requirements compared to traditional sequential processing methods for orbital propagation. This effect was not seen when propagating a low number of satellites propagated due to the data set up for parallel processing taking some initial time. During the initial feasibility study, it was found that for a proposed mega constellation of 500 satellites using parallel processing for orbital propagation required approximately 20% of the computational time required from the sequential orbital propagation method. Very similar results were seen when increasing the period that the satellites were propagated over with an approximately 20% of the computational time required from the sequential propagation method. The reduction for 500 satellites demonstrates the power of parallel processing with the trend showing that parallel processing gets more efficent, relative to the sequential method, as the number of satellites and period of propagation increases. Confirming that the propagation was the cause of this was from the ground station test where the number of ground stations were varied, as this does not have an impact on the propagator being tested. The differences in the parallel and sequential trend in the ground station test was very different from the satellite and period test, confirming the hypothesis that parallel processing increases the computation speed of orbital propagation with out reducing accuracy. One of the most interesting results and visualisation of improvement parallel processing has on orbital propagation comes from the trends of the Dorman Prince 5(4), method for period and satellite testing, Figure 23 and Figure 24 respectfully. In these graphs the trend of computation time trend starts of linearly till they reach a saturation point where the trend is no longer linear. After this saturation point the computation time does not change significantly with an increase in the number of satellites, or propagation period. This demonstrates the power of parallel processing with propagation performing the same instructions on multiple data sets (satellites) it can perform them at the same time as long as it has the hardware resources to do so. This is shown by the plateau after the linear region in both graphs having 500 satellites being propagation with approximately the same computation time as 2,500 satellites. This does have its limits when the amount satellites or propagation period requires more resources than available it is required to do it in steps. This can be seen in Figure 23 with the jump in computation time from ~70 seconds to ~140 seconds after 160 days of propagation, and is not seen in the number of satellites, Figure 24, as more satellites would be required to reach the hardware limit. This is a massively beneficial in the context of SSA where thousands of RSO's are propagated, allowing for more RSO's to be introduced with out the linearly increasing the computation time required. From the beneficial results of the initial feasibility study parallel processing was implemented on a more complex propagation mode, SGP4, as well as, implementing parallel processing in the image simulation of low-resolution space based optical RSO detections.

The improvement of SGP4 architecture to utilising parallel processing has many implications in the space sector and simulated image generation. In the context of this thesis the image simulator requires an orbital propagator to find the locations of the RSO's to see; if they are in the image, and if they are in the image there located in the image. Before implementing this for simulating images a test on the performance of SGP4 vs a parallel implementation of SGP4, called PSGP4, was required. While parallel processing was shown to improve the computation speed of state propagators SGP4 is a Keplerian propagator, to demonstrate that the technology is transferable an implementation and comparison of serial and parallel Keplerian propagators was required. The study in section 3.2 and its results were used to verify and quantify the improvement in computation time of parallel processing methods orbital propagation methods compared to serial methods. To do this orbital maneuver detection algorithms, state and TLE, were tested and compared using both parallel and sequential propagation methods. As was hypothesised it can be seen that the parallel processing version of both the state and TLE maneuver detection performed with less computational time while not reducing the accuracy. In orbital maneuver detection state propagation is a higher fidelity method than TLE but comes with the downside of being significantly more computationally intensive. With the implementation of parallel processing for orbital propagation the required time for state analysis dropped orders of magnitude, with the computation time now being comparable to the TLE analysis method. Comparing the accuracy between the parallel and sequential implementation of the orbital detection algorithms demonstrates that the decrease in computational time has no effect on the accuracy of the results. These results have large implication for the over all SSA, as well as, in the goal of this thesis. Almost all RSO identification and conjunction warning algorithms require orbital propagation of large numbers of RSO's, the introduction of parallel processing not only greatly reduces the computation time but can allow for multiple analysis to take place at once. For conjunction warning algorithms, that commonly use Monty Carlo simulation for high fidelity results, being able to perform the thousands to millions of orbital propagations at once can greatly reduce the computation time required or be leveraged to have more orbits getting higher accuracy results. In the context of this thesis orbital propagation is one of the major bottle necks for the current image simulator architecture, updating this to the PSGP4 propagation method can greatly increase the throughput of the image simulator.

Image simulation is an important but sometimes overlooked aspect of RSO detection, identification, and characterisation. Image simulation allows for the estimation of the performance of sensors, RSO identification, scheduling, and mission planning, as well as the characterisation of RSO properties. Most current image simulators focus on higher resolution imagery where an RSO would be seen as a multipixel object taking up a majority of the screen, unlike low resolution imagery where almost all RSO's are subpixel sized. To overcome this a star field image and RSO simulator was made in MATLAB. Since the image simulators first version, performed by a previous student, it has been completely overhauled to ORBITALS which implements parallel processing methods to reduce the computation time and includes different reflection models for RSO modeling and light curve generation. The industry standard software that best replicates what is being performed in ORBITALS is System Tool Kit's (STK) EOIR toolbox. To demonstrate the power of the newly developed ORBITALS it was compared vs STK EOIR for image simulation using real images from the Fast Auroral Imager onboard the CASSIOPE satellite. The results from the comparison of ORBITALS to STK's EOIR show that ORBITALS generates images with similar accuracy for both stars and RSO location but, an order of magnitude faster than STK's EOIR. Also, STK is a commercial software the requires licences to be bought for the base software, as well as the EOIR toolbox. This is extremely expensive and can create a barrier to entry for smaller companies and research group. ORBITALS does currently require an active MATLAB licence but no external toolboxes, making it a more affordable and accessible option when compared to STK's EOIR. Also, with MATLAB being standard in the space sector a majority of companies and research groups already have a licence, which requires no additional financial investment to use. Another major benefit of ORBITALS is that it automates the labeling of data with the ability for the user to control the data outputted such as position, relative motion, SNR, RSO catalog number, etc. In STK EOIR the user needs to select the pixel and extract the information manually or have a 3rd party code extract the labeled data. When dealing with large data sets the manual labeling of data can be time consuming, the automatic labeling in ORBITALS overcomes this allowing for generation of large data set with limited human interaction. The reduction in computation time and labeling allows for ORBITALS to generate large data sets more efficiently than using the commercially available STK EOIR counterpart.

