MPC-Based Autonomous Assignment Algorithm for a Swarm of Satellites

by

Rupak Dhankhar

to obtain the degree of Master of Science (Vehicle Engineering),

Faculty of Mechanical, Maritime and Materials Engineering (3ME),

Delft University of Technology.

to be defended on Thursday July 8, 2021 at 10:00 AM.

Supervisor: prof. dr. ir. J. Hellendoorn
Co-supervisor: Dr. J. Guo

The thesis committee consisted of:

Prof. dr. ir. J. Hellendoorn,	3ME-Cognitive Robotics, TU Delft
Dr. J. Guo,	Aerospace Engineering, Space Systems Engineering,TU Delft
Dr. J. Alonso Mora,	3ME-Cognitive Robotics, TU Delft
Dr. S.H. Mok,	Defense and Space Center, Agency for Defense Development, South Korea

PREFACE

This thesis was aimed for the Master of Science degree in Mechanical Engineering at The Delft University of Technology. The goal of this research was to formulate an assignment algorithm for autonomous guidance, control, and navigation system for spacecraft formation flying using a Model Predictive Control approach for estimation of fuel expenditure.

This report is split into three parts. First, the earth's gravitation field and its variation are discussed. The relative motion in the orbit around the earth is briefly explained. The derivation of linearized relative dynamics by Clohessy-Wiltshire is presented. In the second part, the assignment problem is explained. The majority of the literature has been using the distance between the deployed state to the target state. A new approach is discussed using fuel as the assignment parameter including the framework for the assignment parameters is established.

For the third part, fuel expenditure estimation using two types of control schemes is explained. The Artificial Potential Function approach is concisely explained. The choice of Model Predictive Control over Artificial Potential Function is discussed. The aim for the third part was the formulation of the control problem using the Model Predictive Control with a linearization dynamics model of the satellites. The system model used in the MPC formulation was constructed based on the Clohessy–Wiltshire relative equations of motion. The demonstrations were established using randomized positions.

The simulations were performed and estimation of the fuel expenditure was explained based on the simulation results. The proposed assignment algorithm was tested using the fuel cost estimations from the MPC-based simulations. The results showed that the algorithm performed as designed. Lastly, the possible future research options are listed out.

Rupak Dhankhar
Delft, July 2021

ACKNOWLEDGEMENT

As a science student, I have always been interested in all things space. I could not believe that I got an opportunity to work on such a project as my Master's Thesis. It has been an insightful, educational and eventful though a bit longer than expected journey at TU Delft so far. I have met some of the most brilliant, supportive, and kind people in my time here. Here is a word of thanks to everyone who has supported and helped me on this road.

First and foremost, I want to thank my parents for always being there, supporting me through thick and thin, always believing in me. I wouldn't even be alive without both of you so, thank you very much.

I want to thank Prof. Hans Hellendoorn for giving me this opportunity to pursue this thesis project. I could not have wished for a better supervisor. He has been a constant bastion to me all this time. He has been patient, critical, and enthusiastic for this long thesis project He has taught me the approach to take for scientific research in academia. He has constantly pushed me to do better. This project would not be the same without him. Furthermore, I want to thank Dr. Jian Guo for assigning me this project, who believed in me when few people would and assigning me this project. This thesis project would not be possible without Dr. Jian. He also introduced me to Ir. Sung Hoon Mok. Sung Hoon has been an integral part of the thesis. The long discussions with him taught me immensely about the space domain. I want to thank him for his crucial direction whenever I was feeling lost during the work. He has been nothing but supportive all this time. He continued to help me even when he decided to go back to Korea. Thank you, Sung Hoon for constant encouragement from halfway around the world.

Next, I would like to thank my counselor Drs. Lourdes Gallastegui Pujana for helping me during the rough time and making me realize that I can pursue space-related projects as a mechanical engineer. I want to say thanks to Yash Raj Khusro and Nishant Chowdhri for helping me with MPC design at crucial moments. Lastly, I want to thank Rishab, Nandu, Abhimanyu, Vishrut, Prabhav, Abhairaj, Manasa, Chinmaya and Karan for supporting and motivating me in everything. I hope our friendship will last a lifetime.

The education in TU Delft has been enlightening, at the very least. I have become a better student, a better engineer in this journey. But most of all, this experience has taught me to be a better human.

Rupak Dhankhar
Delft, July 2021

CONTENTS

List of Figures

LIST OF TABLES

1

INTRODUCTION

1

1.1. INTRODUCTION

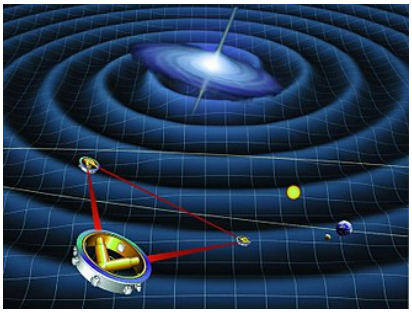
In the last couple of decades, small satellites' capability to carry out complex missions has improved exponentially with the advent of technology. With such recent improvements in small satellites, formation flying has become a lucrative area of research. Using multiple spacecraft provides more flexibility and robustness during a mission in comparison to a singular monolithic satellite. A minor malfunction in a large satellite can result in mission failure. But if such a case occurs in a formation mission, the mission control can replace the faulty satellite in the formation potentially saving large sums of investment. The production and deployment costs alone justify the influx of the research resources.

Recent formation flying missions such as GRACE by NASA [1] where two satellites orbit the earth in tandem to map its gravity field by measuring the mass of polar ice caps, ocean currents, sea-level changes, and so on. The enhanced geoid gravity knowledge will help with a better understanding of weather patterns. Magnetospheric Multiscale (MMS) by NASA [2] employs four spacecraft in tetrahedral formation in the area of interest. MMS was deployed to study the magnetosphere of the earth making crucial discoveries. PRISMA [3] by ESA consists of two spacecraft where one being stationary in the relative frame while the other spacecraft performed Guidance, navigation, and control demonstrations to show formation flying capabilities. TanDEM-X [4] by DLR involves two spacecraft with the primary goal to generate a Digital Elevation Model using synthetic aperture radar interferometry in a polar orbit Along-track formation described by Sabol et al. [5].

Such space missions are demonstrating a glimpse of the potential that formation flying missions hold. The satellite swarms can enable massive sensing capabilities with distributed antennas to cover a larger area than possible with a singular satellite or a scattered array aperture for a telescope to cover a larger field of view in space. There are some intriguing future formation flying missions planned. Recently, there was an exciting discovery in the field of science. For the first time in history, scientists detected gravitation waves using the Light Interferometry Gravitation-wave Observatory (LIGO) based in North America. However, the LIGO is a ground-based observatory limiting the length of the interferometer to 4 km. If the interferometer distance is longer, the observatory will be able to detect even smaller gravitational waves. Hence, NASA has planned a formation flying space mission called Laser Interferometry Space Antenna (LISA) 50 million kilometers behind the earth in its orbit. The interferometer has an arm length of 2.5 million kilometers resulting in more sensitivity to long-period gravitation waves.

Recently, the Event Horizon Telescope (EHT) [6] gathered data of light around the center M87 galaxy for an image of the supermassive black hole. The EHT is a large array of telescopes all around the earth shown in Figure 1.2a. After gathering data using all the telescopes in EHT at precise time instants using an inbuilt hydrogen maser clock, the data were combined and processed to produce the first image of a Blackhole two years ago in 2019. As of March 2021, a new image of the M87 Blackhole has been processed and shows the polarized light around the Blackhole in Figure 1.2b. The EHT is a ground-based array of telescopes. The next step is to set up a space observatory to provide much

(a) LIGO at Livingston



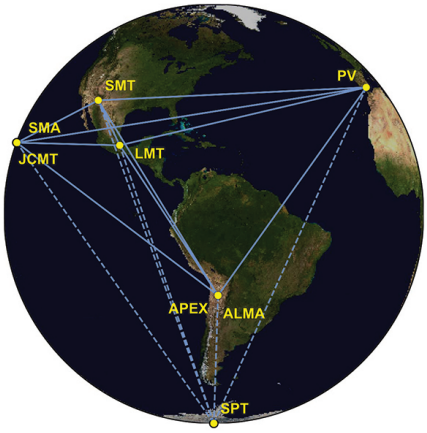
(b) the Laser Interferometry Space Antenna

Figure 1.1: The present and the future of Gravitation wave detection observatories

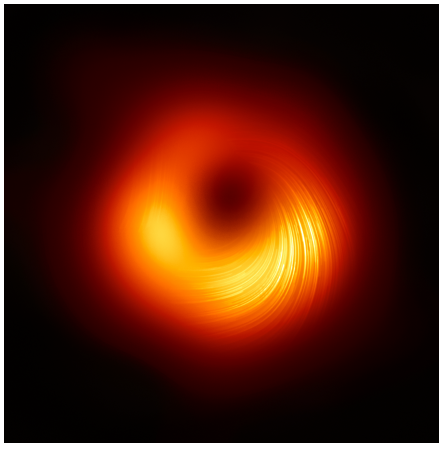
higher capabilities than the Hubble telescope. OLFAR [7] is one such technology under research in the Netherlands. OLFAR aims to deploy a distributed aperture synthesis telescope array using an autonomous system of cube sats to capture the radio frequencies below 50 MHz which is a significant limitation for earth-based observatories.

As the spacecraft increase in number, the satellite swarm becomes extremely complex and can take some spectacular formations. Just consider the structure of solids around us. A few examples are shown in Figure(1.3). These Bravais Lattices are just a few of many possible based on the arrangement of atoms in the matter around us. SpaceX Starlink is a plan where the company plans to blanket the earth with satellites much like the Buckminsterfullerene example.

Izzo and Pettazzi [8] simulate an even more complex procedure, visualized in Figure 1.4.



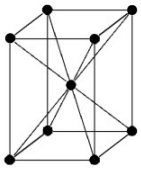
(a) EHT configuration



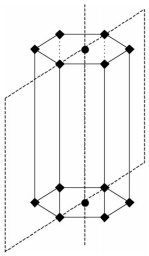
(b) M87 Blackhole in polarized light

Figure 1.2: EHT is an array of telescopes used to capture the first image of a Blackhole at the enter of Galaxy Messier 87

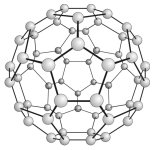
1



(a) Tetragonal-I



(b) Hexagonal-P



(c) Buckminsterfullerene

Figure 1.3: Advanced Lattice structures

The sequential phases show an assembly of a flat surface in the space where 49 aircraft assume the required formation and perform the docking maneuver to achieve the target lattice structure.

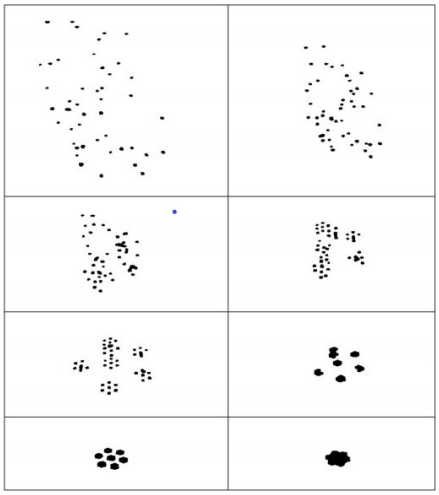


Figure 1.4: Assembly Process of 49 Spacecrafts

However, formation flying comes with its own set of complexities. The swarm of satellites has to operate in cluttered space in close vicinity to carry out the formation maneuvers. Keeping track of multiple satellites and providing assistance from the ground station is much more challenging. Koenig and D'Amico [9] outline some more design constraints such as the small, low-cost satellites incorporated in the formation might be equipped with limited capability hardware. These lower-grade sensors and communication equipment adding that complexity. An autonomous decentralized guidance and control system seems like an appropriate solution but does not solve all the problems. In this work, the autonomous assignment of satellites using fuel as the parameter is explored.

The problem is defined as an Assignment Problem which means assigning spacecraft involved in the formation flying new positions as per the mission requirement. A valuable solution to the problem satisfies the constraints of fuel or time of flight or a combination of both while successfully occupying all the target destinations. In this thesis, the assignment of the destinations is decided based on the fuel parameter such as current fuel level or expected fuel consumption or fuel capacity of the swarm as a whole. The second problem of calculation of expected fuel consumption is explored using preliminary navigation simulations as means for estimating the fuel consumption in the desired maneuver to occupy the target destinations. Two strategies are explored for preliminary navigation simulations. Artificial Potential Function (APF) as a traditional control and guidance approach is discussed. A model-based approach with the Model Predictive Control is explored in-depth with test simulations.

1.2. RESEARCH OBJECTIVES

The goal of this thesis is to formulate an assignment strategy for autonomous reconfiguration maneuvers for space formation flying missions. This thesis aims to tackle the goal through three main objectives:

1. Formulating an algorithm to autonomously assigning satellites a destination from a set of available targets.

2. Formulating a process to estimate fuel consumption using Model Predictive Control.

3. Testing the said autonomous assignment algorithm with a randomized set of initial positions and destinations.

In this thesis report, as discussed earlier, the assignment problem is developed by using fuel expenditure as the assignment parameter with an MPC as the estimation tool. The MPC problem is set up based on a classical two-body model with satellites and the Earth are represented as point masses. The only force experienced by the bodies is the Gravitational force due to earth. The earth is considered symmetrical and all perturbations are neglected in this work. The model used for the MPC is operational in Hill's frame of reference i.e. a rotating frame of reference anchored to the chief orbiting satellite obit. The relative motion of the satellites will be guided by Clohessy-Wiltshire equations [10] in Hill's frame of reference. The model predictive control is set up using the ACADO toolkit for the simulation in the MATLAB SIMULINK environment.

1.3. SCOPE OF THE THESIS

An Autonomous assignment and guidance consist of systems such as separation control, trajectory planning, formation position assignment, setting up a guidance strategy, and finally, obstacle and collision control while in the navigation phase. This thesis report is focused on formulating an assignment algorithm using an MPC as an estimating tool for fuel expenditure. The problems of trajectory planning and navigation are quite broad in themselves, so they will not be covered in this thesis. In terms of MPC formulation, collision avoidance is not considered in the preliminary simulations as the simulations are

targeted towards the estimation of fuel expenditure. The reason for that is the algorithm is approached with a decentralized system in mind. Hence, any cooperating satellite system knowledge is not considered in simulations ran for estimating fuel expenditure. The preliminary simulation demonstrations are carried out in circular orbits thus using the Clohessy-Wiltshire Approximations. The report is structured as follows:

- Chapter 2: The dynamics of the satellites around the Earth as explained in this chapter. The relative frame of reference is defined along with the derivation of relative equations of motion by Clohessy-Wiltshire.
- Chapter 3: This chapter covers the formulation of an assignment algorithm. The assignment problem is defined with illustrations. Some assignment parameters like initial position to destination distance or fuel parameters are discussed. A priority parameter matrix is defined that consists of all the applicable assignment parameters and their relevant weights. An assignment algorithm is formulated using the said parameter matrix.
- Chapter 4: This chapter summarizes the state-space model based on the Clohessy-Wiltshire equations of motions and the control strategies. Artificial Potential Function is briefly discussed. Linear MPC is defined with a brief explanation of the parameters involved in MPC formulation. An approach to calculate dynamic terminal weights using the Algebraic Riccati Equation is mentioned.
- Chapter 5: This chapter presents the simulations and results of this thesis work. The MPC simulations using randomized initial and final positions are demonstrated. The fuel expenditure is estimated based on the MPC simulations. These estimated fuel values are used to demonstrate the formulated assignment algorithm.

2

DYNAMICS AND MOTION

2.1. INTRODUCTION

In this chapter, a brief background to the Earth's gravity is discussed and how the perturbations affect the gravitational field. In section 2.2, the fundamental two-body dynamic equations are discussed and the expression for the force of gravitation is derived. In section 2.3, the Hills frame of reference is established. And in Hill's frame of reference, the Clohessy–Wiltshire equations of motion are derived in section 2.3.2.

The universe is filled with celestial bodies acting on each other by gravity, stars in a galaxy influenced by the black hole, the stars having a solar system around themselves and the planets on those solar systems have moons revolving around them. All the bodies influence each other in this universe, the star of a solar system influences the moon but the moon's motion is defined by the gravitation effect of the parent planet.

$$\mathbf{F_g} = -\frac{GMm}{r^2}\left(\frac{\mathbf{r}}{r}\right) \tag{2.1}$$

Newton coined a law, called *Universal Law of Gravitation*, which describes this phenomenon that every point mass attracts every other point mass by a force acting along the line intersecting the two points. The force, Gravity, is proportional to the product of the two masses, and inversely proportional to the square of the distance between them[11]. So, the equation tells us that the astronomical distances between the bodies are the key factor responsible for the reduced influence that we observe. Another important factor is the mass of the body itself. The heavier body is the stronger its gravitational influence becomes. Planets do no revolve around other planets but the star at the center because it is the heaviest body in a solar system. This phenomenon of attraction plays the most important role while deciphering and understanding the dynamics of the celestial bodies.

