MODELLING AND SIMULATION OF AUTONOMOUS CUBESATS FOR ORBITAL DEBRIS MITIGATION

Christopher Iliffe Sprague

Rensselaer Polytechnic Institute
Mechanical, Aerospace, and Nuclear Engineering
110 8th Street, Troy, New York 12180, USA

ABSTRACT

It is well known that orbital debris about Earth impose increasingly stringent restrictions on the operation and commissioning of both current and future space applications. These orbital debris, which are becoming ever so prevalent, can literally destroy a satellite. Even particles of diminutive stature can result in disastrous ramifications. Many of these endangered satellites, of which humans are reliant upon for supplying the infrastructure necessary to support modern life in the twenty-first century, routinely have to perform avoidance manoeuvres in response to ground data indicating that an object is on a trajectory that could pose a threat, negating away precious finite amounts fuel. These orbital debris are vexatious, with respect to not only a spacecraft's integrity, but also its lifespan. It is imperative that a solution be realised. This study aims to demonstrate the feasibility of utilising modular CubeSats to deorbit space debris. A large high fidelity simulation is constructed with the goal to simulate the CubeSat's orbital dynamics, as well as its autonomous functions. Such autonomous functions will manifest in the development of autonomous control algorithms to execute mitigation procedures, such as path planning and rendezvous, with the intent of substantiating their real-time implementability.

Index Terms— CubeSats, Orbital Debris, Simulation

1. INTRODUCTION

It is now clearer than ever that the future of sustainable space operations is in a dire predicament. This is especially true within the increasingly congested confines of Earth's low earth orbit (LEO) region (160km to 2,000km), where orbital debris are most concentrated. Through humanity's half century of space efforts, an estimated 400,000 and 14,000 objects of 1-10 cm and over 10 cm diameters, respectfully, have congregated in LEO [1]. With a presumably increasing commissioning rate of approximately 70 new satellites annually [2], the probability of catastrophic collisions occurring is ever increasing and worrisome. Responding to this predicament by delaying the progression of space operations would be an insult to the innovative demeanour that humanity has demonstrated through the course of its existence. Without perturbation, orbital debris situated in LEO's most congested region of around 800km can remain in orbit for decades; remaining idle would prove to be inefficacious. In order to continue the efficient progression of technological innovation in space exploration, active debris removal must be implemented.

2. MECHANISM OF ACTION

Such a recalcitrant problem as orbital debris requires an innovative solution. Within this study, attention will be brought to passive aerobraking as a method for deorbiting space debris. Under this paradigm of orbital debris mitigation the spacecraft is to, from any initial orbit, autonomously and intelligently navigate to and rendezvous with orbital debris of specified criteria, subsequently encapsulating the object with an ejected net. Following encapsulation, the spacecraft is to passively transport the object into a lower orbit via deployable drag sails and remaining propulsive resources, only releasing it once it is determined that the object, by itself, will decay in a reasonable amount of time. To illustrate such a novel mechanism of action, an extensive simulation is required.

3. DEVELOPMENT FRAMEWORK

With concern to selection of programming language, it was chosen that the simulation be composed in Python 2.7 [3]. This selection was not without careful consideration to what this study requires, especially in terms of manipulability. The utilisation of commercial astrodynamics programs is akin to using a black box. Although such programs have proven records in regards to industrial usage, it is imperative that the processes by which the results are determined be clearly defined and easily manipulated, especially in a simulation that necessitates modelling of not only the spacecraft's dynamics, but also the environment in which it operates, and its autonomous functions.

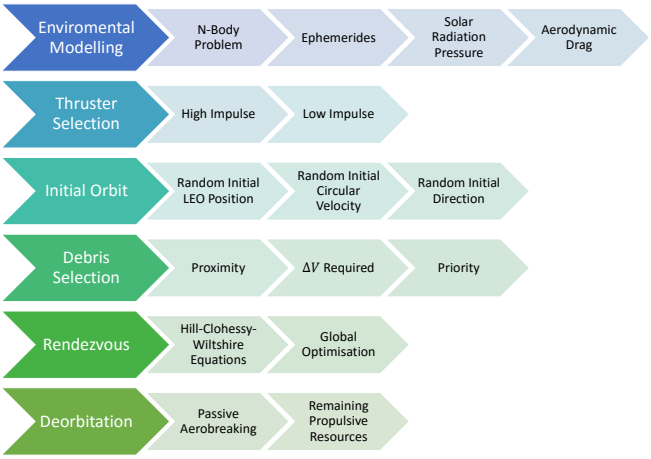


Fig. 1: Simulation Development Road Map

4. EPHEMERIDES

The development of an accurate spacecraft simulation necessitates the modelling of the celestial bodies in order to instantiate their influences on the spacecraft. This requires both the position and velocity of all the solar system's massive bodies; however, in the context of a spacecraft in LEO, it is a proper approximation to invoke only the influences of the Sun, Earth, and Moon. The positions and velocities of all the bodies within the solar system may be obtained through an ephemeris, a tabulation of the celestial bodies' calculated states. In this study, ephemerides were obtained through NASA's Jet Propulsion Laboratory Development Ephemeris, DE423 (used in MESSENGER), accessed through the Python module, '*jplephem 1.2*' [4].

5. PROPULSION

As it is the desire of this study to maximise efficiency through the course of the spacecraft's mission, the effects of its propulsion subsystem will only be intentionally invoked where they are necessarily required, namely the orbital manoeuvring phase in which the spacecraft navigates to the chosen orbital debris objects. In order to maintain this study's adherence to reality, it is necessary to procure the propulsive specifications of actual commercially available propulsion systems. In order to augment the breadth of this study's applicability, attention will be brought to two distinct regimes of propulsion systems, namely high impulse and low impulse propulsion.

5.1. High impulse

Among the most prevalent regimes of spacecraft propulsion is high impulse chemical propulsion. Though chemical propulsion is inherently energy limited, systems of this type are superior with respect to their high thrust and low power requirements as compared to low impulse electric propulsion systems. Such propulsion systems can allow spacecraft to manoeuvre in an agile manner.

There are a variety of cubesat propulsion systems available to the commercial market. In the context of a 4 kilogramme three unit CubeSat, such as that of MERiDIUS [5] (Mitigation of Environmental Risks Due to Impact on Unfortunate Spacecraft), a cubesat designed entirely with commercially available products, one can utilise the one unit *MPS-120 CubeSat High-Impulse Modular Propulsion System (CHAMPS)* of Aerojet Rocketdyne [6], fuelled with hydrazine propellant and equipped with four thrusters. This system has the ability to provide 2.79 Newtons of force per thruster and a potential ΔV of 220 meters per second in the case of MERiDIUS. The use of hydrazine, though highly controversial due to its horrendous health effects, is highly advantageous, being able to supply a favourable specific impulse (I_{sp}) of approximately 220 seconds.

5.2. Low impulse

Similarly to impulsive chemical propulsion systems, low impulse electrically propelled systems have their own limitations. Electric propulsion systems are inherently power limited; however they are contrarily not at all energy limited, ignoring hardware lifetime degradations [7]. In fact, low impulse electrically propelled systems can impart an arbitrarily immense amount of energy, allowing for a greatly elevated specific impulse. Resultantly, such systems provide a low thrust to mass ratio for their given spacecraft. Though this type of propulsion system results in a greatly decreased spacecraft acceleration, it has the ability to produce a prodigious amount of total impulse over time as compared to chemically propelled systems. Similarly to chemical propulsion systems, there are numerous commercially available low impulse propulsion systems available to consumers. In the case of MERiDIUS, one could choose the 1.6 unit *Busek Bit-3 Ion Thruster* [8]. This thruster supplies a specific impulse of 3500 seconds, a thrust of 1.4 milliNewtons an astounding potential ΔV of 5460 meters per second, in the case of a 4.5kg version of MERiDIUS.

5.3. Summary

The propulsive specifications of the two aforementioned commercially available propulsion systems can readily be invoked as metrics within the manoeuvring and decay phase of the spacecraft's simulation. There are three important metrics within the context of a spacecraft simulation, namely: potential delta-V (ΔV_{cap}), specific impulse (Isp) and maximum thrust (T_{Max}). These metrics are summarised for each of the aforementioned propulsion systems in Table 1.