ORBITALS allows for inputs of different bidirectional reflection distribution function for satellite models. This gives more flexibility allowing for models tailored to specific RSO's to be used, as well as allows for the comparison of different models. Initially the industry standard defined model was implemented, this performed well for RSO detection, identification, and characterisation. Using this model and real RSO detections tests were performed to see if space based RSO detections allow for the characterisation of RSO's like terrestrial observations. In a simulation environment the results from section 5.2 show that the standard model was able to estimate the optical properties, attitude, and the optical properties and attitude together for the RCM satellites. As part of this research different BRDF's were tested for satellite facets, specifically the standard model and anisotropic Phong model. While initially the hypothesis was that the implementation of the Phong model would allow for better characterisation of the object, this was not shown to be the case. The increase in parameters in the Phong model compared to the standard model was shown to significantly increase the solution space for the optimisation algorithm to search. The larger solution space lowers the convergence rate of the optimisation algorithm as have more area to search. It should be noted that these results are dependant on the optimizer used, implementation and tuning of other optimisation algorithms could lead to the Phong model possibly outperforming the standard model in the future. More of this is discussed in the Future Work section. During this research different types of optimisation algorithms were implemented and tested for light curve inversion. The results show that PSO performed the best converging quicker and to a more accurate answer than SHC and GA. PSO ability to not require and gradient information or pseudo gradient calculation allowed it to perform well when dealing with the complex response surfaces from the RSO light curve inversion. There is still lots of work to be performed on the testing and tuning of different optimisation algorithms with different BRDF, this will be mentioned in more details in the Future Work section below.

6.2 Future Work

The work performed in this thesis has reduced the technological gaps limiting the use of low-resolution passive space based RSO detections. There is still more work to be performed to enable the operational use of low-resolution passive space based RSO detections, below some of the remaining technological gaps are mentioned with suggested research areas to overcome the technological gaps. The largest and most predominate technological gap is the implementation of RSO characterisation algorithms on real data taken from a space-based platform. Originally in this thesis the goal was to eventually use real world data to compare the results to the simulated data, due to the lack of publicly available labelled data sets this was not possible. Labelled data in the context of this thesis is defined as data where the auxiliary information to replicate the detection (ephemeris data and optical parameters) of the host platform is provided, as well as the identification of what RSO's are visible in the image including their centroid, brightness, and NORAD

Satellite Catalog ID #. To be able to perform RSO characterisation information about the target RSO and host satellite is known, the closest publicly available data set that provides this information is the Fast Aural Imager images. This provides the host satellite information but still has a lack of information on the targets shape, optical properties, and attitude to confirm the results against. In the context of this thesis ORBITALS was used to overcome this lack of data, recently new opportunities for labelled data have become available allowing for RSO detection, identification, and characterisation to be tested with real data. Thanks to partnerships with the DRDC we have been provided the attitude for NEOSSAT during 2016 before and after the on-board failure, along with images of NEOSSAT to allow for light curve extraction. Performing characterisation on NEOSSAT using these images an attitude sequences in the next large step in this research to raising the TRL level to operational level. The analysis of NEOSSAT should look at determining; the optical properties with the given attitude, attitude with estimate optical properties, and both optical properties and attitude given only shape priory information. The results from this analysis will allow the confirmation that RSO characterisation is possible with real data and give initial accuracy results for what can be expected. Comparing this result to currently methods proposed in literature such as [31], [23] , and [52], will help put the accuracy in context and find areas of improvement for RSO light curve inversion algorithms. Comparing the simulated vs real results will also help determine areas where the OBRITALS simulator can be improved. Some limitations of this technology have also been identified with work in the future looking to be overcome these limitations or, when not possible, to find other solutions to augment the currently proposed solution.

The first limitation comes from the length of light curve/ RSO detection. The length of a light curve and how frequently it is sampled sets limits on the bodies rate range and the accuracy that can be found from traditional spin rate analysis and glint analysis of the object. The novel method proposed in this dissertation does not limit the spin rate to being less than the light curve duration which can be seen with traditional spin rate analysis and glint analysis methods. Rather the duration of the light curve effects the accuracy of the estimated attitude and therefore the spin rate of the target RSO. The longer the detection and the more frequent the observations, the more accurately the spin rate can be extracted on the target object. Future work looking at this limitation should aim to test many different light curve durations and observation frequency to see how the accuracy in attitude and other property estimation is related to the length and density of the light curve for different facet models of RSO’s.

Another limitation that was found during this research is it is assumed that the light curve from the target RSO is fully calibrated. Currently the implementation of the light curve inversion is set up is using the absolute magnitude and change in absolute magnitude. Un-calibrated light curves use instrumental magnitude not absolute magnitude, using the instrumental magnitude in place of the absolute magnitude there is a bias introduced. These bias skews the response surface generated for optimisation leading to the convergence to sub-optimum estimates of the attitude, size, and optical properties. To avoid this limitation future work should look to refine the optimisation method to only use the change in magnitude across the light curve as this would be the same for both instrumental and absolute magnitude.

One other limitation to consider is the hardware limitations of the system that the light curve inversion method is running on. As the light curve inversion technique generates thousands of light curves with estimated parameters it can be very computationally intensive. Having a system with limited RAM or CPU capabilities it still can perform the light curve inversion but will take longer for each iteration. This increase in computation time can be offset by reducing the number of estimated light curves, making a trade off between accuracy and computation time which is common in optimisation problems.

The ORBITALS simulator was originally a biproduct of not having enough labelled data but has become one of the largest contributions of the thesis work. The update from the original simulator, developed by Sam Clemens [37], has addressed many of the original limitations including upgrading to parallel processing, improved RSO modeling, and modular functions to allow for easy updates. ORBITALS still has areas that can be improved upon to accomplish its goal of generating simulated space based RSO detections for the training of RSO detection, identification, and characterisation algorithms. I have identified 3 main areas in which ORBITALS can be improved on including implementing a GUI to make it more user friendly, include standard data formats for smooth automation, and improving modules, such as PSF, noise modeling, and BRDF. ORBITALS current implementation requires the user to call the functions from the command line, this more than sufficient for the use in this thesis. As more researchers use ORBITALS to generate data implementing a GUI would; reduce the amount of training researchers require and reduce the accessibility barriers to allow more researchers to use ORBITALS. Currently ORBITALS has a few standard data formats for how the information is inputted into the simulator, preparation of full ICD, as well as, updating the format to match the space standard should be done to keep ORBITALS relevant in the ever change space sector. One good example of this is update from TLE information to the new OMM format should be performed to allow for both historical TLE data and new OMM data to be acceptable inputs. Lastly improvement of the different modules that ORBITALS uses will improve its over all functionality and allow it to be tailored to specific applications. The PSF is one prime example of this, currently the PSF is implemented as a homogeneous 2D gaussian function. For the current application of ORBITALS, the 2D gaussian this is more than sufficient to model the point spread function, as ORBITALS is moved to model different types and resolution of imagers this will need to be updated to properly simulate the image. Another module that greatly impacts the performance of the simulator is the orbital propagator which if upgraded to higher fidelity model will increase the accuracy of RSO placement.