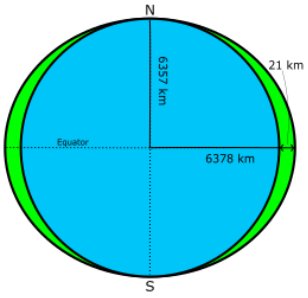


Figure 2.1: Oblate earth causing J2 perturbation

In this report, the focus is on a satellite revolving around the earth and tracking its movement to achieve the desired goal within the gravitational field of the earth which can be represented by the (2.1). However, the earth's gravity is not uniform as the equation suggests. There are a few factors that impact the earth's gravitation field. Earth is not completely a sphere but an oblate spheroid. Because of the rotation of the earth, centrifugal force bulges the equator hence, more of the earth's mass is distributed at the equator

than the poles. Therefore, equatorial orbits experience a larger force of gravity because of increased mass. This perturbation force around equator is called J2 perturbation[12] as in Figure 2.1. This J2 perturbation must be factored in the dynamic system to account for the variation in the gravitation field.

Another topological factor to consider here is the variation due to the Geoid. The Geoid refers to the shape that the ocean surface would take under the influence of gravity and rotation rejecting the effects of winds or tides. The Geoid describes the uneven distribution of mass on the surface of the earth causing variations in the field of its gravity. These gravitation anomalies have been mapped via satellites as recently as 2011 [13] as the Figure 2.2.

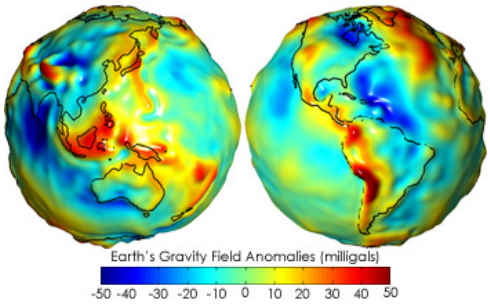


Figure 2.2: Satellite scan of gravity anomalies on Earth

Additionally, Solar winds[14] could also account for additional variation in the force exerted by the earth's gravity field. Solar winds are a continuous stream of charged subatomic particles. Hydrogen and helium are two major ingredients. These winds can travel up to 600-800 km/s and because it is a stream of particles. it exerts pressure on celestial bodies generating force which is considered a disturbance when calculating the dynamics of bodies in space. The force is small compared to gravity but can still be used as propulsion for space travel using solar sails much like ocean-faring ships. The pressure due to solar wind is almost negligible is generally not accounted for.

2.2. TWO BODY DYNAMICS

This project deals with the guidance of satellites around the earth so, the problem in essence is defined as a two-body system of the equation of motion. The Two-body equation is based on Newton's law of gravitation. We will focus on dynamics as a two body problem in our system following texts by Vallado and Curtis [15, 16]. A satellite motion around the earth and it's Inertial frame is described by the state vector $\mathbf{x} = \left[\mathbf{r}^T, \mathbf{v}^T\right]^T$, where $\mathbf{r} = \left[x, y, z\right]^T$ is the position vector and $\dot{\mathbf{r}} = \mathbf{v} = \left[\dot{x}, \dot{y}, \dot{z}\right]^T$ is the velocity vector as in Figure 2.3.

To find out the Two-body equation of motion, the system in Figure 2.3 is placed in a fixed inertial frame of reference shown in Figure 2.4 where m_1 represents the mass of the Earth and m_2 is the mass of the satellite.

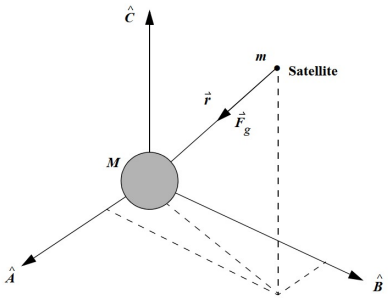


Figure 2.3: Two Body Geometry in Inertial(Earth) Reference Frame

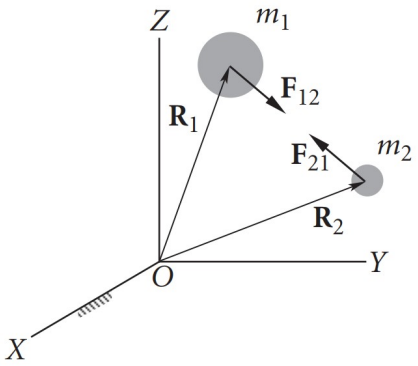


Figure 2.4: Free body diagram of two masses in an Inertial frame of reference

R_1 and R_2 are position vectors for both masses exerting F_{12} and F_{21} force on each other respectively. Let $\mathbf{r}$ be the position vector of m_2 relative to m_1, we get

$$\mathbf{r} = \mathbf{R_2} - \mathbf{R_1} \tag{2.2}$$

and lets also assume $\hat{\mathbf{u}}_\mathbf{r}$ be the unit vector to the position vector r,

$$\hat{\mathbf{u}}_\mathbf{r} = \frac{\mathbf{r}}{|r|} \tag{2.3}$$

Furthermore, the force of gravitation is defined by (2.1). Then, the gravitation force exerted by m_2 in m_1 can be written as

$$\mathbf{F_{21}} = \frac{Gm_1m_2}{r^2}(-\hat{\mathbf{u}}_\mathbf{r}) \tag{2.4}$$

$$m_2\ddot{\mathbf{R}}_2 = \frac{Gm_1m_2}{r^2}(-\hat{\mathbf{u}}_\mathbf{r}) \tag{2.5}$$

$\mathbf{F_{21}}$ suggests that the force is directed towards m_1 hence the $-\hat{\mathbf{u}}_\mathbf{r}$. Similarly, $\mathbf{F_{12}}$ is the force directed towards m_2 which is opposite to $\mathbf{F_{21}}$ from Newton's second law of motion, every force has an equal and opposite reaction that written as

$$\mathbf{F_{12}} = \frac{Gm_1m_2}{r^2}(\hat{\mathbf{u}}_\mathbf{r}) \tag{2.6}$$

$$m_1\ddot{\mathbf{R}}_1 = \frac{Gm_1m_2}{r^2}(\hat{\mathbf{u}}_\mathbf{r}) \tag{2.7}$$

Now, to find the resultant acceleration

$$\ddot{\mathbf{R}}_\mathbf{2} - \ddot{\mathbf{R}}_\mathbf{1} = \frac{Gm_1}{r^2}(-\hat{\mathbf{u}}_\mathbf{r}) - \frac{Gm_2}{r^2}(\hat{\mathbf{u}}_\mathbf{r}) \tag{2.8}$$

$$\ddot{\mathbf{R}}_\mathbf{2} - \ddot{\mathbf{R}}_\mathbf{1} = \frac{Gm_1 + m_2}{r^2}\hat{\mathbf{u}}_\mathbf{r} \tag{2.9}$$

The gravitational parameter μ is defined as

$$\mu = G(m_1 + m_2) \tag{2.10}$$

The mass of the satellite m_2 can be considered negligible for this project as these are man-made satellites which result in

$$\mu = Gm_1 \tag{2.11}$$

Hence, the resultant acceleration $\ddot{r}$ can be defined as

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^2}\frac{\mathbf{r}}{|r|} \tag{2.12}$$

The (2.12) represents the 2nd order non-linear differential equation that governs the relative motion of satellite m_2 with respect to the planet m_1. Few assumptions are made here to get the equations:

1. The bodies are symmetrical so they can be safely assumed as point masses.
2. The mass of the satellite is negligible compared to the parent body.
3. All the perturbation forces like J2, Geoid, etc are neglected. Only the mutual gravitation force is considered.

2.3. RELATIVE MOTION

Since we are looking into a cluster of spacecraft, we are more interested in the dynamics between multiple crafts to each other. The relative motions can be easily imagined by considering real-life examples like rendezvous missions to the International Space Station (ISS) sending supplies or equipment into space. In this project, the guidance of the individual satellites is very similar to the rendezvous operation itself. To find out the relative dynamics equation we shall start with rendezvous operation following Curtis[16].

2.3.1. RELATIVE MOTION IN ORBIT

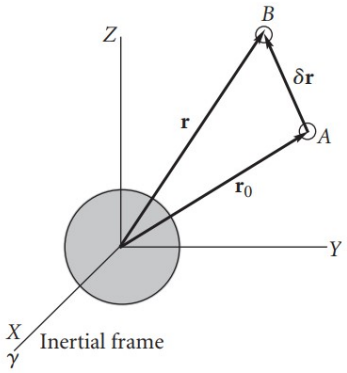


Figure 2.5: Location of satellite B near destination A

Following the Figure 2.5, **A** and **B** are two spacecrafts with position vectors $\mathbf{r_0}$ and $\mathbf{r}$ respectively with $\delta\mathbf{r}$ the small distance between the two spacecrafts. **A** represents the target destination and **B** represents the satellite. So, the position vector $\mathbf{r}$ can be

$$\mathbf{r} = \mathbf{r_0} + \delta\mathbf{r} \tag{2.13}$$

Using the equation above, we can work towards building the dynamics equation for the satellite **B**. We have the two body equation of motion (2.12) for the satellite **B**. Substituting (2.13) in (2.12) we get,

$$\delta\ddot{\mathbf{r}} = -\ddot{\mathbf{r_0}} - \mu\frac{\mathbf{r_o} + \delta\mathbf{r}}{r^3} \tag{2.14}$$

Lets consider $(r \cdot r)$

$$r^2 = \mathbf{r}.\mathbf{r} = (\mathbf{r_0} + \delta\mathbf{r}).(\mathbf{r_0} + \delta\mathbf{r}) = \mathbf{r_0}.\mathbf{r_0} + 2\mathbf{r_0}\delta\mathbf{r} + \delta\mathbf{r}.\delta\mathbf{r}$$

Also, $\delta\mathbf{r}.\delta\mathbf{r} = \delta r^2$ and $\mathbf{r_0}.\mathbf{r_0} = r_0^2$. So

$$r^2 = r_0^2\left[1 + \frac{2\mathbf{r_0}.\delta\mathbf{r}}{r_0^2} + \left(\frac{\delta\mathbf{r}}{\mathbf{r_0}}\right)^2\right]$$

again, $\delta\mathbf{r} << \mathbf{r_0}$ so the last term can be neglected

$$r^2 = r_0^2\left[1 + \frac{2\mathbf{r_0}.\delta\mathbf{r}}{r_0^2}\right] \tag{2.15}$$

Now, $r^{-3} = (r^2)^{-\frac{3}{2}}$, we can substitute (2.15) and get

$$r^{-3} = r_0^{-3}\left[1 + \frac{2\mathbf{r_0}\cdot\delta\mathbf{r}}{r_0^2}\right]^{-\frac{3}{2}} \tag{2.16}$$

using binomial expansion on (2.16) and neglecting terms of higher order, we get

$$r^{-3} = r_0^{-3}\left[1 - \left(\frac{-3}{2}\right).\left(\frac{2\mathbf{r_0}.\delta\mathbf{r}}{r_0^2}\right)\right]$$

which can be written as,

$$\frac{1}{r^3} = \frac{1}{r_0^3} - \frac{3}{r_0^5}\mathbf{r_0}\cdot\delta\mathbf{r} \tag{2.17}$$

Now, we substitute (2.17) in (2.14), we get

$$\delta\ddot{\mathbf{r}} = -\ddot{\mathbf{r_0}} - \mu\left(\frac{1}{r_0^3} - \frac{3}{r_0^5}(\mathbf{r_0}.\delta\mathbf{r})\right)(\mathbf{r_0} + \delta\mathbf{r})$$

Resolving the equation above and neglecting higher order terms,

$$\delta\ddot{\mathbf{r}} = -\ddot{\mathbf{r_0}} - \mu\frac{\mathbf{r_0}}{r_0^3} - \mu\left[\frac{\delta\mathbf{r}}{r_0^3} - \frac{3}{r_0^5}(\mathbf{r_0}\cdot\delta\mathbf{r})\mathbf{r_0}\right] \tag{2.18}$$

But we also know the two- body equation of a satellite (2.12)

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^2}\frac{\mathbf{r}}{|r|}$$

so, we substitute this into (2.18) and we get

$$\delta\ddot{\mathbf{r}} = \frac{\mu}{r_0^5}\left[r_0^2\delta\mathbf{r} - 3(\mathbf{r_0}.\delta\mathbf{r})\mathbf{r_0}\right] \tag{2.19}$$

(2.19) represents the Linearized motion of the satellite for the target in the earth-centric frame when the distance between the target $\delta\mathbf{r}$ is much smaller than the distance from the center of the earth.

2

2.3.2. CLOHESSY–WILTSHIRE RELATIVE MOTION EQUATIONS

We discussed the similarity between the rendezvous maneuver and the guidance of a satellite to its destination to be of a similar approach and we found the equation of motion for spacecraft to its target position in the earth-centric frame. Now, we will explore another approach towards relative motion in the target centric local frame of reference known as Local Vertical Local Horizontal (LVLH) frame of reference also known as Hills frame of reference which is represented by Figure 2.6

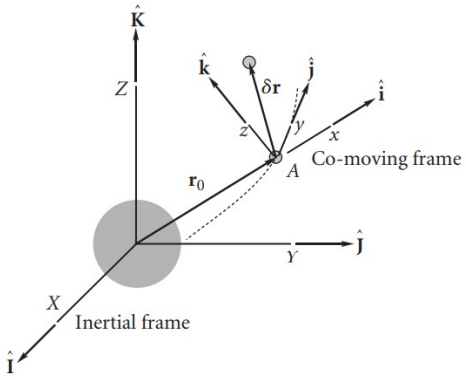


Figure 2.6: Local Vertical Local Horizontal co-moving frame for Relative motion

The same restriction applies as previously with $\delta r << r_0$, The origin for the moving system is at **A**. The x-,y-,z-axis represent the **radial**, **along-track**, **cross-track** directions of motion of deputy spacecraft relative to the chief spacecraft with $(\hat{i}, \hat{j}, \hat{k})$ as the unit vectors in the LVLH frame.

Now, we take the acceleration of a particle in a moving reference also known as five-term acceleration formula [17]

$$\ddot{\mathbf{r}} = \ddot{\mathbf{r_0}} + \dot{\omega} \times \delta\mathbf{r} + \Omega \times (\Omega \times \delta r) + 2\Omega \times \delta\mathbf{v}_{\text{rel}} + \delta\mathbf{a}_{\text{rel}} \tag{2.20}$$

where Ω is the angular velocity of the LVLH frame or the chief spacecraft and $\dot{\Omega}$ is the angular acceleration. The relative position, velocity, and acceleration are defined as follows

$$\delta\mathbf{r} = \delta x\hat{\mathbf{i}} + \delta y\hat{\mathbf{j}} + \delta z\hat{\mathbf{k}} \tag{2.21}$$

$$\delta\mathbf{v}_{\text{rev}} = \delta\dot{x}\hat{\mathbf{i}} + \delta\dot{y}\hat{\mathbf{j}} + \delta\dot{z}\hat{\mathbf{k}} \tag{2.22}$$

$$\delta\mathbf{a}_{\text{rel}} = \delta\ddot{x}\hat{\mathbf{i}} + \delta\ddot{y}\hat{\mathbf{j}} + \delta\ddot{z}\hat{\mathbf{k}} \tag{2.23}$$

Going back to Equation (2.20),

$$\ddot{\mathbf{r}} - \ddot{\mathbf{r}_0} = \dot{\omega} \times \delta\mathbf{r} + \Omega \times (\Omega \times \delta r) + 2\Omega \times \delta\mathbf{v}_{\text{rel}} + \delta\mathbf{a}_{\text{rel}}$$

we consider the chief orbit to be circular in this project so, $\dot{\omega}$ will be zero. (2.20) becomes

$$\delta\ddot{\mathbf{r}} = \Omega \times (\Omega \times \delta\mathbf{r}) + 2\Omega \times \delta\mathbf{v}_{\text{rel}} + \delta\mathbf{a}_{\text{rel}} \tag{2.24}$$

Now, we have assumed the orbit to be circular that means the angular velocity becomes

$$\Omega = n\hat{\mathbf{k}}$$

where n represents the mean motion of the spacecraft which is constant which means

$$\begin{aligned}\Omega.\delta\mathbf{r} &= n\hat{\mathbf{k}}.(\delta x\hat{\mathbf{i}} + \delta y\hat{\mathbf{j}} + \delta z\hat{\mathbf{k}}) \\ &= n\delta z\end{aligned}$$

also,

$$\begin{aligned}\Omega \times \delta\mathbf{v}_{\text{rel}} &= n\hat{\mathbf{k}} \times (\delta x\hat{\mathbf{i}} + \delta y\hat{\mathbf{j}} + \delta z\hat{\mathbf{k}}) \\ &= -n\delta\dot{y}\hat{\mathbf{i}} + n\delta\dot{x}\hat{\mathbf{j}}\end{aligned}$$

substituting these equations into (2.20), we get

$$\delta\ddot{\mathbf{r}} = b\hat{\mathbf{k}}(n\delta z) - n^2(\delta x\hat{\mathbf{i}} + \delta y\hat{\mathbf{j}} + \delta z\hat{\mathbf{k}}) + 2(-n\delta\dot{y}\hat{\mathbf{i}} + n\delta\dot{x}\hat{\mathbf{j}}) + \delta\ddot{x}\hat{\mathbf{i}} + \delta\ddot{y}\hat{\mathbf{j}} + \delta\ddot{z}\hat{\mathbf{k}} \tag{2.25}$$

rearranging the terms in this equation yields,

$$\delta\ddot{\mathbf{r}} = (-n^2\delta x - 2n\delta\dot{y} + \delta\ddot{x})\hat{\mathbf{i}} + (-n^2\delta y + 2n\delta\dot{x} + \delta\ddot{y})\hat{\mathbf{j}} + (\delta\ddot{z})\hat{\mathbf{k}} \tag{2.26}$$