Although low impulse electric propulsion certainly has its merits and accolades from its recent use in space exploration,

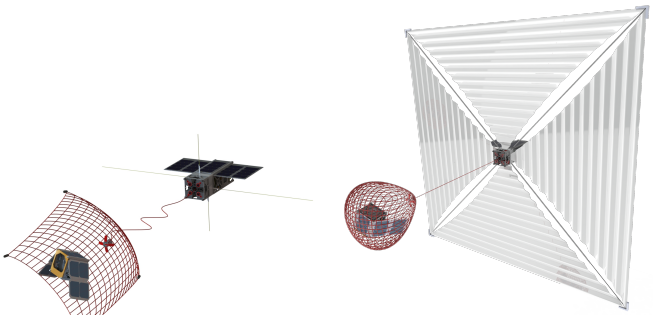
Table 1: CubeSat System Specifications

Type	$\Delta V[\frac{m}{s}]$	I_{sp}[s]	T_M[N]	m_0[kg]	m_f[kg]
High Imp.	220	220	11.16	4	3.7
Low Imp.	5460	3500	1.4e-3	4	3.4

for the sake of this study's intent on demonstrating the feasibility of real-time autonomous operation in the context of active debris removal, this study's focus is herewith restricted to high impulse propulsion.

6. AEROBRAKING

With regard to the unique mechanism of action that this study aims to demonstrate the feasibility of, the spacecraft will utilise a deployable drag sail. Utilisation of such a device will enable the spacecraft, after it has encapsulated its desired object, to passively transport the object into a lower orbit, where it will subsequently realise its demise. Similarly to CubeSat propulsion systems, there is also a variety of deorbit devices available to the commercial market. Such as it is, in the case of MERiDIUS, one may employ the use of the 0.4 unit *Aerodynamic End of Life DeOrbit System for CubeSats (AEOLDOS)* of Clyde Space [9], which is able to supply 3 square metres of area. This aerobraking system is depicted in Figure 2, shown in its deployed state, after the spacecraft has captured its targeted debris object.



(a) Net Caputure (b) Deployed Drag Sail

Fig. 2: MERiDIUS [5]

7. INITIAL ORBIT CHARACTERISATION

In simulating the spacecraft's orbital dynamics with the intention of verifying the robustness of its autonomous control algorithms, it is essential to instill randomness within the parameters of the environment in which the spacecraft operates in. Among the conditions in which this study is conducted are: 1) the spacecraft must only operate in a low Earth orbit with adherence to the task of only seeking orbital debris bodies which are also in LEO and 2) the spacecraft must be initially inserted into a circular orbit of random initial longitude, latitude, and direction for the reason that it will have likely been transported into orbit by a launch vehicle whose intentions are that of a separate mission.

8. INITIAL POSITION

As it is desired to instill randomness into this simulation for the sake of verifying the robustness of the spacecraft's autonomous functions; the initial latitude, longitude, and altitude should be generated at random within this study's constraints.

8.1. Random points on a sphere

It would be incorrect to simply select at random the spacecraft's spherical coordinates from uniform distributions of longitude and latitude: $\theta \in [0, 2\pi)$ and $\phi \in [0, \pi]$, respectively [10]. Randomly selecting from such distributions would result in point clustering towards the poles of the sphere, since any small area element on its surface is a function of ϕ such that $d\Omega = \sin\phi d\theta d\phi$. Because a differential element of solid angle on a sphere is characterised by $d\Omega = \sin\phi d\theta d\phi = -d\theta d(\cos\phi)$, one may ensure a consistent number of available points to select from each small surface area on the sphere, as to allow an equal probability for each point to be selected, by implementing the following formulation:

$$\theta = 2\pi u : u \in (0, 1) \tag{1}$$

$$\phi = cos^{-1}(2v - 1) : v \in (0, 1) \tag{2}$$

where θ and ϕ are respectfully the spacecraft's possible longitudes and latitudes uniformly distributed over the sphere's surface $\mathbb{S}^2$, and $\{u, v\}$ are both uniformly distributed over $(0, 1)$.

8.2. Cartesian coordinates

Assuming the longitude and latitude of the spacecraft's initial orbit have been randomly selected from a uniform point distribution about $\mathbb{S}^2$, the initial altitude of the spacecraft may be randomly generated. Imposing the criteria that the spacecraft may only operate within LEO, its initial altitude may be selected at random from a uniform distribution between 160 and 2,000 km as follows:

$$r = h_{sc} + r_E \mid h_{sc} \in (160\ km, 2000\ km) \tag{3}$$

where h_{sc} is the spacecraft's altitude measured from sea level, r_E is the volumetric mean radius of Earth (6371.0 km), and r is the distance from the spacecraft to the centre of Earth. These parameters are then converted to accommodate the primary Cartesian coordinate system centred at the barycentre of

the solar-system:

$$x = r\sin(\phi)\cos(\theta) \tag{4}$$

$$y = r\sin(\phi)\sin(\theta) \tag{5}$$

$$z = r\cos(\phi) \tag{6}$$

$$\vec{r} = x\hat{i} + y\hat{j} + z\hat{k} \tag{7}$$

$$\vec{R}_{sc} = \vec{R}_E + \vec{r} \tag{8}$$

where $[x, y, z]^T$ forms the vector $\vec{r}$ of the spacecraft's Cartesian coordinates as measured from Earth's centre, $\vec{R}_E$ forms Earth's Cartesian coordinates as measured from the barycentre of the solar system, and $\vec{R}_{sc}$ composes the the barycentric coordinates of the spacecraft.

9. INITIAL VELOCITY

Once the initial position of the spacecraft has been determined, all that remains is to initialise its velocity such that it maintains a circular orbit in the context of a two-body problem with Earth, which of course, due to forces exerted by ancillary bodies such as the Moon and the Sun, and non conservative forces such as aerodynamic drag and solar radiation pressure, may not be entirely accurate; however, for a spacecraft in LEO, such an approximation is suitable for its initialisation.

9.1. Magnitude

The only specific requirement of the spacecraft to maintain a circular orbit is the magnitude of its velocity, which can be determined by assuming the semi-major axis of the spacecraft's orbit is equal to the magnitude of its initial geocentric position, and that it must be directed tangentially along its sphere of orbit about Earth, such that its radial velocity towards Earth is zero, which is summarised accordingly:

$$v = \sqrt{\mu_E(\frac{2}{r} - \frac{1}{a})} \Rightarrow \sqrt{\frac{\mu_E}{r}} : a = r \tag{9}$$

where v is the magnitude of the spacecraft's geocentric velocity, μ_E is the standard gravitational parameter of Earth, r is the magnitude of the spacecraft's geocentric position, and a is the magnitude of the semi-major axis of the spacecraft's orbit.

9.2. Direction

In order to maintain this study's desired randomness in the spacecraft's initial orbital characterisation, its velocity vector must be directed in any direction that is tangential to its sphere of orbit about Earth. In order to do this, firstly the equation of the plane tangent to the spacecraft's sphere of orbit about Earth at its initial instance, whose points are arbitrary, must be obtained accordingly:

$$x(x_{arb} - x) + y(y_{arb} - y) + z(z_{arb} - z) = 0 \tag{10}$$

where $\vec{r} = [x, y, z]^T \in \mathbb{S}^2$ composes the spacecraft's Cartesian geocentric coordinates and $\vec{r}_{arb} = [x_{arb}, y_{arb}, z_{arb}]^T \in \mathbb{T}_P$ composes the geocentric coordinates of an arbitrary point somewhere in the tangent space ($\mathbb{T}_P$) of the spacecraft's orbital sphere manifold ($\mathbb{S}^2$) at it's initial coordinates. Secondly, in accordance to this study's scheme, two arbitrary points lying on the orbital sphere's tangent space are selected at random from a uniform distribution between -1 and 1 such that $\{y_{arb}, z_{arb}\} \in (-1, 1)$, which enables the subsequent determination of z_{arb}, the third remaining arbitrary Cartesian coordinate on the orbital sphere's tangent plane. Thirdly, the direction of the spacecraft's initial velocity vector can be determined by calculating the position vector from the spacecraft to the arbitrary point on the plane tangent to the spacecraft's sphere of orbit about Earth, and then computing its unit vector accordingly:

$$\hat{v} = \frac{\vec{r}_{arb} - \vec{r}}{\|\vec{r}_{arb} - \vec{r}\|} \tag{11}$$

where $\hat{v}$ is the direction of the spacecraft's geocentric velocity vector. Further clarification on the process which led to this result can be attained by observing Figure 3.

