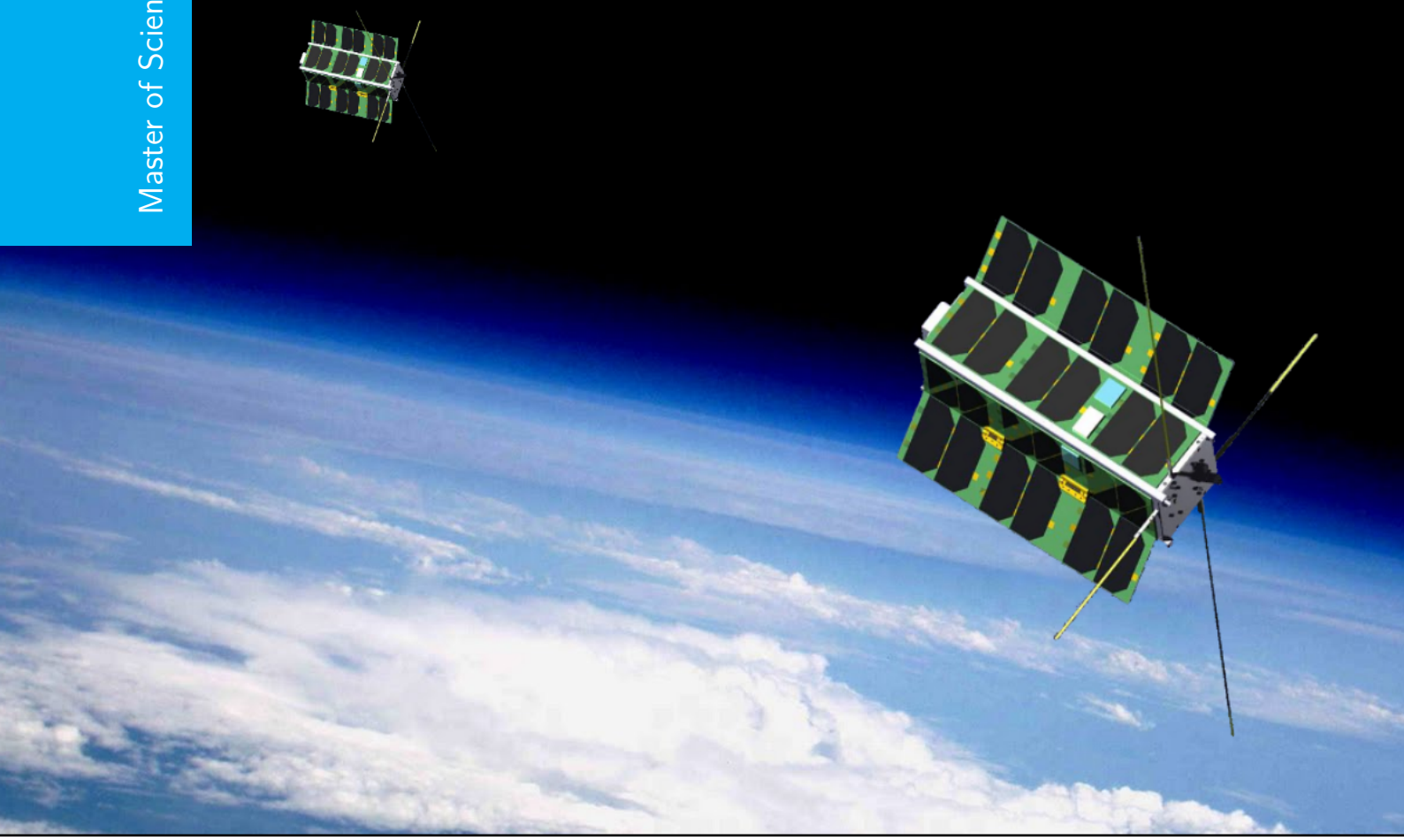
# Maintenance of a Long Baseline Along-Track Formation for the DelFFi Mission

Amy Deeb

Master of Science Thesis



TU Delft Delft University of Technology

Space Systems Engineering

# Maintenance of a Long Baseline Along-Track Formation for the DelFFi Mission

by

Amy Deeb

To obtain the degree of Master of Science in Space Systems Engineering
at Delft University of Technology,
to be defended publicly on Wednesday September 30, 2015.

| | | |
|---|---|---|
| Student number: | 4282329 | |
| Project duration: | March 1, 2015 – September 30, 2015 | |
| Thesis committee: | Prof. Dr. E.K.A. Gill, | TU Delft, supervisor |
| | Dr. J. Guo, | TU Delft, supervisor |
| | Ir. R. Noomen, | TU Delft |

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Faculty of Aerospace Engineering · Delft University of Technology

Cover image modified from Artist's Impression of DelFFi satellites "Curvature Closed" from http://www.delfispace.nl/delffi


# Abstract

Consisting of two three-unit CubeSats, the DelFFi project will study atomic oxygen in the lower thermosphere as part of QB50. To enhance the scientific return, autonomous formation flying will be used to maintain a constant 1000 km along track distance between the two satellites. The use of CubeSats implies strict mass, volume and power budgets, as well as the limited capabilities of the propulsion system, attitude determination and control system and onboard processors. Due to these constraints, differential navigation based on two-line elements (TLEs) has been suggested. To determine if TLEs are suited to the DelFFi mission and formation flying in general, three topics are addressed in this thesis: navigation based on TLEs, formation maintenance control algorithms and CubeSat operations.

Although TLEs are provided at a frequency of 0.6 days, they are generated based on data averaged over the previous five days. Thus, assuming one day latency for transmitting the TLEs to orbit, the formation must passively maintain its control window for a minimum of 6 days. Analyses showed that minimizing differential drag (mainly by reducing the attitude pointing error) and matching mean semi major axis at initialization are critical to maintaining the formation. A set of viable orbits, acceptable attitude pointing error limits and control accuracies for matching the mean orbital elements are established.

Based on this set of viable orbits, MATLAB Simulink models have been developed to study formation maintenance algorithms. Using an established analytical formation maintenance controller for demonstration, the control accuracy was shown to decrease with increasing correction size (control window dimension). Further, the magnitude of the manoeuvres required every six days was shown to be too large to be performed by the current design of the propulsion system and requires better pointing accuracy than the current iteration of the ADCS can provide.

The DelFFi satellites operate in a Chief-Deputy configuration, in which the Deputy actively maintains the nominal separation distance relative to the passive Chief. Fuel balancing by exchanging roles is recommended. An intersatellite link to frequently exchange state data (orbital elements) is also recommended to improve the formation performance.

This research shows that TLE-based formation maintenance is viable in specific cases. However, improvements to the attitude control, propulsion system, and intersatellite communication are critical for successful implementation of TLE-based formation maintenance for DelFFi and on CubeSats in general.

# Acknowledgements

It would not have been possible to complete this thesis without the advice and assistance of the staff and students of the space engineering department of the TU Delft, as well as the love and support of my friends and family. In particular, I would like to thank my supervisor Dr. Jian Guo for helping me to develop this work and for challenging me to improve myself both technically and professionally. Further, I would like to thank the whole DelFFi team, both staff and students, for engaging my curiosity and raising interesting design problems. A special thanks to Prem Sundaramoorthy and Nuno dos Santos for challenging me to broaden my knowledge and to think beyond the standard solutions. Thank you also to Jing Chu, Johan Carvajal Godinez and Adolfo Chaves Jimenez who provided insight and technical background regarding the overlap between their PhD work and my thesis work with unending patience. I'd also like to extend a very warm thanks to the space girls for being around during the good times and bad.

To my parents and sisters – thank you for understanding and supporting me despite my decision to be an ocean away. Thank you, as well, to Shriya Hari, who has shown me true friendship and reminded me of how amazing the world can be. Finally, thank you to Andrew MacMillan for reading countless drafts, pulling numerous all-nighters and always having an optimistic outlook to share.

Thank you all. I could not have completed this thesis without you.

Delft, University of Technology — Amy Deeb
September 16, 2015.

# Table of Contents

# List of Figures

# List of Tables

# Nomenclature

## List of Acronyms

| | |
|---|---|
| **ADCS** | Attitude Determination and Control System |
| **AFF** | Autonomous Formation Flying |
| **APE** | Attitude Pointing Error |
| **CanX** | Canadian Advanced Nanosatellite eXperiment |
| **CCS** | Code Composer Studio |
| **dGPS** | Differential GPS |
| **ECI** | Earth Centred Inertial |
| **FFP** | Formation Flying Package |
| **FIPEX** | Flux-(Phi)-Probe-Experiment |
| **GNC** | Guidance Navigation and Control |
| **GNSS** | Global Navigation Satellite System |
| **GPS** | Global Positioning System |
| **GVE** | Gauss' Variational Equations |
| **HCW** | Hill-Clohessy-Wiltshire |
| **$I^2C$** | Inter-Integrated Circuit |
| **IMU** | Inertial Measurement Unit |
| **ISL** | Intersatellite Link |
| **JGM-3** | Third Joint Gravity Model |
| **KF** | Kalman Filter |

| | |
|---|---|
| **LEOP** | Launch and Early Operation |
| **LQR** | Linear Quadratic Regulator |
| **LTAN** | Local Time of Ascending Node |
| **$\mu$PS** | Micropropulsion System |
| **MIMO** | Multiple-Input Multiple-Output |
| **MMS** | Magnetospheric Multi-Scale mission |
| **MPC** | Model Predictive Controller |
| **N/A** | Not Assessed, Not Applicable |
| **NORAD** | North American Aerospace Defense Command |
| **OBC** | Onboard Computer |
| **ONF** | Onboard Navigation Function |
| **PD** | Proportional-Derivative |
| **P-POD** | Poly-Picosatellite Orbital Deployer |
| **RAAN** | Right Ascension of the Ascending Node |
| **RF** | Radio Frequency |
| **s/c** | Spacecraft |
| **SDRE** | State-Dependent Riccati Equation |
| **SGP4** | Standard General Perturbations 4 |
| **SGP8** | Standard General Perturbations 8 |
| **TDRSS** | Tracking and Data Relay Satellite System |
| **TLE** | Two-Line Element |
| **TU Delft** | Delft University of Technology |
| **USSTRATCOM** | United States Strategic Command |
| **UTC** | Coordinated Universal Time |
| **WGS84** | World Geodetic System (1984) |