(2.26) gives us the components of absolute acceleration. Since we assumed the orbit to be circular, the mean motion is

$$
\begin{aligned}
n &= \frac{v}{r_0} \\
&= \frac{1}{r_0}\sqrt{\frac{\mu}{r_0}} \\
&= \sqrt{\frac{\mu}{r_0^3}}
\end{aligned}
$$

Hence, n can be written as

$$n^2 = \frac{\mu}{r_0^3} \tag{2.27}$$

Also, we can note that

$$\mathbf{r_0} \cdot \delta\mathbf{r} = (r_0)\hat{\mathbf{i}}.(\delta x\hat{\mathbf{i}} + \delta y\hat{\mathbf{j}} + \delta z\hat{\mathbf{k}}) \tag{2.28}$$

$$= r_0\delta x \tag{2.29}$$

Recalling (2.19), we can substitute (2.27), (2.29) and (2.23)

$$\delta\ddot{\mathbf{r}} = -n^2\left[(\delta x\hat{\mathbf{i}} + \delta y\hat{\mathbf{j}} + \delta z\hat{\mathbf{k}}) - \frac{3}{r_0^2}(r_0\delta x)(r_0\hat{\mathbf{i}})\right] \tag{2.30}$$

$$= 2n^2\delta x\hat{\mathbf{i}} - n^2\delta y\hat{\mathbf{j}} - n^2\delta z\hat{\mathbf{k}} \tag{2.31}$$

If we merge (2.31) and (2.26), we obtain

$$2n^2\delta x\hat{\mathbf{i}} - n^2\delta y\hat{\mathbf{j}} - n^2\delta z\hat{\mathbf{k}} = (-n^2\delta x - 2n\delta\dot{y} + \delta\ddot{x})\hat{\mathbf{i}} + (-n^2\delta y + 2n\delta\dot{x} + \delta\ddot{y})\hat{\mathbf{j}} + (\delta\ddot{z})\hat{\mathbf{k}}$$

After rearranging the terms on both left and right side we get

$$(\delta\ddot{x} - 3n^2\delta x - 2n\delta\dot{y})\hat{\mathbf{i}} + (\delta\ddot{y} + 2n\delta\dot{x})\hat{\mathbf{j}} + (\delta\ddot{z} + n^2\delta z)\hat{\mathbf{k}} \tag{2.32}$$

Which can be written as

$$\delta\ddot{x} - 3n^2\delta x - 2n\delta\dot{y} = 0 \tag{2.33}$$

$$\delta\ddot{y} + 2n\delta\dot{x} = 0 \tag{2.34}$$

$$\delta\ddot{z} + n^2\delta z = 0 \tag{2.35}$$

The set (2.35) represents the Clohessy–Wiltshire Equations of motion for deputy spacecraft in the LVLH frame of reference. The terms in (2.35) containing n^2 terms are the centripetal acceleration, $\dot{x}, \dot{y}$ are the Coriolis accelerations and $\ddot{x}, \ddot{y}, \ddot{z}$ terms are the total accelerations. This set of equations are coupled second-order differential equations which can be solved to obtain the analytical solution for the spacecraft.

$$\delta x = (4 - 3\cos nt)\delta x_0 + \frac{\sin nt}{n}\delta \dot{x}_0 + \frac{2}{n}(1 - \cos nt)\delta \dot{y}_0 \tag{2.36}$$

$$\delta y = 6(\sin nt - nt)\delta x_0 + \delta y_0 + \frac{2}{n}(\cos nt - 1)\delta \dot{x}_0 + \left[\frac{4}{n}\sin nt - 3t\right] \tag{2.37}$$

$$\delta z = \cos nt \delta z_0 + \frac{1}{n}\sin nt \delta \dot{z}_0 \tag{2.38}$$

including the change in velocity equations

$$\delta \dot{x} = 3n \sin nt \delta x_0 + \cos nt \delta \dot{x}_0 + 2\sin nt \dot{y}_0 \tag{2.39}$$

$$\delta \dot{y} = 6n(\cos nt - 1)\delta x_0 - 2\sin nt \delta \dot{x}_0 + (4\cos nt - 3)\dot{y}_0 \tag{2.40}$$

$$\delta \dot{z} = -n \sin nt \delta z_0 + \cos nt \dot{z}_0 \tag{2.41}$$

Where the initial conditions at $t = 0$ are given as

$$\delta x = \delta x_0 \qquad \delta y = \delta y_0 \qquad \delta z = \delta z_0$$
$$\delta \dot{x} = \delta \dot{x}_0 \qquad \delta \dot{y} = \delta \dot{y}_0 \qquad \delta \dot{z} = \delta \dot{z}_0$$

These sets of equations remove the requirement for numerical integration in the LVLH frame but the cost of flexibility because these equations are only valid for circular orbits.

2.4. SUMMARY

In this chapter, we discussed some background and variation in the gravitational field of the earth. The classical two-body problem is explored and the relative motion of two bodies is explained for the two-body problem. To reduce the complexity of the problem, the orbits around the earth are limited to the circular shape. Keeping that in mind, a linear approximation derivation is presented known as Clohessy–Wiltshire relative equations of motion. With these basics defined, the guidance problem is discussed next.

3

Assignment Strategy

3.1. INTRODUCTION

This chapter discusses the importance of the assignment phase of the satellite formation execution. The chapter begins with a brief explanation of what the assignment phase means and follows with an example of how a badly planned assignment could result in poor fuel utilization. The chapter further explores how to formulate the assignment algorithm and explains the algorithm execution with the help of an example.

One of the most important questions in the Automated Guidance problem is the destination assignment and the path planning phase. For Automated guidance in 2-D i.e. the surface of the earth, the path planning aspect is more important because the destination is regarded as a preset and the terrain navigation is more difficult. However, in this project, a group of satellites is required to be in a certain formation in Space. This 3-D nature, without much spatial restriction, relieves us from most of the terrain navigation problems for now. Although, the collision and safety problem remains common. The main focus becomes to select a destination minimizing time or fuel. The satellite should be able to figure out what position assume in the formation or what position would be most beneficial in the formation.

3.2. ASSIGNMENT PROBLEM

This assignment phase governs the process of helping the system find a selection order for satellites. According to this selection order, the satellites select a position out of an available destination pool based on individual preference. This availability pool is called the Bag of Destination (BoD), where the number of set elements, the destinations, are equal to $numsat$. $numsat$ attributes to the number of satellites participating in the formation maneuver. However, there might be cases where the number of destinations might be more or less than the $numsats$ depending on the final formation. Next, The *selection order* is a $numsat \times 1$ column vector. The element of the vector refers to the place in the queue of assignment preference.

We consider a simplified example of a particular case where the selection order vector shows the satellite s_2 will get to choose from the available positions shown in the *Destination set (BoD)*. At the end, each satellite has a preferred destination if available in the BoD.

$$\textit{selection order} \Rightarrow \begin{bmatrix} s_5 \\ s_1 \\ s_n \\ \vdots \\ s_2 \end{bmatrix} \qquad \textit{destination set} \in \{1,2,3,4.....n\}$$

In the literature, the most commonly used parameter for assignment is the distance of the satellites from their initial position to the possible destinations. This can be acquired simply by the resultant of the initial position vector and the possible final position vectors for each satellite unit.

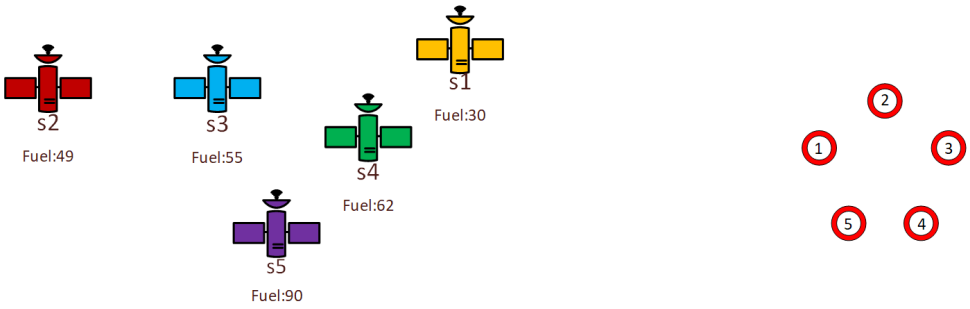


Figure 3.1: The satellites and available formation destinations

Following Figure 3.1, satellite s_2 is farthest of the satellite swarm. Hence, it gets to make the choice first. So, calculating the distance for each satellite to all the possible destinations a matrix can be constructed called a Distance matrix (D_{mat}). The D_{mat} will become the basis for the assignment of the destinations for the satellites.

3.3. ASSIGNMENT PARAMETERS

In the previous section, we introduced the idea of the distance between the satellites and the destinations as an assignment criterion. There can more than one criterion for assignment. Some of them are listed below:

- Total distance
- Reserve fuel level
- Expected fuel consumption
- Obstacle avoidance

Using these criteria, a matrix with priority values for each satellite to the destinations can be constructed. This matrix can be classified as the *Parameter matrix* (P_{mat}) in (3.1) where D_{mat} is the distance matrix as described in the previous section, F_{exp} is the expected fuel consumption matrix, F_{res} is the fuel matrix consisting the reserve fuel levels of the satellites. More assignment criteria can be added to (3.1) as per requirement.

$$P_{mat} = W_d D_{mat} + W_{fe} F_{exp} + W_{fr} \frac{1}{F_{res}} \tag{3.1}$$

The W_d, W_{fe}, and W_{fc} are the weights for the assignment criteria giving us flexibility in cases where one criterion is more important than the other. P_{mat} has the values that represent the priority of destination for each satellite. The higher value points to higher priority to the destination.

3.3.1. TOTAL DISTANCE

This criterion is simplest and straightforward to implement. The information is stored in a $n \times 3$ matrix where n is the number of satellites in the formation. The Parameter matrix based on distance will look as follows:

$$D_{mat} = \begin{array}{c} \\ s_1 \\ s_2 \\ \vdots \\ s_n \end{array} \begin{array}{c} \begin{array}{ccccc} d_1 & d_2 & d_3 & \dots & d_n \end{array} \\ \begin{bmatrix} D_{1,1} & D_{1,2} & D_{1,3} & \dots & D_{1,n} \\ D_{2,1} & D_{2,2} & D_{2,3} & \dots & D_{2.n} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ D_{n,1} & D_{n,2} & D_{3,n} & \dots & D_{n,n} \end{bmatrix} \end{array}$$

The satellites are numbered from s_1 to s_n on the left side of the column matrix, the destinations are from d_1 to d_n on the top, and the matrix is the distance matrix where $(D_{n,n})$ is the distance of each formation position from the current satellite position. Now using this distance matrix, the elements are sorted column-wise to understand what position is farthest/closest for each satellite termed *sorted distance matrix*.

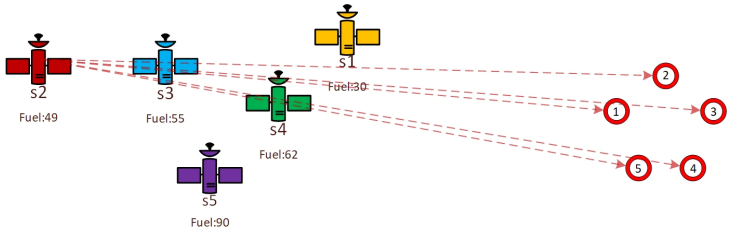


Figure 3.2: Possible trajectories for satellite s_2

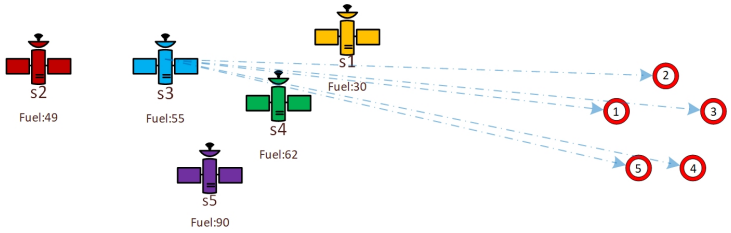


Figure 3.3: Possible trajectories for satellite s_3

The priority order for the satellites is simply from the farthest satellite to the nearest satellite. The farthest satellite from Figure 3.2, the red satellite ie satellite s_2, has all 5 of the formation destinations available to choose from. Hence, the priority for the red satellite is destination 1. Then, the blue satellite s_3 gets to choose from the available 4 formation positions as in Figure 3.3. The preference for the satellite s_3 was also destination 1 but since it was selected by the red satellite s_2, its next preferred destination is 5

which is available and so the process goes on. This is further explained as the assignment algorithm example.

3.3.2. RESERVE FUEL LEVEL

It is the same implementation as the distance-based one explained in the previous section. For this priority sequencing, the reserve fuel capacity (f_{res_n}) of the satellites decides which satellite gets to have a go at the process of selecting their final position out of the bag. The vector comprised of the reserve fuel level of the satellites is represented below as Fuel Level.

$$\text{Fuel Level} = \begin{matrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{matrix} \begin{bmatrix} f_{res_1} \\ f_{res_2} \\ \vdots \\ f_{res_n} \end{bmatrix}$$

$$F_{res} = \begin{matrix} \\ s_1 \\ s_2 \\ \vdots \\ s_n \end{matrix} \begin{matrix} \begin{matrix} d_1 & d_2 & d_3 & \dots & d_n \end{matrix} \\ \begin{bmatrix} f_{res_1} & f_{res_1} & f_{res_1} & \dots & f_{res_1} \\ f_{res_2} & f_{res_2} & f_{res_2} & \dots & f_{res_2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{res_n} & f_{res_n} & f_{res_n} & \dots & f_{res_n} \end{bmatrix} \end{matrix}$$

The F_{res} is a matrix comprised of the reserve fuel levels of the respective satellites arranged in a $numsat \times numsat$ dimensional matrix to satisfy the priority matrix P_{mat} calculation such that the priority parameter includes the factor of the reserve fuel level to each destination. The individual values of the matrix F_{res} are inverted to ensure the low reserve fuel value contributes to a higher priority score for the satellite.

Now, we go back to the image showing all the satellites and the formation positions Figure 3.1, we observe that the yellow satellite s_1 has the lowest amount of fuel left in reserve and the purple satellite s_5 has the highest amount of fuel reserve. In this case, we want the satellite with the lowest fuel reserve to go first because it minimizes the risk that the satellite might be assigned an unfavorable position rendering the satellite cluster incapable of another reconfiguration maneuver if one of the unit satellite fuel is too low. Here the yellow satellite s_1 makes the choice of destination 1 as probable one.

3.3.3. EXPECTED FUEL CONSUMPTION

The lines shown in the figures are just the distance vectors for each available destination from the satellites. However, the actual trajectory is not straight. It follows the celestial dynamics as explained in the previous chapter as shown in Figure 3.5. Hence, assigning the destination to the satellites just based on of distance does not paint the complete picture. Therefore, we define a new parameter called Expected Fuel Consumption (F_{ex}).

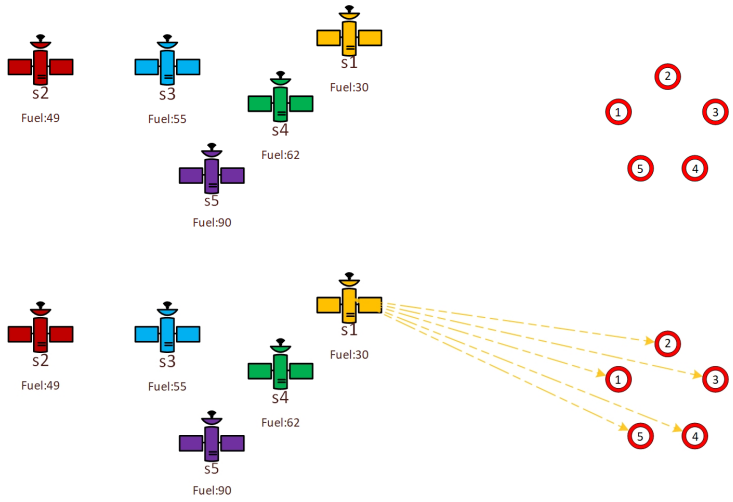


Figure 3.4: Possible trajectories for satellite s_1

The order is set up using this expected fuel consumption (F_{ex}) parameter. This parameter is calculated using an MPC controller where the MPC runs a simplified simulation without any state constraints except the physical limitation of the maximum thrust available from the thrusters. This optimized fuel expenditure profile is used to calculate the fuel consumption parameter ΔV. A more detailed explanation of the process is provided in the next chapters.