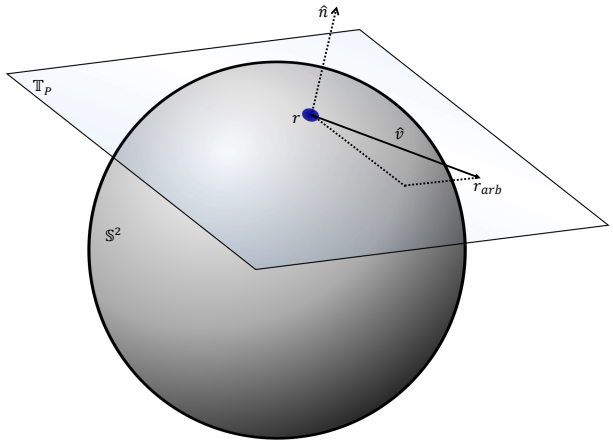


Fig. 3: This figure represents the spacecraft's sphere of orbit $\mathbb{S}^2$ and tangential plane $\mathbb{T}_P$, from which the spacecraft's random initial velocity direction is calculated.

9.3. Vectorisation

Finally, the spacecraft's geocentric and barycentric velocity vectors, can be determined respectively:

$$\vec{v} = v\hat{v} \tag{12}$$

$$\vec{V}_{sc} = \vec{V}_E + \vec{v} \tag{13}$$

where $\vec{v}$ is the spacecraft's geocentric velocity vector, $\vec{V}_E$ is Earth's velocity vector with respect to the barycentre of the solar system, and $\vec{V}_{sc}$ is the spacecraft's barycentric velocity vector.

10. ENVIRONMENTAL CHARACTERISATION

Commonly, in elementary studies of orbital dynamics, only the gravitational effects native to the two body problem are taken into account. However, in reality, a spacecraft orbiting Earth will experience a multitude of accelerations, instilled by not only conservative forces, but also non-conservative forces.

10.1. Conservative forces

Of the conservative forces considered extraneous in the context of basic orbital dynamics, are the gravitational influences of all ancillary bodies in the solar system; however, it is indeed valid to only heed influences of the Earth, Moon and Sun, as the forces felt by the other planets in the solar system whilst orbiting Earth are negligible in comparison. Such as it is, one may construct the spacecraft's conservative acceleration vector as the summation of the gravitational influences of the Sun, Earth and Moon accordingly:

$$\vec{A}_n^C = \vec{A}_n^{Sun} + \vec{A}_n^{Earth} + \vec{A}_n^{Moon} \tag{14}$$

$$\vec{A}_n^j = -\frac{\mu^j(\vec{R}_n^j - \vec{R}_n^{sc})}{\|\vec{R}_n^j - \vec{R}_n^{sc}\|^3} \tag{15}$$

where $\vec{A}_n^C$ is the spacecraft's conservative acceleration vector, $\vec{A}_n^j$ denotes the spacecraft's acceleration due to the gravitational influence of the j^{th} body such that j signifies the index of the observed body, μ^j is the standard gravitational parameter of the j^{th} body, $\vec{R}_n^j$ is the barycentric position vector of the j^{th} body, $\vec{R}_n^{sc}$ is the spacecraft's barycentric position vector, and the subscript n denotes the n^{th} time step within the simulation.

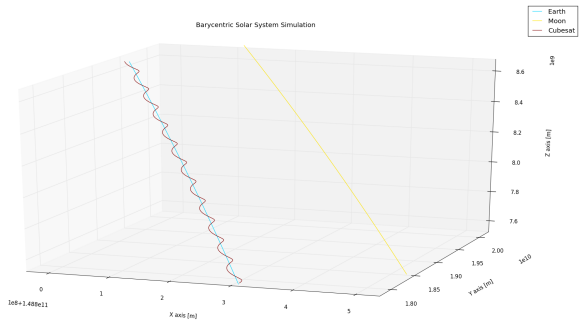


Fig. 4: Barycentric perspective of the spacecraft situated in low Earth orbit, with the moon in view.

10.2. Non-conservative forces

Whilst conservative forces display continuous conservation of energy within the spacecraft's orbital dynamics, non-conservative force cause perturbations that lead to observed deviations from the idealised predictions of the spacecraft's motion. Among the non-conservative forces that the spacecraft may experience are aerodynamic drag and pressure due to solar radiation. One may start by constructing the spacecraft's non-conservative acceleration vector, as the summation of its perceived aerodynamic acceleration and acceleration due to solar radiation pressure, accordingly:

$$\vec{A}_n^{NC} = \vec{A}_n^{Aero} + \vec{A}_n^{Solar} \tag{16}$$

where $\vec{A}_n^{NC}$ is the spacecraft's non-conservative acceleration vector, $\vec{A}_n^{Aero}$ is the acceleration due to aerodynamic drag felt by the spacecraft, and $\vec{A}_n^{Solar}$ is the spacecraft's acceleration due to solar radiation pressure.

10.3. Aerodynamic drag

While the calculation of solar-radiation pressure is relatively straight forward, obtaining the atmospheric density at a given altitude in accordance to an exponential scale model requires a change in scale as the spacecraft descends into each subsequent altitude range as its orbit decays, as can be seen in Figure 5. The vector of the spacecraft's acceleration due to aerodynamic drag can be formulated as follows [11]:

$$\vec{A}_n^{Aero} = -\frac{\rho_n C_n^D A_n^{\perp\hat{v}} v^2 \hat{v}_n}{2m_n} \tag{17}$$

where, as before, the subscript n denotes the n^{th} time step within the simulation, C_n^D is the spacecraft's drag coefficient, $A_n^{\perp\hat{v}}$ is the area of the spacecraft perpendicular to the direction of the spacecraft's velocity, ρ_n is the atmospheric density at the spacecraft's altitude, v is the spacecraft's geocentric velocity, $\hat{v}_n$ is the direction of the spacecraft's velocity, and m_n is the spacecraft's mass.

It should be noted that before the deployment of the drag sail, the spacecraft's area perpendicular to flow is such that $A_n^{\perp\hat{v}} = 10cm^2$; however this area increases significantly once the drag sail is deployed such that $A_n^{\perp\hat{v}} \rightarrow 3m^2$. Furthermore, it is widely accepted to assume that $C_n^D = 2.2$ for a spacecraft. In order to accurately simulate the atmospheric drag that the spacecraft will experience, the atmospheric density at any given altitude must be computed accordingly [12]:

$$\rho_n = \rho_0 e^{\frac{-h_n}{H}} \tag{18}$$

where ρ_0 is the atmospheric density at sea level, H is the atmospheric scale height for the given range that the spacecraft's altitude pertains to, and h_n is the altitude of the spacecraft.

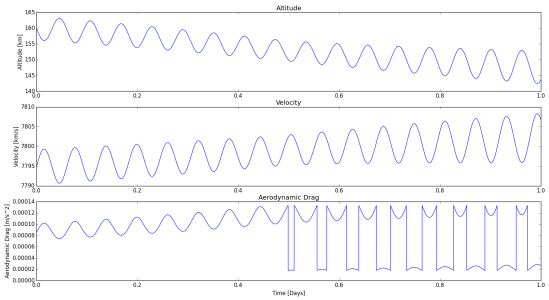


Fig. 5: This figure represents the functioning of the atmospheric scale model. As the spacecraft descends into lower altitudes, its atmospheric scale height drops into a new range. The altitude, velocity, and acceleration due to drag are depicted in the top, middle, and bottom rows, respectively.

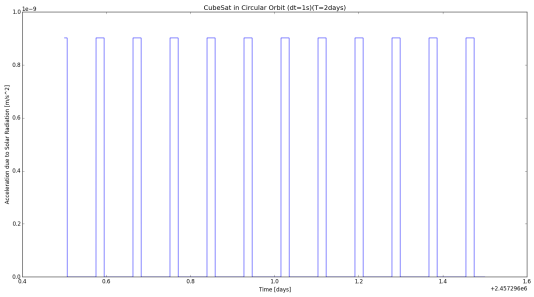


Fig. 6: In this figure, the functioning of the solar radiation pressure and eclipse models can be seen. As the spacecraft's view to the sun becomes blocked by a celestial body, its acceleration due to solar radiation drops.