The implementation of PSGP4 algorithm in place of the standard SGP4 algorithm had a major improvement in computation speed with out a reduction of accuracy. There is still room to improve on the PSGP4 algorithm, as well as implement parallel processing and more complex propagation algorithms to allow for higher fidelity propagation. RSO propagation is almost always a trade off between computation time and accuracy, which means by reducing the computation time of the propagation algorithm more accurate answer can be found. One good example of this is in the space sector is for conjunction warnings where RSO are first propagated by lower fidelity algorithm and if any RSO’s get to close together they go for further conjunction analysis. Commonly for the second propagation a higher fidelity propagation method is used commonly Monte Carlo or HPOP is used. HPOP does not represent one algorithm but rather a sub section of high-fidelity algorithms that different companies implement differently for the needed applications. The benefits from the implementation a HPOP with full parallel processing capability has not been shown in literature yet, but it has the potential to greatly increase RSO propagation technology. One emerging technology that has great potential for improved performance of parallel processing is dynamic allocation. Dynamic allocation allows for a system to utilise hardware resources more optimally by allocating only the required resources allowing the free resources to be used for other process. This has great synergy with PSGP4, the integration of dynamic allocation would allow for the different time steps to run in parallel if there is enough RAM. This will reduce the overall computation time of the system, as propagators are a trade off between speed and accuracy this would allow for more accurate propagation methods to be implemented with little to no increase in computation time.

Wrapping the technologies in this thesis together is a future goal that would provide an end-to-end solution for RSO detection, identification, and characterisation from passive space based low resolution images. In this thesis the work is mostly focused on the post processing of the detected RSO’s, to provide an end-to-end solution some on in-situ processing is required to; determine if there is an RSO in the image, compress the image with RSO to only the needed information, and downlink the information to be analysed on the ground. Currently these different areas of research are being looked at from different academic and industrial partners. On board RSO detection is an emerging field with different analytical and machine learning algorithms being designed to be implemented on microprocessors on board satellites. A group in the nanosatellite research lab at York is currently research this topic and has developed a machine learning algorithm to perform onboard processing from a series of 4 images. This is looking to be implemented on the RSONAR mission which will be mentioned in more details in the paragraph below. Another topic of research in the nano research lab and at Magellan Aerospace is the compression of these images for ground-based processing. Images require lots of data to store, most of which is not needed for RSO identification and characterisation. This is further limited by the downlink capability of the host satellite, in most cases the required downlink size for full frame images is not feasible without disrupting the satellites primary goals. By processing the images in-situ board and extracting the relevant information, in the form of a lit pixel list or other parameters, the amount of data that needs to be downlinked can be significant reduced. The development of an end-to-end system to detect RSO’s, compress the images with RSO’s, and downlink the data for ground analysis will greatly increase the TRL of this technology and prepare it for space demonstration.

Before a complete space-based demonstration can be performed with this technology a ground or low altitude demonstration should be performed. This will be accomplished with the upcoming RSONAR mission at York university funded from CSA’s FAST grant. RSONAR mission plans to put a low-resolution imager, selected by York, to act as a dual-purpose star tracker on a high-altitude balloon. The dual purpose refers to the star tracker performing its job of boresight estimation, as well as RSO detection if an RSO is present in the image. The goal of this mission is to show that star tracker like cameras can passively detect and save images containing RSO’s with out impacting the cameras’ primary goal. This mission which is planned to launch in 2022 will greatly improve the TRL of passive low resolution RSO detections, if successful the next step would be looking to implement passive RSO detections on board a current active satellite.

6.3 Contribution

There are a few technological gaps and solutions that were discussed in this thesis, in this section my contribution to all of these areas and their importance is outlined below.

The largest and most important contribution is the work that was performed on RSO characterisation from passive space based low resolution imagers. In this work a novel study was performed to show the feasibility and expected accuracy of LEO-to-LEO passive low-resolution space based RSO detections and characterisation. Similar studies have been performed on dedicated LEO to LEO, terrestrial to LEO, terrestrial to GEO, and GEO to GEO RSO detections but this is the first study that looks at passive LEO to LEO for RSO characterisation. Studies that do focus on low resolution light curve analysis commonly focus on spin rate analysis instead of characterisation from full light curve inversion. While light curve inversion techniques are performed for higher resolution images this is one of the first time it is applied to space based low resolution images. This was taken further by a comparative study of the performance of optimisation algorithms for the estimation of optimal attitude and optical properties of an RSO. This contribution of demonstrating the feasibility and giving expected accuracy of RSO characterisation from passive space based low resolution demonstrates the benefit these sensors can be to SSA efforts. With the growing need for better SSA augmenting current and future satellites to be passive RSO detectors can be a cost-effective solution. This will help improve the accuracy and frequency of satellite catalog updates which reduces the ever-growing risk of in orbit collisions. Characterising RSO will help determine what type of object they are (active satellite, de-active satellite, rocket body, etc.), if they are adhering to current regulations, and assist in giving priory information for active debris removal. With the recent proliferation of RSO’s this contribution is coming at a time where innovation in the field `is critical for the protection of current and future space assets.

The second most impactful contribution is the development of the ORBITALS simulator. This software allows for a user to replicate space based optical detections of RSO’s with different RSO models. This had two major impacts; the first was it allowed for the replication of image sequences of RSO detections using different RSO models. From the image sequences the light curve was extracted for different RSO models. This enabled the RSO characterisation to be performed by simulating the light curves for different models of RSO’s and comparing them to the real light curve. The second impact of this was it enabled the generation of data sets in a field where there are not many publicly available data sets. One of the biggest challenges I faced in this thesis was not have access to ladled sequences of images from space-based sensors. This made it impossible in some cases to characterise the RSO due to lacking host satellite and sensor parameters. Many other doing similar research in RSO detection, identification, and characterisation from space-based sensors were having the same issue. The development of the ORBITALS simulator over came that by providing labeled data from a controlled environment that could generate large data sets quickly. The ORBITALS simulator has already been used to generate data sets for this thesis, other researchers in the Nano Satellite laboratory, as well as different industry partners. These data sets have been used to; help train and compare the performance of machine learning RSO detection algorithms, test the feasibility of angles only orbital estimation from space-based platforms, as well as test and verify RSO identification and classification algorithms. The development of ORBITALS already has had an impact in the Canadian space sector, and it still has room for updates and optimisation.

The third most important contribution from this thesis was the development of PSGP4 propagator. Leveraging SIMD parallel processing with the SGP4 algorithm made huge improvement in the computation time required for propagation. This was leveraged in the ORBITALS simulator to allow for compete satellite catalog propagation while still take less than a second to generate one image. PSGP4 will drastically reduce the computation time for SSA tasks, such as conjunction warnings, access finding, and satellite identification. As more and more RSO are cataloged in orbit more of them will require propagation, increasing the required computation time and resources. PSGP4 looks to reduce the computation burden from the increasing catalog by making more efficent use of current hardware available.

