MASTER'S THESIS

Learning based resilient manoeuvre planning for efficient orbit insertion targeting for initialization of a large scale constellation

Roger Le fort Ndayambaje

Engineering Physics and Electrical Engineering, master's level (120 credits)
2025

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering

LULEÅ UNIVERSITY OF TECHNOLOGY

[This page intentionally left blank]

Abstract

In applications like communication with a wide-ranging coverage, navigation, and Earth observation, large-scale satellite constellations mark the next frontier in the space community. This research presents an optimal intelligent satellite constellation initialization framework using minimum-energy motion planning. Combining traditional orbital mechanics with supervised learning, it addresses the placement of satellites in designated orbits to achieve efficiency in deployment by analyzing orbital maneuvers. The Hohmann and non-Hohmann transfers are analyzed for their applicability in moving satellites between orbital configurations. The study uses MATLAB simulations to validate the framework, employing neural networks to predict optimal target acquisition while accounting for the initial orbit and transfer trajectories, treated as input-output pairs of initial and target orbital parameters. Results demonstrate the efficacy of the proposed framework in determining minimum-energy paths while accounting for autonomy in high-level decision-making for satellite deployment. This integration of supervised learning provides valuable insight for initiating large-scale constellations, specifically for the deployment stage just after orbit insertion.

Contents

1 Introduction

The rapid expansion of global connectivity and the increasing demand for high-speed, wide-range communication have propelled the development of large-scale satellite constellations. Consisting of multiple satellites in coordinated orbits, these constellations offer unparalleled global coverage, reduced latency, and enhanced redundancy compared to sparsely distributed or objective driven single-satellite systems [22]. The technological advancement in small satellite system development has further accelerated this evolution, propelling the initiative towards Low Earth Orbit (LEO) for more efficient large-scale distributed network of satellites [33], transforming the landscape of space exploration and communication [40]. Satellite constellations provide high-speed Internet access worldwide, particularly benefiting underserved and remote areas, while also enabling continuous Earth observation [36]. Beyond telecommunications, satellite constellations are revolutionizing Earth observation, navigation, and scientific research, paving the way for a more interconnected and data-driven world.

Primarily deployed LEO satellites [14] offer key advantages over traditional geostationary satellites, including lower signal delay, widespread coverage, and cost-effective deployment. Organizations like SpaceX's Starlink and OneWeb are at the forefront of this movement [10, 34], continuously launching satellites to create robust, scalable communication infrastructures. However, the space industry advances, optimizing the deployment and management of these constellations remains a critical challenge, driving innovation in orbital mechanics, artificial intelligence, and autonomous mission planning [5].

As the number of satellite in constellations grow in scale and complexity, the need for higher autonomy becomes increasingly critical. Traditional ground-based control methods, such as Centralized Ground-based Operator in-the-Loop [30], rely heavily on globally distributed infrastructure and frequent communication links, which limit responsiveness and scalability. With increasing number of satellites, the computational load on ground stations rises significantly, and delays in decision-making can lead to inefficiencies in constellation management. To address these challenges, on-board autonomy [38] becomes essential. Recently, with significant progress in the area electronics system the space grade processors are becoming more elegant and space community is moving towards computational approaches for onboard computations [35, 4]. Toward this, there is a highly increasing need for On-

board Orbit Control with Distributed Constellation Guidance, which shifts the decision-making and control tasks to the satellites themselves, reducing reliance on ground operations and improving real-time adaptability. The highest level of autonomy, enables satellites to coordinate and optimize their positions without continuous ground intervention, ensuring efficient operations even in dynamic environments. This shift toward autonomous satellite coordination is crucial for enhancing scalability, resilience, and operational efficiency, making it a key enabler for the future of large-scale satellite constellations.

The constellation management typically considers the following primary operations such as, initialization, station keeping, reconfiguration and collision avoidance [23]. The scope of present research will primarily focus on a learning based autonomy perspective towards the constellation initialization. The initialization of a large-scale satellite constellation is a coordinated process designed to ensure safe deployment, controlled orbital insertion, and efficient phasing of satellites into their designated orbits. The process begins with a dedicated launcher carrying multiple satellites, typically inserting them into a predefined injection orbit.

Upon separation from the launcher, satellites enter a safe passive formation, where they maintain a stable relative distance to avoid collisions. Their solar arrays remain stowed, and onboard GNSS orbit determination enables them to assess their positions independently. While OneWeb follows a structured release pattern to achieve immediate passive formation, Starlink employs a more chaotic separation strategy, relying on differential drag and natural orbital drift for gradual spacing before activating propulsion systems.

Following initial deployment, each satellite undergoes a controlled ascent to its parking orbit, where it remains until verified for operational readiness. Once cleared, ground control sequentially authorizes the transition of satellites from parking to operational orbit, with the actuation commands executing the necessary maneuvers. Communication during this phase is facilitated via Inter-Satellite Links (ISL) and ground station updates, ensuring coordinated phasing into assigned orbital slots.

A crucial aspect of constellation initialization is the minimization of fuel consumption. With satellites gradually achieving their final positions, the constellation adopts a string-of-pearls formation [12], ensuring continuous global coverage and robust inter-satellite communication. Efficient constellation initialization is essential for ensuring the long-term success and sustainability of satellite networks. As the number of satellites in constella-

tions grow in scale, the complexity of their deployment increases, requiring precise coordination to optimize orbital placement while minimizing energy consumption. Traditional deployment methods rely heavily on ground-based control, but with the increasing number of satellites, there is a growing need for intelligent, autonomous strategies to streamline this process [3]. A systematic development of learning based autonomy for an optimized framework for constellation initialization, it is possible to enhance deployment efficiency, reduce operational costs, and extend the lifespan of the satellites. The following section outlines the key contributions of this research, focusing on the development of an energy-efficient and intelligent approach to satellite deployment and maneuvering.

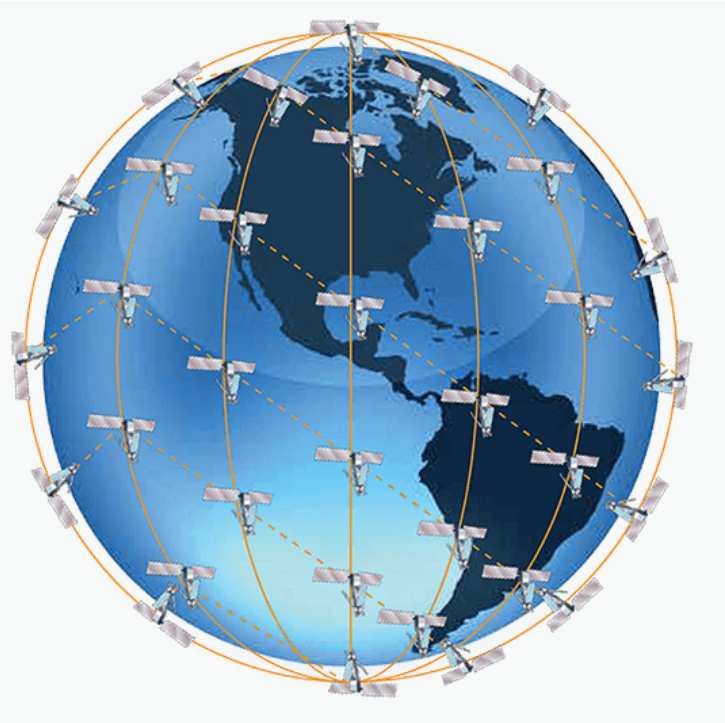

Figure 1: Example of a satellite constellation providing global coverage[41].

1.1 Aim of the Thesis

Satellite deployment involves launching multiple satellites into space using rockets or other launch vehicles. Once in space, these satellites undergo or-

bital transfer maneuvers to place them into their designated orbits. The process of orbital transfer requires precise calculations and adjustments to achieve the desired orbital parameters, such as altitude, inclination, and eccentricity[1, 7]. The aim was to develop an optimal constellation initialization framework by

- Establishing a suitable optimization framework incorporating minimum energy
- Intelligent motion planning for individual satellites, targeting optimal allocation into their respective orbits.
- The verification and validation of the framework in a simulated environment

To fully leverage the potential of these constellations, each satellite must be precisely positioned in a predefined orbital slot, a process known as constellation initialization. This task is complex, involving the precise deployment and maneuvering of each satellite, especially as the number of satellites in constellations increases. The operational demands and cost of these systems underscore the need for efficient methods of constellation initialization [21, 24].

Constellation initialization involves deploying satellites into their designated orbits with minimal energy expenditure, which is critical to maximizing the operational life of each satellite. Delta-v, the measure of velocity change required for orbital maneuvers, directly affects the fuel consumption and cost of these missions[20, 27, 26]

1.2 Learning-Based Maneuver Planning

Advances in machine learning have enabled new possibilities for adaptive, real-time maneuver planning in constellation management. Supervised neural networks learning models can be trained on historical and simulated maneuver data to predict delta-V values for a variety of orbital configurations [19]. Another value from machine learning algorithms is that they are improving satellite positioning and link maintenance, enabling constellations to

adapt autonomously to environmental changes [11]. The integration of machine learning and AI in space exploration has significantly enhanced mission efficiency, accuracy, and autonomy. AI applications have revolutionized satellite operations, data analysis, robotics, and autonomous spacecraft, paving the way for future advancements and ambitious exploration missions[2].

2 Theoretical background

2.1 The laws of orbital dynamics

When calculating how objects move through space, the laws of orbital dynamics are the fundamental principles that guide the motion of those objects. These laws were formed through the work of different physicists by understanding how our solar system came into existence. Later on with Newtons law of gravitation, Johannes Kepler formulated three laws that describes how the planets move around the sun[39].

2.1.1 Newton's Laws of Motion

Newton's three laws of motion are[9]:

1. an object in motion remains in that motion, and an object at rest remains at rest remains so unless external forces are applied to the object

2. an object's acceleration(a) is proportional to the overall applied force($\mathbf{F}$).

$$\mathbf{F} = ma \tag{1}$$

3. Two objects acting on each other with forces subjecting each other equally large but oppositely directed forces.

Newton also formulated a universal law of gravitation[25]:

- every point mass in the universe attracts every other point mass with a force that is directly proportional to the product of their masses and inversely proportional to the square of the distance between them.

$$\mathbf{F} = \frac{G\ M\ m}{r^2} \tag{2}$$

where:

- F is the magnitude of the gravitational force between the two masses
- G is the gravitational constant and has value of $6.6742 * 10^{11} m^3/kg * s^2$
- m and M are the masses of the two objects,
- r is the distance between the centers of the two masses.

2.1.2 Kepler's Laws of Planetary Motion

Kepler's Laws of Planetary Motion are as follows[15]:

1. Planets orbit the Sun in ellipses with the Sun at one focus.
2. Planets sweep out equal areas in equal times, moving faster when closer to the Sun[57].
3. The square of the orbital period is proportional to the cube of the semi-major axis of the orbit.

When calculating how objects move in orbits such as satellites, Kepler's Laws of Planetary Motion and Newton's laws are the basic nature laws followed.