10.4. Solar radiation pressure

Among the remaining non-conservative perturbations that the spacecraft may experience whilst orbiting about Earth is solar radiation pressure. Solar radiation pressure is the force imparted on the spacecraft as a result of electromagnetic particles either being reflected or absorbed by the spacecraft's exterior. The resultant acceleration due to this force may be computed accordingly:

$$\vec{A}_n^{Solar} = -\frac{p_n^s C_n^R A_n^{\perp \hat{s}} \hat{s}_n}{m_n} \tag{19}$$

where p_n^s is the nominal value of solar pressure felt at Earth's location ($\simeq 5.00e-6\frac{N}{m^2}$), C_n^R is the spacecraft's coefficient of reflectivity, $A_n^{\perp \hat{s}}$ is the area of the spacecraft perpendicular to the sun's direction, $\hat{s}_n$ is the direction vector from the spacecraft to the sun and m_n is the spacecraft's mass. The spacecraft coefficient of reflectivity may vary from 0.0 to 2.0 such that $C_n^R \in [0.0, 2.0]$, where 0.0 indicates a translucent surface with no influence, 1.0 indicates a completely absorbing surface and 2.0 indicates a surface that both absorbs and reflects. Additionally, the value of p_n^s is difficult to compute due to unpredictable fluctuations in solar activity. An effective model can be found to indicate whether the spacecraft is in sunlight or being eclipsed by either Earth or another celestial body using geometry [11]:

$$\tau = \frac{\vec{r} \cdot \vec{r} - \vec{s}_E}{|\vec{r} - \vec{s}_E|^2} \tag{20}$$

where $\vec{r}$ is the spacecraft's geocentric position vector and $\vec{s}_E$ is the position vector from Earth to the Sun. If τ is either less than zero or greater than one, than the spacecraft will be in sunlight.

11. NUMERICAL PROPAGATION

In modelling and simulating the spacecraft's orbital dynamics, it is necessary to propagate the effects of the environmental accelerations applied to the spacecraft by way of integrating a set of ordinary differential equations. It is through this integration that the primarily gravitational motion of the spacecraft is characterised. However, it is not only gravity that the spacecraft's motion is caused by; other non-conservative forces, such as aerodynamic drag and solar radiation pressure, also make their effects known. Given that, in regards to the spacecraft's gravitational influences, not only is Earth taken into account, but also the Sun and Moon. It is for this reason that it is necessary to apply the N-Body Method to accommodate more than one body of attraction as follows:

$$\vec{A}_{i,n} = \sum_{j=0, i \neq j}^{N} \vec{A}_{i,n}^{j} \tag{21}$$

where $\vec{A}_{i,n}$ is the ultimate acceleration of the i^{th} body, $\vec{A}_{i,n}^{j}$ is the acceleration of the i^{th} body due to the influence of the j^{th} body, and N is the total number of bodies. All of the spacecraft's environmental influences may culminate in the form of one ultimate acceleration vector that is readily applied to multiple spacecraft simultaneously:

$$\vec{A}_{i,n} = \vec{A}_{i,n}^{Sun} + \vec{A}_{i,n}^{Earth} + \vec{A}_{i,n}^{Moon} + \vec{A}_{i,n}^{Aero} + \vec{A}_{i,n}^{Solar} \tag{22}$$

where the individual accelerations that compose the spacecrafts' accelerations, $\vec{A}_{i,n}$, are defined as they were before. The acceleration of the spacecraft in observance to multiple attracting bodies and non-conservative perturbations can be propagated through the time frame of the spacecraft's simulation with the utilisation of one of the variety of numerical integration methods available to handle ordinary differential equations.

11.1. Forward Euler method

Among the simplest of such methods that are available to integrate the ordinary differential equation that compose the spacecraft's motion is the Forward Euler Method of numerical integration. This method can be summarised as follows:

$$\vec{V}_{i,n+1} = \vec{V}_{i,n} + \delta t \vec{A}_{i,n} \tag{23}$$

$$\vec{R}_{i,n+1} = \vec{R}_{i,n} + \delta t \vec{V}_{i,n} \tag{24}$$

where $\vec{A}$ is an acceleration vector, $\vec{V}$ is a barycentric velocity vector, $\vec{R}$ is a barycentric position vector, δt is the simulation's time step size with units of seconds, i denotes the i^{th} body and n denotes the n^{th} time step whose value may either be n or $n+1$ in order to indicate the next time time step from the spacecraft's current time step.

11.2. Runge-Kutta-Fehlberg method

In order to guarantee accuracy of this study's simulation, it's necessary to implement a more advance integration algorithm. For the case of initial value problems such as the one of this study, where the spacecraft has a random initial position and velocity about Earth, one can ensure numerical integration accuracy by varying the simulation's step size in a dynamic manner. In order to determine how the simulation's time step should be varied as time progresses, one would intuitively have to run the simulation multiple times at different time steps and compare the accuracy of their results. In order to avoid such a computationally intensive process it is advised to implement the Runge-Kutta-Fehlberg (RKF45) method with a dynamic time step.

The Runge-Kutta-Fehlberg algorithm is able to determine the proper step size δt to use by computing the error between two different approximations methods at each time step of the simulation. The algorithm adjusts the step size until the step's two approximations come to agreement within a specified accuracy tolerance. At each step of the simulation, vector coefficients are generated accordingly [13]:

$$(\vec{r_1}, \vec{v_1}) = (\vec{R}_n, \vec{V}_n) \tag{25}$$

$$\vec{a}_1 = \delta t \vec{A}(t_n, \vec{r}_1, \vec{v}_1) \tag{26}$$

$$\vec{v}_2 = \vec{v}_1 + \frac{1}{4}\vec{a}_1 \tag{27}$$

$$\vec{r}_2 = \vec{r}_1 + \frac{1}{4}\vec{v}_1 \tag{28}$$

$$\vec{a}_2 = \delta t \vec{A}(t_n + \frac{1}{4}\delta t, \vec{r}_2, \vec{v}_2) \tag{29}$$

$$\vec{v}_3 = \vec{v}_1 + \frac{3}{32}\vec{a}_1 + \frac{9}{32}\vec{a}_2 \tag{30}$$

$$\vec{r}_3 = \vec{r}_1 + \frac{3}{32}\vec{v}_1 + \frac{9}{32}\vec{v}_2 \tag{31}$$

$$\vec{a}_3 = \delta t \vec{A}(t_n + \frac{3}{8}\delta t, \vec{r}_3, \vec{v}_3) \tag{32}$$

$$\vec{v}_4 = \vec{v}_1 + \frac{1932}{2197}\vec{a}_1 - \frac{7200}{2197}\vec{a}_2 + \frac{7296}{2197}\vec{a}_3 \tag{33}$$

$$\vec{r}_4 = \vec{r}_1 + \frac{1932}{2197}\vec{v}_1 - \frac{7200}{2197}\vec{v}_2 + \frac{7296}{2197}\vec{v}_3 \tag{34}$$

$$\vec{a}_4 = \delta t \vec{A}(t_n + \frac{12}{13}\delta t, \vec{r}_4, \vec{v}_4) \tag{35}$$

$$\vec{v}_5 = \vec{v}_1 + \frac{439}{216}\vec{a}_1 - 8\vec{a}_2 + \frac{3680}{513}\vec{a}_3 - \frac{845}{4104}\vec{a}_4 \tag{36}$$

$$\vec{r}_5 = \vec{r}_1 + \frac{439}{216}\vec{v}_1 - 8v_2 + \frac{3680}{513}\vec{v}_3 - \frac{845}{4104}\vec{v}_4 \tag{37}$$

$$\vec{a}_5 = \delta t \vec{A}(t_n + \delta t, \vec{r}_5, \vec{v}_5) \tag{38}$$

$$\vec{v}_6 = \vec{v}_1 - \frac{8}{27}\vec{a}_1 + 2\vec{a}_2 - \frac{3544}{2565}\vec{a}_3 + \frac{1859}{4104}\vec{a}_4 - \frac{11}{40}\vec{a}_5 \tag{39}$$

$$\vec{r}_6 = \vec{r}_1 - \frac{8}{27}\vec{v}_1 + 2\vec{v}_2 - \frac{3544}{2565}\vec{v}_3 + \frac{1859}{4104}\vec{v}_4 - \frac{11}{40}\vec{v}_5 \tag{40}$$

$$\vec{a}_6 = \delta t \vec{A}(t_n + \frac{1}{2}\delta t, \vec{r}_6, \vec{v}_6) \tag{41}$$

where the members of $(\vec{R}_n, \vec{V}_n)$ are defined as they were before as the spacecraft's barycentric position and velocity vector at the n^{th} time step, δt is the time step size of the simulation and $\vec{A}$ is the acceleration function as defined in equation 22 that takes $\vec{R}$ and $\vec{V}$ as its arguments. After the coefficients have been generated, approximations from both the Runge-Kutta 4^{th} and 5^{th} order methods can be computed accordingly:

$$\vec{V}_{n+1}^{O(4)} = \vec{V}_n + \frac{25}{216}a_1 + \frac{1408}{2565}a_3 + \frac{2197}{4101}a_4 - \frac{1}{5}a_5 \tag{42}$$