7 Appendixes

A. Accuracy Analysis of Simulated Images Using ORBITALS and STK EOIR

Objects	**Orbitals**		**STK-EOIR**		**Orbitals Error** (X ,Y)*	**STK Error** (X ,Y)*
	Row(X)*	Col(Y)*	Row(X)*	Col(Y)*		
2019-06-15 23:35:50						
Ref 6111	143.0	151.2	143.6	151.0	(1.1, -8.6)	(1.7, -8.8)
Ref 6158	121.6	161.3	120.6	160.7	(-4.4, -5)	(-5.4, -5.6)
Ref 6195	101.1	170.2	100	170.0	(-1.5, -4.8)	(-2.6, -5)
RCM-1	22.6	155.0	13.7	152.6	(3.8, 5.1)	(-5.1, 2.7)
RCM-2	35.5	147.8	30.0	144.6	(1.3, 4.8)	(-4.2, 1.6)
RCM-3	49.9	139.6	45.4	136.6	(-0.3, 4)	(-4.8, 1)
2019-06-15 23:35:55						
Ref 6111	143.8	152.3	144.5	150.8	(1.3, -6.7)	(2, -8.2)
Ref 6158	122.5	162.5	122.6	161.6	(-4.2, -3.5)	(-4.1, -4.4)
Ref 6195	102.0	171.5	101.7	171.7	(-0.6, -3.5)	(-0.9, -3.3)
RCM-1	55.5	139.5	50.0	140.6	(1.1, 4.5)	(3.9, 5.6)
RCM-2	-	-	65.6	131.6	-	(2.9, 4.2)
RCM-3	82.9	123.9	79.6	123.0	(-1.6, 3.7)	(4, 2.8)
2019-06-15 23:36:00						
Ref 6111	144.7	153.5	144.7	153.7	(2.5, -5.6)	(2.5, -5.4)
Ref 6158	123.3	163.7	122.0	163.4	(-3, -2.4)	(-4.3, -2.7)
Ref 6195	102.9	171.8	100.6	172.1	(0.2, -3.4)	(-2.1, -3.1)
RCM-1	88.7	123.7	85.0	121.7	(1.6, 3.9)	(2.1, 1.9)
RCM-2	101.5	116.4	100.6	113.6	(-1, 3.7)	(8.1, 1)
RCM-3	115.7	108.6	114.6	106.0	(-1.5, 3.4)	(-2.6, 0.8)
2019-06-15 23:36:05						
Ref 6111	144.2	151.8	144.0	151.6	(1.7, -6.2)	(1.5, -6.4)
Ref 6158	122.9	162.2	122.0	162.6	(-4.3, -3.1)	(-5.2, -2.7)
Ref 6195	102.6	171.4	101.8	171.7	(-0.7, -3.8)	(-1.5, -3.5)
RCM-1	120.3	105.1	116.0	104.8	(-0.4, 1.7)	(-4.7, 1.3)
RCM-2	132.7	97.8	130.6	96.0	(-2.5, 1.4)	(-4.6, 0.4)
RCM-3	146.5	89.6	144.0	87.7	(-3.6, 1.9)	(-6.1, 0)
2019-06-15 23:36:10						
Ref 6111	143.1	148.6	143.6	148.4	(0.2, -9.2)	(0.7, -9.4)
Ref 6158	121.9	159.2	121.6	159.6	(-4.1, -6.6)	(-4.4, -6.2)
Ref 6195	101.7	168.6	100.8	169.6	(-1.2, -6.4)	(-2.1, -5.4)
RCM-1	150.3	84.8	147.7	83.6	(-2.8, -2.4)	(-5.4, -3.6)
RCM-2	161.2	77.6	161.6	74.7	(-6.6, -2.9)	(-6.2, -5.9)
RCM-3	175.3	69.5	174.6	65.4	(-6.5, -3.3)	(-7.2, -7.4)
2019-06-15 23:36:15						
Ref 6111	142.0	145.3	142.0	145.4	(-0.4, -11.8)	(-0.4, -11.7)

Ref 6158	120.9	156.2	119.7	157.3	(-6.6, -10.2)	(-7.8, -9.1)
Ref 6195	100.8	165.8	98.6	167.3	(-2.1, -9.4)	(-4.3, -7.9)
RCM-1	179.0	64.5	178.6	62.4	(-7.1, -7.1)	(-7.5, -9.2)
RCM-2	190.4	57.5	192.3	53.6	(-7.1, -6.1)	(-5.2, -10)
RCM-3	202.6	49.6	205.0	45.5	(-11.6, -7.3)	(-9.2, -11.5)

* Values are given in pixels.

B. References

[1] ESA, "Space Debris by the Numbers," ESA, 01 01 2019 . [Online]. Available: https://www.esa.int/Our_Activities/Operations/Space_Safety_Security/Space_Debris/Space_debris_by_the_numbers. [Accessed 26 03 2019].

[2] I. Nikolova , "Micro-Satellite Advantages. Profability and Return," in *SPACE, ECOLOGY, SAFETY*, Varna, 2005.

[3] K. Woellert, P. Ehrenfreund, A. J. Ricco and H. Hertzfeld, "Cubesats: Cost-effective science and technology platforms for emerging and developing nations," *Advances in Space Reseach,* vol. 47, no. 4, pp. 663-684, 2011.

[4] J. Esper, S. Neeck, J. A. Slavin, J. Leitner, W. Wiscombe and F. H. Bauer, "Nano/Micro Satellite Constellations for Earth and Space Science," *Acta Astronautica*, vol. 59, no. 9-12, pp. 785-792, 2003.

[5] BIS Research, "Global CubeSat Market: Focus on Sizes (1U, 2U, 3U, 6U, and Other Sizes),Subsystems, and End Users (Academic, Commercial, Government, Defense,and Non-Profit Organization) - Analysis and Forecast: 2018-2022," BIS Research, 2018.

[6] H. G. Lewis, J. Radtke, A. Rossi, J. Beck, M. Oswald, P. Anderson, B. Bastida Virgili and H. Krag, "Sensitivity of the Space Debris Enviroment to Large Constellations and Small Satellites," in *European Conference on Space Debris* , 2017.

[7] M. K. Ben Larbi, B. Grzesik, J. Radtke, C. Trentlage and E. Stoll, "Active Debris Removal for Mega Constellations: CubeSat Possible?," in *9th International Workshop on Satellite Constellations and Formation Flying*, Boulder, Colorado, 2017.

[8] R. G. Harrison, "Unpacking the Three C's: Congested, Competitive, Contested Space," *The International Journal of Space Politics & Policy,* pp. 123-131, 2013.

[9] S. Gao, K. Clark, M. Unwin, J. Zackrisson, W. A. Shiroma, J. M. Akagi, K. Maynard, P. Garner, L. Boccia, G. Amendola, G. Massa, C. Underwood, M. Brenchley, M. Pointer and M. N. Sweeting, "Antennas for Modern Small Satellites," *IEEE Antennas and Propagation Magazine,* vol. 51, no. 4, pp. 40-56, 2009.

[10] E. Kulu, "What is a CubeSat & other picosatellites," Nanosats Database, 2019. [Online]. Available: https://www.nanosats.eu/cubesat. [Accessed 27 03 2019].

[11] California Polytechnic State University , "CubeSat Design Specifications," 20 2 14. [Online]. Available: https://static1.squarespace.com/static/5418c831e4b0fa4ecac1bacd/t/56e9b62337013b6c063a655a/1458157095454/cds_rev13_final2.pdf. [Accessed 27 03 2019].