2.2 The motion of a satellite around Earth

The motion of a satellite around Earth can be described using coordinate systems oriented towards orbital mechanics and following the laws of orbital dynamics. there are different coordinate systems and using these systems and the transformations between them, we can accurately describe and predict the satellite's position, velocity, and orbital dynamics around Earth.

2.2.1 The Cartesian frames

When describing the satellite motion around earth, the Cartesian frames are used by looking at a satellite in a three-dimensional coordinate system fixed relative to earth. This way is used for predicting and studying satellite trajectories.

In the inertial frame, the Earth's center is the origin and stationary, The Y-axis and X axis lie in the equatorial plane and are perpendicular to each other with X-axis pointing toward the vernal equinox. The Z-axis points toward

the North Pole and perpendicular to the equatorial plane. The satellite's position and motion are described relative to this fixed coordinate system.

The position of the satellite at any time (t) is given by the position vector

$$\mathbf{r} = (x\ y\ z) \tag{3}$$

and the velocity of the satellite is given by the time derivative of the position vector

$$\dot{\mathbf{r}} = (v_x\ v_y\ v_z) = (\dot{x}\ \dot{y}\ \dot{z}) \tag{4}$$

and the angular momentum can be given by

$$\mathbf{h} = \mathbf{r}\ \mathrm{x}\ \dot{\mathbf{r}} \tag{5}$$

The motion of the satellite is governed by Newton's second law of motion combined with the gravitational force exerted by the Earth[20]. The corresponding acceleration $\mathbf{a}$ can be given by :

$$\mathbf{a} = \ddot{\mathbf{r}} = \frac{d^2\mathbf{r}}{dt^2} \tag{6}$$

Substituting the expression for the gravitational force into Newton's second law gives:

$$\ddot{\mathbf{r}} = -\frac{G\ M}{r^3}\ \mathbf{r} \tag{7}$$

This equation can be broken down into its Cartesian components:

$$\begin{aligned} \ddot{x}(t) &= -\frac{GM}{r^3}x(t) \\ \ddot{y}(t) &= -\frac{GM}{r^3}y(t) \\ \ddot{z}(t) &= -\frac{GM}{r^3}z(t) \end{aligned} \tag{8}$$

These are the differential equations describing the motion of the satellite in each of the Cartesian directions.

The solution to these equations of motion describes the satellite's orbit, which is generally, circular, an ellipse, parabola, or hyperbola, depending on the energy of the satellite. For most satellites in Earth orbit, the orbit is elliptical, with Earth at one focus of the ellipse.

2.2.2 Orbital elements

The shape and orientation of the orbit can also be described by the orbital elements [20]:

- **Semi-major axis (a)**: Describes the size of the orbit.
- **Eccentricity (e)**: Describes the shape of the orbit.
- **Inclination (i)**: The tilt of the orbit relative to the Earth's equatorial plane.
- **Right ascension of the ascending node (Ω)**: The horizontal orientation of the orbit in the equatorial plane.
- **Argument of periapsis (ω)**: The orientation of the orbit within the orbital plane.
- **True anomaly (θ)**: The position of the satellite along the orbit at a specific time.

the path of satellite around earth can be defined in terms of eccentricity and true anomaly shown in equation 9

$$r = \frac{h^2}{G\ M(1 + e\ cos\theta)} \tag{9}$$

Where G, M, and e are constant in a specific orbit.

2.2.3 Satellite Orbits in Cartesian Coordinates

A satellite's orbit can be described as elliptical orbit, which is when the orbit of satellite forms an ellipse and remains bound to earth in one of the focus[20]. In Cartesian coordinates with origin at the center C as shown in figure 2 x and y can be expressed in equation

$$\begin{aligned} x &= ae + rcos\theta = a\frac{e + cos\theta}{1 + ecos\theta} \\ y &= rsin\theta = bsin\theta\frac{\sqrt{1 - e^2}}{1 + ecos\theta} \end{aligned} \tag{10}$$

With satellite's position (x, y), a is the semi-major axis and b is the semi-minor axis, the equation for an ellipse can be given in Cartesian coordinates as the usual formula for an ellipse shown in equation 11

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{11}$$

Elliptical orbit is described by eccentricity(e) $0 < e < 1$ and the semi-major axis(a) as shown in figure 2. Periapsis(r_p) is the closest point to the central body and Apopsis(r_a) is the farthest point. on orbits around earth, they are called perigee and apogee and can be described using equation 9 and that $\theta = 0$ respectively $\theta = 180$:

$$r_p = \frac{h^2}{G\ M(1+e)} \tag{12}$$

$$r_a = \frac{h^2}{G\ M(1-e)} \tag{13}$$

Given r_a and r_p the the semi-major axis(a) can be found by equation 14

$$a = \frac{r_a + r_p}{2} \tag{14}$$

Using equation 13 and equation 12, eccentricity can be derived as shown in equation 15

$$e = \frac{r_a - r_p}{r_a + r_p} \tag{15}$$

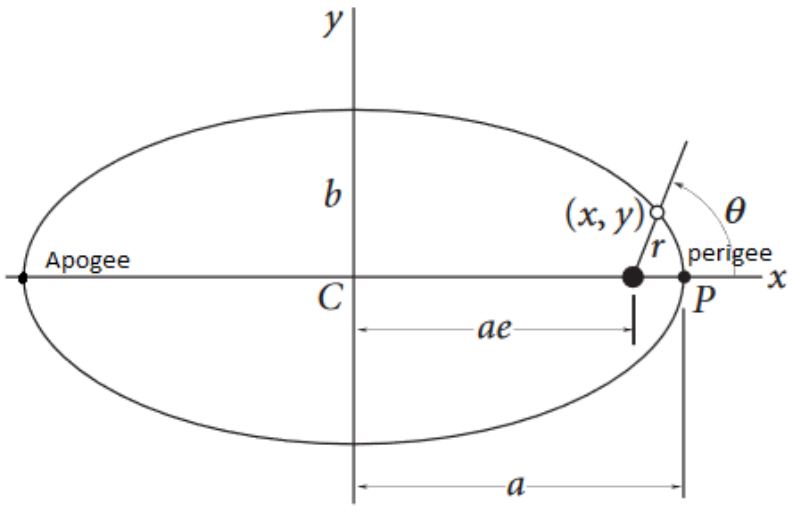

Figure 2: illustration of elliptical orbit in Cartesian coordinates [20]

Conservation of angular momentum in a two-body system explain why a satellite moves slowest at the Apogee and moves the fastest at the Perigee. The velocity$\dot{r}$for the satellite in orbit can be separated in 2 components as shown in figure 3, one component normal to the position vector ($\mathbf{r}$),v_r and the other perpendicular to $\mathbf{r}$, $v_\perp$ [20].

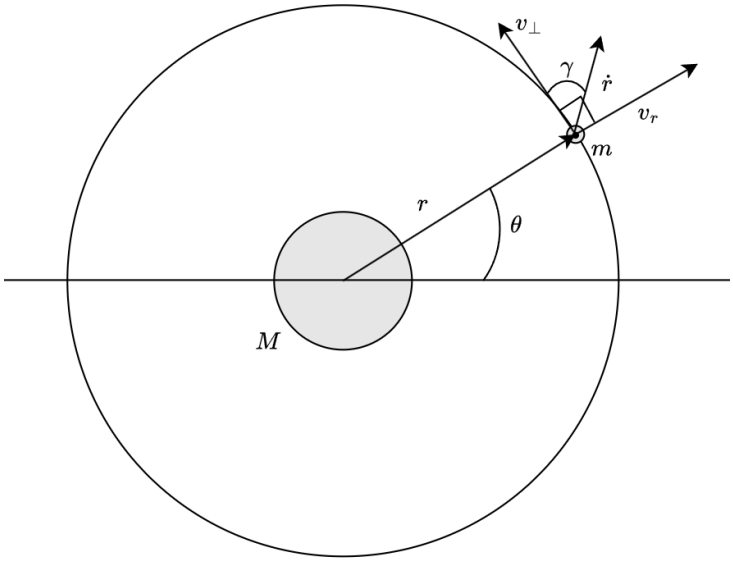

Figure 3: illustration of velocity vectors of satellite with mass m on orbit around earth with mass M

The velocity components can be given as shown in equation16 and 17

$$v_\perp = \frac{h}{r} = \frac{G\ M}{h}(1 + e\ cos\theta) \tag{16}$$

$$v_r = \frac{G\ M}{h} e\ sin\theta \tag{17}$$

the flight path angle γ shown in figure 3 can be calculate by equation 18

$$tan\gamma = \frac{v_r}{v_\perp} = \frac{e\ sin\theta}{1 + e\ cos\theta} \tag{18}$$

And if v_r and $v_\perp$ are given then velocity v can be calculated as shown in equation 19

$$v = \sqrt{v_r^2 + v_\perp^2} \tag{19}$$

There is an another coordinate system called perifocal frame which is very useful in describing the motion of a satellite in orbit. The perifocal frame is centered at the focus of the ellipse, which is earth[8]. The perifocal frame is defined by three orthogonal unit vectors shown in figure 4 :

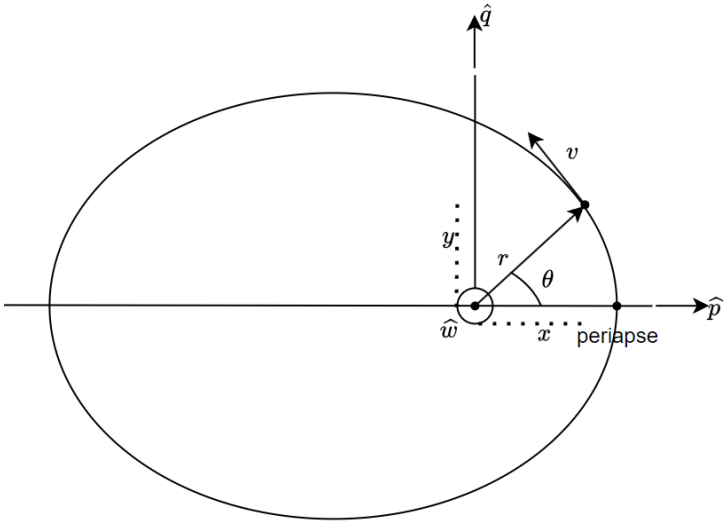

Figure 4: illustration of elliptical orbit in perifocal frames

P-axis $(\widehat{\mathbf{p}})$ points from the focus of the ellipse toward the periapsis.
Q-axis $(\widehat{\mathbf{q}})$ Lies in the orbital plane, perpendicular to the P-axis, and points in the direction of the satellite's motion at periapsis.
W-axis $(\widehat{\mathbf{w}})$ Is perpendicular to the orbital plane.
These three axis together forms a right-handed coordinate system where $\widehat{w}$ is the unit vector in the direction of the angular momentum vector[20].
The position vector $\mathbf{r}$ can be written in perifocal frame as equation 20

$$\mathbf{r} = x\widehat{\mathbf{p}} + y\widehat{\mathbf{q}} = r\ cos\theta\ \widehat{\mathbf{p}} + r\ sin\theta\ \widehat{\mathbf{q}} \tag{20}$$

where the orbital radius r is given by equation 9 and θ is the true anomaly. Putting equation 9 into equation 20, r can be defined as

$$\mathbf{r} = \frac{h^2}{G\ M(1 + e\ cos\theta)}(cos\theta\ \widehat{\mathbf{p}} + sin\theta\ \widehat{\mathbf{q}}) \tag{21}$$