$$\vec{R}_{n+1}^{O(4)} = \vec{R}_n + \frac{25}{216}v_1 + \frac{1408}{2565}v_3 + \frac{2197}{4101}v_4 - \frac{1}{5}v_5 \tag{43}$$

$$\vec{V}_{n+1}^{O(5)} = \vec{V}_n + \frac{16}{135}a_1 + \frac{6656}{12825}a_3 + \frac{28561}{56430}a_4 - \frac{9}{50}a_5 + \frac{2}{55}a_6 \tag{44}$$

$$\vec{R}_{n+1}^{O(5)} = \vec{R}_n + \frac{16}{135}v_1 + \frac{6656}{12825}v_3 + \frac{28561}{56430}v_4 - \frac{9}{50}v_5 + \frac{2}{55}v_6 \tag{45}$$

where $O(4)$ and $O(5)$ denote the 4^{th} and 5^{th} order Runge-Kutta approximations, respectfully. Once the approximations have been computed, the value of the errors between the two methods associated with the spacecraft's barycentric velocity and position may subsequently be determine as follows:

$$E_{\vec{V}} = \frac{1}{360}a_1 - \frac{128}{4275}a_3 - \frac{2197}{75240}a_4 + \frac{1}{50}a_5 + \frac{2}{55}a_6 \tag{46}$$

$$E_{\vec{R}} = \frac{1}{360}v_1 - \frac{128}{4275}v_3 - \frac{2197}{75240}v_4 + \frac{1}{50}v_5 + \frac{2}{55}v_6 \tag{47}$$

where $E_{\vec{V}}$ and $E_{\vec{R}}$ are the errors between the 5^{th} and 4^{th} order Runge-Kutta methods for both the spacecraft's barycentric velocity and position vectors, $\vec{V}_{n+1}$ and $\vec{R}_{n+1}$,

respectively. If either of these error values are outside of the simulation's specified error tolerance criteria such that $\{E_{\vec{R},n}, E_{\vec{V},n}\} > E_T$ or $\{E_{\vec{R},n}, E_{\vec{V},n}\} \ll E_T$, then the coefficients $[a_1, \ldots, a_6]$ and $[v_1, \ldots, v_6]$ are recomputed at time step sizes of $\frac{\delta t}{2}$ or $2\delta t$, respectively, until the error is acceptable such that $\{E_{\vec{R},n}, E_{\vec{V},n}\} < E_T$, where $E_T \simeq 1e-8$. Once it has been determined that the estimated error associated with both the spacecraft's barycentric velocity and position are within the simulation's error tolerance criteria, the values $\vec{V}_{n+1}^{O(5)}$ and $\vec{R}_{n+1}^{O(5)}$ can be accepted, and a conservative estimate of the next iteration's time step can be computed accordingly:

$$\delta t_{opt} = \min \left\{ \delta t \left(\frac{E_T \delta t}{2 \left| E_{\vec{R},n} \right|} \right)^{1/4}, \delta t \left(\frac{E_T \delta t}{2 \left| E_{\vec{V},n} \right|} \right)^{1/4} \right\} \tag{48}$$

where δt_{opt} is the approximated optimal time step for the simulation's next iteration based on the error between the 5^{th} and 4^{th} order Runge-Kutta methods for the current iteration, E_T is the specified error tolerance in the simulation and δt is the accepted time step of the current iteration. This particular integration method is able to anticipate certain areas of the spacecraft's trajectory where more resolution may be needed.

12. SIMULATION VALIDITY

Once the supporting framework of the spacecraft's simulation is entirely constructed, it is important to verify that its dynamics and utilities are behaving as they are supposed to according to astrodynamical theory and user preferences.

12.1. Reference frames

Given that the framework of this study was developed such that its core computations are primarily performed in terms of barycentric coordinates, data can readily be constructed with reference to any bodies of which this study restricts its observance to, mainly Earth and the Sun. Such as it is, visualisations can easily be constructed with reference to either barycentric or geocentric coordinate systems, as can be seen in Figure 4 and Figure 7, respectively.

12.2. Insertion

Firstly, attention should be brought to the method by which the spacecraft's orbit was initiated, developed in Section 7. Adhering to the fundamental principle's that this study adheres to, it was desired that the cubesat be inserted into a completely random orbit, with the only restrictions imposed being that it is circular and within LEO. Noting that a perfectly circular orbit's radial velocity should remain at zero in reference to it's primary attracting body, it can be seen in Figure 8 that

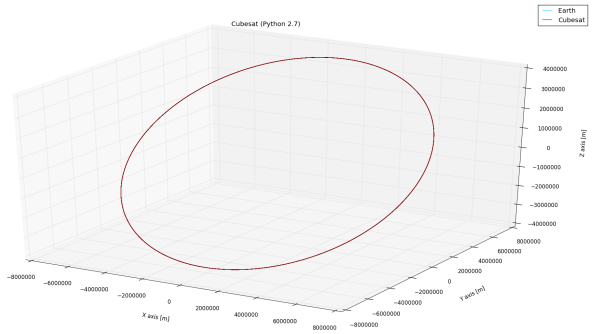


Fig. 7: Geocentric perspective of spacecraft's orbit, integrated with the RKF45 method.

the spacecraft's range of radial geocentric velocities are minuscule in comparison to the magnitude of its overall geocentric velocity ($\sim 7.8\ km/h$) at anytime; though it should be noted that Figure 8 was constructed using the Forward Euler method, while in contrast the RKF45 method has constructed a very fine line in Figure 7, indicating much less error.

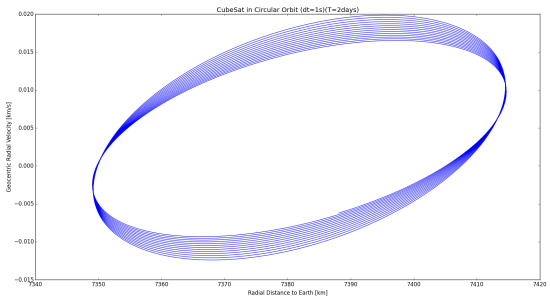


Fig. 8: Radial Velocity vs. Geocentric Position, integrated with the Forward Euler method.

12.3. Perturbations

The two non-conservative forces modelled in Section 10.2 were aerodynamic drag and solar radiation pressure. It can be seen in Figure 9 that the spacecraft experiences its greatest atmospheric drag at its closest approach to Earth just as it should in accordance to Equation 17. Additionally, the intended behaviour of the exponential scale density model constructed in Equation 18 can be seen in Figure 5; observe how the curve of the aerodynamic drag shifts for each subsequent altitude range. In addition to aerodynamic drag, the success of the solar radiation pressure model formed in Equation 19 can be seen in Figure 6, where the imparted acceleration of the spacecraft sharply drops off once it has been determined that the spacecraft's view of the sun is obscured.

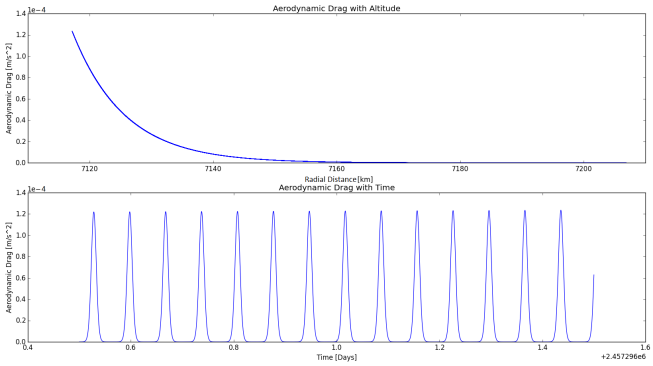


Fig. 9: Aerodynamic Drag with Geocentric Position and Time

13. RENDEZVOUS

Assuming that the spacecraft has been inserted into a randomly characterised orbit in order to simulate a CubeSat's lack of jurisdiction in regard to its orbital insertion, it is necessary to bring attention to how the cubesat will fulfil its mission goals by rendezvousing with a target debris object. In order to do this it is necessary to simulate not only the cubesat's motion, but also the plausible motion of the debris it intends to seek out. It is entirely possible that the targeted space debris within this simulation may have an elliptical orbit; however, attention will only be brought debris remain in the confines of LEO. It was chosen that this simulation include debris whose orbits may be elliptical in order to give credence to the fact that such problematic bodies often behave chaotically, originating from collisions that are becoming ever so frequent. In this particular study, attention will be brought directly towards the debris of Fengyun-1C, the Chinese weather satellite that was destroyed in 2007 by China as part of an anti-satellite missile test. This event created more dangerous orbital debris than any other space mission in history [14].