# List of Symbols

| | |
|---|---|
| $\Delta$v | Change in Velocity, Delta-v [m/s] |
| $\Delta v_{AT}$ | Along-Track Change in Velocity [m/s] |
| $\Delta t_{ho}$ | Time Spent in the Higher Orbit [s] |
| $\epsilon$ | Error [varies] |
| $\lambda, u$ | True Latitude [rad] |
| $\mu$ | Standard Gravitational Parameter (for Earth unless otherwise specified) [$m^3/(kg{\cdot}s^2)$] |
| $\Omega$ | Right Ascension of the Ascending Node [rad] |
| $\omega$ | Argument of Perigee [rad] |
| $\phi$ | Attitude Pointing Error (Total) [rad] |
| $\pi$ | Ratio of a Circle's Circumference to its Diameter [-] |
| $\psi$ | Attitude Pointing Error in Pitch [rad] |
| $\rho$ | Density [$kg/m^3$] |
| $\rho_0$ | TLE standard density [$kg/m^3$] |
| $\mathbf{A}$ | Relation Between Current and Future State [-] |
| $\mathbf{B}$ | Relation Between External Forces and Future State [-] |
| $\mathbf{V}_{rel}$ | Satellite Velocity Relative to the Atmosphere [m/s] |
| $\mathbf{x}, \mathbf{y}, \mathbf{z}$ | Coordinate Frame Axes (ECI if not stated) [m] |
| $\theta$, $f$ | True Anomaly [rad] |
| $^\circ$, [deg] | Degrees |
| $A$ | Frontal Area [$m^2$] |
| $a$ | Semi-Major Axis [m] |
| $AT_{dd}$ | Along-track Drift due to Differential Drag [m] |
| $B$ | Ballistic Coefficient [$m^2/kg$] |
| $B^*$ | BSTAR - Variation of Ballistic Coefficient Provided in TLEs [$m^{-1}$] |
| $C_d$ | Coefficient of Drag [-] |
| $d_x, d_y, d_z$ | Disturbance Accelerations in the x, y and z axes [$m/s^2$] |
| $d_{AT}$ | Along-Track Correction Dimension [m] |
| $d_{ref}$ | Reference/Nominal Separation Distance [km] |
| $d_{sep}$ | Separation Distance [km] |
| $E$ | Eccentric Anomaly [rad] |
| $e$ | Eccentricity [-] |
| $E_{orb}$ | Energy [J] |
| $F$ | Thrust [N] |
| $f_0$, $f_{Theo}$ | Nominal Oscillation Frequency [Hz] |
| $F_x, F_y, F_z$ | Forces in the x, y and z axes [N] |
| $h$ | Magnitude of the Angular Momentum Vector [$kg{\cdot}m^2/s$] |
| $i$ | Inclination [rad] |

| | |
|---|---|
| $J$ | Term of Potential Function Describing the Earth's Gravity Field [-] |
| $k$ | Parabolic Curvature Constant [Hz/$^\circ C^2$] |
| $L$ | Long Side Length of CubeSat [m] |
| $l$ | Short Side Length of CubeSat [m] |
| $M$ | Mean Anomaly [rad] |
| $m$ | Mass [kg] |
| $M_0$ | Mean Anomaly at Epoch [rad] |
| $m_{sat}$ | Total Satellite Mass [kg] |
| $n$ | Mean Motion [$s^{-1}$] |
| $N_{orbs}$ | Number of Orbits [-] |
| $p$ | Semilatus Rectum [m] |
| $q_1$ | Non-singular Orbital Element $q_1 = e\cos(\omega)$ [-] |
| $q_2$ | Non-singular Orbital Element $q_2 = e\sin(\omega)$ [-] |
| $r$ | Radial Position [m] |
| $t$ | Time [s] |
| $t_b$ | Manoeuvre Duration, Burn Time [s] |
| $u_e$ | External Accelerations [m/$s^2$] |
| $u_x, u_y, u_z$ | Control Accelerations in the x, y and z axes [m/s$^2$] |
| $v$ | Velocity [m/s] |
| $x$ | State Vector [varies] |
| [°C] | Degrees Celsius |
| [d] | Days |
| [hr] | Hours |
| [Hz] | Hertz |
| [kg] | Kilograms |
| [m] | Metres |
| [min] | Minutes |
| [N] | Newtons |
| [Pa] | Pascals |
| [ppm] | Parts Per Million |
| [rad] | Radians |
| [rev] | Revolutions |
| [s] | Seconds |
| $\mathbf{\Delta V}$ | Delta-v Vector (radial, tangential and out-of-plane) [m/s] |
| M | Number of Half Orbits Between the First and Second $\Delta$v's in Analytical Controller Solution (must be odd) [-] |
| N | Number of Half Orbits Between the First and Third $\Delta$v's in Analytical Controller Solution (must be even) [-] |

# Chapter 1

# Introduction

The newest addition to the Delfi program at the Delft University of Technology (TU Delft) is the DelFFi mission. DelFFi consists of a pair of three-unit CubeSats, which will be launched into a near circular low Earth orbit to study atomic oxygen distribution in the lower thermosphere as part of the QB50 project [1]. The two DelFFi satellites will carry an Autonomous Formation Flying (AFF) payload, which will allow correlated measurements by maintaining a constant relative position between the two satellites.

As a pair of CubeSats in low Earth orbit, the DelFFi satellites have a different set of constraints than most formation flying missions. In particular, the power, volume and mass budgets associated with CubeSat missions place restrictions on both the sensors and actuators available to the system. For this reason, a controller that can function without requiring a dedicated sensor (for example global positioning system (GPS) receivers, cameras, rangefinders etc.) would be desirable. This has led to a recommendation to use two-line elements (TLEs) as the primary navigation data source. The challenges and opportunities that this creates will be elaborated on in Section 1.2.

This introduction will provide a brief description of the mission context, followed by a review of the existing formation flying research, the definition of the research objective and questions, and then conclude with an outline of this thesis.

## 1.1 Mission Context

The DelFFi satellites will be launched as part of the QB50 project. The QB50 Consortium intends to study atomic oxygen distribution in the lower thermosphere (90-320 km) using a network of 50 CubeSats, each carrying identical sensors [2]. By building such a network the individual CubeSats can together develop a model for the atmosphere that no CubeSat could complete alone. It is expected that these atmospheric sensors will require high pointing

accuracy, pointing stability and position knowledge accuracy.

While the other satellites that make up the QB50 network will have a constantly changing relative state (in this case the state consists of the satellite's position and velocity), the two DelFFi satellites will keep a nominally constant relative position throughout the autonomous formation flying demonstration [1]. This constant relative position translates to a constant temporal resolution between the two sensors, which will enhance the scientific value of the data [2]. For the QB50 mission, a separation distance of 1000 km (with a control window of 100 km), which corresponds to a 130 s sampling time between the two satellites, has been chosen for DelFFi, based on the QB50 science objective [2].

## 1.2 Formation Flying Review

In general, the formation flying problem is a specific form of a guidance, navigation and control (GNC) problem. These three modules, which determine a desired trajectory (guidance), estimate the current absolute and relative positions (navigation) and then generate control inputs needed to maintain the relative orbit (control), form the basis of the research presented in this thesis. This section will discuss the current state of the art of formation flying algorithms (Section 1.2.1) in order to identify opportunities that this research can address (Section 1.2.2).

### 1.2.1 State of the Art

Formation flying has been studied from several viewpoints depending on the mission under study. In general, formation flying is considered for missions in which strict relative position requirements are placed on the system - either due to a science requirement or to avoid collision between closely spaced constellations [3]. Recent formation flying missions have included the Morning Constellation, GRACE, the A-train Constellation, TerraSar-X/TanDEM-X, PRISMA, CanX-4/5, Swarm and MMS. Of these missions, only CanX-4 and CanX-5 are nanosatellites, however, unlike DelFFi, they do not follow the CubeSat standard. Further details on each of these missions are provided in Appendix A.

Recent developments have been driven by increasing mission performance needs - for example for a multi-satellite telescope, synthetic aperture radar or gravitational modelling missions [4]. These developments can be grouped into two areas: metrology (state estimation/navigation) and guidance and control strategies.

Technologies that allow for better estimation of the formation's state (position and velocity of each satellite as well as their relative positions and velocities) have been developed both for relative (direct) measures (optical, radio and laser ranging sensors) and for absolute measures (GPS algorithms) [4]. These absolute measures are then used to estimate the relative state with differential navigation algorithms (an indirect measure). The broad categories of these