The velocity is given as time derivative of the position in equation

$$\dot{\mathbf{r}} = \dot{x}\widehat{\mathbf{p}} + \dot{y}\widehat{\mathbf{q}} = \frac{G\ M}{h}((-sin\theta\ \widehat{\mathbf{p}}) + (e + cos\theta\ )\widehat{\mathbf{q}}) \tag{22}$$

And for one orbit, the orbital period can be expressed in term of a shown in equation 23

$$T = \frac{2\pi}{\sqrt{GM}} a^{3/2} \tag{23}$$

A satellite in orbit has also a specific mechanical energy ϵ which is constant for the orbit, derived in [42], ϵ is shown in equation 24

$$\epsilon = \frac{v^2}{2} - \frac{GM}{r} = -\frac{GM}{2a} \tag{24}$$

The velocity for the satellite in its orbit can then be written as:

$$v = \sqrt{GM\left(\frac{2}{r} - \frac{1}{a}\right)} \tag{25}$$

For circular Orbit, the distance r is constant, the eccentricity ($e = 0$) and the motion in the X, Y, and Z directions follows simple harmonic motion. Suppose a satellite is in a circular orbit around Earth with a radius r. Then using equation 9 and setting $e = 0$, r becomes

$$r_{circular} = \frac{h^2}{GM} \tag{26}$$

The velocity becomes constant at every point on the orbit. Since for circular orbits $a = r$,velocity can be described using equation 25 as:

$$v_{circular} = \sqrt{\frac{GM}{r}} \tag{27}$$

Then in Cartesian coordinates with true anomaly θ, the position might be expressed as:

$$\begin{aligned} x &= rcos\theta \\ y &= rsin\theta \end{aligned} \tag{28}$$

Since the circumference of circle is known as $2\pi r$, The time for one orbit can be in terms of circumference as shown in equation 29:

$$T_{circular} = \frac{2\pi}{\sqrt{GM}} r^{3/2} \tag{29}$$

2.3 Orbital maneuvers

In orbital mechanics, maneuver planning involves calculating the series of burns (thrusts) required to move a satellite from one orbit to another. The key metric in evaluating maneuver efficiency is delta-v, which represents the change in velocity. Delta-v is directly related to fuel consumption[6]; hence, minimizing delta-v is critical for cost-effective constellation deployment.[20]

2.3.1 Hohman transfer

One of the fundamental maneuvers in orbital dynamics is the Hohmann transfer, named after German engineer Walter Hohmann[28]. A Hohmann transfer involves moving a spacecraft from one circular orbit to another by performing two engine burns: one to increase the spacecraft's velocity and raise its orbit, and another to decrease velocity and lower its orbit. This maneuver is highly efficient in terms of delta-v and is commonly used for interplanetary missions[20, 55].

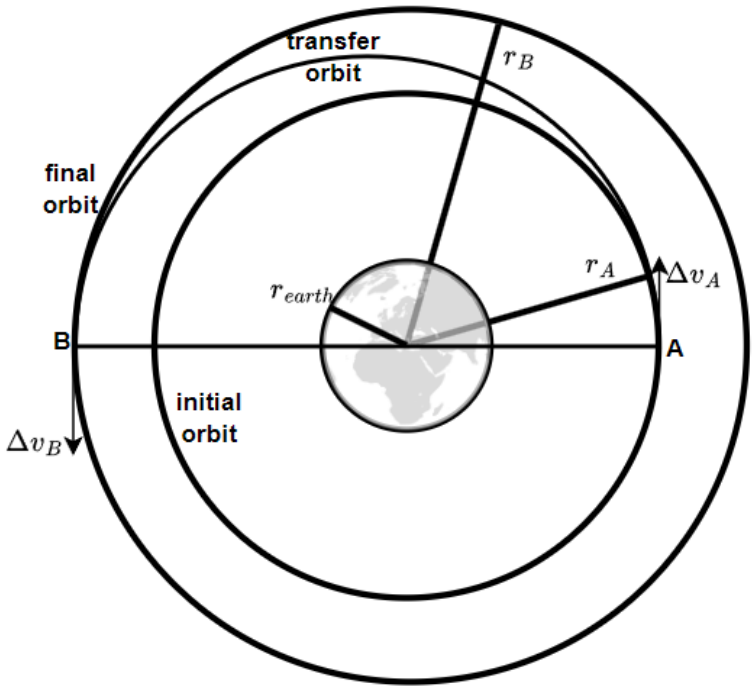

Figure 5: illustration of Hohmann transfer from position A to position B

Suppose there is two circular orbit as illustrated in figure 5, where the initial orbit has radius r_A, from the center of earth, and the final orbit is r_B, then the semi-major axis of the orbit,a_{ht}, of a Hohmann transfer's orbit can be defined by equation 30

$$a_{ht} = \frac{r_B + r_A}{2} \tag{30}$$

By using equation 30 and defining apogee as r_B is and perigee as r_A, the transfer time $T_{transfer}$ can be calculated using equation 23 and taking into account that only half of orbit is needed and that the time it ta takes from perigee to apogee is the same from apogee to perigee, then only half of elliptical period is needed as shown in equation 31

$$T_{transfer} = \frac{T}{2} = \pi\sqrt{\frac{a_{ht}^3}{GM}} \tag{31}$$

For velocities and changes in velocities between two circular orbits, it is

calculated by following equations[42]:
For initial velocity on position A:

$$v_A = \sqrt{\frac{GM}{r_A}} \tag{32}$$

For transfer orbit velocity on position A:

$$v_{ht_A} = \sqrt{GM\left(\frac{2}{r_A} - \frac{1}{a_{ht}}\right)} \tag{33}$$

The the changes in velocities in position A is:

$$\Delta v_A = v_{ht_A} - v_A \tag{34}$$

For final velocity on position B:

$$v_B = \sqrt{\frac{GM}{r_B}} \tag{35}$$

For transfer orbit velocity on position B:

$$v_{ht_B} = \sqrt{GM\left(\frac{2}{r_B} - \frac{1}{a_{ht}}\right)} \tag{36}$$

The the changes in velocities in position B is:

$$\Delta v_B = v_B - v_{ht_B} \tag{37}$$

This finally gives the required total delta-v Δv_{tot} for the the trajectory:

$$\Delta v_{tot} = \Delta v_A + \Delta v_B \tag{38}$$

2.3.2 Example Hohman transfer for one Satellite

An example of Hohmann transfer is presented here, where figure 6 and figure 7

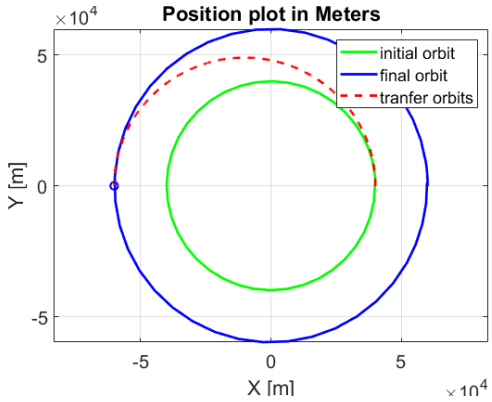

Figure 6: Hohmann transfer between two circular orbits when putting on only Δv_A

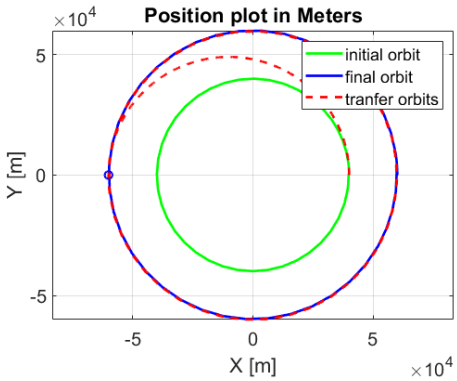

Figure 7: Hohmann transfer between two circular orbits after putting on only Δv_B

Figure 6 shows Hohmann maneuver between two circular orbits, where the initial position is on lower orbit than the target orbit. The Δv_{total} was calculated to be 1.81 km/s.

2.3.3 Non-Hohman transfer with common apse line

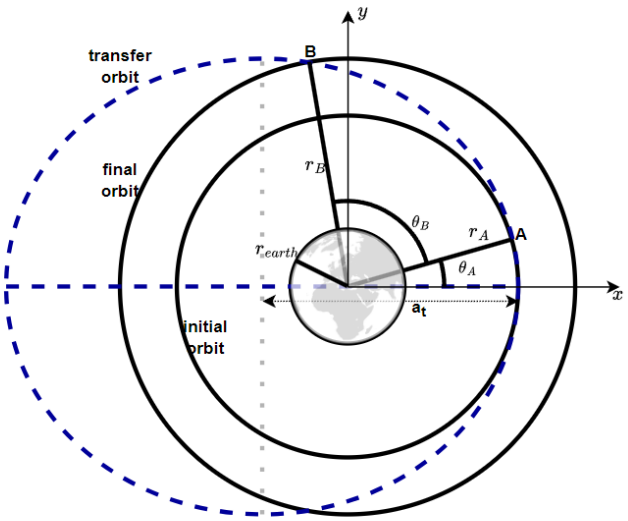

Figure 8: illustration of Non-Hohmann transfer between two circular orbits from position A to position B. r_A represents the radius of initial orbit, r_B represents the radius of final orbit. Blue orbit represents the transfer orbit

Another transfer maneuver used is a non-Hohmann transfer with a common apse line, meaning that the periapsis and apoapsis of the initial and final orbit are aligned as shown in figure 8. This method also works well with circular orbits and it is done by making the initial position the peri-apse of the transfer orbit and the transfer orbit is chosen so that both the two position respective orbit crosses through the transfer orbit. Then equations 41 and 42 are used to find the angular momentum and eccentricity[56].

Another reason non-Hohmann transfer was considered was because when launching multiple satellite from initial orbit, it may happen that the available slots on the final orbit may not be possible with Hohmann transfer even if it is fuel efficient, this maneuver can then be more flexible and sometimes more efficient when if the orbits share the same apsidal line. The key equation for this type of maneuver is equation 25.