13.1. Hill-Clohessy-Wiltshire equations

In the case that the spacecraft is fortunate enough to find itself in close proximity to its orbital debris target where the spacecraft's position with respect to its target δr is much smaller than both the geocentric positions of the spacecraft r_{sc} and the orbital debris object r_{od}, such that $\delta r \ll \{r_{sc}, r_{od}\}$, one can what should be implemented the Hill-Clohessy-Wiltshire (HCW) equations [15]:

$$\delta\ddot{x} - 3n^2\delta x - 2n\delta\dot{y} = \frac{T_x}{m} \tag{49}$$

$$\delta\ddot{y} + 2n\delta\dot{x} = \frac{T_y}{m} \tag{50}$$

$$\delta\ddot{z} + n^2\delta z = \frac{T_z}{m} \tag{51}$$

$$n = \sqrt{\frac{\mu_E}{r_{sc}^3}} \tag{52}$$

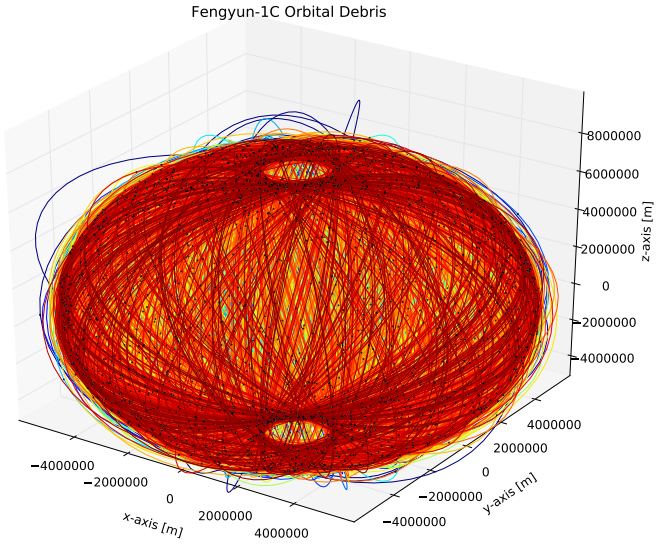


Fig. 10: Every debris fragment of Fengyun-1C as of January 1^{st}, 2016.

where n is the spacecraft's mean motion, r_{sc} is the geocentric position of the spacecraft, δx is the radial position of the spacecraft relative to the target in the direction $\hat{i}$, δz is the cross-track position of the spacecraft relative to the target in the direction $\hat{k}$, δy is the along-track position of the cubesat relative to the target in the direction $\hat{j}$, and $[T_x, T_y, T_z]^T$ composed the spacecraft's thrust vector in LVLH space.

13.2. LVLH reference frame

These equation are constructed in the realm of the LVLH reference frame centred about the target, where direction vectors describing the target's frame from which the chaser's relative motion are formed accordingly:

$$\hat{i} = \frac{\vec{r}}{\|\vec{r}\|} \tag{53}$$

$$\hat{k} = \frac{\vec{h}}{\|\vec{h}\|} \tag{54}$$

$$\hat{j} = \hat{k} \times \hat{i} \tag{55}$$

where $\vec{r}$ denotes the target's position vector with respect to the inertial frame centred about Earth and $\vec{h}$ denotes the target's angular momentum vector. The direction cosine matrix converting from the geocentric reference frame centred about Earth $\hat{X}$ to the LVLH reference frame centred about the the target space debris object $\hat{x}_{od}$ can then be constructed accordingly:

$$\mathbf{Q}_{\hat{X}\to\hat{x_{od}}} = \begin{bmatrix}\hat{i}\\ \hat{j}\\ \hat{k}\end{bmatrix} = \begin{bmatrix}\dots\hat{I} & \dots\hat{J} & \dots\hat{K}\\ \dots\hat{I} & \dots\hat{J} & \dots\hat{K}\\ \dots\hat{I} & \dots\hat{J} & \dots\hat{K}\end{bmatrix} \tag{56}$$

where $[\hat{i}, \hat{j}, \hat{k}]^T$ characterises the orientation of the LVLH frame's principles axes with respect to the geocentric frame centred about Earth, denoted by $\{\dots \hat{I}, \dots \hat{J}, \dots \hat{K}\}$.

13.3. Two-impulse rendezvous

The closed form solutions of the Hill-Clohessy-Wiltshire Equations are particularly useful in the implementation of real-time control. From these equations, the required $\Delta\vec{V}$, in the target's reference frame, for the cubesat to rendezvous with the object can be obtained and executed. The closed-form solution to these equations, from which the required $\Delta\vec{V}$ will be obtained, can be constructed accordingly:

$$\delta\vec{r}(\delta t) = \mathbf{\Phi}_{rr}(\delta t)\delta\vec{r}_0 + \mathbf{\Phi}_{rv}(\delta t)\delta\vec{v}_0 \tag{57}$$

$$\delta\vec{v}(\delta t) = \mathbf{\Phi}_{vr}(\delta t)\delta\vec{r}_0 + \mathbf{\Phi}_{vv}(\delta t)\delta\vec{v}_0 \tag{58}$$

$$\mathbf{\Phi}_{rr}(\delta t) = \begin{bmatrix} 4 - 3\cos n\delta t & 0 & 0 \\ 6(\sin n\delta t - n\delta t) & 1 & 0 \\ 0 & 0 & \cos n\delta t \end{bmatrix} \tag{59}$$

$$\mathbf{\Phi}_{rv}(\delta t) = \begin{bmatrix} \frac{\sin n\delta t}{n} & \frac{2(1-\cos n\delta t)}{n} & 0 \\ \frac{2(\cos n\delta t - 1)}{n} & \frac{(4\sin n\delta t - 3n\delta t)}{n} & 0 \\ 0 & 0 & \frac{\sin n\delta t}{n} \end{bmatrix} \tag{60}$$

$$\mathbf{\Phi}_{vr}(\delta t) = \begin{bmatrix} 3n\sin n\delta t & 0 & 0 \\ 6n(\cos n\delta t - 1) & 0 & 0 \\ 0 & 0 & -n\sin n\delta t \end{bmatrix} \tag{61}$$

$$\mathbf{\Phi}_{vv}(\delta t) = \begin{bmatrix} \cos n\delta t & 2\sin n\delta t & 0 \\ -2\sin n\delta t & 4\cos n\delta t - 3 & 0 \\ 0 & 0 & \cos n\delta t \end{bmatrix} \tag{62}$$

where $\{\delta\vec{r}_0, \delta\vec{v}_0\}$ and $\{\delta\vec{r}, \delta\vec{v}\}$ are respectively defined as the initial and time-variant LVLH referenced position and velocity vectors of the spacecraft, n is defined as it was in Equation 52 and δt is the elapsed time of the manoeuvre such that $\delta t = t_f - t_0$.

These solutions can offer an idealised solution to the spacecraft's rendezvous with its targeted space debris object; however, it can be seen in Figure 11 that these solutions may result in drift. Drift of the solutions provided by the closed-form of the HCW equations can arise when the orbit of the spacecraft's target does not entirely fit the criteria for use of the HCW equations, namely: 1) the chaser's orbit must be in the 'neighbourhood' of the target 2) the target's orbit must be circular and 3) the distance between the chaser and the target are much smaller than either of their geocentric positions. Although these equations uncompromisingly require specific conditions to be met, sometimes they may suffice even if their criteria is not entirely met. The HCW equations can be periodically invoked through the course of the spacecraft's trajectory in conjunction with a sampling routine, allowing these equations to operate somewhat accurately in the situation that the target's orbit is slightly elliptical, whilst maintaining real-time operation.

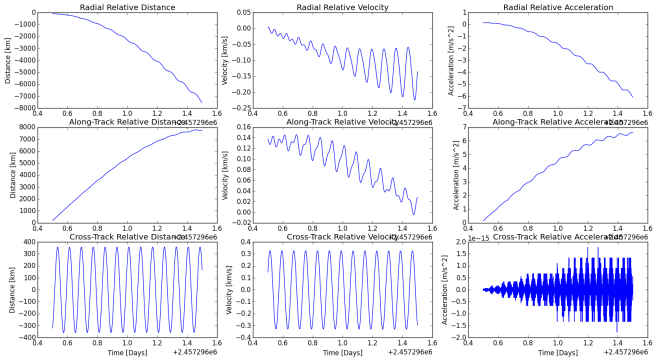


Fig. 11: An example of the drift resulting in a solution provided by the HCW equations for a target whose characteristics do not entirely fit the idealised criteria.