[12] H. J. Kramer , "Flock 1," ESA, 2019. [Online]. Available: https://directory.eoportal.org/web/eoportal/satellite-missions/f/flock-1#mission-status. [Accessed 27 03 2019].

[13] D. J. Kessler, N. L. Johnson, J. C. Liou and M. Matney, "The Kessler Syndrome: Implications to Future Space operations," *Advances in the Astronautical Sciences,* vol. 137, no. 8, p. 2010, 2010.

[14] M. R. Ackermann, R. R. Kiziah, P. C. Zimmer, J. T. McGraz and D. D. Cox, "A Systematic Examination of Ground-Based and Space-Based Approaches to Optical Detection and Tracking of Satellites," in *31st Space Symposium, Technical Track*, Albuquerque, New Mexico, 2015.

[15] D. L. Pechkis, N. S. Pacheco and T. W. Botting, "Statistical Approach to the Operational Testing of Space Fence," *IEEE Areospace and Electronic Systems Magazine,* vol. 31, no. 11, pp. 30-39, 2016.

[16] P. Maskell and L. Oram, "Sapphire: Canada's Answer to Space-Based Surveillance of Orbital Objects," in *The Advanced Maui Optical and Space Surveillance Technologies Conference*, 2008.

[17] L. Scott, B. Wallace, M. Sale, M. Levesque and S. Thorsteinson, "Toward Microsatellite Based Space Situational Awareness," in *The Advanced Maui Optical and Space Surveillance Technologies Conference* , 2013.

[18] ESA, "About Space Debris," ESA, 21 2 2018. [Online]. Available: http://www.esa.int/Our_Activities/Operations/Space_Safety_Security/Space_Debris/About_space_debris. [Accessed 27 03 2019].

[19] National Research Council, Limiting future collision risk to spacecraft: an assessment of NASA's meteriod and orbital debris programs., National Academies Press, 2011.

[20] G. A. McCue, J. G. Williams and J. M. Morford, "Optical Characteristics of Artificial Satellites," *Planet. Space Sci,* vol. 19, pp. 851-868, 1971 .

[21] N. L. Eisner, C. J. Lintott and S. Aigrain, "LATTE: Lightcurve Analysis Tool for Transiting Exoplanets," *The Journal of Open Source Software,* vol. 5, no. 49, p. 2101, 2020.

[22] M. Kaasalainen, T. Johanna and M. Karri, "Optimization Methods for Asteriod Lightcurve Inversion: II. The complete inverse problem," *Icarus,* vol. 153, no. 1, pp. 37-51, 2001.

[23] R. Linares, J. L. Crassidis and M. K. Jah, "Particle Filtering Light Curve Based Attitude Estimation for Non-Resolved Space Objects," in *24th AAS/AIAA Space Flight Mechanics Meeting, 2014*, 2014.

[24] N. Koshkin, S. Strakhova and S. Leonid, "Remote Sensing of the EnviSat and Cbers-2B satellites rotation around the centre of mass by photomerty," *Advances in Space Research,* vol. 58, no. 3, pp. 358-371, 2016.

[25] C. Fruh, "Identification of Space Debris," in *Proceeding of the international Astronautical Congress* , 2011.

[26] A. Friedman, S. Fan, C. Frueh and T. Schildknecht, "Observability of Light Curve Shape Inversion Based on Optical Data," in *First International Orbital Debris Conference* , 2019.

[27] 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.

[28] R. Clark, S. Dave, J. Wawrow and R. Lee, "Performance of Parameterization Algorithms for Resident Space Object (RSO) Attitude Estimates," *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies,* vol. 1, no. 1, pp. 15-18, 2020.

[29] M. D. Hejduk, "Specular and Diffuse Components in Spherical Satellite Photometric Modeling," *Hejduk, M. D. (2011, September). Specular and diffuse components in spherProceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference* , vol. 1, no. 1, pp. 1-11, 2011.

[30] M. Ashikhmin and P. Shirley, "An Anisotropic Phong Light Reflection Model," *Journal of Graphic Tools*, vol. 5, no. 2, pp. 25-32, 2000.

[31] B. K. Bradley and P. Axelrad, "Lightcurve Inversion For Shape Estimation of GEO Objects from Space-Based Sensors," in *Univ. of Colorado. International Space Symposium for Flight Dynamics*, 2014.

[32] F. Siwei, A. Friedman and C. Frueh, "Satellite Shape Recovery from Light Curves with Noise," in *Advanced Maui Optical and Space Surveillance Technologies* , Maui , 2019.

[33] C. Bonnal, J.-M. Ruault and M.-C. Desjean, "Active debris removal: Recent progress and current trends," *Acta Astronautica,* vol. 85, pp. 51-60, 2013.

[34] F. Schiemenz, J. Utzmann and H. Kayal, "Survey of the operational state of the art in conjunction analysis," *CEAS Space Journal,* vol. 11, no. 3, pp. 255-268, 2019.

[35] T. Maclay and D. Mcknight, "Space environment management: Framing the objective and setting priorities for controlling orbital debris risk," *Journal of Space Safety and Engineering,* vol. 8, no. 1, pp. 93-97, 2021.

[36] A. Rossi, "Population models of space debris," *Proceedings of the International Astronomical Union* , pp. 427-438, 2004.

[37] S. Clemens, "On-Orbit Resident Space Object (RSO) Detection Using Comerical Grade Star Trackers," 2019.

[38] D. C. Pak, "Linearized equations for J2 perturbed motion relative to an elliptical orbit," Sn Jose State University, 2005.

[39] H. D. Curtis, Orbital Mechanics for Engineering Students, 2014.

[40] Lasunncty, "Orbital Elements Wikipedia," 19 02 2019. [Online]. Available: https://commons.wikimedia.org/w/index.php?curid=8971052. [Accessed 28 03 2019].

[41] M. I. Ross, "Linearized Dynamic Equations for Spacecraft Subject to J2 Perturbations," *J Guidance,* vol. 26, no. 4, pp. 657-659, 2003.

[42] B. K. Bradley, "Numerical Algorithms for Precise and Efficient Orbit Propagation and Positioning," University of Colorado at Boulder, 2015.

[43] D. A. Vallado and P. J. Cefola, "Two-Line Element Sets - Practise and Use," in *63rd International Astronautical Congress*, Naples, Italy , 2012.

[44] The Consultative Committee for Space Data Systems , "CCSDS RECOMMENDED STANDARD FOR ORBIT DATA MESSAGES," Consultative Committee for Space Data Systems , 2009.

[45] D. A. Vallado and P. Crawford, "SGP4 Orbit Determination," in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit* , 2008.

[46] D. A. Vallado, P. Crawford, R. Hujask and T. S. Kelso, "Revisiting Spacetrack Report #3: Rev 2," in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2006.

[47] F. R. Hoots and R. L. Roehrich, "SpaceTrack Report No. 3," Project Spacetrack Reports, Office of Astrodynamics, Areospace Defence Center, 1980.