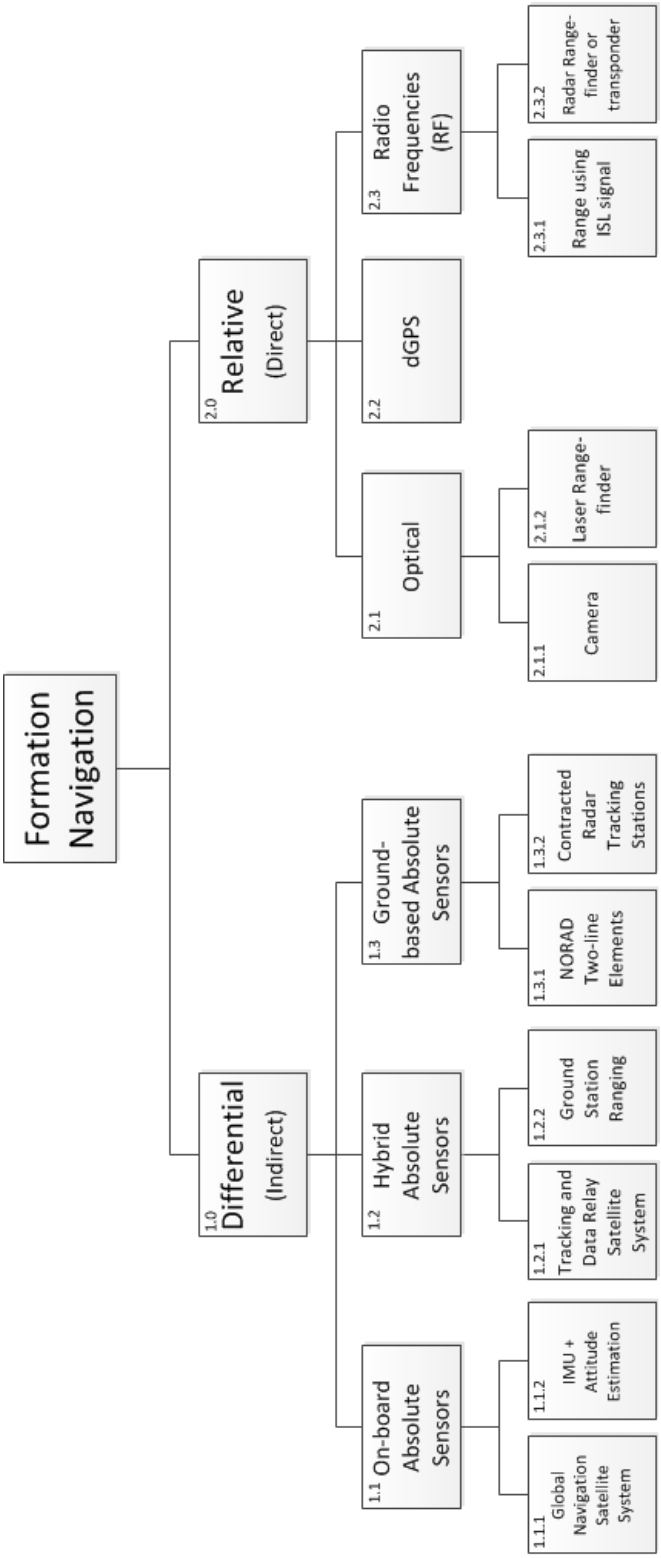



**Figure 1.1:** Formation Navigation Design Option Tree.

two types of formation flying navigation algorithms are shown in Figure 1.1.

Although relative navigation generally provides better performance, [3], all of the methods identified require dedicated sensors (Cameras, range-finders, or transponders) or computationally intensive algorithms (dGPS (Differential GPS), Intersatellite Link signal processing). Note that dGPS (option 2.2) is not the same as the differential GNSS algorithms identified in option 1.1.1. Instead, it uses dedicated antennae that are more similar to the RF navigation options (options 2.3.1 and 2.3.2) to produce a position signal relative to the chief satellite, rather than to the GNSS (GPS) network. On CubeSats, having dedicated sensors is undesirable due to the extra power and volume required. Thus, the dedicated algorithms are a preferable option for relative navigation on CubeSats, however significant development work will be necessary to use these methods for DelFFi.

Differential navigation options can be grouped into three main categories based on where the absolute position sensors are located: onboard, on the ground or both. Of these options, contracting radar stations to track two CubeSats or using the Tracking and Data Relay Satellite System (TDRSS) would be prohibitively expensive [3]. This leaves only TLEs as generated by NORAD as a viable ground-based solution. The onboard option of using an inertial measurement unit (IMU) combined with the attitude estimate to integrate the accelerations for state estimation is not possible for two reasons. First, DelFFi is not equipped with an IMU (though integration of one would not be a significant variation as the Attitude Determination and Control System (ADCS) microcontroller is designed to accept one) and second, the current angular pointing error modelled by the ADCS team is too large to allow accurate position estimate in this way [5]. The use of a global navigation satellite system (such as GPS) is a possible solution to the CubeSat navigation problem, however issues relating to the drop-out of GPS connections on CubeSats and the time required to re-establish the connection [6], as well as the limited access to GPS hardware and software due to governmental restrictions, poses some risk to a design that relies on GPS alone.

Improvements on GPS algorithms - and in particular, differential GPS algorithms - while outside the scope of this thesis, were considered by, for example, [3] and [7], within the context of formation flying missions. Outside the field of formation flying, improvement of position and state estimates using two-line elements (TLEs) have also been under study by [6], [8], [9], [10] and [11] among others. In particular, TLE accuracy for CubeSats was assessed by Coffee, who compared GPS data with estimates propagated from corresponding TLEs to give a better understanding of the expected errors [12].

Guidance and control approaches can be categorized in many ways. One is to use five control configurations as recommended by [13]: Multiple-Input Multiple-Output (MIMO), Leader/Follower, Virtual Structures, Cyclic and Behavioural. The surveys completed by Scharf in 2003 on guidance and control give a detailed background in how these classifications were chosen and what developments were made in those areas up until that time [14], [13]. The most popular of these approaches has been and continues to be the Leader/Follower configuration (also called Chief/Deputy or Master/Slave) in which the orbit of one satellite (designated the leader) is treated as the reference trajectory for all of the other satellites (called followers) in

the formation [13]. Cyclic controllers have also been of recent interest, as they ensure fuel is balanced between the satellites of the formation by alternating which satellite is the 'leader' in a structured way, such as recommended in [15]. This approach to fuel balancing will be revisited in Chapter 6.

Another way to group controllers is between impulsive control and continuous control. Continuous control has the advantage of much smaller control windows on the nominal separation distances between satellites, however, operating a thruster continuously may be problematic to science instruments that require more stable operating conditions [16]. This is also the case for satellites that are more limited in capabilities such as CubeSats that either cannot maintain attitude sufficiently accurately or operate a thruster for a sufficiently long time for a continuous control scheme [17]. Fixed-impulse controllers provide both simplicity and transparency by constraining the time (or equivalently the position along an orbit) when a thrust can be planned and in some cases also the direction for the thrust [16], [18]. Thus only the magnitude is optimized by the controller which significantly reduces the controller's complexity, though this comes at a cost of a larger control window and the separation distance between the satellites [16]. The general fixed-impulse method as described in [18]) has been expanded upon by considering single-input controllers (uni-directional thrust vectors, for example along the geomagnetic field vector) [19], and its sensitivity to sensor and actuator errors has been assessed in comparison with other impulsive controllers [20]. Many fixed-impulse controllers have linearized the equations of motion (using variations on the Hill-Clohessy-Wiltshire equations [18]), however it is also possible to use non-linear relative motion equations as shown by [21]. These equations will be further detailed in Chapter 3.

The research in this thesis focuses on formation maintenance algorithms, however automating reconfiguration and acquisition of formations is also a topic of interest that can be explored further in [18], [22], [23], [24], [25] and [26] among others. There has also been interest in various formation shapes (projected circular orbits, side-by-side flight, tetrahedrals etc.) which are beyond the scope of this thesis, which focuses on along-track formations. Some discussions of other formation shapes can be found in [18], [22], [26], [27] and [28].

Reviews of the development of formation flying controllers have been compiled by Scharf in 2003 [13], Vaddi in 2005 [22], Eyer and Kristiansen in 2009 [26], [29], and Alfriend et. al. in 2010 [18]. The emerging controller designs have followed two distinct approaches. Several controllers have pushed towards increasingly high requirements on navigation accuracy and relative position accuracy, which has lead to improved dynamics models for linear quadratic regulators and fixed-impulse controllers ([21], [30], [31]) and new methods using approaches such as $H_\infty$ controllers ([32]), Sliding Mode controllers ([33], [34], [35], [36]), State-Dependent Riccati Equation (SDRE) controllers ([37], [38], [39]), Model Predictive Controllers (MPC) ([20], [40]), or adaptive controllers ([41], [42], [43], [44]). On the other hand, reduction in actuation complexity has driven research towards single-input controllers such as the bi- and uni-directional controllers developed by Guerman et. al. [19] and the underactuated systems such as [24], or towards concepts using the orbital perturbations to their advantage. Along a different line of development, the control force for formation maintenance has been achieved by (1) controlling the ballistic coefficient of a satellite to use aerodynamic drag ([36], [45]), (2) controlling solar flaps to use solar radiation pressure ([36], [46]) or (3) controlling the net

charge on the vehicle to use the Coulomb force ([47], [48], [49], [50]). These are alternatives to the more traditional propulsive control.

This divergence in approach between position accuracy and actuation complexity is expected to become less pronounced as processors become more capable on smaller systems - allowing simple systems (such as CubeSats) to provide increasingly capable relative position maintenance.

### 1.2.2 Opportunity for Development

Previously, CubeSats have not been seen as a useful platform due to their limited dimensions and power, however more recently CubeSat networks have shown that they can provide additional capabilities. The adoption of formation flying would further enhance the scientific capabilities of such a network by allowing the collection of either temporally or spatially correlated data. This application of formation flying algorithms is now possible due to the improvements in CubeSat propulsion technologies, as well as a better understanding of TLE accuracy for nanosatellites and CubeSats. This is the technology that the DelFFi mission aims to demonstrate.

This technical challenge of transferring formation technology onto the CubeSat platform includes the need to address the limited rate and accuracy of the TLE as a navigation sensor, the thrust and pointing accuracy limitations of a micropropulsion system and the limited processing power available onboard. These technical challenges will further lead to operational issues that must be planned for early in the mission.

With the QB50 project (as further detailed in Chapter 2), DelFFi has been provided with a payload which provides some additional constraints on the system. The most critical of which, is the desire for a larger baseline between the satellites than most formation flying missions. The desired separation distance for the DelFFi mission is 1000 km (corresponding to 130 s between the two satellites) [1]. This is considered a widely-spaced formation (in comparison with closely spaced formations such as CanX-4/5 with separation distances between 50 m and 1000 m) [26]. This ensures a large safety margin against collision but adds additional challenges to the passive relative motion due to differential gravitational perturbations.

This opportunity to use TLEs as the primary source of navigation data for a widely-spaced along-track formation consisting of three-unit CubeSats is one that could prove to be valuable to other teams looking to enhance the scientific value of their CubeSat missions without significantly increasing the mission complexity.

## 1.3 Research Objectives

The aim of this research project is to develop and characterize the formation maintenance algorithm of the AFF payload for the DelFFi mission. Specifically, this research will address

a TLE-based formation maintenance algorithm for a widely-spaced, along-track formation consisting of two 3-unit CubeSats in low Earth orbit. From this aim, the objective questions and approach to the research are defined.