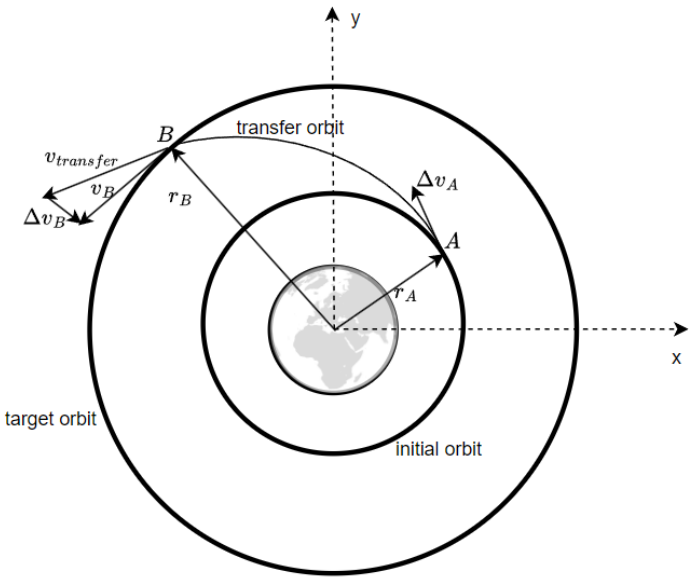

Figure 9: illustration of Non-Hohmann transfer between two circular orbits from position A to position B

Suppose there is two circular orbits in the same plane as shown in figure 9. Since both orbits are circular, then it is granted that the transfer orbit is elliptical. The semi-major axis can be chosen to be bigger than the Hohmann transfer orbit, with true anomalies θ_A and θ_B on transfer orbit. Where position A is the initial position and B is the destination on the target orbit, then using equation 9, position can be calculated as shown in equations 39 and 40:

$$r_A = \frac{h_t^2}{G\ M(1 + e_t\ cos\theta_A)} \tag{39}$$

$$r_B = \frac{h_t^2}{G\ M(1 + e_t\ cos\theta_B)} \tag{40}$$

where h_t and e_t is the angular momentum and eccentricity of the transfer orbit. h_t and e_t can be solved as:

$$e_t = \frac{r_B - r_A}{r_A\ cos\theta_A - r_B\ cos\theta_B} \tag{41}$$

$$h_t = \sqrt{G\ M\ r_A r_B}\sqrt{\frac{cos\theta_A - cos\theta_B}{r_A\ cos\theta_A - r_B\ cos\theta_B}} \tag{42}$$

The velocity on transfer orbit for position A and position B is calculated using equation 25, and velocity for initial and target orbit can be calculated using equation 27.

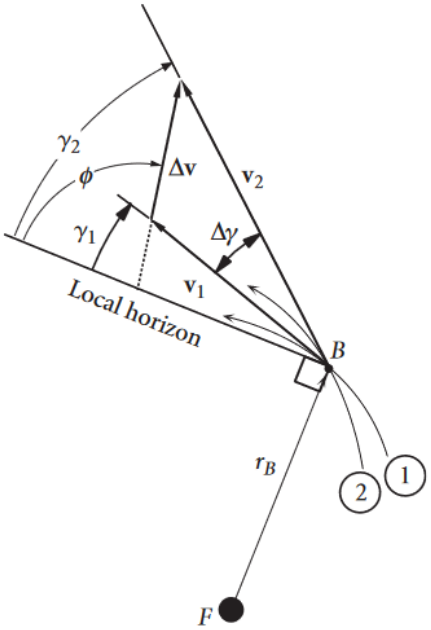

Figure 10: illustration of velocity changes at the intersection between two orbits, described in terms of vector diagram and flight angle [20]

Since the flight angles on the intersection point for both orbits are not parallel like in Hohmann transfer, then the Δvs calculated by equation 43 and the geometry is illustrated in figure 10, where $\mathbf{v_1}$ and $\mathbf{v_2}$ are the velocity vectors from the two orbits and $\Delta\mathbf{v}$ is the magnitude of change in velocity . [20]

$$\Delta v = ||\mathbf{v_2} - \mathbf{v_1}|| \tag{43}$$

Position A can be chosen as the Periapse for transfer orbit the ΔV_A becomes

$$\Delta v_A = |v_{t_A} - v_1| \tag{44}$$

Where v_1 is velocity for initial orbit on position A and v_{t_A} is velocity for transfer orbit on position A.

For intersection on Position B, angles $\gamma 1$ and $\gamma 2$ are flight path angles for orbits at the intersection as shown in figure 10, velocity for transfer orbit on position B is shown as $v_2 = v_{t_B}$ and $v_1 = v_{2_B}$ is velocity for target orbit on target position B.
Using geometry from figure 10, where $\Delta\gamma = \gamma 2 - \gamma 1$, $v_1 = ||\mathbf{v_1}||$ and $v_2 = ||\mathbf{v_2}||$. The law of cosine can be used to find that equation 43 can be rewritten as

$$\Delta v_B = \sqrt{v_2^2 + v_1^2 - 2v_1 v_2 cos(\Delta\gamma)} \tag{45}$$

After calculating Δv_A and Δv_B The total Δv (Δv_{tot}) can be calculated using equation 46 below.

$$\Delta v_{tot} = \Delta v_A + \Delta v_B \tag{46}$$

For every initial position multiple target positions were given and the minimum Δv_{tot} was calculated. The position belonging to the minimum Δv_{tot} was chosen as the final target position since minimum Δv_{tot} also corresponds to the minimum energy. When each initial position had the final target orbit with the minimum energy, both these results were used as data for supervised learning model.

2.3.4 Example of non-Hohmann transfer with multiple target positions

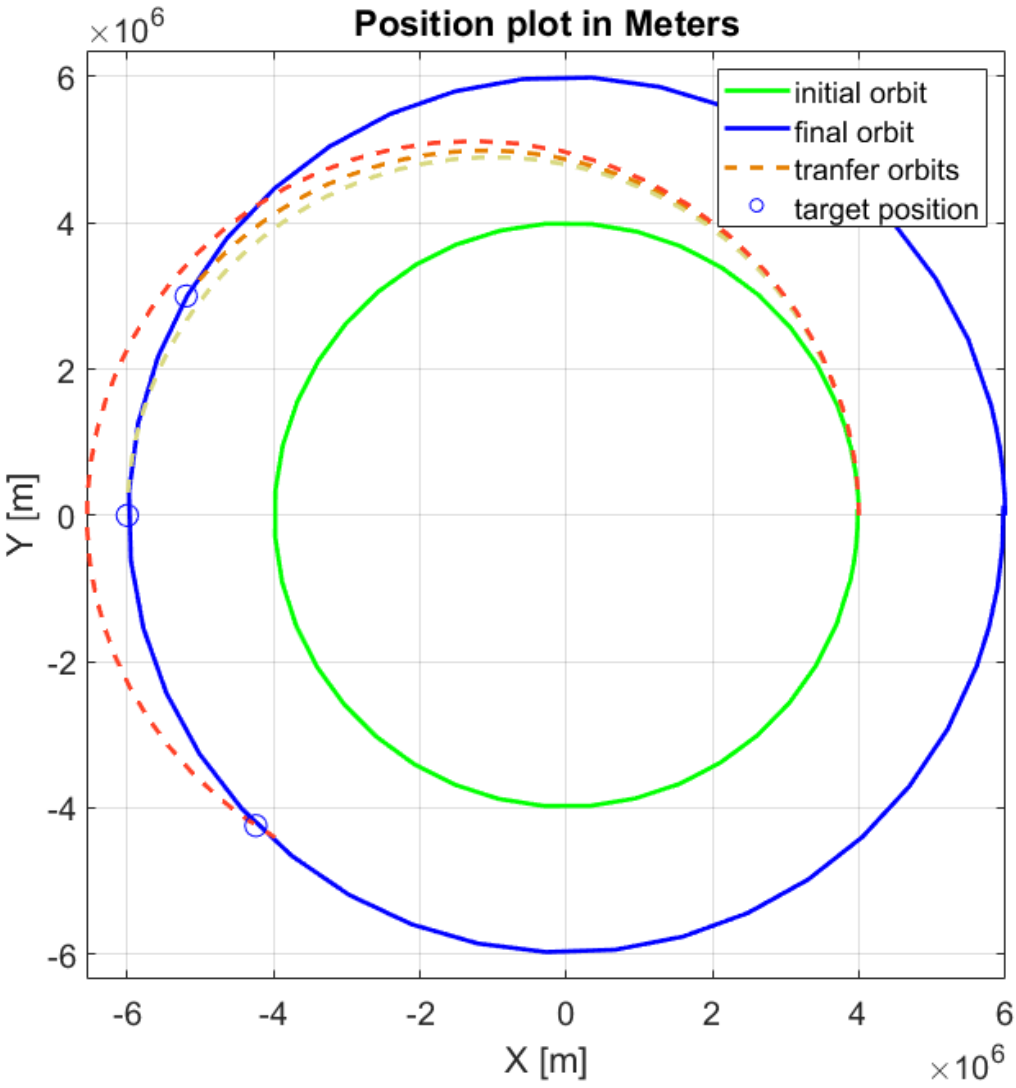

Figure 11: Non-Hohmann while adding Δv_A on orbit 1, the green orbit shows the initial orbit, the blue is the final orbit and the dashed ones are the transfer orbits from lower orbit to a higher orbit.

Figure 11 shows three target position given only one initial position, in this case then non-n transfer is used to calculate the trajectories, Δv_A is added

then when the trajectories reach the final positions, Δv_B is added to match the velocity from the final orbit shown figure 12.