13.4. Dynamic greedy travelling salesperson problem

In order to give credence to the feasibility of utilising economical autonomous CubeSats to perform active debris removal, it is germane to direct this study's attention towards the use of the closed form solution of the Hill-Clohessy-Wiltshire Equations, as they form the basis of many real-time autonomous rendezvous control systems today.

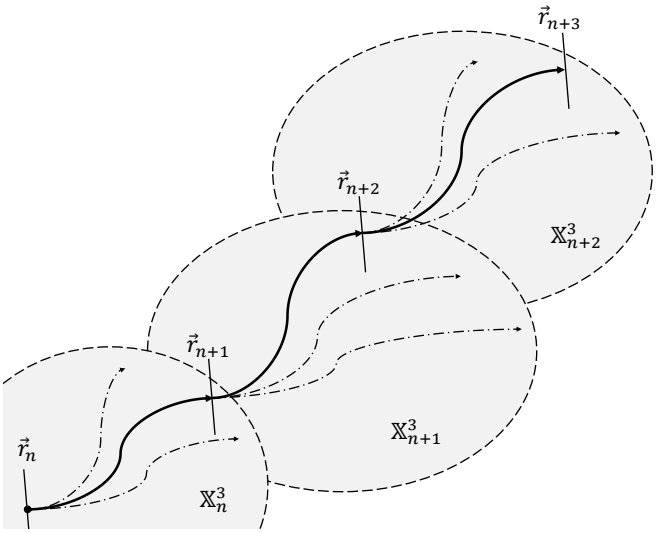


Fig. 12: This figure illustrates the method by which the spacecraft: 1) declares the space within its vicinity where the HCW equation's use would be permissible (grey ellipses), 2) searches its space for neighbours (lines), and 3) computes the cheapest neighbour to manoeuvre to (bold lines).

13.5. Problem

With the shear amount of debris in low Earth orbit, as can be seen in Figure 10, it is not exactly clear how one should proceed in order to perform active debris removal in an eco-

nomic and effective manner. Such as it is, it is necessary to perform an in depth analysis in order to ensure the reduction of propellant expenditure whilst maximising the amount of debris fragments mitigated. The clearest way to proceed is to construct a travelling salesperson problem. A travelling salesperson problem essential asks: given a list of cities and their relative distances, what is the is the shortest possible route in order to visit each city exactly once? In the case of active orbital debris mitigation, where each debris fragment symbolises a city, the becomes much more complicated, as the cities are not only dynamic with time, but also governed by physics whose manifestation is quite foreign to the terrestrial realm.

Although the sophistication of today's continually developing technological is quite remarkable, the on-board computational power of most spacecraft remains constrained by the harsh space environment. Any significant orbital manoeuvre performed by a spacecraft within low Earth orbit most likely with the assistance of a ground command.

13.6. Development

Under the assumption that the spacecraft has found itself to be fortunate enough to be initially situated upon an orbital debris fragment, it is to:

1. Determine the space within its vicinity $\mathbb{X}^3$, in which the implementation of the Hill-Clohessy-Wiltshire Equations would be permissible. (**HCW Space**)

   (a) **HCW Space** acts as a function that determines the space within the spacecraft's vicinity $\mathbb{X}^3$ in which a rendezvous manoeuvre through the use of the Hill-Clohessy-Wiltshire Equations is permissible under its assumptions, sometimes reffered to as 'pruning'. These assumptions outlined in Section 13.1

2. Search within that space for possible debris targets. (**NNS**)

   (a) **NNS** is a function that is able to rapidly return the most similar neighbours of any point within a data set utilising a k-d tree, a space-partitioning data structure for organising point in a k-dimensional space. The function is readily implemented by use of the Python class, '*scipy.spatial.KDTree*' [16]. This function is invoked once the size of the spacecraft's three dimensional search space is minimised, as to expedite the algorithms progress.

3. Compute the cost to rendezvous with each feasible debris. (**MCW2I**)

   (a) **MCW2I** returns the ΔV required for the spacecraft to rendezvous with a specified target debris fragment from its initial position.

4. Rendezvous with the debris fragment requiring the least ΔV.

5. Repeat 1-4 for each subsequent location

These steps are qualitatively summarised in Figure 12. Each vertical cross bar denotes each of the spacecraft's subsequent locations $[\vec{r}_n, \ldots, \vec{r}_N]$, each ellipse represents the spacecraft's vicinity in which it searches for feasible debris targets at each of it's subsequent locations, and the dotted trajectory lines represent the trajectories that were under consideration at each location. Additionally, this method is mathematically formulated in Algorithm 1. Note, once the optimal j^{th} debris to rendezvous with is determined, the spacecraft repeats the computations again from it's new location, where it's old final position and velocity become its new position and velocity.

The position, velocity, and area of each Fengyun-1C debris fragments can be obtained through two-line elements as supplied by CelesTrack [17]. After the ephemerides of the debris fragments are obtained, they will then need to be converted from their Keplerian format to a more convenient Cartesian format.

13.7. Results

Algorithm 1 was invoked for various rendezvous manoeuvre leg durations Δt and fuel capacities ΔV_{cap}. Through multiple runs of this simulation, it was soon realised that the quasi optimal transfer time was $\Delta t \approx 3\ hrs$. It can be seen in Table 2, with a ΔV allowance of $220\ m/s$, significant mitigation of debris can take place in a mere $3\ hrs$, the highest two accumulated areas were $14.185\ m^2$ and $16.031\ m^2$.

It seems as though the optimal way in which to mitigate orbital debris may not be aggressive. Observing Figure 14, in particular, gives off the thought that the spacecraft was just in the right places at the right time; which is indeed true. However, this also encourages that thought that perhaps today's efforts to mitigate orbital debris need to shift their methods to a different paradigm. The attempts to mitigate debris should develop a more opportunistic attitude. Meaning that, instead of chasing the debris, let the debris come to the spacecraft.

14. ORBITAL DECAY

Assuming that the spacecraft has successfully achieved rendezvous with one or more orbital debris objects, it would then execute it's final task. Once the spacecraft has collected as much debris as possible on its limitations, it would then deploy its drag sail. Once its drag sail deploys, its planaform area perpendicular to the flow would dramatically increase from a mere $0.01\ m^2$ to a colossal $3m^2$. From the moment of drag sail deployment onward, the spacecraft's drag increases drastically, and therefor it's time to decay (T_D) decreases significantly.

Algorithm 1: Computes the cheapest feasible active debris removal tour from each debris fragment. **HCW Space** computes the feasible search space, **NNS** assembles a list of n feasible targets via a nearest neighbour search, and **HCW2I** computes the ΔV required to rendezvous from one debris to another.

Data:
Debris' Ephemerides $\rightarrow \{\vec{r}_i, \vec{v}_i\} \forall i \in I = [i, \ldots, N]$
Propellant Capacity $\rightarrow \Delta V_{cap}$
Leg Duration$\rightarrow \Delta t$
Epoch$\rightarrow t_0$
Result:
Indicies of Feasible Tour Sequences from Various Starting Debris $S = \{S_i, \ldots, S_N\} = \{[\ldots], \ldots, [\ldots]\}$
begin
  $S = \{\ldots\}$
for $i \in I$ **do**
  $\{\vec{r}_0, \vec{v}_0\} \leftarrow \{\vec{r}_i, \vec{v}_i\}(t_0)$
  $s = [\ldots]$
  repeat
    $\mathbb{X}^3 \leftarrow$ **HCW Space**$(\vec{r}_0, \vec{v}_0)$
    $\{\vec{r}_j, \vec{v}_j\} \leftarrow$ **NNS**$(\mathbb{X}^3) \forall j \in J = [j, \ldots, n]$
    for $j \in J$ **do**
      $\{\vec{r}_f, \vec{v}_f\} \leftarrow \{\vec{r}_j, \vec{v}_j\}(t_0)$
      $\Delta V \leftarrow$ **HCW2I**$(\vec{r}_0, \vec{v}_0, \vec{r}_f, \vec{v}_f, \Delta t)$
      if $\Delta V < \Delta V_{opt}$ *or* $j = 0$ **then**
        $\Delta V_{opt} \leftarrow \Delta V$
        $j_{opt} \leftarrow j$
    $\Delta V_{spent} \leftarrow \Delta V_{spent} + \Delta V_{opt}$
    Append$(j_{opt}) \rightarrow s$
    $\{\vec{r}_0, \vec{v}_0\} \leftarrow \{\vec{r}_{j_{opt}}, \vec{v}_{j_{opt}}\}$
    $t_0 = t_0 + \Delta t$
  until $\Delta V_{spent} > \Delta V_{cap}$
  Append$(s) \rightarrow S$
return S

Using the RKF45 method of numerical integration with an adaptive time step, simulating the spacecraft's deorbitation becomes quite expedient in comparison to the performance of the forward Euler method. Such as it is, the time it takes for a 5 kilogramme cubesat to decay is easily tabulated in Table 3 for various starting altitudes, with and without the drag sail deployed. The efficacy of utilising passive aerodynamic drag to deorbit space debris is obvious from observing the table's values.