[48] R. Clark and R. Lee, "Parallel Processing for Orbital Maneuver Detection.," *Advances in Space Research,* vol. 66, no. 2, pp. 444-449, 2020.

[49] S. Clemens, R. Lee, P. Harrison and W. Soh, "Feasibility of Using Commercial Star Trackers for On-Orbit Resident Space Object Detection," in *Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)* , 2018.

[50] S. Dave, R. Clark, G. Chianelli and R. Lee, "Machine Learning Implementation for In-Orbit RSO Orbit Estimation Using Star Tracker Cameras," in *Advanced Maui Optical and Space Surveillance Technologies Conference*, Maui, 2020.

[51] A. D. Dianetti and J. L. Crassidis, "Light Curve Analysis Using Wavelets," in *AIAA Guidance Navigation, and Control Conference*, 2016.

[52] Y. Matsushita, R. Arakawa, Y. Yoshimura and T. Hanada, "Light Curve Analysis and Attitude Estimation of Space Objects Focusing on Glint," *LPI Contributions 2109,* 2019.

[53] J. C. Hinks and J. L. Crassidis, "Angular Velocity Bounds via Light Curve Glint Duration," in *AIAA Guidance, Navigation, and Control Conference* , 2016.

[54] Z. Kopal, "Fourier Analysis of the Light Curves of Eclipsing Variables, V," *Astrophysics and Space Science,* vol. 38, no. 1, pp. 191-241, 1975.

[55] P. Martinez and C. Koen, "Period searching by least squares in sparsely sampled light curves with non-sinusoidal oscillations," *Monthly Notices of the Royal Astronomical Society* , vol. 267, no. 4, pp. 1039-1044, 1994.

[56] R. F. Stellingwerf, "Period determination using phase dispersion minimization," *The Astrophysical Journal* , vol. 224, pp. 953-960, 1978.

[57] J.-C. Lu, J.-Y. Wang, T. An, J.-M. Lin and H.-B. Qiu, "Periodic radio variability in NRAO 530: phase dispersion," *Research in Astronomy and Astrophysics,* vol. 12, no. 6, p. 643, 2012.

[58] E. Schafer, "Stereoscopic Light Curve Analysis of Space Debris Objects," *Diss. Julius Maximilians Universität Würzburg, Luleå Tekniska Universitet,* 2017.

[59] W. E. Krag, "Visible Magnitude of Typical Satellites in Synchronous Orbits," in *Massachusetts inst of tech lexington Lincoln Lab*, 1974.

[60] F. Gasdia, "Optical Tracking and Spectral Characterization of Cubesats for Operational Missions," 2016.

[61] T. Kelecy, D. Hall, K. Hamada and D. Stocker, "Satellite Maneuver Detection using Two-line Element (TLE) Data," in *Advanced Maui Optical and Space Surveillance Technologies (AMOS) Conference*, 2007.

[62] R. P. Patera, "Space Event Detection Method," *Journal of Spacecrafts and Rockets,* vol. 45, no. 3, pp. 554-559, 2008.

[63] S. Lemmens and H. Krag, "Two-Line-Elements-Based Maneuver Detection Methods for Satellites in Low Earth Orbit," *Journal of Guidance, Control and Dynamics,* vol. 37, no. 3, pp. 860-868, 2014.

[64] R. P. Patera, "Trajectory Estimation Based on Radar Data," *Aerospace ForumonSpaceDebris, CollisionAvoidance, andReentry Hazards,* 2000.

[65] J. H. Seago and D. A. Vallado, "Coordinate Frames of the U.S. Space Object Catalogs," in *Astrodynamics Specialist Conference* , 2000.

[66] A. I. Zakharov, M. E. Prokhorov, M. S. Tuchin and A. O. Zhukov, "Minimum Star Tracker Specification Required to Achieve a Given Attitude Accuracy," *Astrophysical Bulletin,* vol. 68, no. 4, pp. 481-493, 2013.

[67] V. S. Lobanov, N. V. Tarasenko and V. N. Zboroshenko, "Orientation and Stabilization Systems of Space Vehicles for Different Purposes: Lones of Development," *Gyroscopy and Navigation* , vol. 7, pp. 50-57, 2016.

[68] F. Curti, D. Spiller, V. Schiattarella and R. Orsi, "Recognition of Orbiting-Objects Through Optical Measurements of Light-Reflecting-Targets by Using Star Sensors," in *IAA Conference on Space Situational Awareness*, 2017.

[69] D. O. Fulcoy, K. I. Kalamaroff and F. K. Chen, "Determing Basic Satelltie Shape from Photometric Light Curves," *Journal of Spacecraf and Rockets* , vol. 49, no. 1, 2012.

[70] S. AL, I. Karagoz and A. Dogan, "An Improved Star Detection Algorithm Using a Combination of Statistical and Morphological Image Processing Techniques," in *2018 IEEE Applied Imagery Pattern Recognition Workshop*, 2018.

[71] P. W. Kervin, D. Hall, M. Bolden and J. Toth, "Phase Angle: What is it Good For?," in *Advanced Maui Optical Space and Serveillance Technologies Conference*, Maui, 2020.

[72] D. Hoffleit and W. H. Warren Jr., "Astronomical Data Center Bulletin 1.4," 1987.

[73] S. L. Kenyon and J. W. V. Storey, "A Review of Optical Sky Brightness and Extinction at Dome C, Antarctica," *Publications of the Astronomical Society of the Pacific,* vol. 118, no. 841, p. 489, 2006.

[74] M. Anastasiou, J. H. Seiradakis, J. Evans, S. Drougou and E. Kyriakos, "Position astronomy and the Parapegma of the Antikythera Mechanism," *From Antikythera to the Square Kilometre Array: Lessons from the Ancients*, 2012.

[75] J. R. Shell, "Optimizing Orbital Debris Monitoring with Optical Telescope," AIR FORCE SPACE INNOVATION AND DEVELOPMENT CENTER SCHRIEVER AFB, 2010.

[76] M. Caligaris, G. Rodriguez and L. Laugero, "Designing Tools for Numerical Integration," *Procedia-Social and Behavioral Sciences* , vol. 176, pp. 270-275, 2015.

[77] A. Rocha, "Numerical Methods and Tolerance Analysis for Orbit Propagation," San José State University, 2018.

[78] J. M. Aristoff and A. B. Poore, "Implicit Runge-Kutta Methods for Orbit Propagation," in *AIAA/AAS Astrodynamics Specialist Conference* , 2012.

[79] J. M. Aristoff, J. T. Horwood and A. B. Poore, "Orbit and Uncertainty Propagation: A Comparison of Gauss-Legendre, Dormand-Prince and Chebyshev-Picard Based Approaches," *Celestial Mechanics and Dynamical Astronomy,* vol. 118, no. 1, pp. 19-28, 2013.

[80] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, Numerical Recipies The Art of Scientific Computing, Cambridge university press: Cambridge, 2007.