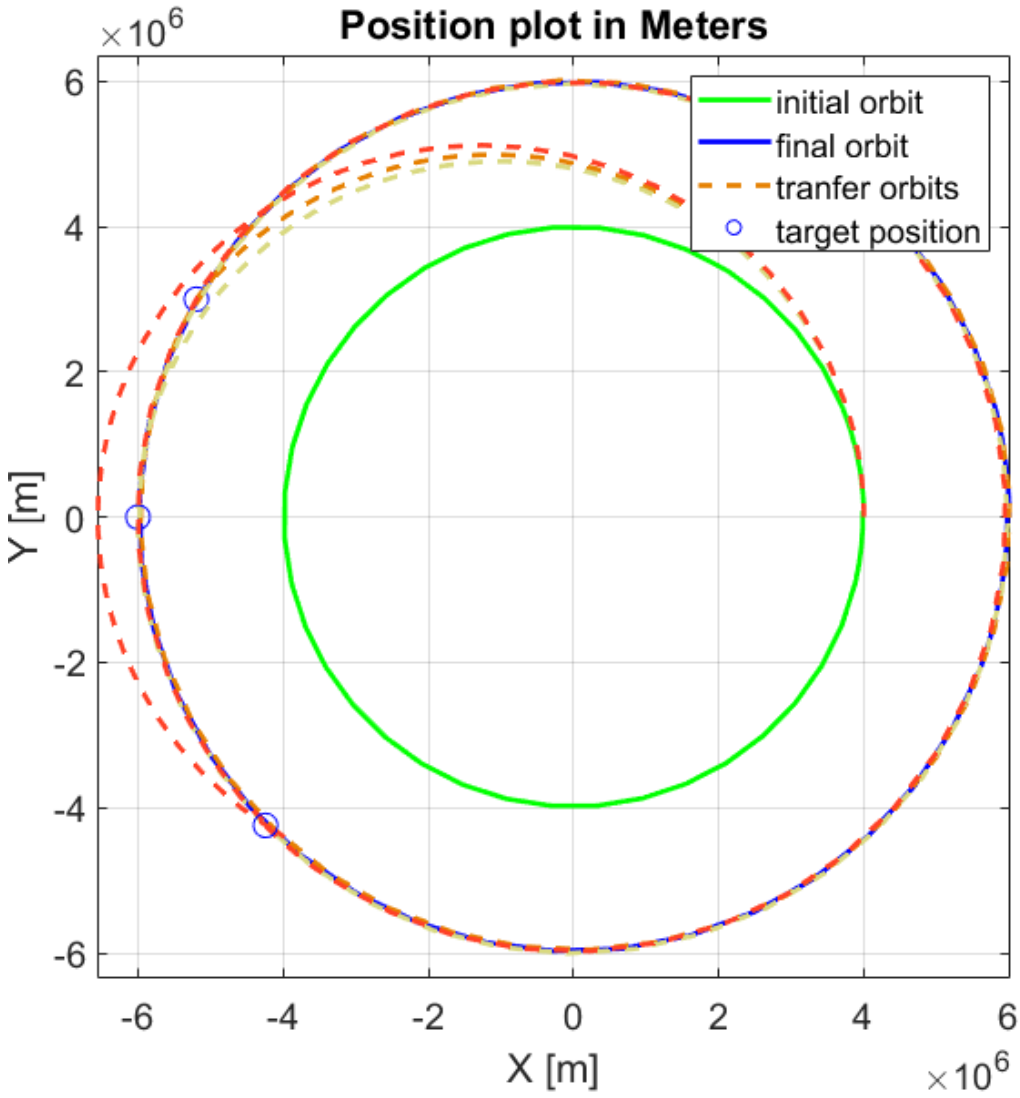

Figure 12: Non-Hohmann while adding Δv_B on orbit 2. The green orbit shows the initial orbit, the blue is the final orbit and the dashed ones are the transfer orbits from lower orbit to a higher orbit

Table 1: The specific Δvs on specific orbital maneuver shown in figure 11

angle(°)	$\Delta v_A(km/s)$	$\Delta v_B(km/s)$	$\Delta v_{total}(km/s)$
135	1.06	0.696	1.76
180	0.953	0.860	1.81
225	1.15	0.575	1.72

Table 1 shows the Δv_{total} calculated from three Non-Hohmann maneuver between two circular orbits, where there is only one initial position on lower orbit then three targets on the final orbit.

2.4 Supervised learning

Supervised learning is used in this research, It is a type of machine learning where the training model takes in known inputs and outputs[18, 31, 54], where each output corresponds to its own input as shown in figure 13.

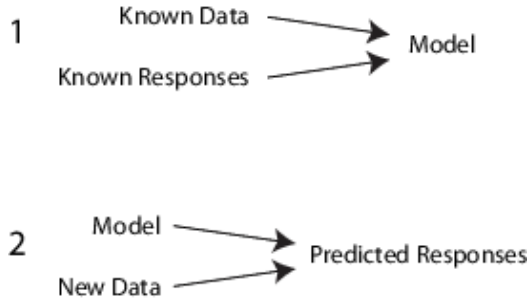

Figure 13: Supervised Learning model [45]

This training data with input-output pairs is used to learn correlation between inputs and outputs. By using internal weights to minimize the error between actual outputs and predicted outputs, the model can accurately predict the outputs for new inputs[19]. The data is usually split between training sets, explained above, validation sets and testing sets as shown in figure 14. This evaluates how well the model predicts new data[46]. Usually the training sets have the most percentage, over 60% of the dataset, and the rest is divided between validation and testing sets[47]. MATLAB offers good and powerful toolbox for neural networks[48], and one used in this research is MATLAB's neural network fitting tool(nftool) which is good when

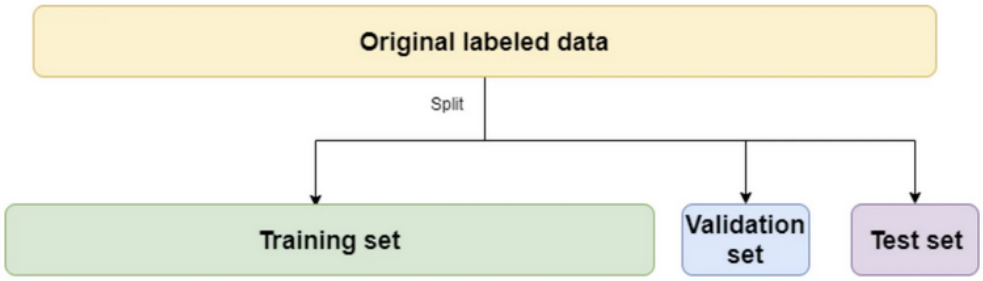

Figure 14: Separation of the datasets in the model[58]

a neural network is learning to map between continuous input and output data and uses linear regression to predict the relation betweeen inputs and outputs[17, 49]. There are different algorithms are designed to minimize the error between predicted and actual values, enabling the neural network to learn complex nonlinear relationships from data[29].

The default algorithm used in NFTool is the Levenberg-Marquardt algorithm[16], which is particularly well-suited for small- to medium-sized datasets. This method combines the benefits of gradient descent and Newton's method, striking a balance between fast convergence and robustness against local minima. Its ability to efficiently solve nonlinear least squares problems makes it a preferred choice for many regression tasks[51].

For datasets that may be noisy or prone to overfitting, NFTool offers Bayesian regularization as an alternative training method[37]. Bayesian regularization modifies the loss function by introducing regularization terms, penalizing overly complex models. This approach automatically adjusts the model complexity during training, ensuring that the network generalizes well even when the data contains significant variability. It is particularly effective for applications where noise in the data could otherwise impair the network's predictive accuracy[50].

Another training option available in NFTool is the scaled conjugate gradient (SCG) algorithm[32]. SCG is a memory-efficient optimization technique that employs a conjugate gradient approach to update weights and biases. Unlike the Levenberg-Marquardt algorithm, SCG is well-suited for larger

datasets where memory constraints may limit the applicability of second-order methods. By efficiently managing memory usage and computational overhead, SCG enables NFTool to handle training tasks involving extensive datasets[52].

Each of these algorithms is complemented by automatic data preprocessing steps, such as normalization and dataset splitting, ensuring stable and efficient training. You can choose to select Levenberg-Marquardt for its speed, Bayesian regularization for its robustness, or scaled conjugate gradient for its scalability, nftool offers users the flexibility to customize training algorithms to the specific needs of their regression problems. These algorithms collectively empower NFTool to deliver reliable and high-performing models for a wide range of applications.

The structure for neural network fitting tool has input layer, Hidden layers and output layer as shown in figure 15.

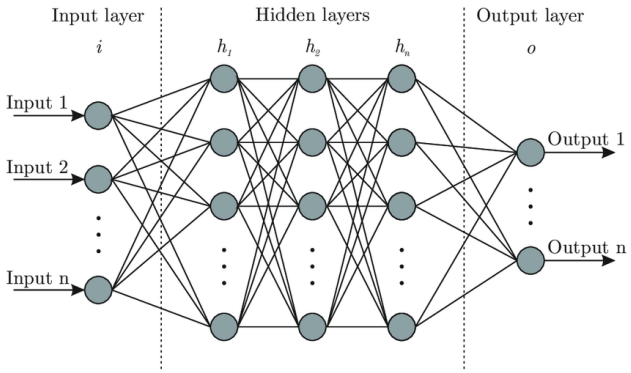

Figure 15: illustration of different layers in supervised learning [43]

Input layer takes in the inputs and the amount of neurons corresponds to the amount of inputs. The hidden layers have neurons that apply weights, biases and activation functions as shown in figure 16, the amount of hidden layers can be specified as the user wishes. The output layer is used to directly map the outputs to continuous values. The training network involves also rearranging the weights and biases by comparing the predicted outputs and the actual desired outputs[53].

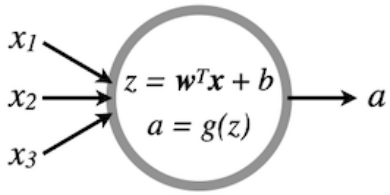

Figure 16: illustration of inside hidden layer showing applied weights, biases and activation functions on the inputs[43]

These weights are parameters that adjust the input signals, determining their influence on the neuron's output. Biases are additional parameters that allow the activation function to be shifted, enabling the model to fit the data more accurately. Activation Functions are the functions that introduce non-linearity into the model, enabling it to learn complex patterns[13].

In neural network training a fundamental concept that refers to a full cycle through the dataset is called a epoch. By training over multiple epochs, neural networks can adjust parameters iteratively, leading to better performance as they learn from complex, high-dimensional data[44]. During an epoch, the network processes each sample in the dataset, calculating the error based on the difference between its predicted outputs and the actual labels. Using this error information, the network then adjusts its parameters (weights and biases) to improve its accuracy. This process of iterating through the dataset is repeated multiple times to allow the network to refine its understanding of patterns within the data[19].

3 Methods

To optimize satellite placement in their designated orbits, two primary maneuver strategies are implemented, the Hohmann transfer and the Non-Hohmann transfer. The Hohmann transfer is calculated to determine the ΔV_{total} required for moving satellites between two circular orbits. For scenarios where Hohmann transfer is less effective, a Non-Hohmann transfer is employed. This strategy introduces greater flexibility, particularly when multiple satellites need to be placed in specific orbital slots or when orbits share a common apsidal line. The Non-Hohmann method involves solving for angular momentum and eccentricity to design elliptical transfer orbits that achieve the desired transitions efficiently. By computing the semi-major axis of the transfer orbit and the associated velocities at initial and final positions, the required energy expenditure is minimized. To validate these strategies, MATLAB simulations are performed. These simulations calculate trajectories, Δv requirements, and orbital dynamics with various initial and target positions, providing a robust dataset for further analysis.

3.1 Supervised Learning Integration

Supervised learning is integrated into the maneuver planning process. A dataset of input-output pairs is generated from the MATLAB simulations, where the initial and target orbital parameters serve as inputs. This dataset is used to train a neural network model using the MATLAB Neural Network Fitting Tool (*nftool*). The neural network comprises an input layer representing orbital parameters, hidden layers applying adaptive weights and biases, and an output layer predicting Δv and target positions. The model is trained over multiple iterations, with the dataset divided into training, validation, and testing subsets. Through this iterative training process, the model learns to generalize from the data, enabling it to predict optimal maneuvers for new orbital configurations.

The effectiveness of the supervised learning model is evaluated by comparing the predictions of the model with the results of MATLAB simulations. The Δv values for the predicted trajectories are analyzed for accuracy and efficiency. The quality of the model is examined to identify potential areas for improvement.

4 Result

Figure 17 shows the results of this work in which the supervised model is trained multiple times before giving these results. The model is given 100 initial positions and 100 new target positions were given out.