A fuel parameter matrix can be obtained that consists of the expected fuel expenditure parameter for each destination in the availability pool. Following previous example of Figure 3.4, satellite s_1 has the lowest amount of fuel left.

$$F_{exp} = \begin{array}{c} \\ s_1 \\ s_2 \\ \vdots \\ s_n \end{array} \begin{array}{c} \begin{array}{ccccc} d_1 & d_2 & d_3 & \dots & d_n \end{array} \\ \begin{bmatrix} f_{1,1} & f_{1,2} & f_{1,3} & \dots & f_{1,n} \\ f_{2,1} & f_{2,2} & f_{2,3} & \dots & f_{2.n} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ f_{n,1} & f_{n,2} & f_{3,n} & \dots & f_{n,n} \end{bmatrix} \end{array}$$

Now, an MPC gives us some possible trajectories in Figure 3.5 [1]. The generation of trajectories is discussed later in Chapter 4. Based on these trajectories, the choice would be between destination '4' or '5' with a minute difference in ΔV values and further availability in the bag of destinations. By sequentially executing this strategy, the satellites can be assigned destinations.

[1] The path lines in the figure are for illustration purposes only. The lines DO NOT signify actual trajectories.

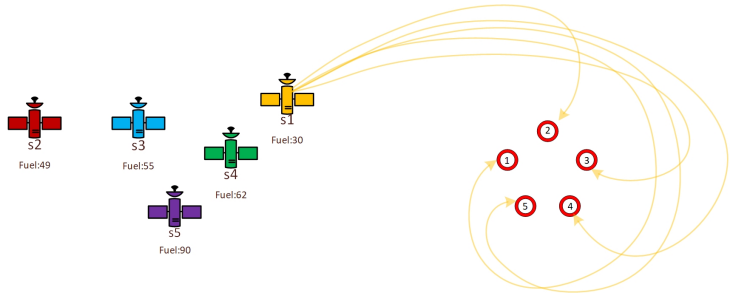


Figure 3.5: Possible trajectories for satellite s_1

In this matrix, the elements $f_{n,n}$ represent the expected fuel consumption values for each satellite s_n to each formation position. This matrix looks the same as the distance parameter matrix except the elements are the ΔV values for all the available positions for each satellite.

3.3.4. OBSTACLE AVOIDANCE

Here the obstacle is defined as any celestial body that is not part of this formation manoeuver. The obstacle avoidance scenario would include an additional parameter to the priority parameters in the P_{mat} matrix. This avoidance parameter will cover the possibility of obstacles that are known to the system, meaning the state of the obstacles' position and velocity is an available set of data that can be used to further improve the accuracy of the predicted trajectory giving us an even more precise fuel expenditure profile. This allows us to make better assignments ensuring the chance to save even more fuel.

However, the obstacle data set availability and quality are hard to determine. The probability of a scenario where every debris is tracked is very low because a large percentage of obstacles encountered are unknown such as debris of a destroyed spacecraft, out-of-commission satellite, or a random piece of rock. The parameter for obstacle dataset availability and quality could become an essential part of the future but as of now, it is not included either in the literature or in this document.

3.4. ASSIGNMENT ALGORITHM

Now that some of the viable parameters have been discussed, the assignment algorithm is the next step of the process. At the start of the assignment, all the destinations are available. This set of available destinations will be referred to as Bag of Destinations (BoD) as explained at the start of the chapter.

Now, the flexibility of the Parameter matrix (P_{mat}) is a very important part. The algorithm is designed to work with any kind of parameter be it distance measurements, fuel measurements or any other factors/scales. The initial positions and the available destinations have been randomized between $[-2,2]$kms and $[-50,50]$m to ensure the algorithm works on any kind of initial and final position dataset.As an example, the distance will be considered as the parameter so the parameter matrix P_{mat} looks as follows:

$$P_{mat} = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \begin{array}{c} \begin{array}{ccccc} d_1 & d_2 & d_3 & d_4 & d_5 \end{array} \\ \begin{bmatrix} 2.6202 & 2.6074 & 2.6637 & 2.6239 & 2.5778 \\ 2.7494 & 2.7763 & 2.7194 & 2.7600 & 2.7825 \\ 1.8560 & 1.8489 & 1.8048 & 1.8354 & 1.9018 \\ 2.0480 & 1.9983 & 2.0302 & 2.0053 & 2.0675 \\ 0.5702 & 0.5319 & 0.5623 & 0.5368 & 0.6011 \end{bmatrix} \end{array}$$

To start, P_{mat} is sorted using the MATLAB sort operation twice. Once row-wise, sort the parameter in ascending/descending order to each destination of each satellite. Considering distance as the assignment parameter, executing this row-wise sorting operation gives us the order of closest to farthest destinations and each satellite. This order helps us identify the selection order for each satellite as to when a satellite's turn comes to select, the preferred destination can be selected quickly out of the available destination from the bag.

$$P_{mat_{stored}} = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \begin{bmatrix} 2.5778 & 2.6074 & 2.6202 & 2.6239 & 2.6637 \\ 2.7194 & 2.7494 & 2.7600 & 2.7763 & 2.7825 \\ 1.8048 & 1.8354 & 1.8489 & 1.8560 & 1.9018 \\ 1.9983 & 2.0053 & 2.0302 & 2.0480 & 2.0675 \\ 0.5319 & 0.5368 & 0.5623 & 0.5702 & 0.6011 \end{bmatrix}$$

Further, the indices of the now sorted destinations can be extracted into a $numsat \times numsat$ matrix called Destination order (D_o) using the same MATLAB sort function. Following the same distance example, D_o looks as shown. Each row index of D_o represents the respective satellite and columns' slots in each row represent the destination index in priority.

$$D_o = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \begin{bmatrix} 5 & 2 & 1 & 4 & 3 \\ 3 & 1 & 4 & 2 & 5 \\ 3 & 4 & 2 & 1 & 5 \\ 2 & 4 & 3 & 1 & 5 \\ 4 & 3 & 1 & 5 & 2 \end{bmatrix}$$

The next task is the sorting of the Parameter matrix again but column-wise this time matrix of $numsat \times numsat$ dimensions again. This column-wise operation puts the satellites in the right order from farthest to closest thereby setting up the selection queue of the satellites.

$$P_{mat_{stored}} = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \begin{array}{c} \begin{array}{ccccc} d_1 & d_2 & d_3 & d_4 & d_5 \end{array} \\ \begin{bmatrix} 2.7494 & 2.7763 & 2.7194 & 2.7600 & 2.7825 \\ 2.6202 & 2.6074 & 2.6637 & 2.6239 & 2.5778 \\ 2.0480 & 1.9983 & 2.0302 & 2.0053 & 2.0675 \\ 1.8560 & 1.8489 & 1.8048 & 1.8354 & 1.9018 \\ 0.5702 & 0.5319 & 0.5623 & 0.5368 & 0.6011 \end{bmatrix} \end{array}$$

The indices of the sorted matrix give us the selection order matrix where the first element in the first column is the first satellite to pick a destination position from the Bag of Destination (BoD). In this case, all columns are identical in the A_o matrix showing the satellite s_2 is the farthest to each destination.

$$A_o = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 \\ 4 & 4 & 4 & 4 & 4 \\ 3 & 3 & 3 & 3 & 3 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix}$$

The assignment order matrix (A_o) and the destination order matrix (D_o) have been obtained, we move on to the next step of the Assignment as shown in Algorithm 1.

The result of this algorithm is the assignment of a destination for each satellite to one of the positions in the final formation. The required data set includes the set of assignable Destinations (BoD), the Parameter matrix (P_{pref}), The Assignment order matrix (A_o), and the Destination order matrix (D_o).

As we follow the example, the A_o matrix's 1st element shows satellite (s_2) gets destination selection priority out of the five satellites. Following, we shift our focus to D_o. In the D_o matrix, we check the second row which represents the priority of the destinations for satellite (s_2) which in this example is the destination (d_3). Now, the algorithm checks if d_3 is available in the bag of destinations (BoD). Since it is the first assignment, all the destinations are available. So, satellite s_2 is assigned its first preference i.e. d_3, and is ready to begin its journey. Subsequently, d_3 is removed from the bag of destinations. Now, there are $numsat - 1$ destinations available, that is 4 in the example we are following, for the other satellites to choose from.

Let us continue with another pass of the loop. The next satellite in order is s_1 from the Assignment matrix (A_o). Following the same procedure, the row corresponding to satellite s_1 is checked in the Destination matrix (D_o) i.e., the first row. The first priority of satellite s_1 is destination d_5. This destination d_5 is available in the bag of Destination (BoD) so, d_5 is assigned to the satellite s_1 and again is deleted from the bag of destinations. Now, there are $numsat - 2$ destinations available that is available destinations are down to 3 for 3 remaining satellites.

3

Algorithm 1: Destination Assignment

Result: D_{ass}, Assignment array with destination for each satellite
Data: Bag of destination(BoD) is a set of destinations, Parameter matrix(P_{pref})
begin
    Sorting operations
    Assignment order(A_o) = sort(P_{pref}, column-wise, ascending/descending)
    Destination order(D_o) = sort(P_{pref}, row-wise, ascending/descending)
    $D_{ass} = numsat \times 1$ null array
    for $i \leftarrow 1$ **to** $numsat$ **do**
        for $j \leftarrow 1$ **to** $numsat$ **do**
            for $k \leftarrow 1$ **to** $|BoD|$ **do**
                flag is 0
                if $A_o(D_o((i), j)) = Bod(k)$ **then**
                    $D_{ass}(D_o(i)) = A_o(D_o((i), j))$
                    remove the respective element from BoD
                    flag is 1
                    break
                end
            end
            if *flag* = 1 **then**
                display 'Assigned!'
                display D_{ass}
            end
        end
    end
end

In the next pass, satellite $s4$ is inline to select its destination and follows the same procedure. The satellite s_4 is assigned its first preference as well which is the destination d_2 because it was available and d_4 is also removed from the bag of destinations. Now, the following pass is the interesting one. Next up is satellite s_3 in the order as we keep following the Assignment matrix (A_o). The Destination matrix (D_o) row three shows the first preference for satellite s_3 is destination d_3. However, d_3 was also the priority of satellite s_2 and therefore, taken by s_2. Since d_3 is not in the bag of destination anymore, satellite s_3 will move on to its second preference which is destination d_4. Destination d_4 is currently available in the bag of destination(BoD). Hence, it can be and is assigned to the satellite s_3 and removed afterward from the bag.

In the final pass of the assignment algorithm loop, the satellite s_5 is the last satellite without a destination. The corresponding row to the satellite s_5 in the D_o matrix shows its first preference being destination d_4 which, unfortunately, is already taken by satellite s_3 in the previous loop. Its next favorable destination is d_3, which is again taken by satellite s_2 at the start of the assignment algorithm. Now, satellite s_5 has to consider its third preference, which is the destination d_1. Destination d_1 is available in the BoD and hence, is assigned to satellite s_5 and removed from the bag of destinations.

Following this assignment algorithm, the bag of destinations would ideally be empty because the number of available destinations is the same as the number of satellites. In the case where there are more destinations available than the total number of satellites, the algorithm would still work because the assignment occurs from a satellite's first to last preference. By the end of the algorithm, all the satellites would have an assigned destination to reach in the formation. In this explanation, the distance example is taken as it is the simplest one. The Parameter matrix would consist of fuel statistics as parameters in the following because it is the primary focus of this project.

3.5. SUMMARY

At the start of this chapter, the assignment problem was defined with appropriate illustrations. From the initial positions, satellites are assigned a target destination from a set called Bag of Destinations. The assignment is executed with priority parameters. The various possible choices of parameters are listed and explained briefly. The construction Parameter matrix P_{mat} is defined using the possible assignment parameters with appropriate weights. After P_{mat} construction, the proposed assignment algorithm is laid out. A demonstration is shown based on the satellite distance to the target destinations. With the assignment algorithm defined, we move on to the procedure to estimate the fuel expenditure which will be used as the major priority parameter.

4

GUIDANCE AND CONTROL

4.1. INTRODUCTION

The development of Control and Guidance techniques has garnered significant research in the last few years in spacecraft formation flying with a key focus on collision avoidance and optimizing fuel consumption. Multiple control strategies have been researched to achieve those goals from Multiple glideslope transfer [18], sliding mode control [19, 20, 21, 22] to Artificial Potential Function [23, 8] and Model Predictive control [24, 25, 26]. Guidance comes with an important and complex factor of the trajectory planning problem brilliantly explained by Hu et al. [27]. This project will focus on guiding a satellite using an MPC to its destination. The trajectory planning is out of scope for this project.

In this chapter, the state-space model is explained in Section 4.2. Then a couple of control strategies are discussed starting with Artificial Potential function(APF) in Section 4.3 and the basic terms and strategy are explained for Model Predictive Control keeping this project in reference in section 4.4. In section 4.4.1, we discuss the role of sampling time in MPC formulation for projects in space. In section 4.4.2, we examine how the prediction parameters in the MPC impact the controller performance. Further, the cost function is constructed based on the project in section 4.4.3 followed by system constraints in section 4.4.4 as they are tied to each other. With these basics, the Optimal Control Problem is formulated in section 4.5.

4.2. STATE-SPACE SYSTEM MODEL

We consider an autonomous formation reconfiguration maneuver for the 'deputy' satellites to the target destinations using a guidance strategy that is discussed in this chapter. We set up the simulation to calculate the expected fuel consumption ΔV used in the Parameter matrix construction for the Assignment of satellites as discussed in the previous chapter. The dynamics for the satellites are defined using the Clohessy-Wiltshire equations derived in the previous Chapter. The CW equations are as follows:

$$\begin{aligned} \delta\ddot{x} - 3n^2\delta x - 2n\delta\dot{y} &= 0 \\ \delta\ddot{y} + 2n\delta\dot{x} &= 0 \\ \delta\ddot{z} + n^2\delta z &= 0 \end{aligned}$$

where the mean motion n is defined as:

$$n = \sqrt{\frac{\mu}{a^3}}$$

The dynamics system can be represented in a state-space model [28, 29] as follows:

$$\dot{X} = AX + BU \tag{4.1}$$

The state matrix X is defined as:

$$X = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \\ \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{bmatrix}$$

The Control matrix U consists of the control input in the 3-dimensions:

$$U = \begin{bmatrix} u_x(t) \\ u_y(t) \\ u_z(t) \end{bmatrix}$$

The system matrices based on the CW equations are as following:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -3n^2 & 0 & 0 & 0 & -2n & 0 \\ 0 & 0 & 0 & 2n & 0 & 0 \\ 0 & 0 & n^2 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

With the Problem and the Model defined, we take a look at the preferred guidance strategies.

4.3. ARTIFICIAL POTENTIAL FUNCTION

Artificial potential functions offer simple mathematical guidance laws that are implemented in real-time and do not require any prior assumptions concerning the system dynamics. The objective of this control algorithm is to guide the element to the target with the help of a negative gradient. In the literature, one of the better research papers is written by Theodore Wahl and Kathleen Howell [30]. They explain it as the space from the element to target position is defined as an artificial potential zone. This approach is accomplished by creating the target position at the lowest potential point in the potential field and all obstacles are placed at high potential. Now the element follows the path of negative potential gradient avoiding any high potential areas to the required target lowest potential location. Figure 4.1 shows the idea behind the said obstacles made to be potential peaks and the system has to define the path around them.

A basic approach is explained by Wahl and Howell [30]. The potential field is generated by incorporating attractive and repulsive pieces. A Lyapunov function is chosen as the quadratic attractive potential function (ϕ_a) that is based on the separation between the spacecraft in Hill's frame (ρ) and target position (ρ_t) described as

$$\phi_a = \frac{k}{2}(\rho - \rho_t)' Q (\rho - \rho_t) \tag{4.2}$$

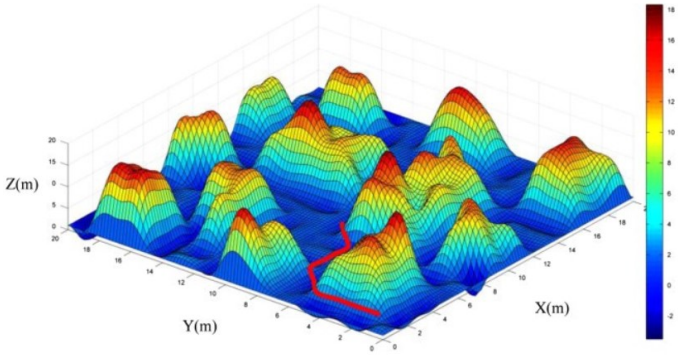


Figure 4.1: Artificial Potential Function guidance

k is a scalar weighting factor and Q is a positive-definite matrix describing the shape of the potential and is spacecraft symmetrical. The repulsive potential for a single obstacle is defined as

$$\phi_r = \frac{K}{2} \frac{(\rho - \rho_t)' Q (\rho - \rho_t)}{(\rho - \rho_0)' P (\rho - \rho_0) - 1} \tag{4.3}$$

where K is a scalar weighting factor, ρ_0 is the initial position, and P is the positive definite matrix describing the shape of an ellipsoid. This shape takes care of the uncertainty of the element position and shape. If there are multiple objects, say n obstacles, $\rho_{0,i}$ describes the i^{th} obstacle position and the repulsive potential is a sum of all n obstacles that need to be avoided.

$$\phi_r = \frac{K}{2} \sum_{i=1}^{n} \frac{(\rho - \rho_t)' Q (\rho - \rho_t)}{(\rho - \rho_{0,i})' P (\rho - \rho_{0,i}) - 1} \tag{4.4}$$

The obstacles can be known spacecraft within the formation or other nearby operations and can be some unknown debris. The total sum of the potential portions is

$$\phi = \phi_a + \phi_r$$

. The desired velocity is the negative gradient of the total potential

$$v_d = -\nabla\phi = -\nabla\phi_a - \nabla\phi_r \tag{4.5}$$

This is the relative velocity definition in Hill's frame. Now, the spacecraft must match the target velocity where the error ϵ_v is defined as the difference in spacecraft and target velocity vector. Based on this error, the algorithm defines the control input ΔV as the difference in the current spacecraft velocity and the error.

$$\Delta V = v_d - \epsilon_v \tag{4.6}$$

However, the APF control method is not inherently optimal because it does not consider the system dynamics. This results in unnecessarily large ΔV i.e. the thrust as maneuver requirement further resulting in more correction maneuvers. This drawback is discussed by Izzo and Pettazzi [8]. Their study shows that APF uses the most amount of propellant compared to Q-guidance or SMC discussed in the simulation. The reason is the APF technique is shown to be robust but the correctional control usage is quite high which in turn results in excessive fuel usage.