The research objective for this project is to make recommendations regarding the implementation of formation maintenance for the DelFFi AFF payload, by characterizing the performance of a TLE-based controller in light of the DelFFi requirements. This objective results in three main phases: determining the performance of a TLE-based navigation algorithm, designing and characterizing a formation maintenance controller and detailing the operational considerations for the AFF payload within the DelFFi mission.

This object can be captured by the following three research questions:

1. What are the capabilities and limitations of a two-line element based navigation algorithm for CubeSats?
    a) What are the characteristics of two-line elements?
    b) In what ways do the initial orbital parameters affect the passive relative motion of the satellites? What passive relative motion is expected for the DelFFi satellites?
    c) What is the expected navigation accuracy for an algorithm based on two-line elements executed on a CubeSat?
2. What impact will CubeSat technologies have on the implementation of a formation maintenance controller for the DelFFi mission?
    a) To what degree do existing formation maintenance control algorithms meet the DelFFi requirements?
    b) Given the limitations of TLE-based navigation and existing CubeSat technologies used on the DelFFi satellites, what performance can be expected of a formation maintenance algorithm?
    c) What requirements must be placed on the actuator (thruster and attitude determination and control system) in order to support the formation maintenance controller?
3. In what ways are the operations of the satellite affected by the Autonomous Formation Flying payload?
    a) What communication/coordination is required between the two satellites and in what way should this be accomplished?
    b) What are the modes associated with the formation maintenance operational phase of the DelFFi mission?

In order to answer these questions, MATLAB Simulink will be used to model the onboard software (formation guidance, navigation and control algorithms). This will also include actuator and sensor models as well as an orbit propagator to allow simulations of the formation maintenance algorithm. The advantage of using Simulink to develop these models, is that the

model of the onboard formation maintenance software can be converted into C for testing on the target microprocessor using the Simulink Coder. This Simulink model (as described in Chapter 5 shall be developed in a modular way such that it can be used to consider different guidance, navigation and control architectures, as well as various actuators and sensors for future projects.

## 1.4 Thesis Structure

This thesis consists of seven chapters including this introduction, as shown in Figure 1.2. First, background relating to the DelFFi Autonomous Formation Flying payload (Chapter 2) and the characteristics of formation flying (Chapter 3) are established. Then, Chapters 4, 5 and 6 address the three research questions: TLE-based navigation, formation maintenance controllers and operational considerations for formation flying on CubeSats. Finally, conclusions, recommendations and future work are provided in Chapter 7.

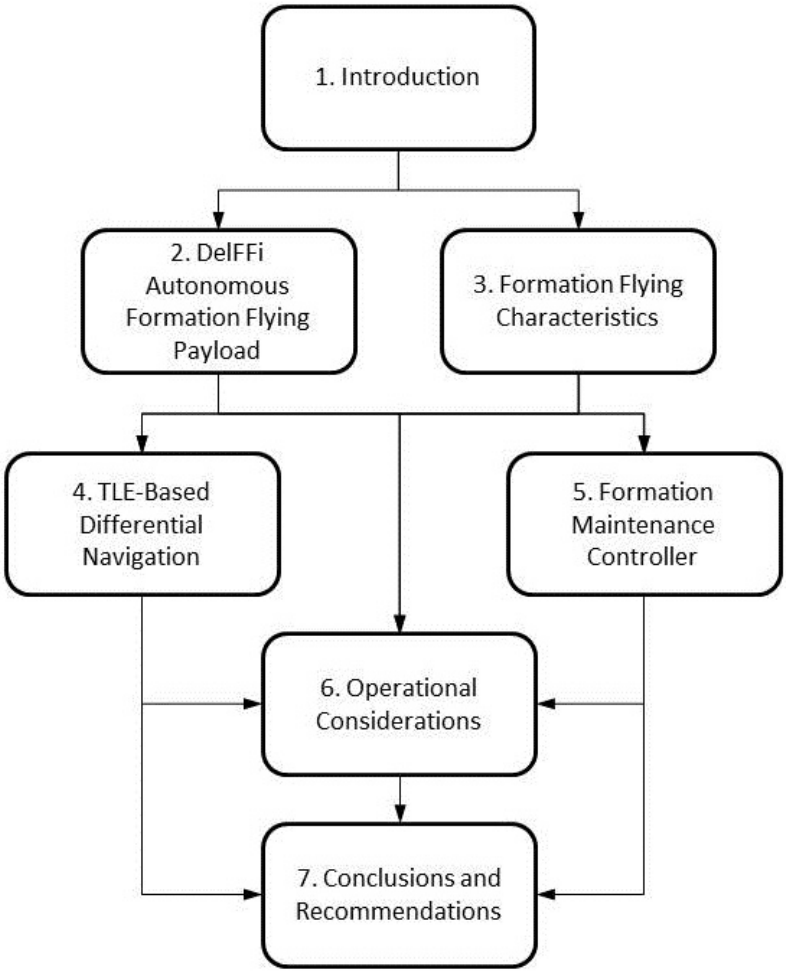



**Figure 1.2:** Visualization of Thesis Chapters.

# Chapter 2

# DelFFi Autonomous Formation Flying Payload

With the objective of providing hands-on training for students and advancing the development of nanosatellites, the Delfi Space Program at TU Delft launched its first satellite (Delfi-C$^3$) in 2008 and its second (Delfi-n3Xt) in 2013 [1]. The current project, DelFFi, aims to build upon these past successes and further enhance the capabilities of CubeSats by demonstrating autonomous formation flying. This section will describe relationship between the DelFFi design and the QB50 Project and identify the enabling technologies within the DelFFi platforms relevant to the Autonomous Formation Flying (AFF) payload. It will then define the requirements on the Formation Flying Package (FFP) and select the relevant performance parameters.

## 2.1 QB50 Project Context

In order to enhance the scientific return of the atmospheric sensors onboard the QB50 satellites, DelFFi will include an AFF payload that will maintain a constant along-track separation distance between the satellites. Two orbits have been identified for the QB50 mission. The nominal orbit is a 380 km altitude circular, sun synchronous orbit. For the purposes of modelling, an eccentricity of 0.001 is used as a perfectly circular orbit will not be achievable in reality, and a Right Ascension of the Ascending Node (RAAN) of -15° will be used [51]. A secondary option is the QB50 elliptical orbit. It is a 400 km by 600 km orbit, which will be modelled with the same inclination and RAAN as the nominal orbit and an argument of perigee of 0°, since these values are not known at this time. The RAAN of the elliptical orbit will vary at a rate of four hours per year meaning that the orbit at which the satellites are inserted, will not be the one in which formation acquisition occurs. Since the main purpose of this work is to control the along-track separation, the choice of RAAN will have a small impact on the design developed in this work. Further, there is no clear indication of the RAAN of the elliptical orbit, so the RAAN will be arbitrarily chosen to match

the RAAN of the circular orbit. Once the orbital parameters have been established, the relative motion model should be iterated to a more precise level. For both of these orbits, the target separation distance is 1000 km based on the scientific objective as established in [2].

Although the primary DelFFi mission is aligned with the QB50 project, formation flying using CubeSats is an interesting design problem outside of the QB50 framework, as correlated data is useful for a variety of scientific missions as mentioned in the previous chapter. For this reason, this research will not only consider the constraints of the DelFFi project on the AFF payload, but also address the more generalized problem of a pair of CubeSats in a widely-spaced (500 km - 1000 km separation) along-track formation. The orbital elements for the reference orbits under study in this research are listed in Table 2.1 below for clarity.

**Table 2.1:** Mean orbital elements of the two orbits of interest for the DelFFi mission.

| Element | Circular Orbit | Elliptical Orbit |
|---|---|---|
| Semi-Major Axis ($a$) [km] | 6758 | 6878 |
| Eccentricity ($e$) [-] | 0.001 | 0.015 |
| Inclination ($i$) [deg] | 96.96 | 96.96 |
| RAAN ($\Omega$) [deg] | -15 | -15 |
| Argument of Perigee ($\omega$) [deg] | 0 | 0 |
| Mean Anomaly at Epoch ($M_0$) [deg] | 0 | 0 |
| Nominal Separation Distance ($d_{sep}$) [km] | 1000 | 1000 |

## 2.2 CubeSats

In an effort to simplify launch procedures for very small satellites, Stanford University's Space Systems Development Laboratory developed a standard for nanosatellites called a CubeSat [2]. A one-unit CubeSat measures 10 x 10 x 10 cm and with a nominal mass of 1 kg [2]. Extensions of this are the two-unit CubeSats (10 x 10 x 20 cm and 2 kg mass) and three-unit CubeSats (10 x 10 x 30 cm and 3 kg mass). Although these strict dimensions pose limitations on the CubeSat's capabilities (due to power, mass and form factor limits [2]), the advantage is the ability to use a standard Poly-Picosatellite Orbital Deployer (P-POD) to maximize compatibility with launch vehicles [2]. Once on orbit, the CubeSat is deployed from the P-PODs using a spring which results in an exit velocity of 1.6 m/s or 1 m/s depending on the P-POD provider [2]. The standard form factor is beneficial for a formation flying mission, as it reduces variations between the satellites, which results in additional (undesired) relative motion.

## 2.3 Enabling Technologies

The two DelFFi satellites (called Delta and Phi) will be identical CubeSats, developed based on the Delfi-n3Xt bus [1]. The mass budget indicates a total mass of 3.64 kg per satellite

[52]. Details on the overall satellite bus design are provided in [1]. The AFF payload is at its core a set of algorithms bundled in what will be called the Formation Flying Package (FFP) running on the Attitude Determination and Control (ADCS) processor. In order to perform formation maintenance, the FFP will need to interface with the attitude determination and control subsystem (ADCS), the micropropulsion system ($\mu$PS), the inter-satellite link (ISL), and the global positioning system (GPS) receiver. The relationships between these aspects are shown in the context diagram in Figure 2.1.