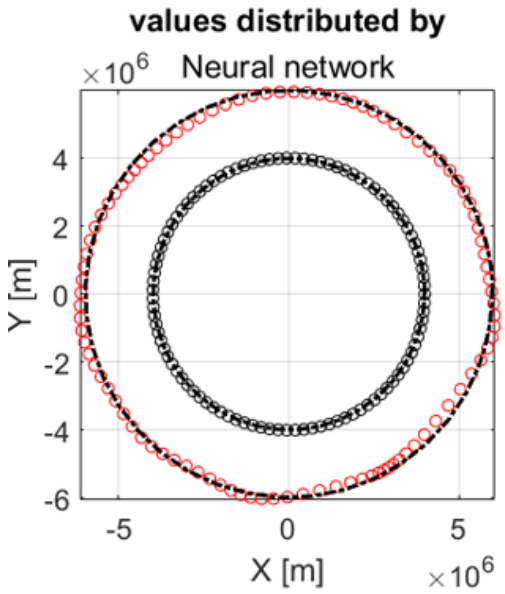

Figure 17: Distribution made by supervised model where for ever initial orbit, there is a target position belonging to that initial position. The model is given hundred initial position and associated target position were given out. The black inner small round circles represents initial positions and The red outer small round circles represent target positions

4.1 Simulations

This section shows the result calculated from the simulations, The result are given by using two circular orbit in xy-coordinates. The initial orbit has a radius of $r1$ and the target orbit has a radius of r_2 from the earth's surface.

4.1.1 Constants and Initial Setup

The key constants used are as follow

$$
\begin{aligned}
r_e arth &= 6.371e6\,\text{m} \quad \text{(radius of Earth)} \\
r_1 &= (4,000,000 + r_e arth)\,\text{m} \quad \text{(initial orbit radius)} \\
r_2 &= (6,000,000 + r_e arth)\,\text{m} \quad \text{(target orbit radius)} \\
M &= 5.972 \times 10^{24}\,\text{kg} \quad \text{(mass of Earth)} \\
G &= 6.67408 \times 10^{-11}\,\text{Nm}^2/\text{kg}^2 \quad \text{(gravitational constant)}
\end{aligned}
$$

When making the orbital maneuvers, it is assumed that the satellites can accelerate in any direction when need to. The positions on the initial orbit are evenly distributed in the orbit. On the target orbit it is also evenly distributed and the amount of position on target are equal to the amount on initial as shown in figure 18. These 100 positions are distributed by giving creating a vector of evenly distributed $\theta_i = [0, 3.636, 7.273, 10.909, \ldots, 360]$ where $i = [1, 2, 3, ..., 100]$.

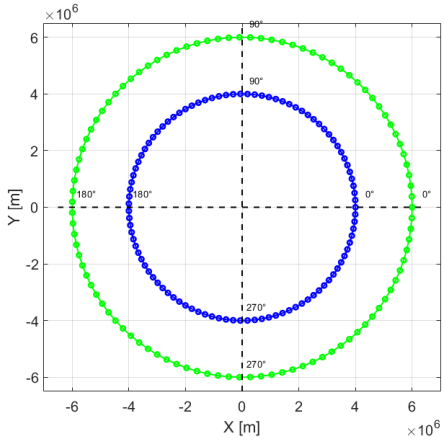

Figure 18: 100 evenly distribution of initial an target positions on the orbits. Blue inner figure shows the positions on the initial orbit. Green outer figure shows the positions on the target orbit.

After distribution of the position, every initial position different trajectories are calculated using Non-Hohmann transfer as described in 2.3.3 and the trajectory with minimum change in velocity is chosen as the true trajectory for that specific initial position. Equation 7 is the equation of motion and is

used to calculated the positions of the satellite in orbit. The initial positions are given by equation 47

$$\begin{aligned} x_{1i} &= r * cos(\theta_{1i}) \\ y_{1i} &= r * sin(\theta_{1i}) \\ z_{1i} &= 0 \end{aligned} \tag{47}$$

The velocity on positions on circular orbits is calculated using circular velocity shown in equation 48

$$v_{circular} = \sqrt{\frac{GM}{r}} \tag{48}$$

$$\begin{aligned} \ddot{x}_{1i} &= -\frac{GM}{r^3} x_{1i} \\ \ddot{y}_{1i} &= -\frac{GM}{r^3} y_{1i} \\ \ddot{z}_{1i} &= -\frac{GM}{r^3} z_{1i} \end{aligned} \tag{49}$$

Acceleration and velocity are integrated over orbital period to find new position at every instant. The eccentricity and angular momentum for transfer orbits are solved using equations 39 and 40 and becomes:

$$e = \frac{r_2 - r_1}{r_1 \cos(\theta_{1i}) - r_2 \cos(\theta_{target})}$$

$$h_t = \sqrt{G\ M\ r_1 r_2} \sqrt{\frac{cos\theta_{1i} - cos\theta_{target}}{r_1\ cos\theta_{1i} - r_2\ cos\theta_{target}}}$$

This is used to calculate Velocity Components both perpendicular and radial velocity components of the transfer orbit on initial position:

$$v_{A,\text{per}} = \frac{h}{r_1}, \quad v_{A,\text{rad}} = \frac{\mu \cdot e \cdot \sin(\theta_{1i})}{h}$$

The velocity on initial position depended on transfer orbit becomes:

$$v_{At} = \sqrt{v_{A,\text{per}}^2 + v_{A,\text{rad}}^2}$$

The velocity components at the target orbit are similarly calculated:

$$v_{B,\text{per}} = \frac{h}{r_2}, \quad v_{B,\text{rad}} = \frac{\mu \cdot e \cdot \sin(\theta_{target})}{h}$$

The velocity on initial position depended on transfer orbit becomes:

$$v_{Bt} = \sqrt{v_{B,\text{per}}^2 + v_{B,\text{rad}}^2}$$

where θ_{1i} is the angle on initial orbit and θ_{target} is the angle on target orbit. Velocity on is calculated using equation Δvs, which represents the required velocity change, is calculated for both burns on position A and position B:

$$\Delta v_A = \sqrt{v_A^2 + v_1^2 - 2v_1 v_A \cos(\gamma_1)}$$
$$\Delta v_B = \sqrt{v_2^2 + v_B^2 - 2v_B v_2 \cos(\gamma_2))}$$

where γ_1 is the different of flight angle between transfer position and initial position and γ_2 is the different of flight angle between transfer position and target position shown as in figure 10. These values represent the velocity adjustments needed to initiate and complete the transfer. The total delta-V(Δv_{tot}) is the sum of these two values. The velocity on initial orbit is found to be 6199 m/s and on target orbit, velocity is 5676 m/s. And the acceleration of the satellites is calculated using equation 49.

4.1.2 Planning for different initial position

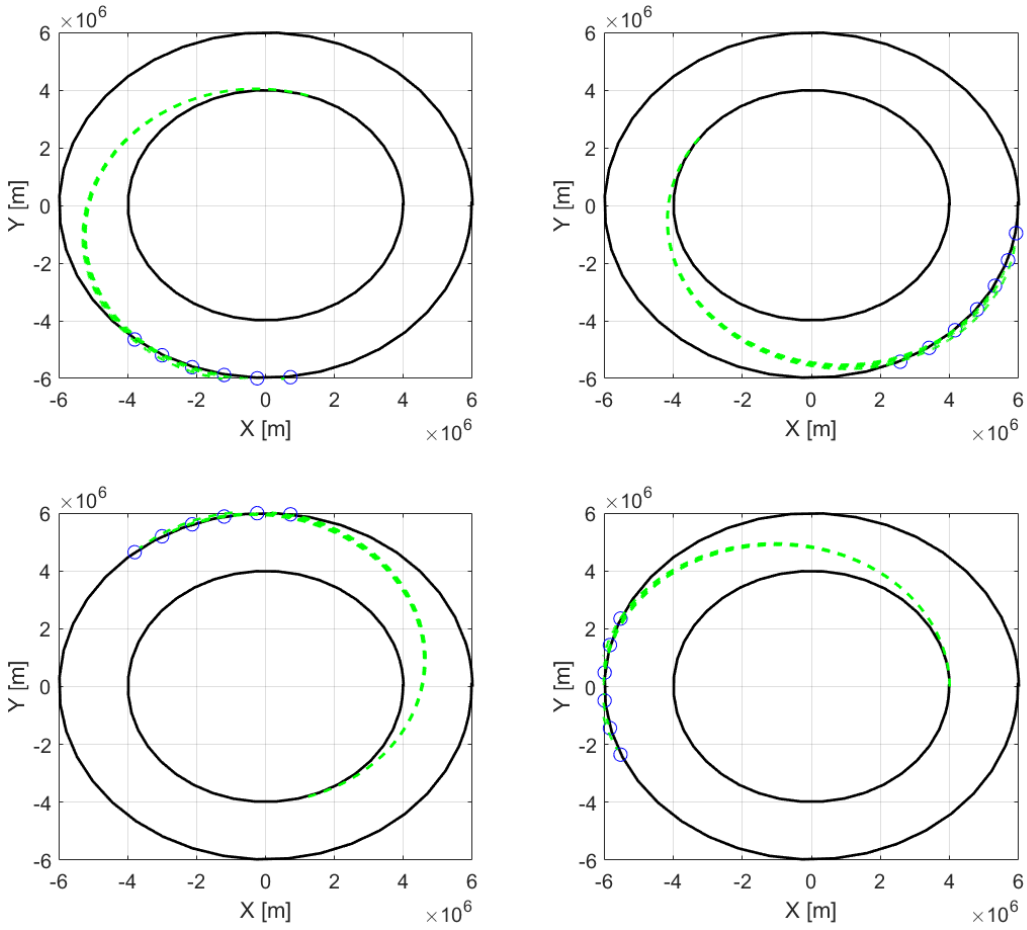

Figure 19: adding different Δv_1 on initial positions, every initial position has multiple target position

Figure 19 shows Non-Hohmann maneuver between two circular orbits, where the initial position is on lower orbit than the target orbit. There is multiple initial position and every initial position has multiple target positions.

4.2 Result from supervised learning model

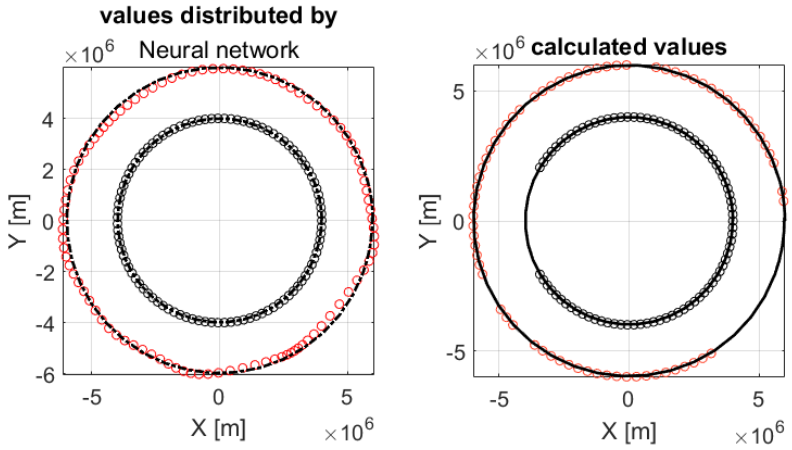

Figure 20: Result showing target positions given 100 initial orbit calculated before supervised learning model and after supervised learning model

For the training a supervised learning model, a significant portion of data is used to obtain a good model to use. To train the model 700 hundred different position is used for the initial and target orbit, each position on target orbit being result to a specific position for initial orbit.