An extension is proposed again by Wahl and Howell [30] based on the work of Josue Munoz [31] called The Adaptive Artificial Potential Function(AAPF). Munoz explains in his dissertation that APF is simple and shows favorable convergence characteristics. APF can be modified by embedding the dynamics and performance index in its algorithm using a state transition matrix to adapt the system relative to the dynamics. These studies successfully showed the ability of APF and AAPF to guide the spacecraft to their designated locations but fail to remark on the key point of fuel efficiency.

Wahl and Howell further go to show in their research [32, 30, 23] another guidance approach using the Model Predictive control for better control over fuel consumption. In the next section, we explain how MPC formulation works.

4.4. LINEAR MPC-BASED GUIDANCE

At its root, MPC is an optimal control problem where a dynamic system model is used to optimize the prediction resulting in the best decisions. MPC is a form of control strategy where the general design objective is the computation of the control action trajectory by solving an optimal control problem at each time step for the optimized future behavior. The optimization is carried out inside a limited time frame where its current state is its initial state at the start of the time window. The MPC computes the control action online which differentiates MPC from the conventional control strategies where the control law is precalculated offline [33].

Furthermore, Liuping Wang in his book [34] briefly points out the parameters for an MPC described as following:

- Moving horizon sampling time window T_p is a constant time window from an arbitrary point t_i to $t_i + T_p$.
- Prediction Horizon (N_p) is the parameter that governs how far ahead the prediction takes place. Same length as moving horizon window T_p.
- Receding horizon control referring to predicting the full horizon but using only the first prediction value in the next time step and repeating the process.
- For prediction computation, state information vector x_i is needed at the current time step t_i directly measured or estimated.

- A precise dynamic model is a requirement for the MPC to consistently perform during operation.

- The best decision is made based on the criterion which is an error function between suitable and real response. The optimal control is found by minimizing the said error function

4.4.1. SAMPLING TIME

For an MPC implementation, the continuous system is discretized according to sampling time (T_s), which in turn refers to the sampling frequency ($1/T_s$). The Sampling frequency is the interval when data is collected and hence, quite an important factor for real-time applications. A higher sampling frequency i.e. lower sampling time means the data is being collected more frequently. The high rate of data collection results in a high computational load. The computation factor is often the bottleneck for a Control system's efficacy. The higher frequency for data collection allows the controller to follow the system dynamics more closely. So, for a system with dynamics that changes within a millisecond, a computer with high computational capacity is a necessity otherwise the control system will tend to miss the dynamics resulting in poor performance. For this project, the satellites are operated in low-earth orbit with the primary objective of minimizing fuel consumption. Given reasonably ample time to finish the operation, the sampling time for the system is usually in minutes [32, 8, 18].

The satellites being small and starved for space, the onboard computers are traditionally not powerful ones. So, designing a control strategy with low computational power high sampling time is chosen. Furthermore, the fuel consumption can also be reduced by spacing the times the control action is applied. Hence, keeping sampling high also increases the interval where the control action can be applied. However, the sampling time cannot be a large interval either because a large sampling time interval can destabilize the controller. A compromise is necessary to achieve both objectives, minimizing fuel consumption and accuracy of the reconfiguration operation.

4.4.2. PREDICTION AND CONTROL

MPC involves future prediction to calculate the optimal control input. There are two types of predictions made [36]. System state predictions are sometimes referred to as set-points and control predictions. These predictions on the future state are based on the Plant system model. The system is optimized for each sampling interval throughout multiple samples called Prediction Horizon (N_p). While using the MPC, Optimal control input is calculated over the prediction termed as Control Horizon (N_c). Over the control horizon, the control input from the next time step is applied. In the next iteration step, the Prediction and Control horizon are moved to another time step making a new prediction calculation for states and optimal control. The system has a terminal state to achieve, the control system drives the plant system towards it with each time step until it encounters the terminal state in the prediction horizon. This is called a Receding Horizon control as the prediction horizon recedes to the terminal state. The process is aptly described in Figure 4.2.

Now, for each step of the prediction horizon, the optimized control input is calculated. If it's longer than the prediction horizon, the system cannot optimize control for states that are not predicted yet. The Control Horizon (N_c) has to be shorter or the same as the Prediction Horizon (N_p).

$$N_c \leq N_p \tag{4.7}$$

Furthermore, The prediction time window (T_p) can be defined as the time frame in which prediction is carried out for each sampling time (T_s), written as follows:

$$T_p = N_p.T_s \tag{4.8}$$

Following Figure 4.2, it is shown N_c is the same as N_p. In this way, the control inputs are available for each prediction state. Only the first control input value $u_c(k|k)$ is used by the MPC for each iteration. However, to simplify calculations MPC computes the change in the input (Δu_c) and calculates u_c afterward. When $N_c < N_p$, the change in control input (Δu_c) is assumed as zero for the time steps between N_c and N_p.

Selecting the length of the Predication Horizon is very important. The system must be allowed to predict in significant future anticipating any problems. So, the prediction horizon should not be too short as it could result in system instability and bad reference tracking. On the other hand, the benefits of a long prediction horizon are also evident. The farther ahead in time the control system knows the state the better decisions can be made in the present. This comes at the cost of increased computational capacity as discussed earlier and the risk of compounding errors or uncertainties over a long prediction

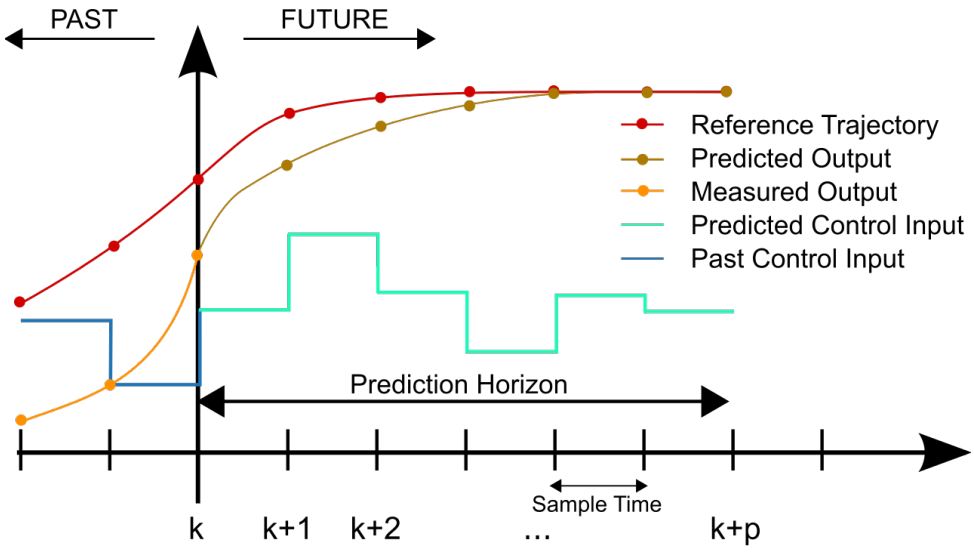


Figure 4.2: MPC working with Prediction and Control Horizon [35]

horizon. The prediction horizon should be long enough so that the system is stable and not starved for resources due to high computation demand.

4.4.3. COST FUNCTION

We have established that MPC is a receding horizon control scheme using a feedback loop for position and velocity to compute the predictive solution over the moving horizon. The dynamic model used for this project is linear defined as following:

$$x_{k+1} = Ax_k + Bu_k \tag{4.9}$$

where x_k is the state vector at time t_k, u_k is the control vector, A is the system matrix and B is the control matrix. A standard cost function consists of tracking error and control action along the moving horizon N_p. The control action is the first element of a sequence obtained by minimizing the cost function at each time instant. The system is updated and the next step is computed and so on. The objective function can be defined as:

$$J(U_k, x_k, x_k^{ref})$$

where U_k is the stacked control vector, x_k is the current state vector at t_k and x_k^{ref} is the reference tracking signal as explained by Gros et. al [37]. The objective function can be written as the sum of two functions J_1 and J_2:

$$J(U_k, x_k) = J_1(x_{k+N_p-1}) + J_2(x_{k+N_p}) \tag{4.10}$$

where J_1 is defined as quadratic positive semi-definite stage cost:

$$J_1(x_k, x_k^{ref}, u_k, u_k^{ref}) = \sum_{k=0}^{N_p-1} \frac{1}{2} \begin{bmatrix} x_{i,k} - x_{i,k}^{ref} \\ u_{i,k} - u_{i,k}^{ref} \end{bmatrix}' W_{i,k} \begin{bmatrix} x_{i,k} - x_{i,k}^{ref} \\ u_{i,k} - u_{i,k}^{ref} \end{bmatrix} \tag{4.11}$$

$$W_{i,k} = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix}$$

Where $W_{i,k}$ represents the augmented matrix made of Q and R weight matrices for states and control input respectively. Furthermore, the terminal cost J_2 is expressed as:

$$J_2(x_{k+N_p}, x_{term}) = (x_{k+N_p} - x_{term})' W_{term} (x_{N_p} - x_{term}) \tag{4.12}$$

where W_{term} represents the positive semi-definite terminal cost matrix on the final time step. Now, we have an Objective function around which the system can run an optimization for the control input.

4.4.4. SYSTEM CONSTRAINTS

The significant advantage of MPC over other control strategies is the application of explicit constraints on the control inputs and system output. There are generally three types of constraints that can be put on the system, two of them on the control input $u(k)$ and the third type is state $x(k)$ or output $y(k)$ constraint as explained by Liuping Wang [34]. These hard constraints on the amplitude of the control input can be defined as:

$$u_{min} \leq u_{i,k} \leq u_{max}$$

where u_{min}, u_{max} are the limits on the amplitude and the constraint on the change in the control input is the second-hand constraint that can be applied as:

$$\Delta u_{min} \leq \Delta u_{i,k} \leq \Delta u_{max}$$

As discussed earlier, the satellites have a large bias towards keeping the weight low where the compromise for the amount of fuel and the thrust available comes into play. There are many types of thrusters available to choose from as NASA [38] and ESA [39].The available thrust can vary from micro-Newtons to tens of Newtons but in small satellites, thrust output is mainly between milli-newtons(mN) to Newtons(N).

Now, the output or state constraints are put on the system to define the operating range similarly:

$$x_{min} \leq x_{i,k} \leq x_{max}$$

In addition to this, the system dynamics themselves also act as constraints in the MPC problem. The problem itself is defined as linear or non-linear based on the constraints and the system dynamics.

4.5. OPTIMAL CONTROL PROBLEM FORMULATION

From the previous sections, we can say that the Linear Model Predictive Control is a fitting control strategy for the Assignment and Guidance problem to handle the dynamics and constraints for the problem. The controller is set up to accommodate the non-linearities that can be added to the dynamics or the constraints. The major issue with the implementation of an onboard MPC is the high computational cost and computation time required to get a solution. In recent times, this problem has been overcome with the advancement in onboard chips becoming faster and getting smaller at the same time. The researchers have not been sitting idle either. The developments in algorithm speed and computational efficiency have been significant making even non-linear controllers affordable for onboard computations [40, 41, 42].

Hence, a MPC problem can be defined in (4.13). The control input u_i is defined for a state estimate $\hat{x}_i$ at every time instant i.

$$QP_{MPC}(\hat{x}_i, x_i^{ref}, u_i^{ref}, x_{term}, u_{term}) = \tag{4.13}$$
$$arg \min_{\Delta x_i, \Delta u_i} \sum_{k=0}^{N-1} \frac{1}{2} \begin{bmatrix} \Delta x_{i,k} \\ \Delta u_{i,k} \end{bmatrix}' W_{i,k} \begin{bmatrix} \Delta x_{i,k} \\ \Delta u_{i,k} \end{bmatrix} + (x_{k+N} - x_{term})' W_{term} (x_N - x_{term})$$

$$s.t. \quad \Delta x_{i,0} = \hat{x}_i - x_{i,0}^{ref}$$
$$\Delta x_{i,k+1} = A_{i,k} \Delta x_{i,k} + B_{i,k} \Delta u_{i,k} + r_{i,k} \qquad k = 0 \cdots N-1, \tag{4.14}$$
$$C_{i,k} \Delta x_{i,k} + D_{i,k} \Delta u_{i,k} + h_{i,k} \leq 0 \qquad k = 0 \cdots N-1, \tag{4.15}$$

4

where x_i^{ref}, u_i^{ref} are the reference values at time instance i with the prediction starting at current time instance. The system is constrained with dynamics and constraints on input using (4.14) where A, B represent the system matrix and the control input matrix. The actuator limitations or obstacle avoidance is included in (4.15) where C, D represent the output matrix and feed forward matrix. The notation given here $x_{i,k}$ suggests to the k^{th} element from $k = 0, \cdots, N-1$ in the sequence x_i provided at time instance i. The error of the predicted and reference values are provided at time i as:

$$\Delta x_{i,k} = x_{i,k} - x_{i,k}^{ref} \qquad k = 0 \cdots N \tag{4.16}$$
$$\Delta u_{i,k} = u_{i,k} - u_{i,k}^{ref} \qquad k = 0 \cdots N-1 \tag{4.17}$$

At every time instance i, the input is applied to the system:

$$u_i^{MPC} = u_{i,0}^{ref} + \Delta u_{i,0} \quad (\Delta x_i, \Delta u_i) = QP_{MPC}(\hat{x}_i, x_i^{ref}, u_i^{ref}) \tag{4.18}$$

The problem formulation in (4.13) is a structured Quadratic Program where $A_{i,k}, B_{i,k}$ are constant in the Linear Time-Invariant model or the Linear Time-Variant model where $A_{i,k}, B_{i,k}$ are developing with time. The affine terms $r_{i,k}$ are zero if the reference values x_i^{ref}, u_i^{ref} satisfy the dynamics of the system. The weights $W_{i,k}$ and W_{term} are positive semi-definite diagonal matrices.

The next important factors are the weights used in the optimization problem. The $W_{i,k}$ matrix is an augment matrix made of matrices S representing the penalty matrix on the difference in states from the reference values except for the final state, and R is the penalty matrix on the control cost. The weight matrix W_{term} is the semi-definite matrix imposing penalties on the difference in the terminal state of the satellite.

This terminal weight can also be calculated using Discrete-time Algebraic Riccati Equation(DARE) as the solution of the unconstrained closed-loop system leads to asymptotic stability given R and $\bar{S}$ being positive definite matrices [34, 23, 43, 44]. The terminal weights can be represented as:

$$W_{term} = A' W_{term} A + \bar{S} - H'(R + B' W_{term} B) H \tag{4.19}$$
$$H = (R + B' W_{term} B)^{-1} B' W_{term} A$$

The Algebraic Riccati Equation solves for the W_{term} at the start of each time step using MATLAB algorithm `eDARE`. Theodore [23] states that the terminal state of the satellite becomes a state penalty instead of a hard terminal constraint forcing the controller to build an optimal control history. With the problem completely defined, we can carry on with the selection of the simulation toolbox in MATLAB SIMULINK.

The MATLAB SIMULINK provides an MPC environment to work with which is a solid solution for problems that have a defined reference but it does not offer the framework regarding the terminal state computations and constraints. In this regard, the ACADO toolbox [45] provides a thorough implementation of the MPC including non-linear application. ACADO comes with discretization and linearization algorithms for non-linear systems and solution computation algorithms. Sequential Quadratic Programming(SQP) is used to compute the solution. SQP is generally used for solving constrained non-linear problems by sequentially approximating the optimization problem providing direction to Newton's steps towards the solution from an available guess [46]. For this project, ACADO is used to solve the MPC problem as per the settings provided in the table 4.1. These chosen settings are further explained.