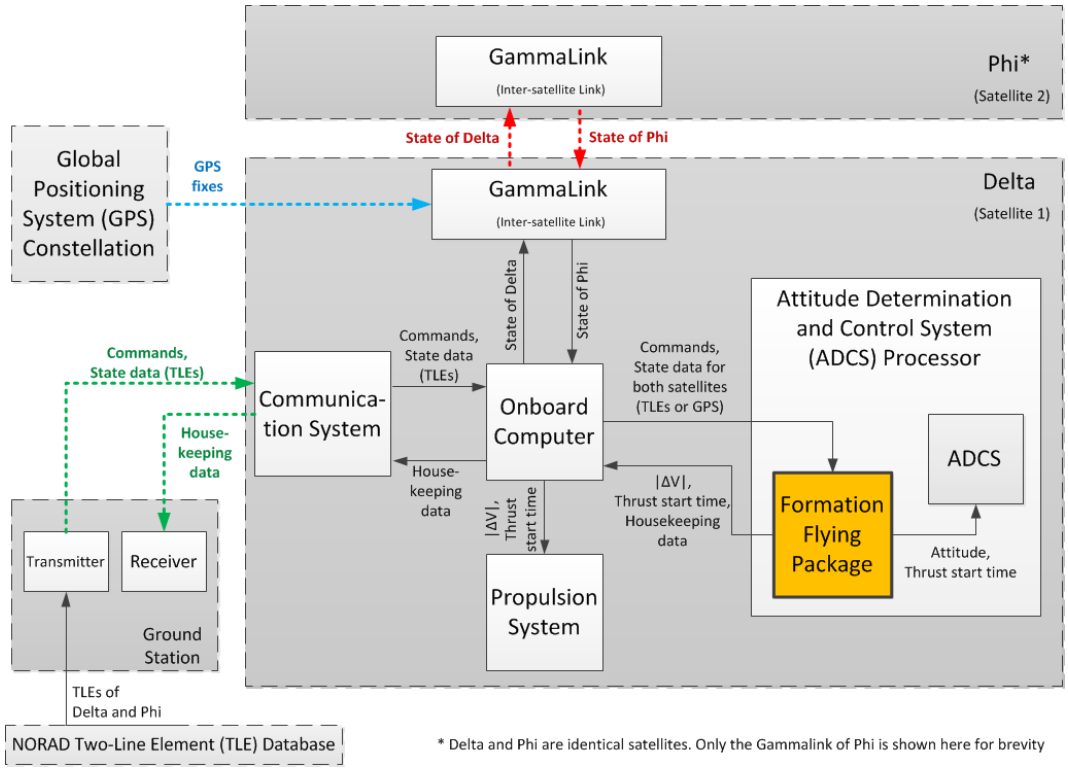



**Figure 2.1:** DelFFi Formation Flying Package Context Diagram.

The FFP contains the software necessary to perform the autonomous formation maintenance, including guidance, navigation and control algorithms as well as a team management to coordinate between the satellites. The design of the FFP is governed by the requirements described in the Section 2.4 and is discussed in Chapters 4 and 5. The design of the navigation sensors, propulsion system, communication links, attitude determination and control subsystem (ADCS) and ADCS microcontroller, as they relate to the FFP, are described here.

**Navigation Sensor** The inclusion of a relative position sensor on each of the CubeSats is prohibited by the limited volume of the satellite. Instead, primary position information is provided by two-line elements (TLEs) from the ground (Refer to Chapter 4 for further details on TLEs). In addition, global positioning system (GPS) receivers integrated in the GAMALINK hardware, will be used on each satellite to determine its absolute position, which can then be exchanged with the other satellite (using either the ground link or the

intersatellite link) to determine their relative positions [1]. Due to risks relating to the low maturity of the GAMALINK GPS receivers (at TU Delft) and potentially limited availability of GPS fixes, it is not desirable to rely solely on GPS for the mission.

**Propulsion System** After a detailed selection process discussed in [53], the design of the $\mu$PS micropropulsion system was conceptualized. It is a micro-resistojet which will provide the required 15 m/s $\Delta$v while fitting within the CubeSat [54]. The theoretical design of Poyck suggests that 50 g of propellant stored at between 4.5 bar (beginning of life) and 2.5 bar (end of life), will provide a thrust of between 1.4 mN (beginning of life) 0.8 mN (end of life) [55]. The variation in thrust over time implies that the to provide the same $\Delta$v at the beginning and end of the mission, the burn time will have to increase as the mission proceeds (this will be discussed further in Chapter 5). This design has not yet been tested in its final state and, as such, noise and performance data will be taken from experience gained from the thesis work of Migliaccio on the T$^3\mu$PS that was used on Delfi-n3Xt [56]. Based on this experience, the design calls for the instantaneous thrust to be estimated based on measurements from a pressure sensor, allowing the burn time to be adjusted in real time to ensure the desired manoeuvre $\Delta$v is provided. The accuracy of the thrust is based on the accuracy of the pressure sensor which is estimated to be no more than 0.5% (as per conversation with A. Cervone, August 19, 2015).

The micro-resistojet design requires the system to be pre-heated before each burn, which is expected to last well under ten minutes (as per conversation with A. Cervone, August 19, 2015). A power budget has been developed for the DelFFi satellites that allows for one 30 minute propulsion system activation (pre-heating and thrust) during the sunlit portion of every second orbit [57]. Thus allowing for up to 20 minute burns every other orbit. The thermal design identified the optimal burn time to be between 30 s and 300 s (as per conversation with T. van Wees, April 2015), based on the desire for the burn to not be inefficiently short relative to the pre-heating, nor cause over-heating to surrounding systems. These burn durations are not hard limits and they can be tailored to the requirements of the formation maintenance controller if its needs vary from these guidelines.

**Communication Links** In addition to the uplink/downlink to the ground station, the DelFFi satellites will be equipped with the GAMALINK system, which provides an S-band intersatellite link (ISL) [1]. The maturity of this system is very low and the procurement process poses significant risks. Design options in which the communications between satellites are relayed through ground stations shall be considered.

**Attitude Determination and Control Subsystem (ADCS)** Since the thruster provides thrust only in one direction (along the satellite's axis), the ADCS is required to point that axis in the desired thrust direction based on the formation maintenance controller. This means that not only must the pointing direction be controlled, but also that the rate of rotation must be minimized such that the platform remains stable during the burn. Attitude pointing errors during formation maintenance (in the ADCS Vector Pointing Mode) are required to be $\leq 10^o$ ($\leq 1^o/s$) [58]. A second ADCS mode, Thrust Vector Control Mode, uses a reaction wheel to further improve performance to an attitude pointing error $\leq 2^o$ ($\leq 1^o/s$), however this has not been studied in detail.

**Microcontroller** The FFP will be incorporated onto the ADCS microcontroller which is based on an ARM-9 processor with the Android 2.3 operating system [1]. The selected processor is a Texas Instruments TM4C1294NCPDT, which is available for preliminary testing using the Tiva C Series Connected LaunchPad Evaluator Kit [59]. This microcontroller will run both the ADCS algorithms as well as the FFP, so timing constraints must be considered. Although the design is only in the preliminary stages, a one second loop has been selected, with at most half of that time allotted to each of the two packages [59]. This means that the FFP must be able to complete one cycle of calculations in under 500 ms or else the structure of the code must be divided to execute over multiple cycles (loops). The clock frequency for the system will be 100 MHz.

## 2.4 Formation Flying Package Requirements

Requirements on the FFP have been developed based on the DelFFi mission-level requirements and the existing constraints on the FFP from other subsystems. Each requirement is composed of the requirement identifier, a brief statement of the requirement and the method by which it will be verified. The requirement identifier has been kept short for the sake of brevity. It consists of three clauses: the system the requirement applies to, the type of requirement and the requirement number. Five requirement types are identified based on the European Space Agency (ESA) standard ECSS-E-ST-10-06C [60]. The requirement numbers are divided evenly between these five requirement types, thus the requirement number alone is sufficient to identify the requirement within the FFP. The format is defined below in Table 2.2.

**Table 2.2:** Requirement Identifier Structure.

| First Clause | Second Clause | Number |
|---|---|---|
| AFF | - Mission (M) | - 01 – 19 |
| | - Functional (F) | - 20 – 39 |
| | - Design (D) | - 40 – 59 |
| | - Operational (O) | - 60 – 79 |
| | - Interface (I) | - 80 – 99 |

The FFP requirements are provided in detail in Appendix B and are summarized in Tables 2.3 to 2.7 for clarity.

The three verification methods identified in Tables 2.3 to 2.7 have been taken from the ESA standard ECSS-E-ST-10-02C [61], and their definitions can be found therein. Requirements that are verified by test are the most reliable while those verified by analysis or review of design are less so. Since the majority of requirements cannot be verified by test due to the available resources at this time, a risk assessment was performed to highlight critical development risks for the FFP, the AFF payload and the DelFFi mission itself. This risk analysis and mitigation plan are included in Appendix C.

**Table 2.3:** Formation Flying Package Mission Requirements.

| ID | Requirement | Method | | |
|---|---|---|---|---|
| | | Test | Analysis | Review |
| AFF-M-01 | In the formation keeping phase, the formation flying package shall maintain a nominal inter-satellite distance (in the along-track direction). | | x | |
| AFF-M-02 | The formation keeping phase of the mission shall last no less than 20 days [OPTIONAL: 30 days]. | | x | |

Keeping track of all of the requirements throughout the design is unrealistic. Instead, three driving requirements are identified that summarize the critical requirements for the FFP. These are: AFF-F-24 (concerning relative position control accuracy), AFF-F-26 (concerning total propellant budget), and AFF-F-33 (concerning processing time).

### 2.4.1 Performance Parameters

During development and testing a set of performance parameters should be used to ensure the FFP design is progressing in line with the requirements identified here. In prior work of the author [62], existing formation flying missions were studied and their verification procedures were summarized. Although existing missions used a variety of combinations of simulation and hardware tests, the parameters measured by the tests were consistent: (1) Relative Position Accuracy, (2) Fuel Consumed (total for all satellites in formation over a representative period) and (3) Processing Time. These three parameters also align well with the driving DelFFi FFP requirements identified at the end of Section 2.4.