[81] J. D. Little, K. G. Murty, D. W. Sweeney and C. Karel, "An algoirthm for the Traveling Salesman Problem," *Operations Research* , vol. 11, no. 6, pp. 972-989, 1963.

[82] A. Globus, J. Crawford, J. Lohn and A. Pryor, "Scheduling Earth Observing Satellites with Evolutionary Algorithms," 2003.

[83] S. Baluja, "An Empirical Comparison of Seven Iterative and Evolutionary Heuristics for Static Function Optimization," in *Eleventh International Conference on Systems Engineering* , 1996.

[84] D. S. Johnson and L. A. McGeoch, "The Traveling Salesman Problem: A Case Study in Local Optimization," *Local search in combinatorial optimization,* vol. 1, no. 1, pp. 215-310, 1997.

[85] L. Ansalone, "A Search Algorithm for Stochastic Optimisation in Initial Orbit Determination," *PhD Thesis, Univerity of Rome,* 2014.

[86] A. Corana, M. Marchesi, C. Martini and S. Ridella, "Minimizing Multimodal Functions of Continous Variables with the "Simulated Annealing" Algorithm," *ACM Transactions on Mathmatical Software,* vol. 13, no. 3, pp. 262-280, 1987.

[87] Y. Nourani and B. Andresen, "A comparison of simulated annealing cooling strategies," *Journal of Physics A: Mathematical and General,* vol. 31, no. 41, p. 8373, 1998.

[88] D. Weld, "CSE 473: Artificial Intelligence Power Point," 2016.

[89] W. J. Wolfe and S. E. Sorensen, "Comparison of three scheduling algorithms," *SPIE,* vol. 3203, pp. 172-181, 1997.

[90] W. J. Wolfe and E. S. Sorensen, "Three Scheduling Algorithms Applied to Earth Observing System Domain," *Management Science* , 2000.

[91] X. Zhai, X. Niu, H. Tang, L. Wu and Y. Shen, "Robust Satellite Scheduling Approach for Dynamic Emergency Tasks," *Mathematical Problems in Engineering* , 2015.

[92] R. Cheng, M. Gen and Y. Tsujimura, "A tutorial survey of job-shop scheduling problems using genetic algorithms, part II: hybrid genetic search strategies," *Computers & Industrial Engineering,* vol. 30, no. 4, pp. 983-997, 1996.

[93] T. Mok, H. Liu, Y. Ni, F. F. Wu and H. Ron, "Tuning the fuzzy damping controller for UPFC through genertic algorithm with comparison to the gradient descent training," *International Journal of Electrical Power & Energy Systems,* vol. 27, no. 4, pp. 275-283, 2005.

[94] Y. Shi, M. Mizumoto, N. Yubazaki and M. Otani, "A learning algorithm for tuning fuzzy rules based on the gradient descent method," *Proceedings of IEEE 5th International Fuzzy Systems,* vol. 1, pp. 55-61, 1996.

[95] F. Piltan, B. Boroomand, A. Jahed and H. Rezaie, "Performance-Based Adaptive Gradient Descent Optimal Coefficient Fuzzy Sliding Mode Methodology," *International Journal of Intelligent Systems and Applications,* vol. 4, no. 1, p. 40, 2012.

[96] S. Pesme, A. Dieuleveunt and N. Flammarion, "On convergence-diagnostic based step sizes for stochastic gradient descent," in *International Conference on Machine Learning*, 2020.

[97] M. Abido, "Simulated annealing based approach to PSS and FACTS based stabilizer tuning," *International Journal of Electrical Power and Energy Systems,* vol. 22, no. 4, pp. 247-258, 2000.

[98] T. A. Yanar and A. Zuhal, "Fuzzy model tuning using simulated annealing," *Expert Systems with Application,* vol. 38, no. 7, pp. 8159-8169, 2011.

[99] R. Bellio, S. Ceschia, L. Di Gaspero, A. Schaerf and T. Urli, "Feature-based tuning of simulated annealing applied to the curriculum-based course timetabling problem," *2016,* vol. 65, pp. 83-92, Computers and Operations Research.

[100] A. Varsek, T. Urbancic and B. Filipic, "Comparison of PID controller tuning methods with genetic algorithm for FOPTD system," *Genetic algorithms in controller design and tuning* , vol. 23, no. 5, pp. 1330-1339, 1993.

[101] T. L. Seng, M. B. Khalid and R. Yusof, "Tuning of a neuro-fuzzy controller by genetic algorithm," *IEEE Transactions on Systems, Man, and Cybernetics Part B,* vol. 29, no. 2, pp. 226-236, 1999.

[102] M. E. H. Pedersen and A. J. Chipperfield, "Simplifying particle swarm optimization," *Applied Soft Computing,* vol. 10, no. 2, pp. 618-628, 2010.

[103] T.-H. Kim, I. Maruta and T. Sugie, "Robust PID controller tuning based on the constrained particle swarm optimisation," *Automatica,* vol. 44, no. 4, pp. 1104-1110, 2008.

[104] L. K. Newman, R. Frigm and D. McKinley , "It's Not A Big Sky After All: Justification for a Close Approach Prediction and Risk Assessment Process," in *AAS/AIAA Astrodynamics Specialist Conference*, 2009.

[105] J. A. Haimerl and G. P. Fonder, "Space Fence System Overview," in *AMAdvanced Maui Optical and Space Serveillance Technology Conference,* Maui, 2014.

[106] B. Neta, D. A. Danielson, S. Ostrom and S. K. Brewer, "Preformance of Analytic Orbit Propagators on a Hypercube and a Workstation Cluster," in *Astrodyanmics Conference,* 1994.

[107] N. Beny, "Parallel Version of Special Perturbations Orbit Propagator," NAVAL POSTGRADUATE SCHOOL MONTEREY CA DEPT OF MATHEMATICS, 1998.

[108] M. Lin, M. Xu and X. Fu, "A Parallel Algorithm for the Initial Screening of Space Debris Collisions Prediction Using the SGP4/SDP4 Models and GPU Acceleration," *Advances in Space Research,* vol. 59, no. 9, pp. 2398-2406, 2017.

[109] F. J. San-Juan, I. Perez, M. San-Martin and E. P. Vergara, "Hybrid SGP4 Orbit Propagator," *Acta Astronautica,* vol. 137, pp. 254-260, 2017.

[110] D. Koblick, M. Poole and P. Shankar, "Parallel High-Precision Orbit Propagation Using the Modified Picard-Chebyshec Method," in *ASME 2012*, 2012.

[111] M. Shen, X. Guo, P. Gao, D. Yang, Y. Zhao, Z. Fan, J. Yu and Y. Ma, "Application of Parallel Computing for Space Debris Close Approach Analysis," in *International Conference on Computer Science and Service System*, 2012.

[112] W. Zhang, Z. Jiang, H. Zhang and J. Luo, "Optical Image Simulation System for Space Surveillance," in *Seventh International Conference on Image and Graphics*, 2013.