Parameter	**ACADO Setting**
Hessian Approximation	Gauss-Newton
Discretization Type	Multiple Shooting
Sparse QP solution	Full Condensing N2
Integrator Type	Explicit RK4 integrator
Number of Integrator steps	2N
QP solver	qpOASES3
Levenberg Maquardt Parameter	$1e^{-4}$

Table 4.1: ACADO toolbox settings for the MPC solver

Commonly used approaches for the discretization of system dynamics are based on boundary value problem-solving techniques. These techniques are divided into single shooting or multiple shooting methods. The direct multiple shooting method divides the time interval $(t_0, t_o + T)$ into several smaller intervals $(k, k+1)$ for $k = 0 \cdots N_p - 1$ and solves the initial value problem for each interval. The constraints over the state are discretized over the same interval as the states and controls. On the other hand, the single shooting method solves the boundary problem over a single interval. The multiple shooting method has been shown to perform superior to single shooting in modeling and numerical stability [47, 48]. Hence, the Multiple shooting method is chosen for this project.

The linearization of non-linear constraints or models requires accurate approximations by the integrator to ensure feasible prediction for the system. ACADO provides vari-

ous integrators implicit and explicit in nature. In this project, we are using a non-stiff dynamic so an explicit Range-Kutta integrator is a good choice for this MPC problem computation.

Another important choice here is the integration steps and order of the integrator. This choice again depends on the available computation effort. A smaller step size or a higher order of integrator will result in more accurate computations at the cost of processing power and time. Since we are using a linearized dynamics model, we can afford some computational leeway and go for the Range-Kutta integrator order 4 with a step size of $2 * N$ i.e. twice the prediction horizon. The next choice we have to make is the Quadratic Problem solver. The QP 4.13 can be solved using various online solvers included in the ACADO toolbox and are categorized based on their constraint handling approach.

The interior-point(IP) method of optimization solver uses a barrier function called barrier method IP where the barrier function replaces the limits and is added to the optimization problem. When these barriers are violated, the cost of function increases. Afterward, an optimal solution is calculated using Newton's method. Another approach to the QP solution is using the Active Set (AS) method where at the current state, the algorithm finds a working set of active constraints and computes the resulting QP problem with equality constraints. The algorithm is repeated until optimality is achieved. The IP method uses less iterations than AS however, the AS algorithm gets faster with each calculation [40, 49]. Farreau et al. [40] further propose a qpOASES QP solver that employs AS algorithm for constraint handling.

With the ACADO ready to go according to this explanation, we move on to see how the system performs based on these parameters and set up the simulation for the MPC Control Strategy.

4.6. SUMMARY

The chapter introduces the various control strategies employed in the literature to solve a navigation problem. Artificial Potential Function is briefly explained. But an MPC is chosen as the control strategy of choice due to its inherent optimization strategy. Next, the dynamics model is constructed using the CW equations of motions for the MPC controller. Furthermore, the linearized MPC strategy is discussed along with all the control parameters and constraints. An approach of introducing dynamic terminal weights by using Algebraic Riccati Equation is introduced. The ACADO toolkit is employed as the interface to set up the MPC controller in the MATLAB environment. The ACADO's parameter choices are displayed and explained briefly. With the controller ready to estimate the ΔV values, we go on ahead with simulations to test the proposed assignment algorithm.

5

SIMULATIONS AND RESULTS

5.1. INTRODUCTION

In this chapter, we analyze the working of the algorithm in tandem with the fuel values extracted from the MPC simulations. The randomized initial conditions are discussed and the MPC simulated trajectories are highlighted. Furthermore, the expected fuel consumption extracted from the MPC simulations for each satellite to each destination is also analyzed. Based on this information, the algorithm is tested and the destination for each satellite is assigned. It should be noted that the tuning and setup of the MPC and the assignment algorithm are kept the same throughout the simulations.

5.2. SIMULATION FRAMEWORK

Now, that the background is established, the simulation setup is defined in this section with spacecraft specs, initial condition, MPC parameters, and constraints. After the simulation is carried out with the mentioned conditions, the results are analyzed.

5.2.1. SPACECRAFT SPECIFICATIONS

The spacecraft chosen for this project is a small satellite. All the satellites in the cluster are of the same type and dimensions i.e. a Homogenous cluster of satellites. Similar to that of the Starlink operation [50] by SpaceX. Table 5.1 shows the specifications for the satellites considered in this project. Furthermore, the satellites are assumed to have intercommunication capabilities throughout the simulation with common data access with their stats available at all times during the simulation.

Orbit	Orbital Period	Mass	Thrust
Low earth-300 km	5431 s	100 kg	1 N

Table 5.1: Satellite Specifications

5.2.2. INITIAL CONDITIONS

The satellites are deployed via a rocket into the low earth orbit. The rocket is launched into orbit and the payload fairings are jettisoned into space as explained by NASA [51]. Then, the spacecraft must make its way to the desired orbit using its thrusters. This initial deployment will result in randomly distributed satellites near the destination orbit which is the Chief Orbit. For this project, the satellites must make their way themselves to the destinations.

Satellite	Radial	Along-Track	Cross-track
s_1	-1747	-1675	848
s_2	-97	1265	-1772
s_3	-936	-435	-191
s_4	1131	1427	635
s_5	-398	1161	-1722

Table 5.2: Satellite initial positions [m]

The initial and final positions for the satellite are decided using the randomize function

available in MATLAB. The algorithm was tested on randomized scenarios to emphasize its ability to deal with any general proximity reconfiguration maneuvers. For this simulation demonstration, the initial positions for the satellites are randomized between ±2000 m as shown in Table 5.2.

Destination	Radial	Along-Track	Cross-track
d_1	-40	-21	-21
d_2	-18	-28	29
d_3	-30	27	42
d_4	34	-10	-17
d_5	32	38	36

Table 5.3: Satellite final destinations [m]

The limits for the randomization were chosen to be 2000 m because the dynamics model used here is the Clohessy-Wiltshire relative model which is linearized to simplify the formulation. As the distance is increased to tens of Kilometers, the trajectories were observed to be inaccurate. The 2000 m limit is far enough to be still considered as proximity and close enough that the model captures the dynamics to acceptable accuracy.

Similarly, the destinations are also randomized between ±50 m as shown in Table 5.3. The limits for the final designated randomization were chosen as 50 m to show the system can deliver the satellite to meters accurately.

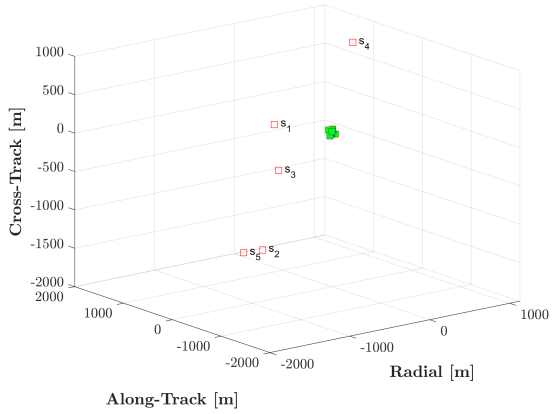
The set of randomized positions were chosen such that the satellites are spread in space in ±2000 m. Figure 5.1 shows the initial state of the simulation showing the relative initial positions to the destinations in HIll's frame of reference where the black star represents the chief location as the origin of the relative frame. The red squares represent the initial positions for satellites s_1 to s_5 and the green squares represent the available destinations d_1 to d_5.

5.2.3. MPC PARAMETERS AND CONSTRAINTS

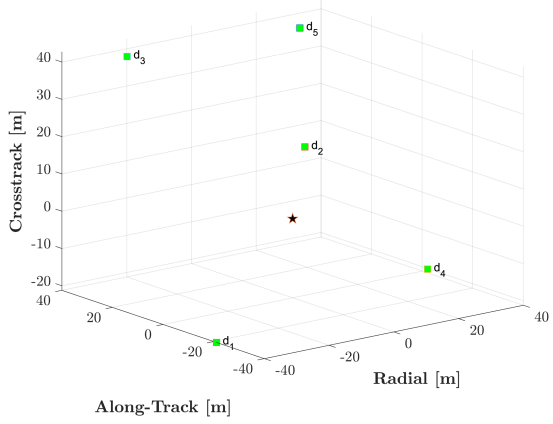
For this project, no obstacles are considered during the MPC simulation because as explained previously, this is a preliminary simulation to estimate the fuel consumption of a satellite to the available destinations. There are no state transition constraints either because we do not have a reference trajectory only a reference final position. Hence, a terminal constraint is added to the MPC.

The next system constraint is for the maximum allowed acceleration to the system accounting for the physical limiting capability of the satellite thrusters as in Table 5.4. Thrusters here are considered to be cold gas thrusters with maximum available thrust at 1 N. The thrusters are assumed to provide maximum thrust instantaneously in the required x,y, or z-direction.

As we discuss in previous Chapter 3, the choice of the MPC parameters for the available computation power and time for onboard calculation is reiterated here in Table 5.5. In addition to this, the weights for the MPC as shown in the same table. The state error cost is kept low because, as mentioned previously, we do not have a state trajectory to follow



(a) Initial positions for satellites s_{1-5}



(b) Available Destinations d_{1-5}

Figure 5.1: Initial satellite positions depicted as red squares and available destinations are green squares with the black star as the origin of chief orbit reference frame

Contorl Input	
u_{min}	$-0.01\,\mathrm{m\,s^{-2}}$
u_{max}	$0.01\,\mathrm{m\,s^{-2}}$

Table 5.4: Control input Constraints

only the final destination. On the other hand, the control cost is kept high to emphasize the optimization of fuel consumption to be minimum to the reference which is taken to be zero. Control cost any higher than the selected value would break the control.

Now, The terminal constraint is used in two ways in the simulations for comparison. As discussed in the previous chapter, the terminal cost is not applied but controlled with terminal cost weights. Another method is adding the destination as a terminal constraint at the time of exporting MPC block for SIMULINK where it becomes a hard constraint. In

Sampling Time (T_s)	Prediction Horizon (N)
60 s	50

State error Cost(Q)	Control Cost(R)	Terminal Cost (W_{term})
$1 \times 10^{-8} * I_{6\times6}$	$1 \times 10^{3} * I_{3\times3}$	$1 \times 10^{10} * I_{6\times6}$

Table 5.5: MPC parameters and weights [m]

Figure 5.2, it can be observed the effect of terminal cost addition to a constant terminal cost weight scenario.

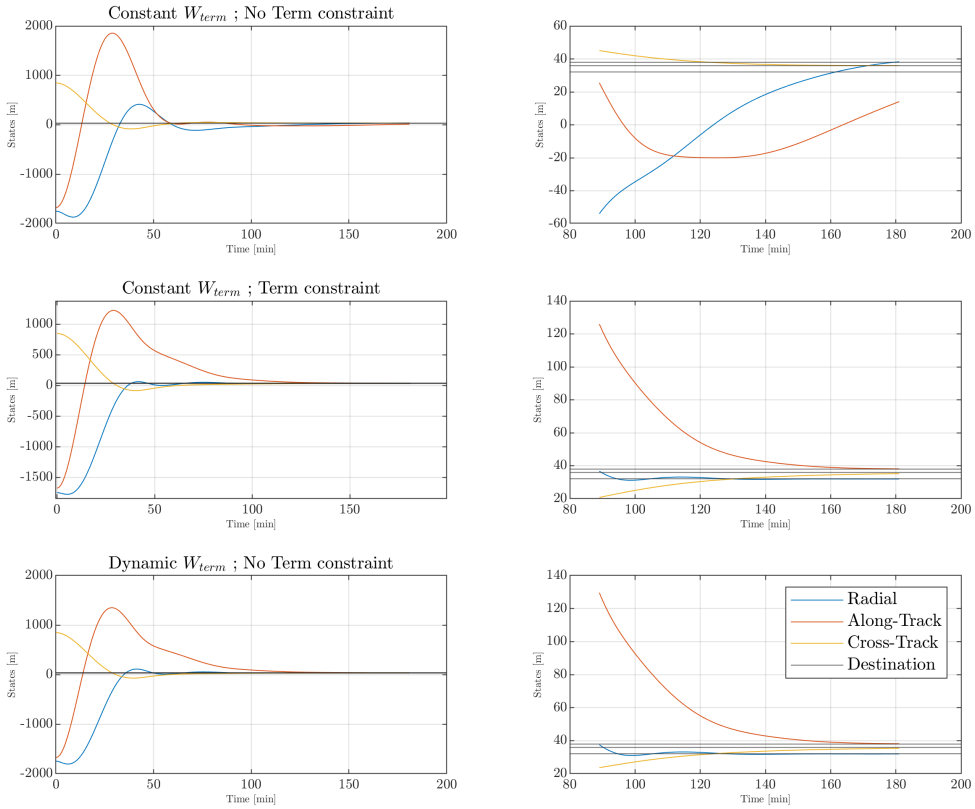


Figure 5.2: The Comparison between scenarios with Terminal constraint constant terminal cost and Constant or Dynamic terminal cost weights without terminal constraint

The scenario with constant terminal cost weights shows is unable to converge in time of the simulation with a larger overshoot in comparison to the scenario where the controller is additionally dealing with a terminal constraint. The scenario with terminal constraint performs much better and can achieve the target state in the given time. However, adding a hard terminal constraint to the controller also increases computation load and time. This scenario with terminal constraint has another drawback while

simulating using the ACADO toolkit. The ACADO toolkit adds the terminal constraint as a hard constraint while compiling the s-function. Hence, the user is forced to compile a new s-function every time the destination is changed making the whole process much more complex and time-consuming. Another important observation here to make is the performance similarity for the scenario with terminal constraints and Dynamic weights without terminal constraints. Both scenarios converge to target states within the given time.

These dynamic weights are calculated using the DARE Equation for each time step as explained in Chapter 4. The Introduction is dynamic weights alleviates the problem with the application of hard constraints providing flexibility to the application without changing the s-function with each iteration.

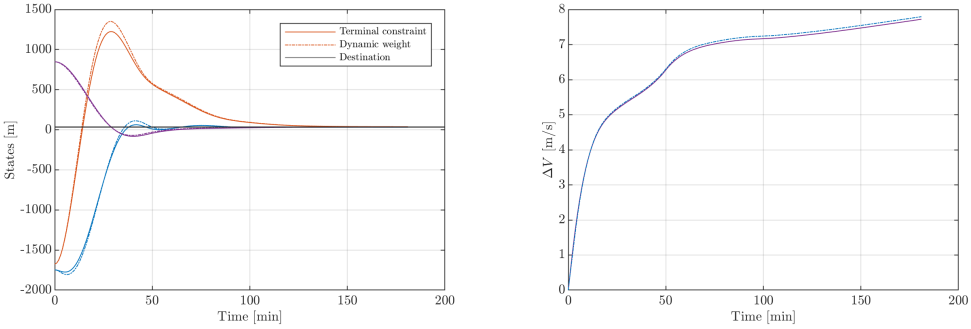


Figure 5.3: Performance comparison for scenarios with Dynamic weights and Constant weights with terminal cost.

Furthermore, It is observed in Figure 5.3 the controller with dynamic weights can closely keep up with the controller with the terminal constraint with state transition and the fuel expenditure. Hence, the controller with the dynamic weights is a viable option to be considered for controller design. With the setup defined, we carry on with the simulation scenarios.

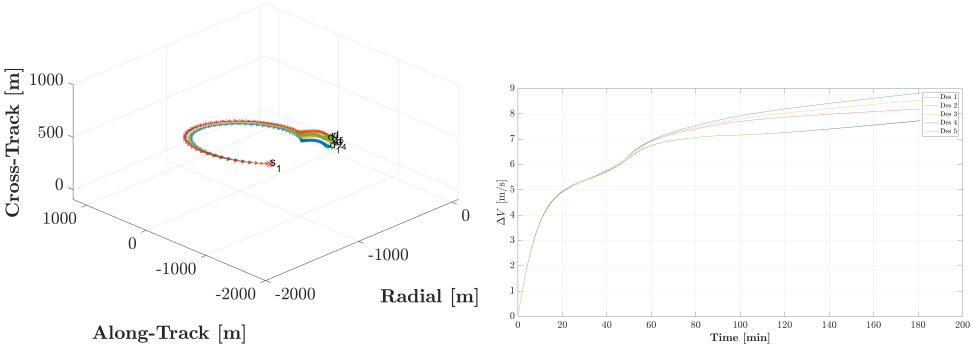
5.3. SIMULATIONS

The simulations are carried out in SIMULINK using the exported s-function from the ACADO toolkit. The simulation time limit is 360 min that is four times the Orbital period(T) of the chief orbit which is 90 minutes at 300 km. If the satellite does not reach its destination in the time frame, simulation is considered unsuccessful. Also, when the velocity for the satellite reaches 1 cm s^{-1}, the simulation is considered successful. At this timestamp, the ΔV for the satellites is logged to be used for the Assignment algorithm.