## 2.5 Conclusions

This chapter described the DelFFi AFF payload to which this research will contribute. The designs of the satellite bus and supporting subsystems have not yet been completely defined, thus allowing this research to provide input to those designs, to better meet the DelFFi mission objectives. This research will also allow many of the requirements defined in the Appendix B to be verified at an early stage of the design, thus increasing the AFF payload's technology readiness level and decreasing the system risk. Three performance parameters have also been selected for study throughout this research: relative position accuracy, total fuel consumed, execution time of the onboard software. These three measures will be revisited throughout this work, in particular in Chapter 5. This research will aim to develop a system that meets the requirements set in this chapter, while future work may optimize the system for better performance.

**Table 2.4:** Formation Flying Package Functional Requirements.

| ID | Requirement | Method | | |
|---|---|---|---|---|
| | | Test | Analysis | Review |
| AFF-F-20 | The AFF payload shall implement relative guidance and navigation using ground and/or onboard information. | | | x |
| AFF-F-21 | The AFF payload shall generate and implement relative control commands. | | | x |
| AFF-F-22 | The satellite shall determine its position to within 10 km [OPTIONAL: 1 km] accuracy (TBC). | | x | |
| AFF-F-23 | The nominal inter-satellite distance shall be 1000 km. | | | x |
| AFF-F-24 | The nominal inter-satellite distance shall be maintained to within 10% of its value - ie. 1000 km $\pm$ 100 km. | | x | |
| AFF-F-25 | The attitude vector (the desired control acceleration direction) shall be requested no less than 15 minutes [5 minutes if reaction wheel is active] (TBC) in advance of a propulsive manoeuvre. | | x | |
| AFF-F-26 | The AFF payload shall require no more than 15 m/s delta-v during the mission lifetime to maintain the required control window. | | x | |
| AFF-F-27 | The burn time required from the thrusters shall be no less than 30 seconds (TBC) and no more than 300 seconds (TBC). | | x | |
| AFF-F-28 | The thruster activation shall be requested no less than 300 seconds (TBC) in advance of the thrust start time. | | x | |
| AFF-F-29 | [OPTIONAL] The propulsive manoeuvres should be timed to occur during the second half of sun-lit periods only. | | x | |
| AFF-F-30 | [OPTIONAL] The propulsive manoeuvres should be separated by at least one non-thrusting sun-lit period (ie. two orbital periods). | | x | |
| AFF-F-31 | [OPTIONAL] The AFF payload should maximize the time available for scientific observation (non-firing, in-track pointing). | | x | |
| AFF-F-32 | The AFF payload shall provide housekeeping parameters (to be detailed) at a rate of 1 Hz (TBC) to the On-Board Computer (OBC). | | | x |
| AFF-F-33 | [OPTIONAL] The formation flying package should require no more than 500 ms per cycle when executed on the ADCS microcontroller. | x | | |

**Table 2.5:** Formation Flying Package Design Requirements.

| ID | Requirement | Method | | |
|---|---|---|---|---|
| | | Test | Analysis | Review |
| AFF-D-40 | The formation flying package will be identical on both satellites. | | | x |
| AFF-D-41 | The AFF payload shall be compatible with the nominal QB50 orbit. | | x | |
| AFF-D-42 | [OPTIONAL] The AFF payload should be compatible with a generic near-circular low Earth orbit with the ranges of parameters (as specified in Appendix B and in Table 2.1). | | x | |
| AFF-D-43 | The formation flying package shall remain functional despite loss of the inter-satellite link. | | x | |
| AFF-D-44 | [OPTIONAL] The formation flying package should remain functional despite loss of GPS fixes on one or both satellites. | | x | |

**Table 2.6:** Formation Flying Package Operational Requirements.

| ID | Requirement | Method | | |
|---|---|---|---|---|
| | | Test | Analysis | Review |
| AFF-O-60 | [OPTIONAL] The AFF payload should allow the eight GNC Architectures, as defined in the Phase A Study. | | x | |
| AFF-O-61 | The AFF payload shall facilitate autonomous operation (without human interaction). | | | x |

**Table 2.7:** Formation Flying Package Interface Requirements.

| ID | Requirement | Method | | |
|---|---|---|---|---|
| | | Test | Analysis | Review |
| AFF-I-80 | The Formation Flying Package shall be compatible with the ADCS microcontroller. | x | | |
| AFF-I-81 | The AFF payload shall be compatible with a clock signal which uses Coordinated Universal time (UTC) as the time reference. | | | x |

# Chapter 3

# Formation Flying Characteristics

This chapter will first introduce the formation flying architectures used in this research, then describe the relative motion models necessary to understand the passive relative motion between the satellites.

## 3.1 Formation Flying Architectures

Two architectures are relevant to discussing the structure of the Formation Flying Package (FFP) developed in this research: the guidance navigation and control (GNC) architecture and the operational architecture.

### 3.1.1 Guidance Navigation and Control Architecture

A formation flying algorithm is a specific example of a GNC algorithm and can be characterized through the generic GNC architecture shown in Figure 3.1.

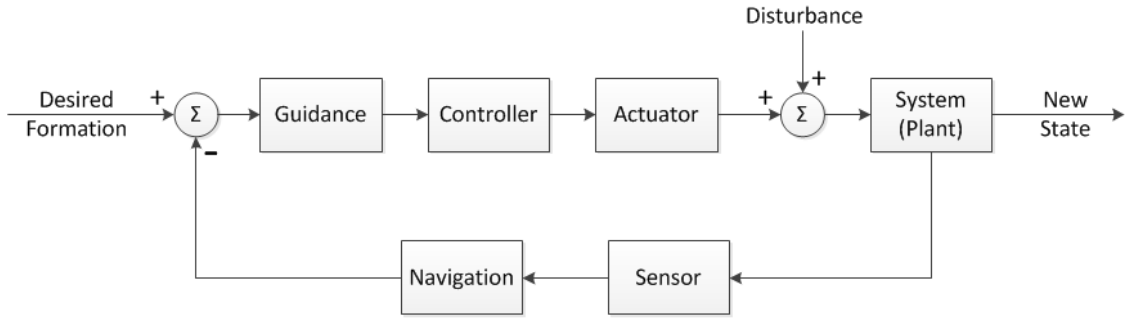



**Figure 3.1:** General architecture of a Guidance Navigation and Control (GNC) loop.

This architecture has driven the design of the simulation model with one variation. The DelFFi formation will be an along-track formation between two satellites in the same orbit

with a constant separation distance. Consider, for example, a circular orbit. If the relative positions are characterized through classical orbital elements, this corresponds to a constant offset in the true anomaly. As such, the guidance algorithm becomes trivial and it is convenient to include it within the control algorithm rather than isolate it. Despite this combination, this architecture still lends itself to more complex formations (involving more spacecraft, reconfiguration manoeuvres or the like), which would normally require dedicated guidance algorithms.

### 3.1.2 Operational Architecture

From an operational view, the DelFFi team has previously identified eight scenarios with varying levels of autonomy for the FFP to perform. These eight scenarios characterize the operational architecture that will be followed for the DelFFi mission and are provided in Table 3.1.

**Table 3.1:** GNC Scenarios for the Autonomous Formation Flying Payload [63]

| Scenario Name | Guidance | Navigation | Control |
|---|---|---|---|
| 1. Basic | On ground | TLE uploaded to 1 s/c | Single thruster |
| 2. Onboard | Onboard | TLE uploaded to 1 s/c | Single thruster |
| 3. Distributed | On ground | TLE uploaded to 2 s/c | Single, dual |
| 4. Distributed onboard | Onboard | TLE uploaded to 2 s/c | Single, dual |
| 5. Distributed coordinated | Onboard | Relative TLE onboard and exchanged via ground | Single, dual |
| 6. Onboard ISL-based relay | Onboard | Relative TLE computed onboard, relayed via ISL | Single, dual |
| 7. Distributed coordinated ISL-based | Onboard | Relative TLE computed onboard and exchanged with ISL | Single, dual |
| 8. Full autonomous formation flying | Onboard | Determined onboard by exchange GPS via ISL | Single, dual |

These scenarios provide a wide range of demonstration capabilities, but in general do not change the functional design of the FFP itself. In each case the FFP is provided with a navigation source from the onboard computer (OBC), the control input is calculated by each satellite, then it is executed as planned. For scenarios 1 and 3, the control input calculated onboard is over-written by the ground command. This variation does not change the algorithm but indicates a flag should be included to allow ground input to be used. In scenarios 5, 6 and 7 the satellites relay the relative TLE either through the ground station or the ISL rather than each receiving the TLEs. This extra step will mean an additional delay in receiving the TLEs, however the delay will be on the order of an orbit while the TLE update rate is on the order of days (TLE update frequency is approximately 0.6 days) [64] (for further detail see Chapter 4). For the sake of the simulations conducted here, scenario 4 – in which both satellites receive both TLEs from the ground, the guidance, navigation and control inputs are all calculated onboard and either one or both thrusters can be used for the manoeuvre – will be considered, based on the analysis in Chapter 6.

## 3.2 Relative Motion Models

### 3.2.1 References Frames

The most important reference frames (or coordinate frames) for Formation flying are the Earth Centred Inertial (ECI) and the Hill frame. These are standard reference frames that are defined in several sources, but are taken here from [18] and [26] for consistency.

**Earth Centred Inertial**
The ECI frame is geocentric (centred on Earth) with the **x-y** plane through the equator and the **z** vector towards the geographic north pole (Celestial Ephemeris Pole [26]) [18]. The **x** vector is oriented towards the vernal equinox and the **y** vector completes the right-handed Cartesian frame. As the name implies the frame is inertial – both non-rotating and non-accelerating. Figure 3.2 shows a satellite in this frame.