The model has been trained multiple times before it was able to give values as indicated in figure 17, figure 21 one of the models where training data was chosen to be 85%, validation data was 10% and test data was 5%.

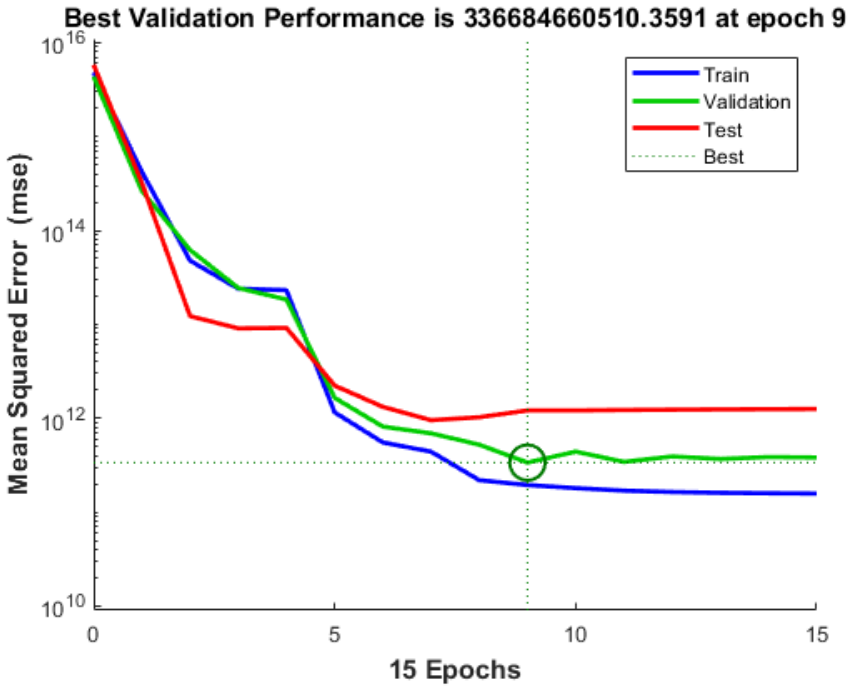

Figure 21: Performance of the supervised model under training, training data was chosen to be 85%, validation data was 10% and test data was 5%

When the model was trained, a new set of 100 initial positions was sent in the model, which it has not seen before, and the model gave good result as shown in figure 20. Since training supervised learning model the data used needs to be as accurate as possible, result calculated were chosen carefully. The model was trained using different distribution in dataset, figure 22 shows when training data was chosen to be 70%, validation data was 20% and test data was 10%.

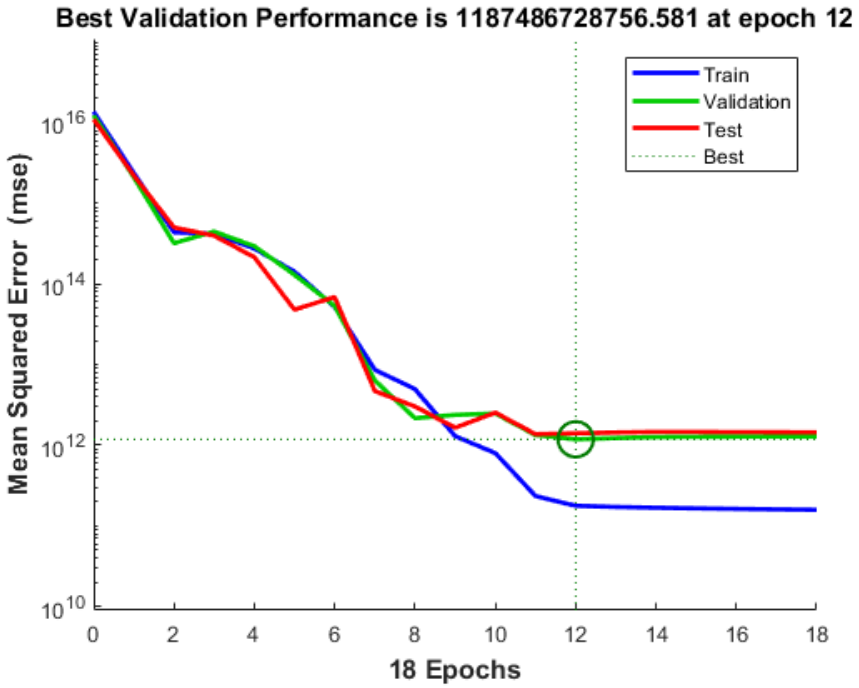

Figure 22: Performance of the supervised model under training, training data was chosen to be 70%, validation data was 20% and test data was 10%

Matlab's neural net fitting app was used and the result was compared to the calculated result shown table 2, from initial orbit, ten positions were taken and the results from supervised learning model and the calculated were compared. This comparison includes the total change in velocity(Δv_{total}) and their position on target orbit. The model gave target positions that are reasonable and lay on target orbit, Δv_{total} was calculated for the ten responses as shown table 2, the Δv_{total} from the positions distributed by supervised model were higher than those calculated before but not unreasonable. With farther observation it can be noticed that the target positions from the supervised model are closer from initial position than the results given before, which is why the Δv_{total} from supervised model are much higher than those calculated before.

Table 2: Comparing results from supervised model with results from simulations in Matlab

	Matlab simulations		**supervised learning**	
initial angle(°)	final angle(°)	$\Delta v_{total}(m/s)$	final angle(°)	$\Delta v_{total}(m/s)$
0	202.5	179.2	150.9	720.6
21.25	225	178.9	171.7	724.7
42.5	247.5	178.6	192.9	725.3
67.5	270	178.1	213.3	732.8
85	292.5	177.7	234.3	735.5
106.25	315	180.5	255.8	732.7
127.5	45	180.7	280.4	701.9
148.5	67.5	180	300.1	715.5
170	90	179.3	3.009	730.9
191.25	112.5	179.6	23.54	731.5

5 Discussion

This research addresses the complex task of satellite constellation initialization, particularly focusing on efficient orbit insertion using minimal-energy trajectories. The combination of classical orbital mechanics with supervised learning in maneuver planning provides a promising framework to enhance the efficiency and resilience of satellite constellation management. Traditional methods, such as Hohmann and non-Hohmann transfers, were foundational for this work, with Hohmann transfers offering fuel-efficient solutions for simpler orbital adjustments. However, the study found that non-Hohmann transfers with a common apse line provide greater flexibility, accommodating more varied target positions and reducing fuel expenditure in specific configurations. This flexibility is crucial for managing large constellations, where fixed maneuver types may lack adaptability for dynamic deployment needs.

One of the most significant contributions of this research is the application of supervised learning to predict optimal maneuver trajectories. The neural network model trained on data generated through MATLAB simulations demonstrated the potential to generalize across multiple initial and target orbits, effectively selecting minimum-fuel paths. This approach leverages

machine learning to overcome limitations associated with manual planning, including the time and computational resources required for large-scale simulations. The results from the supervised learning model were compared with MATLAB simulations, showing reasonable accuracy in predicting trajectories with slightly higher delta-v values. This slight discrepancy is likely due to the limitations inherent in approximations made by neural networks. However, the results indicate that supervised learning models can reliably approximate optimal trajectories, reducing the operational burden of constellation deployment while maintaining efficiency.

The integration of learning-based approaches with established principles of orbital mechanics has broader implications for satellite constellation management. By predicting delta-v values and suitable maneuver strategies based on input-output pairs of orbital parameters, the framework supports real-time decision-making, which is essential for autonomous constellation operations. As constellations expand to hundreds or thousands of satellites, this capability will be crucial for maintaining precise orbital configurations while minimizing operational costs and extending satellite lifespans. Additionally, this approach promotes resilience in satellite networks by allowing adaptive maneuver planning that can respond dynamically to changes in environmental conditions, such as unexpected orbital debris or anomalies in individual satellite performance.

The research also emphasizes the importance of accurate and diverse training data in developing robust neural network models for satellite applications. The results suggest that data from various orbital scenarios enhance the model's predictive accuracy, supporting the model's adaptability to a wide range of deployment configurations. However, one area for further exploration is the model's scalability to more complex constellations and diverse orbital geometries, such as elliptical and inclined orbits. Future work could focus on training models with broader datasets to enhance their generalization, making them suitable for multi-orbital constellations that combine low, medium, and high Earth orbits.

While this study achieved positive results, some limitations should be noted. The training data relied on MATLAB simulations, which, although accurate, may not fully capture the operational complexities encountered in real-space environments. Additionally, the research primarily examined circular or-

bits, limiting the applicability of the framework to more diverse orbit types. The adaptation of the supervised learning model to handle varied orbital parameters, including inclined orbits, could further broaden its application. Integrating reinforcement learning methods could also improve the mode, l's ability to optimize trajectories in real-time, enhancing its utility for increasingly complex and autonomous satellite constellations.

In conclusion, this research provides an innovative approach to satellite constellation initialization, blending traditional orbital mechanics with machine learning techniques to achieve efficient and resilient orbit insertion. By utilizing both Hohmann and non-Hohmann maneuvers, this study provides foundational insights into the advantages and limitations of different transfer types, demonstrating how non-Hohmann transfers offer enhanced flexibility for larger constellations. The integration of supervised learning enables the autonomous prediction of minimum-fuel maneuvers, allowing for real-time, adaptive constellation management that aligns with the increasing scale and complexity of modern satellite networks.

The framework presented in this study has the potential to impact various applications, from global communications to Earth observation, by providing an effective method for initializing and managing satellite constellations in a fuel-efficient manner. The results indicate that learning-based models can approximate optimal maneuvers with reasonable accuracy, highlighting the feasibility of machine learning in reducing the computational demands and operational costs associated with constellation management. Furthermore, the adaptability of supervised learning models to new data suggests their utility in handling the dynamic nature of space environments, ultimately promoting resilience and autonomy in satellite operations.

References

[1] Feifei Li et al. "Analysis of Initial Deployment Strategy for Low-Orbit Large-Scale Constellations". In: *International Journal of Aerospace Engineering* 2022 (2022).

[2] Shaik Arif and Sri Velagapudi Veda. "Leveraging the Potential: Artificial Intelligence Applications in Space Exploration". In: *International Journal of Innovative Research in Technology* 10.5 (2023). Accessed: 10.06.2024. ISSN: 2349-6002. URL: https://ceur-ws.org/Vol-3105/paper21.pdf.