5.3.1. EXPECTED TRAJECTORIES AND FUEL CONSUMPTION

First part of the simulations is the satellites translating to the each destination. The satellites are guided by an MPC controller to the designations as the terminal constraint. The destinations are a single set of randomized $[x, y, z]$ coordinates for each satellite. Figures 5.4-5.8 show the trajectory in 3-D space the satellite takes to each destination. Figure

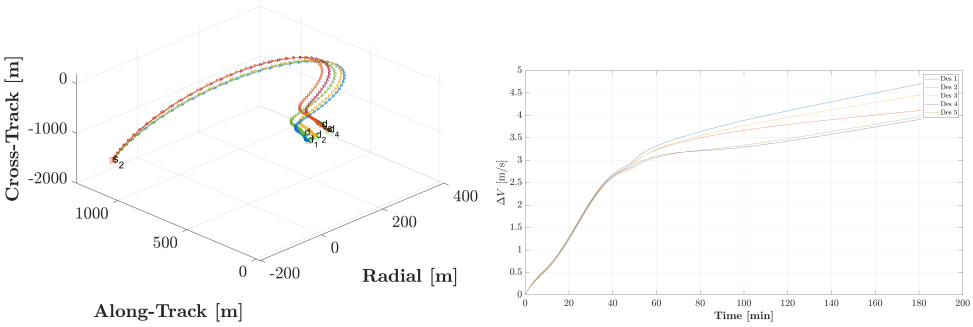
5.4a shows the satellite s_1 from its initial position $[-1747, -1675, 848]$ to the destinations d_1 to d_5. Figure 5.4b shows the fuel consumption in terms of ΔV for each destination again.



(a) Trajectories

(b) Fuel consumption

Figure 5.4: Satellite 1 trajectories and the ΔV for each destination



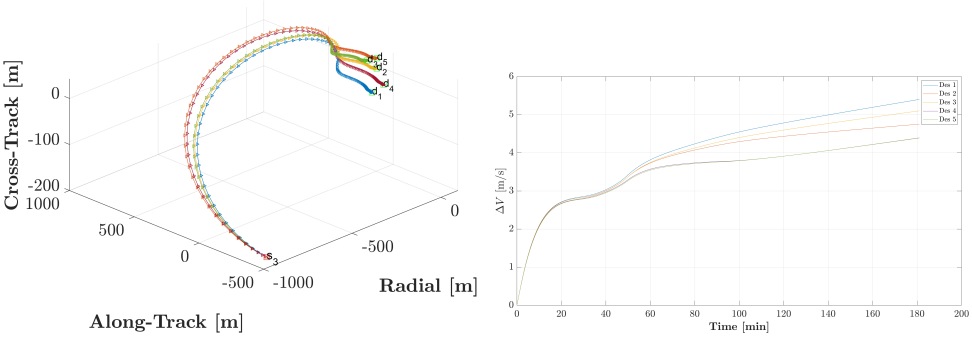
(a) Trajectories

(b) Fuel consumption

Figure 5.5: Satellite 2 trajectories and the ΔV for each destination

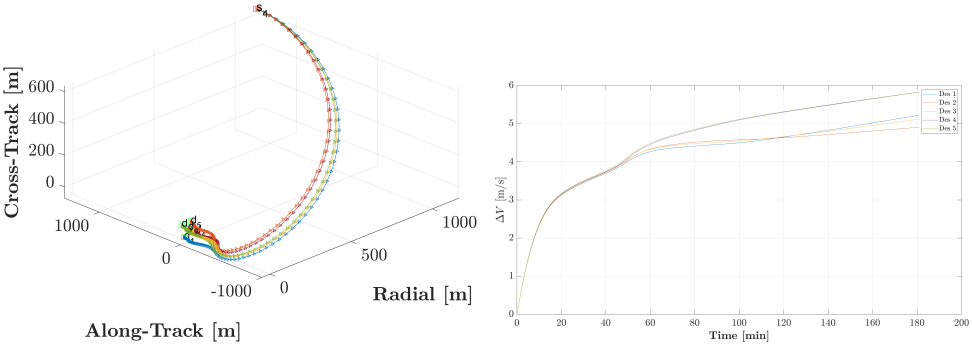
At the start of the maneuver, the fuel expenditure is similar and the diff is observed when the satellite closes in on one of the destination positions. The reason for the similar fuel expenditure is that the destinations are in range of ±50 m which is a small space compared to the initial positions of the satellites. Therefore, the starting leg of the journey 30–40 min is similar for all the available destinations and the trajectories diverge when the satellites get close to the destination space. Hence, you observer the splitting of plots in figure 5.4b and the difference in ΔV is much clearer.

Notice here that maximum fuel consumption takes place at the early stage of the translation i.e. the first 20% of the journey. The rest of the time is spent making sure the destination is reached. The time taken by each satellite to finish the maneuver is also similar. The farther the satellite seems to be, the higher acceleration is allowed by the MPC to complete the maneuver in similar Tof as shown by Walh and Howell [23]. Hence, the



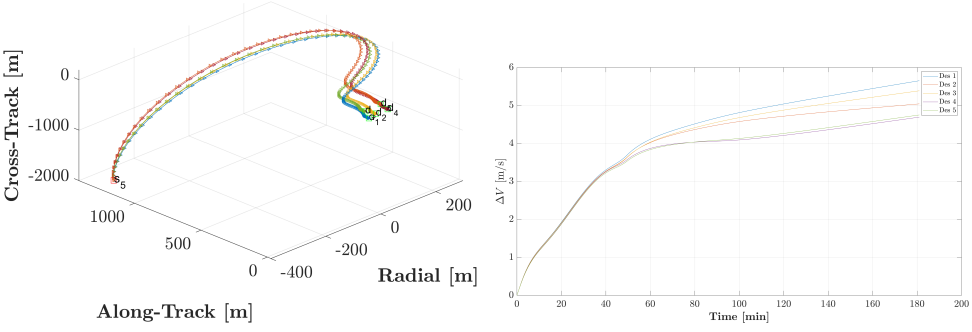
(a) Trajectories (b) Fuel consumption

Figure 5.6: Satellite 3 trajectories and the ΔV for each destination



(a) Trajectories (b) Fuel consumption

Figure 5.7: Satellite 4 trajectories and the ΔV for each destination



(a) Trajectories (b) Fuel consumption

Figure 5.8: Satellite 5 trajectories and the ΔV for each destination

ΔV values are different allowing us a great parameter to differentiate the destinations with.

A similar Tof is between the 90–110 min where the rate of change of ΔV seems to be constant i.e. accelerations have become constant. But the velocities at this point are around 1 cm for each component essentially implying the satellite has come to a stop as shown in Figure 5.9. The acceleration does not go to zero even when the satellite is relatively stationary because the natural motion of a satellite is to orbit in equilibrium due to gravity. Since here the satellite is kept stationary, the constant slope of ΔV refers to the station-keeping fuel expenditure.

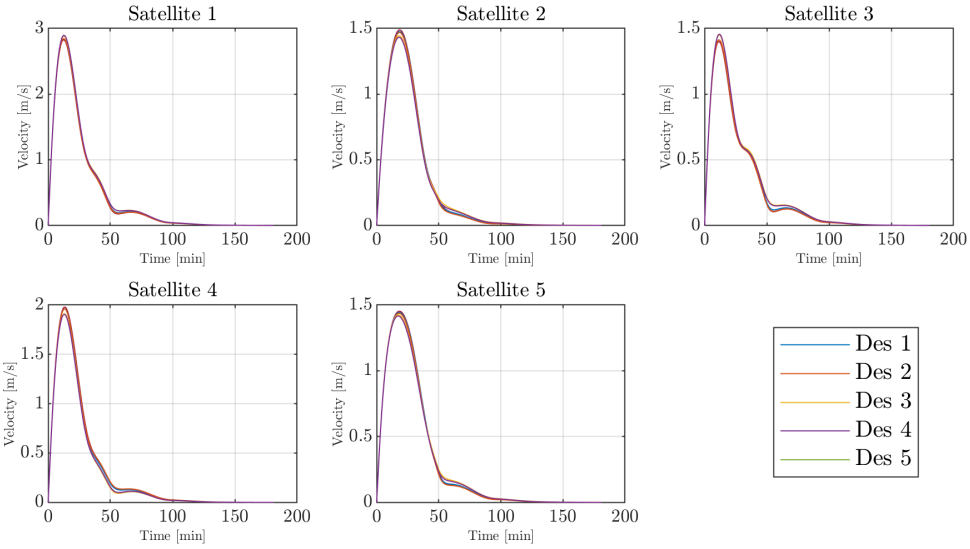


Figure 5.9: Resultant velocity of each satellite corresponding to each destination

Figure 5.10 shows the expected fuel expenditure for each satellite to destinations $d_1 - d_5$. It can be observed from both Figure 5.10 and 5.9 around 100 min mark the satellites are converging to their destination. Thus, at timestamp 100 min the ΔV values are considered for the assignment algorithm. Using this data F_{exp} matrix is constructed which looks as following:

$$F_{exp} = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \begin{array}{c} \begin{array}{ccccc} d_1 & d_2 & d_3 & d_4 & d_5 \end{array} \\ \begin{bmatrix} 7.9099 & 7.6894 & 7.7935 & 7.1632 & 7.1651 \\ 3.8761 & 3.6601 & 3.7590 & 3.2852 & 3.3267 \\ 4.5338 & 4.2844 & 4.3879 & 3.7920 & 3.7885 \\ 4.4929 & 4.5737 & 4.5441 & 5.0874 & 5.1056 \\ 4.7999 & 4.5699 & 4.6697 & 4.0896 & 4.1273 \end{bmatrix} \end{array}$$

5.3.2. THE ASSIGNMENT

For the assignment phase, we go back to equation 3.1 in Chapter 3. With the P_{mat} equation, we will consider two cases. The first case where only the expected fuel consumption

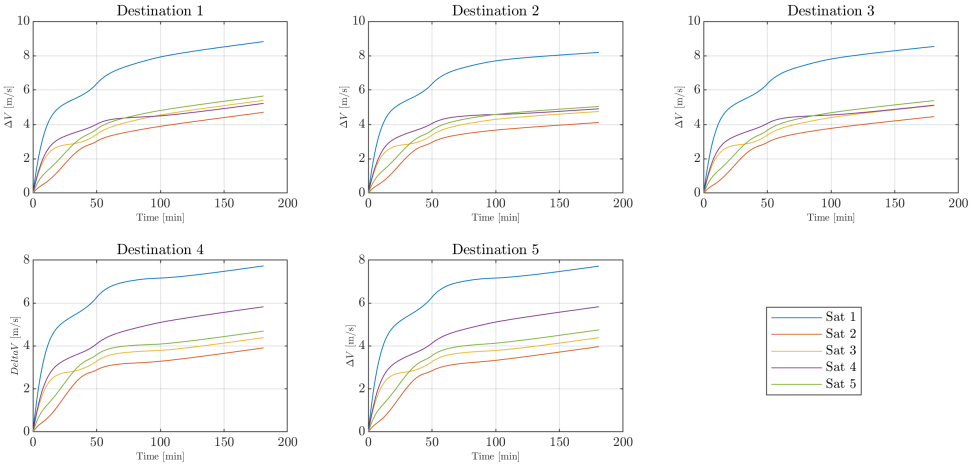


Figure 5.10: Fuel Expenditure of each satellite corresponding to each destination

5

matrix F_{exp} is used for the assignment. In the second case, the expected fuel matrix F_{exp} and the reserve fuel level F_{res} are both used for the assignment.

$$P_{mat} = W_d D_{mat} + W_{fe} F_{exp} + W_{fc} \frac{1}{F_{cr}}$$

Either of the cases does not consider the distance matrix D_{mat}; the weight W_d is zero for D_{mat}. The expected fuel consumption indirectly accounts for the distance because the fuel consumption is measured using Δv ($\mathrm{m\,s^{-1}}$) where the distance to the destinations is part of the calculation.

CASE I

Now, we will assign the destinations based on the expected fuel consumption values. So, W_{fe} will be one making sure we only consider the expected fuel consumption as the parameter. The W_{fe} is only non-zero weight, in this case; thus W_{fe} is 1. Therefore, the parameter matrix P_{mat} is same as the expected fuel consumption matrix in this case:

$$P_{mat} = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \begin{array}{c} \begin{array}{ccccc} d_1 & d_2 & d_3 & d_4 & d_5 \end{array} \\ \begin{bmatrix} 7.9099 & 7.6894 & 7.7935 & 7.1632 & 7.1651 \\ 3.8761 & 3.6601 & 3.7590 & 3.2852 & 3.3267 \\ 4.5338 & 4.2844 & 4.3879 & 3.7920 & 3.7885 \\ 4.4929 & 4.5737 & 4.5441 & 5.0874 & 5.1056 \\ 4.7999 & 4.5699 & 4.6697 & 4.0896 & 4.1273 \end{bmatrix} \end{array}$$

Following the same assignment procedure as the Algorithm 24 in Chapter 3, next step is to sort the P_{mat} in descending order to find out what satellite has highest priority for each particular destination. After sorting, the Parameter matrix look as following:

$$P_{mat_{sort}} = \begin{matrix} \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} d_1 & d_2 & d_3 & d_4 & d_5 \end{matrix} \begin{bmatrix} 7.9099 & 7.6894 & 7.7935 & 7.1632 & 7.1651 \\ 4.7999 & 4.5737 & 4.6697 & 5.0874 & 5.1056 \\ 4.5338 & 4.5699 & 4.5441 & 4.0896 & 4.1273 \\ 4.4929 & 4.2844 & 4.3879 & 3.7920 & 3.7885 \\ 3.8761 & 3.6601 & 3.7590 & 3.2852 & 3.3267 \end{bmatrix}$$

In addition to this sorted P_{mat}, the MATLAB sort function also gives us the assignment order of the satellites that are the index values of P_{mat} repented by matrix A_o:

$$A_o = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 5 & 4 & 5 & 4 & 4 \\ 3 & 5 & 4 & 5 & 5 \\ 4 & 3 & 3 & 3 & 3 \\ 2 & 2 & 2 & 2 & 2 \end{bmatrix} \tag{5.1}$$

Another sorting row-wise operation on P_{mat} gives us the destination priority order for each satellite D_o as following:

$$D_o = \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{matrix} \begin{bmatrix} 4 & 5 & 2 & 3 & 1 \\ 4 & 5 & 2 & 3 & 1 \\ 5 & 4 & 2 & 3 & 1 \\ 1 & 3 & 2 & 4 & 5 \\ 4 & 5 & 2 & 3 & 1 \end{bmatrix} \tag{5.2}$$

Now, Assignment Algorithm is executed using the information obtained above from the MPC simulations. The Assignment Algorithm gives us the final Assignment Order A_o based on the Parameter matrix P_{mat}. The destinations for each satellite are set according to the Assignment order A_{fin} vector.

$$A_{fin} = \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{matrix} \begin{bmatrix} 4 \\ 3 \\ 2 \\ 1 \\ 5 \end{bmatrix} \tag{5.3}$$

CASE II

For this case, we consider both F_{exp} and F_{res} for the assignment. The weights W_{fc} and W_{fr} are set at 1 and 1×10^3 respectively. The large W_{fr} signifies the importance of reserve fuel in this case. The fuel values of the satellites are as following:

$$\begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{matrix} \begin{bmatrix} 41 \\ 41 \\ 36 \\ 45 \\ 38 \end{bmatrix}$$

The fuel levels are in terms of the Δv. The fuel value indicated the amount of Δv the satellite can expend. The base value of reserve fuel for each satellite is 50. In this case, the reserve fuel level is lower than 50 to account for the assumed previous reconfiguration or any other maneuvers the satellites might have executed. Based on the fuel levels, the F_{res} can be defined as follows:

$$F_{res} = \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{matrix} \begin{bmatrix} 41 & 41 & 41 & 41 & 41 \\ 41 & 41 & 41 & 41 & 41 \\ 36 & 36 & 36 & 36 & 36 \\ 45 & 45 & 45 & 45 & 45 \\ 38 & 38 & 38 & 38 & 38 \end{bmatrix}$$

The F_{res} has been defined, with F_{exp} from the previous case, the new P_{mat} is calculated. Note that the higher value means a higher priority.

$$P_{mat} = \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{matrix} \begin{bmatrix} 32.3001 & 32.0796 & 32.1837 & 31.5534 & 31.5553 \\ 28.2663 & 28.0503 & 28.1492 & 27.6754 & 27.7169 \\ 32.3116 & 32.0622 & 32.1657 & 31.5698 & 31.5662 \\ 26.7151 & 26.7959 & 26.7663 & 27.3096 & 27.3278 \\ 31.1156 & 30.8857 & 30.9855 & 30.4054 & 30.4431 \end{bmatrix}$$

The sorting operations are carried out again on the new P_{mat}. The new A_o and D_o are shown below.

$$D_o = \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{matrix} \begin{bmatrix} 4 & 5 & 2 & 3 & 1 \\ 4 & 5 & 2 & 3 & 1 \\ 5 & 4 & 2 & 3 & 1 \\ 1 & 3 & 2 & 4 & 5 \\ 4 & 5 & 2 & 3 & 1 \end{bmatrix} \tag{5.4}$$

$$A_o = \begin{bmatrix} 3 & 1 & 1 & 3 & 3 \\ 1 & 3 & 3 & 1 & 1 \\ 5 & 5 & 5 & 5 & 5 \\ 2 & 2 & 2 & 2 & 2 \\ 4 & 4 & 4 & 4 & 4 \end{bmatrix} \tag{5.5}$$

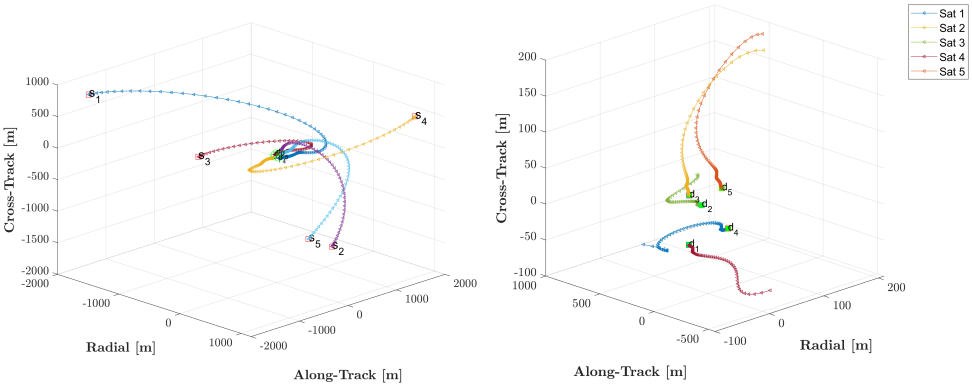
These new A_o and D_o are further fed into the assignment algorithm which gives us the final assignments A_{fin} as following:

$$A_{fin} = \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{matrix} \begin{bmatrix} 4 \\ 3 \\ 5 \\ 1 \\ 2 \end{bmatrix} \tag{5.6}$$

The results show the D_o matrices are the same for both cases. The result is as expected because the destination order for each steatite is decided based on the expected fuel consumption, which is the same for both cases. However, the assignment priority order has changed in Case II. The A_o representative matrices in (5.1) and (5.5) show a shift in the assignment order with inclusion of the reserve fuel levels F_{res}. The satellites s_2 and s_3 have moved up in the priority order.