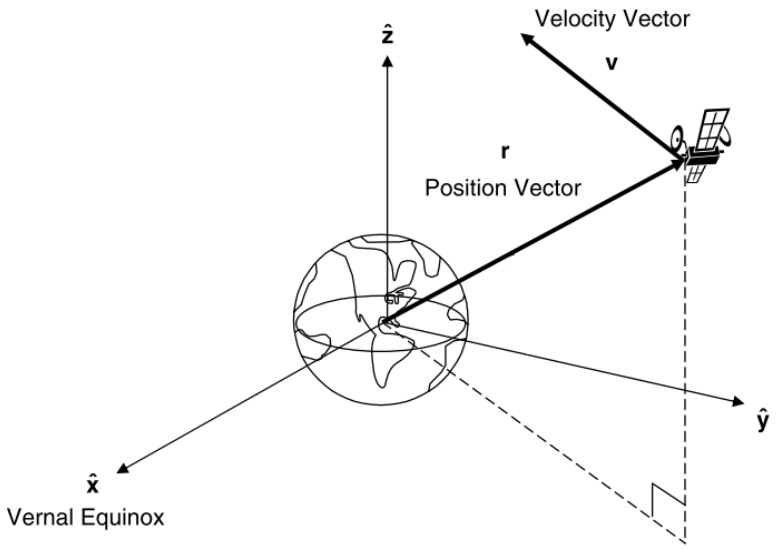



**Figure 3.2:** The Earth-centred-inertial (ECI) Coordinate system as well as the position and velocity vectors. Adapted from [18].

**Hill Frame**
The Hill frame is a local-vertical/local-horizontal rotating reference frame that is popular for expressing the relative position and velocity of one satellite with respect to another [26]. The frame is centred on the chief (also called leader, primary or main satellite) with the **x** axis in the same direction as the radial vector, the **z** axis in the direction of the orbit normal (or angular momentum vector) and the **y** axis completing the right-handed system [18], as shown in Figure 3.3.

**Conversion Between Frames** Rotation matrices are used to transfer from one reference frame to another. For detail of their structure, formation and use, refer to an astrodynamics

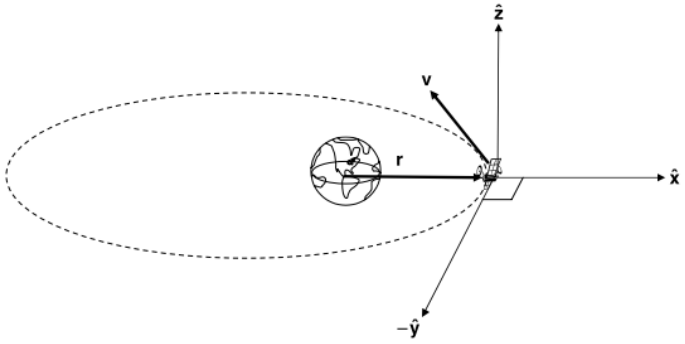



**Figure 3.3:** The Hill coordinate frame as well as the position and velocity vectors [18].

text, [18] or [26].

### 3.2.2 Orbital Elements

Several methods of defining an orbit have been used in past work. This section will discuss Keplerian orbital elements, classical orbital elements, a set of non-singular orbital elements and the difference between mean and osculating orbital elements.

**Keplerian Orbital Elements**
The standard means of representing the position of a satellite in an orbit is through the Keplerian orbital elements. They define the size and shape of the satellite's orbit as well as its position around the orbit. The six elements used to define the orbit are the semi-major axis, $a$, the eccentricity, $e$, the inclination, $i$, the right ascension of the ascending node, $\Omega$, the argument of perigee, $\omega$, and the true anomaly, $\theta$. These are shown below in figure 3.4.

**Classical Orbital Elements**
It is also common to define the mean motion, $n = \sqrt{\frac{\mu}{a^3}}$, and the mean anomaly, $M = M_0 + n(t - t_0)$ [18]. The term $M_0$ is the mean anomaly at epoch, which is a constant of motion in the Keplerian two-body problem [18]. This forms the set of classical orbital elements: $[a, e, i, \Omega, \omega, M_0]$ [18].

**Non-Singular Orbital Elements**
However, these classical orbital elements have a singularity for circular orbits [18]. To mitigate this, a set of non-singular orbital elements have been suggested using the terms $q_1$, $q_2$ and $\lambda$ (the mean argument of latitude), as given in equations 3.1, 3.2 and 3.3, in place of $e$, $\omega$ and $M$ [18]. The mean argument of latitude $\lambda$ is also represented by $u$ in some sources including the controller detailed in Chapter 5.

$$q_1 = e\cos\omega \tag{3.1}$$

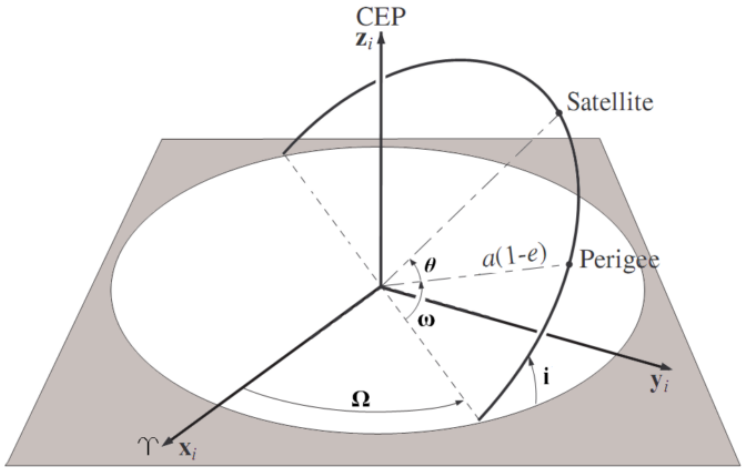



**Figure 3.4:** The definition of the Keplerian orbital elements [26].

$$q_2 = e \sin \omega \tag{3.2}$$

$$\lambda = \omega + M \tag{3.3}$$

This set of elements $[a, q_1, q_2, i, \Omega, \lambda]$ is still singular for equatorial orbits [18], but will be sufficient for the sun-synchronous orbits addressed in this research.

**Mean and Osculating Orbital Elements**

When describing orbits using orbital elements (in particular when determining passive relative motion or in formation flying control algorithms), it is important to distinguish between two types of orbital elements: osculating (or instantaneous) orbital elements and mean orbital elements. The osculating orbital elements are those that would be found by measuring the instantaneous position and velocity vectors (as would be done using a GPS receiver) and transforming them into orbital elements. Mean orbital elements are found through an averaging algorithm such as the SGP4 (Standard General Perturbations 4 [11], [65]) used to generate TLEs as discussed in Chapter 4. If two satellites have the same mean orbital elements, their relative drift will be small, since their orbit's will have the same energy, so differential perturbations are small (see Section 3.2.3). On the other hand, two satellites, having the same osculating elements at a certain point, could correspond to completely different mean orbits. This would lead to large relative drift - especially if the mean orbits had different semi-major axes (due to the effect of differential drag at low altitudes).

When the two satellites are inserted into orbit, they will have nearly identical osculating elements, however depending on the early differential velocity, the satellites may initialize to different mean orbits. The formation acquisition phase must place the satellites into the same

mean orbit. If this were not completed, the frequency of manoeuvres required to maintain the control window would be prohibitive. The rate of manoeuvres will be detailed in Chapter 4.

It is often preferable to use the mean orbital elements for formation flying control algorithms, as they remove the periodic variations due to gravitational perturbations (see Section 3.2.3 for more details on the gravitation perturbations) [66]. This means the control algorithm can correct the secular drift between the satellites without responding to the periodic variations [67]. Note that this is only useful if the magnitude of the periodic variations are within acceptable bounds for the mission. For example, the CanX-4/5 mission required the use of the osculating orbital elements to avoid collisions at close proximity (CanX-4/5 maintained separation distances as small as 50 m) [26]. The accuracy of the mean orbital elements depends on the mapping (averaging) algorithm used. When two-line elements (TLEs) are the source of the mean orbital elements, the SGP4 algorithm used by NORAD to generate the mean elements should be used to return to the osculating elements, as is described by the Onboard Navigation Function (ONF) developed for the Delfi-n3Xt satellite [64]. For this research work, the mean orbital elements have been provided as estimates of the QB50 orbits, so Lyddane's modifications to Brouwer theory will be used to map between these mean and osculating elements as given in Appendix G of [68]. This algorithm is accurate to the order of $J_2$. For example, if the mean anomaly has an accuracy of 0.001 radians (the order of magnitude of $J_2$ from the WGS84 model), this corresponds to a 6.8 km along-track error at the 380 km altitude. The effect of this and other estimation errors are assessed in Chapters 4 and 5.

### 3.2.3 Perturbations

Although several perturbations effect satellite motion in Earth orbit, the most prominent ones are atmospheric drag and $J_2$ (Earth oblateness gravitational perturbation) [69]. While, the higher order geopotential terms were also found to have some effect, the magnitude of the effects of solar radiation pressure and lunar and solar third-body perturbations were negligible in comparison [69].

**Atmospheric Drag** In low Earth orbit, the atmosphere has a low but non-negligible density that reduces the velocity, and thus semi-major axis, of a satellite [26]. The force due to atmospheric drag is given by equation 3.4 below.

$$\mathbf{f}_{drag} = -B\rho V_{rel}\mathbf{V}_{rel} \tag{3.4}$$

Where $\mathbf{V}_{rel}$ is the velocity of the satellite relative to the atmosphere, $V_{rel}$ is the magnitude of $\mathbf{V}_{rel}$, $\rho$ is the density and $B$ is the ballistic coefficient as given by equation 3.5 from [64].

$$B = \frac{1}{2}C_d\frac{A}{m} \tag{3.5}$$

Here, $C_d$ is the drag coefficient of the satellite (2.3 for a CubeSat according to the von Karman Institute's estimations in [70]), $A$ is its cross-sectional area, and $m$ is its mass. Assuming that

the two satellites in formation are not separated by too great a distance, the atmospheric density and the velocity relative to the atmosphere can be assumed to be the same for the two satellites [69]. Thus, the most important factor affecting drift due to differential drag is the Ballistic Coefficient. For two identical satellites that are non-symmetric (such as the pair of three-unit CubeSats of DelFFi), the variation in cross-sectional area due to their relative attitude is the most important consideration [69]. This relative area will be assessed in Chapter 4. To get an idea of how large the difference between ballistic coefficients for two 'identical' satellites can be, the GRACE mission found a typical maximum difference of 1% that varied over the mission time [71].