[3] D Athauda et al. "Intelligent motion planning for collision free autonomous docking of satellite emulation platform using reinforcement learning". In: *IFAC-PapersOnLine* 56.2 (2023), pp. 3354–3359.

[4] Avijit Banerjee et al. "On the design, modeling and experimental verification of a floating satellite platform". In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 1364–1371.

[5] Avijit Banerjee et al. "Resiliency in space autonomy: a review". In: *Current Robotics Reports* 4.1 (2023), pp. 1–12.

[6] Daniel Brown and Jennifer Miller. "Satellite Constellation Deployment Strategies and Optimization". In: *Journal of Space Systems* 15.3 (2020), pp. 199–213. DOI: 10.2514/1.JS34567.

[7] Bruce R. Elbert. *introduction to satellite communication*. third edition. ARTECH HOUSE, INC, 2008.

[8] Bryan Weber. *Classical Orbital Elements: Perifocal Frame*. Accessed: 06.10.2024. 2024. URL: https://orbital-mechanics.space/classical-orbital-elements/perifocal-frame.html.

[9] NASA Glenn Research Center. *Newton's Laws of Motion*. URL: https://www1.grc.nasa.gov/beginners-guide-to-aeronautics/newtons-laws-of-motion/. (accessed: 22.08.2024).

[10] Elton Chang. *Space Segment in Satellite Networks*. URL: https://telecomworld101.com/space-segment-satellite-networks/. (accessed: 06.11.2024).

[11] Y. et al. Chen. "Deployment Strategies for Mega-Constellations". In: *Space Engineering Review* (2020).

[12] Nathaniel Choo. "Design and Analysis of Asymmetric "String-of-Pearls" Common Repeating-Ground-Track Satellite Constellations for Missions Requiring Regional Coverage". In: (2023).

[13] Andrea D'Agostino. *Introduction to neural networks — weights, biases and activation*. URL: https://medium.com/@theDrewDag/introduction-to-neural-networks-weights-biases-and-activation-270ebf2545aa. (accessed: 22.10.2024).

[14] Brandon Escamilla et al. "A Concurrent Methodology for Optimizing Constellation Deployment and Launcher Selection". In: *10th European Conference for Aerospace Sciences (EUCASS)*. July 2023. DOI: 10.13009/EUCASS2023-994.

[15] NASA Solar System Exploration. *Orbits and Kepler's Laws*. URL: https://science.nasa.gov/resource/orbits-and-keplers-laws/. (accessed: 22.03.2024).

[16] Henri P. Gavin. "The Levenberg-Marquardt Algorithm for Nonlinear Least Squares Curve-Fitting Problems". In: *Department of Civil and Environmental Engineering, Duke University* (2024).

[17] GeeksforGeeks. *Linear Regression in Machine learning*. Accessed: 23.11.2014. 2024. URL: https://www.geeksforgeeks.org/ml-linear-regression/.

[18] GeeksforGeeks. *Supervised Machine Learning*. Accessed: 23.11.2014. 2024. URL: https://www.geeksforgeeks.org/supervised-machine-learning/.

[19] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[20] Howard D. Curtis. *Orbital Mechanics for Engineering Students*. First edition. Elsevier Butterworth-Heinemann, 2005.

[21] James R. Wertz , Wiley J. Larson. *Space Mission Analysis and Design*. third edition. Microcosm, Inc., 1999.

[22] Jeffery John Puschell James Richard Wertz David F. Everett. *Space Mission Engineering: The New SMAD*. Microcosm Press, 2011.

[23] Alexander Kyuroson et al. "Towards fully autonomous orbit management for low-earth orbit satellites based on neuro-evolutionary algorithms and deep reinforcement learning". In: *European Journal of Control* 80 (2024), p. 101052.

[24] W. J. Larson and J. R. Wertz. *Space Mission Analysis and Design*. Microcosm Press, 2000.

[25] Lumen Learning. *Newton's Universal Law of Gravitation*. Accessed: 10.05.2024. URL: https://courses.lumenlearning.com/suny-physics/chapter/6-5-newtons-universal-law-of-gravitation/.

[26] Hang Woon Lee et al. "Optimization of satellite constellation deployment strategy considering uncertain areas of interest". In: *Acta Astronautica* 153 (Apr. 2018). DOI: 10.1016/j.actaastro.2018.03.054.

[27] Qizhang Luo et al. "Orbital Maneuver Optimization of Earth Observation Satellites Using an Adaptive Differential Evolution Algorithm". In: *Remote Sensing* 14.9 (2022). ISSN: 2072-4292. URL: https://www.mdpi.com/2072-4292/14/9/1966.

[28] Mark Wade. *Hohmann Transfer*. Accessed: 06.10.2024. 2024. URL: http://www.astronautix.com/h/hohmann.html.

[29] MathWorks. *Neural Network Fitting Tool (NFTool) Documentation*. Accessed: 16.04.2024. URL: https://se.mathworks.com/help/deeplearning/ref/nftool.html.

[30] Giovanni Minelli. "Resource-Constrained Autonomous Operations of Satellite Constellations and Ground Station Networks". PhD thesis. Monterey, CA; Naval Postgraduate School, 2018.

[31] Moumita Mukherjee, Avijit Banerjee, and George Nikolakopoulos. "An auto-encoder enabled fault detection and isolation scheme for enabling a multi-sensorial distributed pose estimation". In: *2022 IEEE 31st International Symposium on Industrial Electronics (ISIE)*. IEEE. 2022, pp. 281–288.

[32] Martin Møller. "Moller, M.F.: A Scaled Conjugate Gradient Algorithm For Fast Supervised Learning. Neural Networks 6, 525-533". In: *Neural Networks* 6 (Dec. 1993), pp. 525–533. DOI: 10.1016/S0893-6080(05)80056-5.

[33] NASA Small Satellite Institute. *Integration, Launch, and Deployment*. Accessed: 16.11.2024. 12.02.2024. URL: https://www.nasa.gov/smallsat-institute/sst-soa/integration-launch-and-deployment/#10.4.2.

[34] Ogutu B Osoro and Edward J Oughton. "A techno-economic framework for satellite networks applied to low earth orbit constellations: Assessing Starlink, OneWeb and Kuiper". In: *IEEE Access* 9 (2021), pp. 141611–141625.

[35] Radhakant Padhi et al. "Computational guidance using model predictive static programming for challenging space missions: an introductory tutorial with example scenarios". In: *IEEE Control Systems Magazine* 44.2 (2024), pp. 55–80.

[36] Digital regulation platform. *Regulation of NGSO Satellite Constellations*. URL: https://digitalregulation.org/regulation-of-ngso-satellite-constellations/. (accessed: 22.10.2024).

[37] Freedom Preetham. *Bayesian Regularization for NeuralNetworks*. URL: https://medium.com/autonomous-agents/bayesian-regularization-for-neuralnetworks-2f2d34f03adc. (accessed: 06.11.2024).

[38] Lorraine E Prokop et al. "Case studies in verifying spacecraft autonomy". In: *2021 IEEE Aerospace Conference (50100)*. IEEE. 2021, pp. 1–18.

[39] J. E. Prussing and B. A. Conway. *Orbital Mechanics*. Oxford University Press, 2012.

[40] Jason Rainbow. *SIA reports more record growth for the global commercial satellite industry*. Accessed: 10.08.2024. 2024. URL: https://spacenews.com/sia-reports-more-record-growth-for-the-global-commercial-satellite-industry/.

[41] Roadpost. *Iridium Communications Network*. URL: https://www.roadpost.com/iridium-satellite-network. (accessed: 22.10.2024).

[42] Roger R. Bate, Donald D. Mueller, and Jerry E. White. *Fundamentals of Astrodynamics*. First edition. Dover Publications, 1971.

[43] I.H Sarker. *Machine Learning: Algorithms, Real-World Applications and Research Directions*. URL: https://link.springer.com/article/10.1007/s42979-021-00592-x. (accessed: 28.05.2024).

[44] Simplilearn. *What is Epoch in Machine Learning?* URL: https://www.simplilearn.com/tutorials/machine-learning-tutorial/what-is-epoch-in-machine-learning. (accessed: 06.11.2024).

[45] Inc. The MathWorks. *Divide Data for Optimal Neural Network Training*. URL: https://se.mathworks.com/help/deeplearning/ug/divide-data-for-optimal-neural-network-training.htm. (accessed: 16.09.2024).

[46] Inc. The MathWorks. *Fit Data with a Shallow Neural Network*. URL: https://se.mathworks.com/help/deeplearning/gs/fit-data-with-a-neural-network.html. (accessed: 16.09.2024).

[47] Inc. The MathWorks. *Supervised Learning Workflow and Algorithms*. URL: https://se.mathworks.com/help/stats/supervised-learning-machine-learning-workflow-and-algorithms.html#bswluh9. (accessed: 16.09.2024).

[48] The MathWorks, Inc. *Deep Learning Toolbox Documentation*. Accessed: 16.04.2024. URL: https://se.mathworks.com/help/deeplearning/index.html.

[49] The MathWorks, Inc. *Linear Regression*. Accessed: 26.11.2024. URL: https://se.mathworks.com/help/matlab/data_analysis/linear-regression.html.

[50] The MathWorks, Inc. *trainbr: Bayesian Regularization Backpropagation*. Accessed: 26.11.2024. URL: https://se.mathworks.com/help/deeplearning/ref/trainbr.html.

[51] The MathWorks, Inc. *trainlm: Levenberg-Marquardt Backpropagation*. Accessed: 26.11.2024. URL: https://se.mathworks.com/help/deeplearning/ref/trainlm.html.

[52] The MathWorks, Inc. *trainscg: Scaled Conjugate Gradient Backpropagation*. Accessed: 26.11.2024. URL: https://se.mathworks.com/help/deeplearning/ref/trainscg.html.

[53] W3Schools. *Artificial Intelligence Tutorial*. Accessed: 10.08.2024. 2024. URL: https://www.w3schools.com/ai/default.asp.

[54] Yasi Wang, Hongxun Yao, and Sicheng Zhao. "Auto-encoder based dimensionality reduction". In: *Neurocomputing* 184 (2016), pp. 232–242.

[55] Bryan Weber. *Hohmann Transfer*. Accessed: 10.05.2024. 2024. URL: https://orbital-mechanics.space/orbital-maneuvers/hohmann-transfer.html.

[56] Bryan Weber. *Non-Hohmann Transfers*. Accessed: 10.05.2024. 2024. URL: https://orbital-mechanics.space/orbital-maneuvers/non-hohmann-transfers.html.

[57] Bryan Weber. *Orbital Plane and Kepler's Second Law*. Accessed: 10.05.2024. 2024. URL: https://orbital-mechanics.space/constants-of-orbital-motion/orbital-plane-and-keplers-second-law.html.

[58] Computer Vision Wiki. *Dataset Split in Machine Learning*. Accessed: 10.05.2024. URL: https://wiki.cloudfactory.com/docs/mp-wiki/splits/data-splitting-in-machine-learning.