[113] S. P. Haherty and H. Benton Ellis Jr., "A High Fidelity Approach to Data Simulation for Space Situational Awareness Missions," in *Advanced Maui Optical and Space Surveillance Technologies Conference*, 2016 .

[114] L. Varela, L. Boucheron, N. Malone and N. Spurlock, "Streak detection in wide field of view images using convolutional neural networks (CNNs)," in *Advanced Maui Optical and Space Surveillance Technologies,* Maui, 2019.

[115] I. W. McQuaid, "Autonomous Association of GEO RSO Observations Using Deep Neural Networks," *AIR FORCE INSTITUTE OF TECHNOLOGY WRIGHT-PATTERSON AFB*, 2018.

[116] B. Liu, L. Yao and D. Han, "Harnessing ontology and machine learning for RSO classification," *SpringerPlus,* vol. 5.1, pp. 1-22, 2016.

[117] S. Hagerty and K. Ansari, "An Innovative Near-Real-Time Approach to Space SItualtional Awareness (SSA) Data Simulation," in *35th Space Symposium, Technical Track.*, Colorado Springs, 2019.

[118] L. Cogger, A. Howarth, A. Yau and A. White, "Fast Auroral Imager (FAI) for the e-POP Mission," *Space Science Review,* vol. 189, no. 1-4, pp. 15-25, 2014.

[119] B. Jia, K. D. Pham, E. Blasch, Z. Wang, D. Shen and G. Chen, "Space Object Classification using Deep Neural Networks," in *IEEE Aerospace Conference*, 2018.

[120] J. Fitzmaurice, D. Bedard, C. H. Lee and P. Seitzer, "Detection and correlation of geosynchronous objects in NASA's Wide-field Infrared Survey Explorer images," *Acta Astronautica*, vol. 183, pp. 176-198, 2021.

[121] D. Burandt, D. Hampf, J. Rodmann and W. Riede, "INTERPRETATION OF LIGHT CURVES BASED ON SIMULATION SOFTWARE," in *International Astronautical Congress (IAC)*, Adelaide, 2017.

[122] C. R. Singletary and F. K. Chen, "Simulating Complex Satellites and a Space-Based Surveillance Sensor Simulation," in *AMOSTech*, 2007.

[123] B. Lal , A. Balakrishnan, B. M. Caldwell, R. S. Buenconsejo and S. A. Carioscia, "Global Trends in Space Situational Awareness (SSA) and Space Traffic Management," IDA SCIENCE & TECHNOLOGY POLICY INSTITUTE, 2018.

[124] B. Wallace , R. Scott and A. Spaans, "The DRDC Ottawa Space Surveillance Observatory," in *Advanced Maui Optical and Space Surveillance Technologies Conference*, Maui, 2007.

[125] L. Scott, S. Thorsteinson, D. Bedard, B. Cotten and R. E. Zee, "Canadian Ground Based Optical Observations of the CANX-7 Sail Deployment," in *CASI/ASTRO*, 2018.

[126] University of Calgary , "e-POP data set website," [Online]. Available: https://epop-data.phys.ucalgary.ca/.

[127] W.-C. Lin and D.-Y. Liao, "A Tabu Search Algorithm for Satellite Image Scheduling," in *2004 IEEE International Conference on Systems, Man and Cybernetics*, 2004.

[128] S. Fan and C. Frueh, "A Direct Light Curve Inversion Scheme in the Presents of Measurement Noise," *The Journal of the Astronautical Sciences,* 2019.

[129] K. Subbarao and L. Henderson, "Obervability and sensitivity analysis of lightcurve measurement models for use in space situational awareness," *Inverse Problems in Science and Engineering,* vol. 27, no. 10, pp. 1399-1424, 2019.

[130] W. Baur, O. Romberg, W. Carsten and D. Gerhard, "Development of in-situ space debris detector," *Advances in Space Research,* vol. 54, no. 9, pp. 1858-1869, 2014.

[131] F. Gasdia, A. Barjatya and S. Bilardi, "Multi-Site Simultaneous Time-Resolved Photometry with a Low Cost Electro-Optics System," *Sensors,* vol. 17, no. 6, p. 1239, 2017.

[132] K. B. Bernander, "A Method for Detecting Resident Space Objects and Orbit Determination Based on Star Trackers and Image Analysis," Uppsala Universitet, Sweeden, 2014.

[133] T. Payne, A. Chaudhary, S. Gregory, J. Brown and M. Nosek, "Signature Intensity Derivative and its Application to Resident Space Object Typing," 2009.

[134] R. Lambour, R. Bergemann, C. von Braun and E. M. Gaposchkin , "Space-Based Visible Space Object Photometry: Initial Results," *JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS,* vol. 23, no. 1, 2000.

[135] N. B. Cowan and E. Agol, "Inverting phase functions to map exoplanets," *The astrophysical Journal Letters,* vol. 678, no. 2, p. L129, 2008.

[136] X. Guo, J. Tang, J. Li, C. Shen and J. Liu, "Attitude Measurement Based on Imaging Ray Tracking Model and Orthographic Projection with Iteration Algorithm," *ISA Transactions,* vol. 95, pp. 379-391, 2019.

[137] C. J. Wetterer and M. Jah, "Attitude Estimation form Light Curves," *Journal of Guidance, Control, and Dynamics* , vol. 32, no. 5, 2009.

[138] V. Pesce, M. F. Haydar, M. Lavagna and M. Lovera, "Comparison of FIltering Techniques for Relative Attitude Estimation of Uncooperative Space Objects," *Aerospace Science and Technology* , vol. 84, pp. 318-328, 2019.

[139] R. Cheng, "A Study on Genetic Algorithms-Based Optimal Scheduling Techniques," 1997.

[140] I. M. Oliver, D. J. Smith and J. R. Holland, " A Study of Permutation Crossover Operators on the Traveling Salesman Problem," in *International Conference on Genetic Algorithm*, 1987.

[141] E. Falkenauer and S. Bouffouix, "A Genetic Algorithm for Job Shop," in *IEEE International Conference on Robotics and Automation*, 1991.

[142] Germany, United States, UN. Commiottee on the Peaceful Uses of Outer Space, "National research on space debris, safety of space objects with nuclear power sources on board and problems relating to their collision with space debris," Vienna, 2003.

[143] D. E. Goldberg, GA in search, optimization, and machine learning, Boston: Kluwe r Academic Publishers, 1989.

[144] L. Davis, "Applying Adaptive Algorithms to Epistatic Domains," *IJCAL*, vol. 85, pp. 162-164, 1985.

[145] G. Syswerda, "Uniform Crossover in Genetic Algorithms," *ICGA*, vol. 3, pp. 2-9, 1989.

[146] T. Yamada and R. Nakano, "A Genetic Algorithm Applicable to Large-Scale Job-Shop Problems," *Parallel Problem Solving from Nature*, vol. 2, pp. 281-290, 1992.