15. CONCLUSIONS

After having analysed the results of this study, several things can be concluded. Most significantly, it begs the question of whether the act of mitigating orbital debris should be opportunistic rather than targeted. With the technological capability

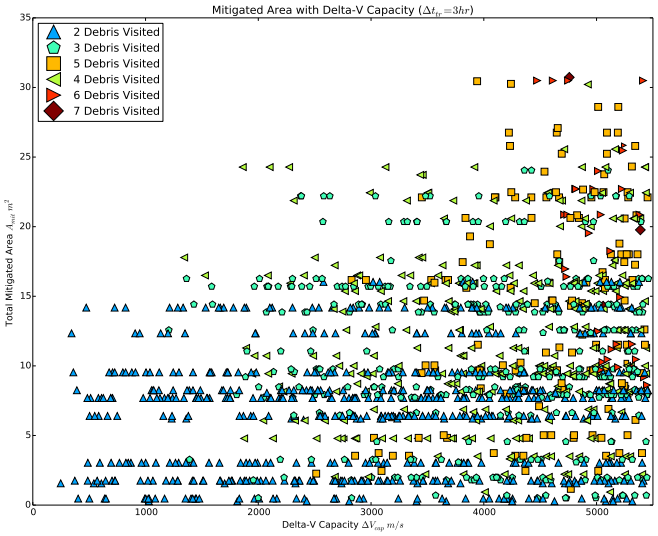


Fig. 13: This figure illustrates the variety of different solutions that Algorithm 1 will return with a delta-V budget of $\Delta V_{cap} = 5400 m/s$ when **HCW Space** is not called. This causes the spacecraft to seek rendezvous purely based on who the closest neighbour is. Thus, the more debris visited per ΔV, the more likely the data point is a feasible solution.

of most on-board spacecraft computers today, the most optimal complex targeted trajectories are simply just not easily attainable for a CubeSat in the realm of autonomy, as of yet. Because CubeSats have an incredibly small physical area in comparison to other spacecraft, they make great candidates for opportunistic orbital debris mitigation, as they can loiter without having to spend any considerable propellant to maintain themselves (granted a collision does not occur).

16. FUTURE WORK

In the very near future the global optimisation methods will be implemented in this simulation. In order to do this, the simulation's structure will need to be altered and made more efficient. Furthermore it is desired that the code built to form this simulation be parallelised to use multiple CPU cores to render individual simulations expeditious or utilise the method of GPU computing.

It is the ultimate goal for this simulation to simulate not only the cubesat's astrodynamics, but also each of its subsystems and their interactions with each other. This will include power consumption, propellant consumption, attitude control, environmental torques, communications, etc. This will allow optimal path planning and intelligent decision making with respect to multiple subsystems' criteria. It is then that further optimisation in the context of rendezvous may be conducted. Ultimately this simulation will culminate to form a full-blown cubesat testbed.

s

Table 2: Feasible HCW Rendezvous Tours with $\Delta V_{cap} = 220m/s$ and $\Delta t = 3hrs$

1^{st} Debris→ 2^{nd} Debris International Designator		$\Delta V_{spent}\ [m/s]$	$A_{deb}\ [m^2]$
1999-025AY	1999-025BMF	159.8777	1.752
1999-025BQ	1999-025ANC	206.9559	1.752
1999-025AEF	1999-025BXU	132.8574	1.752
1999-025AJX	1999-025BRF	197.5427	1.752
1999-025AZV	1999-025ZM	142.8967	0.277
1999-025BLJ	1999-025CT	182.6922	7.688
1999-025BMF	1999-025AY	199.7687	1.752
1999-025CAH	1999-025BLV	213.9889	7.688
1999-025CBT	1999-025ALT	99.7949	6.402
1999-025CFR	1999-025DLT	105.7221	14.185
1999-025DDV	1999-025DYC	126.0361	16.031
1999-025DVV	1999-025AZB	129.6952	8.249

Table 3: Drag Sail Orbital Decay Comparison

	$A_D = 0.01\ [m^2]$	$A_D = 3\ [m^2]$
$Altitude\ [km]$	Time to Deorbit $T_D\ [hrs]$	
200	136.56	2.42
250	1185.61	7.24
300	21806.44	21.36
350	80219.76	79.23
400	275637.64	275.76
450	275637.62	928.84
500	887488.56	2968.56

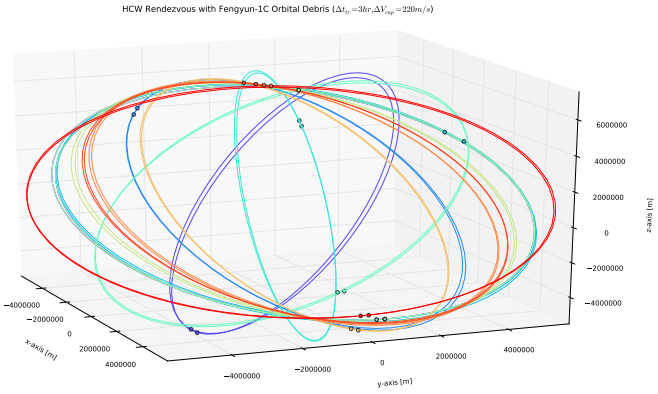


Fig. 14: $\Delta V_{cap} = 220m/s$

Fig. 15: This figure shows the variety of feasible trajectories with a potential delta-V of $\Delta V_{cap} = 220m/s$ in which the spacecraft can rendezvous with at least one additional debris fragment after it has departed from it original fragment.

17. ACKNOWLEDGEMENTS

Fundamental to this study's performance was the use of *Python Programming Language* [3]. Open source software truly is a amazing. Anyone who possesses a computer has a the potential to create something. In addition to Python, credit should also be given to *Matplotlib* [18] and *NumPy* [11]; the detailed figures included in this paper and the math that supported them would be much more difficult to produce otherwise. Lastly, credit should be given to PyEphem [19] for its TLE reader utily.

18. REFERENCES

[1] David Wright, "The Current Space Debris Situation," .

[2] Esa, "About space debris," .

[3] G V Rossum, "The Python Language Reference Manual," 2003.

[4] "jplephem 2.5 : Python Package Index," .

[5] Christopher Iliffe Sprague, Patrick Martin, Zac Amicucci, Jacob Hartley, Daniel Sparer, Adam Ware, and Hans Ofer, "Meridius," Tech. Rep., Rensselaer Polytechnic Institute, December 2015, Active Debris Removal CubeSat Design Report.

[6] Derek T Schmuland, Robert K Masse, and Charles G Sota, "SSC11-X-4 Hydrazine Propulsion Module for CubeSats," .

[7] "ESA Science & Technology: Electric Spacecraft Propulsion," .

[8] "3cm RF Ion Thruster BIT-3," Tech. Rep., Busek Co. Inc.

[9] Patrick Harkness, Malcolm McRobb, Paul Lützkendorf, Ross Milligan, Andrew Feeney, and Craig Clark, "Development status of AEOLDOS A deorbit module for small satellites," *Advances in Space Research*, vol. 54, no. 1, pp. 82–91, jul 2014.

[10] Eric W. Weisstein, "Sphere Point Picking," .

[11] David A. Vallado and David Finkleman, "A critical assessment of satellite drag and atmospheric density modeling," *Acta Astronautica*, vol. 95, pp. 141–165, feb 2014.

[12] "Atmospheric properties," Tech. Rep., Rocket and Space Technology, Tabulation of Atmospheric Density Scale Heights.

[13] Erwin Fehlberg, "Classical eight-and lower-order runge-kutta-nystroem formulas with stepsize control for special second-order differential equations," 1972.

[14] T S Kelso, "Analysis of the 2007 Chinese ASAT Test and the Impact of its Debris on the Space Environment," .

[15] "Terminal Guidance System for Satellite Rendezvous," *Journal of the Aerospace Sciences*, vol. 27, no. 9, pp. 653–658, sep 1960.

[16] Eric Jones, Travis Oliphant, Pearu Peterson, et al., "SciPy: Open source scientific tools for Python," 2001–, [Online; accessed 2016-03-07].

[17] "CelesTrak," .

[18] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[19] "PyEphem," .