In Case I, satellite s_1 was the first one assigned a value, and Case II gives priority to the satellite s_3 corresponding to the change in P_{mat} priority values due to added reserve fuel level consideration. The change in priority order matrix A_o shows a change in the final assignment order A_{fin}. Both cases show different assignments for the satellites s_3 and s_5. Similarly, more parameters can be added to the priority calculations for a wider consideration at the assignment stage of the problem.

With all the set information about initial positions and the calculated destinations, the satellites are ready to be deployed into the Guidance phase using any control scheme chosen by design. One important note is that we started these simulations with just the target destinations as the reference. However, the trajectories generated by the MPC in these simulations can be used as the reference trajectories by the guidance controller in the absence of a good trajectory generator as shown in 5.11.



(a) Satellite trajectories to the assigned positions

(b) Zoomed in view showing satellites converging to the assigned destination

Figure 5.11: Simulation showing successful demonstration of trajectory generation by MPC controller and convergence to destinations assigned using the autonomous assignment algorithm

5

5.4. SUMMARY

This chapter details the simulations and tests the proposed assignment algorithm. At the start, the randomized approach to the initial positions and target destinations is explained. It is chosen in this work to generalize the assignment approach with randomness instead of treating each case separately. After that, the simulation parameters are defined from the satellite dimensions, weight, fuel capacity to the operational orbit. The MPC parameters and constraints are chosen with an explanation for that choice. Simulation is executed with constant terminal weights and dynamic terminal weights. The results from both simulations are compared and it is observed that the dynamic constraints offer much better results than the constant terminal weight scenario. Furthermore, the simulations for each satellite are performed with results showing satellites are expected to reach the target between 90 to 110 min mark. In that time frame, the value is extracted to be used in the assignment algorithm. Using these ΔV values, the assignment algorithm is successfully executed. In the next section, some future research options are discussed.

6

Conclusion and Future Work

6.1. CONCLUSION

An autonomous assignment algorithm is proposed and presented in this thesis. The assignment algorithm autonomously assigns the satellites an optimal destination based on the priority parameter. Various assignment parameters such as the distance of initial position from the available destinations, the current fuel capacity, and the expected fuel expenditure demonstrate the versatility of the algorithm. The simulations to destinations for each satellite using an MPC algorithm provide the expected fuel expenditure data.

The assignment algorithm autonomously assigning the optimal destinations to the satellites is demonstrated using the fuel parameters from the MPC algorithm. Furthermore, it is essential to note that the simulations were without a reference trajectory. These pre-guidance phase MPC simulations carried out to compute the expected fuel consumption also generate a reference trajectory. When a dedicated trajectory generator is lacking, the reference trajectory generated by the MPC can be convenient in the guidance phase of the delivery problem.

6.2. FUTURE WORK

The work represented in this thesis is considered preliminary and can serve as a stepping stone for more comprehensive development. The next step should include a blend of decentralized parallel assignment algorithm instances with accurate autonomous navigation and delivery methods. Some potential improvements that would provide a course towards an autonomous assignment and delivery system as proposed as following:

- This thesis focused on computing expected fuel consumption to and from an arbitrary position. Thus, the delivery of satellites to a planned formation was not considered at the time of assignment. A further step would be testing the algorithm in a well-defined arrangement with computed final position orbits and check the performance.
- In addition to the previous point, the thesis deals with the assignment problem. A guidance system built in tandem with this assignment algorithm is the most natural step in future research.
- In this work, Linearized Clohessy-Wiltshire relative equations of motion is the dynamics model. The Yamanaka-Ankersen (YA) dynamics approximation will be an upgrade to CW equations. The YA approximations are non-linear in nature and applicable to elliptical orbits. The YA implementation will widen the scope of the application. Furthermore, the non-spherical elements of the earth like $j2$ perturbation can also be included to more accurately define the relative motion.
- The satellite swarm is assumed to be in communication at all times. However, the distance between the deputy satellites limits the connection in a real case. The chief satellite acts as the centralized entity for the communication and the assignment information decimation. For true decentralization, the deputy satellites must be able to operate on their own. Therefore, an assignment and a guidance algorithm can be developed with communication limitations as a design constraint.

- Safety is a decisive factor to consider for the autonomous delivery system. The use of debris/spacecraft tracking data would further improve the assignment algorithm by accounting for possible collisions or collision avoidance maneuvers at the time of the assignment.

It is evident from the complexity of the autonomous guidance problem of a swarm of spacecraft, and this thesis work was a small part of a big puzzle. Although the thesis demonstrates the feasibility of a specific parameter set-based assignment algorithm, the algorithm can achieve onboard functionality by further studying the constraints and uncertainties in the delivery algorithm.

REFERENCES

[1] B. D. Tapley et al. “The gravity recovery and climate experiment: Mission overview and early results”. In: *Geophysical Research Letters* 31.9 (2004), n/a–n/a. DOI: 10.1029/2004gl019920.

[2] J. L. Burch et al. “Magnetospheric Multiscale Overview and Science Objectives”. In: *Space Science Reviews* 199.1-4 (2015), pp. 5–21. DOI: 10.1007/s11214-015-0164-9.

[3] Eberhard Gill, Simone D’Amico, and Oliver Montenbruck. “Autonomous Formation Flying for the PRISMA Mission”. In: *Journal of Spacecraft and Rockets* 44.3 (2007), pp. 671–681. DOI: 10.2514/1.23015.

[4] Gerhard Krieger et al. “TanDEM-X: A Satellite Formation for High-Resolution SAR Interferometry”. In: *IEEE Transactions on Geoscience and Remote Sensing* 45.11 (2007), pp. 3317–3341. DOI: 10.1109/tgrs.2007.900693.

[5] Chris Sabol, Rich Burns, and Craig A. McLaughlin. “Satellite Formation Flying Design and Evaluation”. In: *Journal of Spacecraft and Rockets* (), pp. 99–212.

[6] Kazunori Akiyama et al. “First M87 Event Horizon Telescope Results. II. Array and Instrumentation”. In: *The Astrophysical Journal Letters* 875 (Apr. 2019).

[7] Marinus Jan Bentum et al. “OLFAR: the orbiting low frequency array, how a cube sat swarm becomes a novel radio astronomy instrument in space”. Undefined. In: *Vonk* 25 (2010), pp. 1–5. ISSN: 0925-5427.

[8] Dario Izzo and Lorenzo Pettazzi. “Autonomous and Distributed Motion Planning for Satellite Swarm”. In: *Journal of Guidance, Control and Dynamics* 30 (2007).

[9] Adam W. Koenig and Simone D’Amico. “Safe spacecraft swarm deployment and acquisition in perturbed near-circular orbits subject to operational constraints”. In: *Acta Astronautica* 153 (2018), pp. 297–310. DOI: 10.1016/j.actaastro.2018.01.037.

[10] W. H. CLOHESSY and R. S. WILTSHIRE. “Terminal Guidance System for Satellite Rendezvous”. In: *Journal of the Aerospace Sciences* 27.9 (1960), pp. 653–658. DOI: 10.2514/8.8704. eprint: https://doi.org/10.2514/8.8704. URL: https://doi.org/10.2514/8.8704.

[11] Alfred North Whitehead and Bertrand Russell. *Principia Mathematica to *56*. Cambridge University Press, 2010.

[12] 2021. URL: https://ai-solutions.com/_freeflyeruniversityguide/j2_perturbation.htm.

[13] 2021. URL: https://en.wikipedia.org/wiki/Gravity_anomaly.

[14] How Stuff Works. *Solar Wind Perturbation*. https://science.howstuffworks.com/dictionary/astronomy-terms/perturbation-info.htm, accessed 2019-10-17.

[15] David A. Vallado. *Fundamentals of Astrodynamics and Applications*. New York: McGraw-Hill, 1997.
[16] Howard D Curtis. *Orbital mechanics for engineering students*.
[17] 2021. URL: http://www.kwon3d.com/theory/rotsys/cent.html.
[18] Hari B. Hablani, Myyorn L. Tapper, and David J. Dana-Bashian. "Guidance and Relative Navigation for Autonomous Rendezvous in a Circular Orbit". In: *Journal of Guidance, Control and Dynamics* 25 (), pp. 99–212.
[19] Rajarshi Ghosh Dastidar. "On the Advantages and Limitations of Sliding Mode Control for Spacecraft". In: *AIAA SPACE 2010 Conference & Exposition* (2010). DOI: 10.2514/6.2010-8777.
[20] Elisa Capello et al. "Sliding-Mode Control Strategies for Rendezvous and Docking Maneuvers". In: *Journal of Guidance, Control, and Dynamics* 40.6 (2017), pp. 1481–1487. DOI: 10.2514/1.g001882.
[21] Hancheol Cho and Gaetan Kerschen. "Satellite Formation Control Using Continuous Adaptive Sliding Mode Controller". In: *AIAA/AAS Astrodynamics Specialist Conference* (2016). DOI: 10.2514/6.2016-5662.
[22] Ranjith Ravindranathan Nair and Laxmidhar Behera. "Robust adaptive gain higher order sliding mode observer based control-constrained nonlinear model predictive control for spacecraft formation flying". In: *IEEE/CAA Journal of Automatica Sinica* 5.1 (2018), pp. 367–381. DOI: 10.1109/jas.2016.7510253.
[23] Theodore Wahl. "Autonomous Guidance Strategy for Spacecraft Formations and Reconfiguration Maneuvers". PhD thesis. Purdue University, West Lafayette, Indiana, 2017.
[24] Avishai Weiss et al. "Model Predictive Control for Spacecraft Rendezvous and Docking: Strategies for Handling Constraints and Case Studies". In: *IEEE Transactions on Control Systems Technology* 23.4 (2015), pp. 1638–1647. DOI: 10.1109/tcst.2014.2379639.
[25] Hyeongjun Park, Stefano Di Cairano, and Ilya Kolmanovsky. "Model Predictive Control for Spacecraft Rendezvous and Docking with a Rotating/Tumbling Platform and for Debris Avoidance". In: *American Control Conference*. 2011.
[26] Utku Eren et al. "Model Predictive Control in Aerospace Systems: Current State and Opportunities". In: *Journal of Guidance, Control, and Dynamics* 40.7 (2017), pp. 1541–1566. DOI: 10.2514/1.g002507.
[27] Qinglei Hu, Jingjie Xie, and Chenliang Wang. "Dynamic path planning and trajectory tracking using MPC for satellite with collision avoidance". In: *ISA Transactions* 84 (2019), pp. 128–141. DOI: 10.1016/j.isatra.2018.09.020.
[28] Mirko Leomanni, Eric Rogers, and Stephen B. Gabriel. "Explicit Model Predictive Control Approach for Low-Thrust Spacecraft Proximity Operations". In: *Journal of Guidance, Control and Dynamics* 37.6 (2014).
[29] Julian Scharnagl, Panayiotis Kremmydas, and Klaus Schilling. "Model Predictive Control for Continuous Low Thrust Satellite Formation Flying". In: *IFAC-PapersOnLine* 51.12 (2018), pp. 12–17. DOI: 10.1016/j.ifacol.2018.07.081.
[30] Theodore Wahl and Kathleen C. Howell. "Autonomous Guidance Algorithm for Multiple Spacecraft and Formation Reconfiguration Maneuvers". In: *Advances in the Astronautical Sciences Spaceflight Mechanics* 158 (2016).

[31] Jouse David Munoz. “Rapid Path-Planning Algorithms For Autonomous Proximity Operations Of Satellites”. PhD thesis. University Of Florida, 2011.
[32] Theodore Wahl and Kathleen Howell. “Autonomous Guidance Algorithms for Formation Reconfiguration Maneuvers”. In: *Advances in the Astronautical Sciences AAS/AIAA Astrodynamics Conference 2017* (2018).
[33] James Blake Rawlings, Moritz M Diehl, and David Q Mayne. *Model predictive control*. Nob Hill, 2017.
[34] Liuping Wang. *Model predictive control system design and implementation using MATLAB*. Springer, 2009.
[35] 2021. URL: https://en.wikipedia.org/wiki/Model_predictive_control.
[36] Dale E. Seborg et al. *Process Dynamics and Control*. Wiley, 2017.
[37] Sebastien Gros et al. “From linear to nonlinear MPC: bridging the gap via the real-time iteration”. In: *International Journal of Control* 93.1 (2016), pp. 62–80. DOI: 10.1080/00207179.2016.1222553.
[38] 2021. URL: https://www.nasa.gov/smallsat-institute/sst-soa-2020/in-space-propulsion.
[39] 2021. URL: https://www.esa.int/Education/Thrusters.
[40] Hans Joachim Ferreau et al. “qpOASES: a parametric active-set algorithm for quadratic programming”. In: *Mathematical Programming Computation* 6.4 (2014), pp. 327–363. DOI: 10.1007/s12532-014-0071-1.
[41] Janick V. Frasch, Sebastian Sager, and Moritz Diehl. “A parallel quadratic programming method for dynamic optimization problems”. In: *Mathematical Programming Computation* 7.3 (2015), pp. 289–329. DOI: 10.1007/s12532-015-0081-7.
[42] Alexander Domahidi et al. “Efficient interior point methods for multistage problems arising in receding horizon control”. In: *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)* (2012). DOI: 10.1109/cdc.2012.6426855.
[43] Alberto Bemporad et al. “The explicit linear quadratic regulator for constrained systems”. In: *Automatica* 38.1 (2002), pp. 3–20. DOI: 10.1016/s0005-1098(01)00174-1.
[44] Matthew Brand et al. “A Parallel Quadratic Programming Algorithm for Model Predictive Control”. In: *IFAC Proceedings Volumes* 44.1 (2011), pp. 1031–1039. DOI: 10.3182/20110828-6-it-1002.03222.
[45] Boris Houska, Hans Joachim Ferreau, and Moritz Diehl. “ACADO toolkit-An open-source framework for automatic control and dynamic optimization”. In: *Optimal Control Applications and Methods* 32.3 (2010), pp. 298–312. DOI: 10.1002/oca.939.
[46] Daniel Morgan, Soon-Jo Chung, and Fred Y. Hadaegh. “Model Predictive Control of Swarms of Spacecraft Using Sequential Convex Programming”. In: *Journal of Guidance, Control, and Dynamics* 37.6 (2014), pp. 1725–1740. DOI: 10.2514/1.g000218.
[47] M. Kiehl. “Parallel multiple shooting for the solution of initial value problems”. In: *Parallel Computing* 20.3 (1994), pp. 275–295. DOI: 10.1016/s0167-8191(06)80013-x.
[48] Yash Raj Khusro. “MPC-based Motion Cueing Algorithm for a 6 DOF Driving Simulator with Actuator Constraints”. PhD thesis. Delft University of Technology, 2020.

[49] R.A. Bartlett, A. Wachter, and L.T. Biegler. “Active set vs. interior point strategies for model predictive control”. In: *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334)* (2000). DOI: 10.1109/acc.2000.877018.

[50] Wikipedia. *Starlink (satellite constellation)*. https://en.wikipedia.org/wiki/Starlink_(satellite_constellation), accessed 2019-11-26.

[51] 2021. URL: https://solc.gsfc.nasa.gov/modules/launch/mainMenu_textOnly.php.