It is also important to note that even if the ballistic coefficients are identical at the start of the mission, if one satellite uses more fuel than the other, their masses will diverge which will further impact the differential drag. For example, the DelFFi satellites will each carry 50 g of propellant compared to the total satellite mass of 3.64 kg - corresponding to 1.4% [52]. Thus, without fuel balancing (ie. if only one satellite is propulsively active while the other is passive until the first exhausts its fuel supply), the satellites' ballistic coefficients will differ by 1.4% from this effect alone. This will be discussed further in Chapter 6.

**Gravitational Perturbations**

As the Earth is not a perfect sphere, the uneven distribution of mass causes perturbations to the satellite's orbit. To describe the Earth's gravitational field a potential function is defined that divides the earth into zones and sectors as shown in Figure 3.5 [26].


**Figure 3.5:** Types of spherical harmonics used in describing the geopotential field [26].

In selecting which of the geopotential terms to include in the gravity model, it has been shown that the zonal terms have a larger impact on the accuracy than the tesseral and sectorial terms [69]. This is why for the CanX-4/5 model, only (a subset of) the zonal terms (specifically $J_2$, $J_3$, $J_4$, $J_5$, $J_6$) were considered [26]. These terms are constants that have been measured accurately on-orbit and are used to calculate the perturbing accelerations for a satellite, as was done in the appendix of [26]. A different modelling level for the geopotential field was used in developing an orbit propagator within the Space Systems Engineering department. Here, the JGM-3 (the third Joint Gravity Model) coefficients truncated to 20 by 20 (400 coefficients) was used [72]. This model will be referred to hereafter as $J_{20,20}$.

Gravity is a conservative force, meaning that the orbital energy is not directly changed (decreased) by the gravitational perturbation. The energy of an orbit ($E_{orb}$) is given by Equation

3.6. Further, Equation 3.7 gives the orbit velocity (known as the Vis-Viva equation).

$$E_{orb} = \frac{v^2}{2} - \frac{\mu}{r} \tag{3.6}$$

$$v^2 = \mu \left( \frac{2}{r} - \frac{1}{a} \right) \tag{3.7}$$

The two terms in Equation 3.6, corresponding to the kinetic and potential energies respectively, are related solely to the instantaneous position ($r$), the orbit's semi-major axis ($a$) and the standard gravitational parameter ($\mu$) of the body being orbited. This means that the orbit's energy is dependent upon the shape of the orbit (eccentricity) and the size of the orbit (semi-major axis), but not the orientation relative to the Earth (inclination, RAAN, argument of perigee). Further, the mean anomaly expresses the instantaneous position of the satellite around the orbit and changing the initial position along the orbit (mean anomaly at epoch) does not change the orbit's energy.

In practice, both periodic and secular variations are seen in the classical orbital elements as a result of gravitational perturbations. In propagating an orbit under the effects of $J_2$, secular variations are seen in the RAAN ($\Omega$), the argument of perigee ($\omega$) and the mean anomaly at epoch ($M_0$) [73]. On the other hand, periodic variations are seen in all six classical orbital elements. The major impact on an along-track formation due to gravitational perturbation is the indirect affect of drag due to the periodic variation of the semi-major axis. When the semi-major axis is (instantaneously) at a lower position, the satellite experiences an increase in drag and consequently slows down, further decreasing its altitude. Since these variations in semi-major axis occur at different times for the two satellites, there is an apparent differential acceleration, which causes them to drift.

As mentioned previously, since the periodic variation in the orbital elements is small relative to the separation distance (see Chapter 4), these periodic variations are often removed for formation maintenance controllers. This is done by using the mean orbital elements instead of osculating elements to determine the error in relative position.

### 3.2.4 Equations of Relative Motion

Generally, equations of motion relate the current state of a system to its state at some time in the future. This section presents some commonly used equations for the relative motion of two satellites in formation. Although the motion is non-linear, it is typical to select a set of linearized equations that sufficiently model the system to reduce the computational load (for an analysis of relative errors by selecting linear over non-linear models for specific orbit parameters see [18]). The equations of relative motion can be used to map the system's state at one time to another time, and the solution to a set of equations of relative motion can be used to determine reference trajectories for the satellites to follow. A generic linear system can be represented by the following equation:

$$\dot{x} = \mathbf{A}x + \mathbf{B}u_e \tag{3.8}$$

Where $x$ is the state of the system (for example position and velocity), $\mathbf{A}$ relates the current state to the future state without external forces, $u_e$ are the external thrusts and/or disturbances (usually expressed as perturbing accelerations), and $\mathbf{B}$ describes the effect of the external forces (thrust or disturbances) on the future state. Determining the $\mathbf{A}$ and $\mathbf{B}$ matrices has been done in many different ways by different teams studying formation control algorithms. In particular, the Hill-Clohessy-Wiltshire (HCW) equations are often used to build the matrices assuming circular orbits. For elliptical orbits, the HCW equations were extended by Lawden, Tschauner and Hempel. These equations both use cartesian coordinates to express the relative motion. To address the relative motion in terms of orbital elements, Gauss' Variational Equations (GVEs) are typically used. This section will present the HCW equations, Lawden's Elliptical Equations and the GVEs as background for the formation maintenance controllers that are discussed in Chapter 5.

### Hill-Clohessy-Wiltshire Equations

The Hill-Clohessy-Wiltshire equations express the motion of the deputy satellite relative to the chief satellite in the Hill Frame. The derivation assumes there are no perturbations, that the chief is in a circular orbit and that the separation distance between satellites is small [26]. The HCW equations as derived in [26] and [18] are given below in equations 3.9, 3.10 and 3.11.

$$\ddot{x} - 2n\dot{y} - 3n^2x = 0 \tag{3.9}$$

$$\ddot{y} + 2n\dot{x} = 0 \tag{3.10}$$

$$\ddot{z} + n^2z = 0 \tag{3.11}$$

Recall the Hill frame for the directions of $x$, $y$ and $z$, and $n$ is the mean motion (also called mean orbital rate) as defined in Figure 3.3. These equations, called the homogeneous equations, do not include any control or disturbing accelerations [18]. The solutions to the homogeneous equations are given in equations 3.12, 3.13 and 3.14 below [26].

$$x(t) = \frac{\dot{x_0}}{n}\sin(nt) - \left(3x_0 + \frac{2\dot{y_0}}{n}\right)\cos(nt) + \frac{2}{n}(2nx_0 + \dot{y_0}) \tag{3.12}$$

$$y(t) = \frac{2\dot{x_0}}{n}\cos(nt) + 2\left(3x_0 + \frac{2\dot{y_0}}{n}\right)\sin(nt) - \frac{2\dot{x_0}}{n} - 3(2nx_0 + \dot{y_0})t + y_0 \tag{3.13}$$

$$z(t) = \frac{\dot{z_0}}{n}\sin(nt) + z_0\cos(nt) \tag{3.14}$$

In the above equations $t$ is the time since initiation ($t_0$), and $x_0$, $y_0$, $z_0$, $\dot{x_0}$, $\dot{y_0}$ and $\dot{z_0}$ are the initial conditions for relative position and velocity. These solutions show that the out of

plane ($z$) motion has been decoupled from the in-plane ($x$, $y$) motion. Also, that these are periodic solutions with the exception of one term in the equation for the $y$ motion. Since a periodic or bounded solution is desired for formation flight (to ensure no secular drift between the satellites) it is desirable to set the term $2nx_0 + \dot{y}_0 = 0$ [26].

To consider the effects of perturbing and control forces, the non-homogeneous equations can be used as given in equations 3.15, 3.16 and 3.17 [18].

$$\ddot{x} - 2n\dot{y} - 3n^2x = d_x + u_x \tag{3.15}$$

$$\ddot{y} + 2n\dot{x} = d_y + u_y \tag{3.16}$$

$$\ddot{z} + n^2z = d_z + u_z \tag{3.17}$$

In these equations, $d_x$, $d_y$ and $d_z$ are the disturbance accelerations in their respective axes and $u_x$, $u_y$ and $u_z$ are the control accelerations [18]. The solutions for the non-homogeneous equations are less useful for closely spaced formations in which the differential perturbation forces are negligible [26], however they may be valuable for missions with moderate to large separation distances. These solutions are given by equations 3.18-3.23 [18].

$$x(t) = \left[4x_0 + \frac{2\dot{y}_0}{n}\right] + \frac{\dot{x}_0}{n}\sin(nt) - \left[3x_0 + \frac{2\dot{y}_0}{n}\right]\cos(nt) \tag{3.18}$$

$$y(t) = -[6nx_0 + 3\dot{y}_0]t + \left[y_0 - \frac{2\dot{x}_0}{n}\right] + \left[6x_0 + \frac{4\dot{y}_0}{n}\right]\sin(nt) + \frac{2\dot{x}_0}{n}\cos(nt) \tag{3.19}$$

$$z(t) = \frac{\dot{z}_0}{n}\sin(nt) + z_0\cos(nt) \tag{3.20}$$

$$\dot{x}(t) = \dot{x}_0\cos(nt) + [3x_0n + 2\dot{y}_0]\sin(nt) \tag{3.21}$$

$$\dot{y}(t) = -[6nx_0 + 3\dot{y}_0] + [6nx_0 + 4\dot{y}_0]\cos(nt) - 2\dot{x}_0\sin(nt) \tag{3.22}$$

$$\dot{z}(t) = \dot{z}_0\cos(nt) - z_0n\sin(nt) \tag{3.23}$$

Several authors have suggested alternative methods of linearization or methods to account for the various perturbations (most notably $J_2$), including, for example, the Schweighart-Sedwick model used in the analysis of [17]. These have not been considered in this research, but should be addressed in future work.

### Lawden and Tschauner-Hempel Equations

While the HCW equations consider only the relative motion where the chief satellite is in a circular orbit, it is common in practice for the orbit to be at least somewhat elliptical [26]. Lawden, Tschauner and Hempel all independently developed a set of equations without the HCW equations' circular restriction [18]. Lawden's elliptical equations, as derived in [74], are parameterized in terms of the true anomaly, $\theta$, instead of time and are presented in equation 3.24 in their state space form as given in [26].

$$\begin{bmatrix} x' \\ y' \\ z' \\ x'' \\ y'' \\ z'' \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{3+e\cos\theta}{G} & \frac{-2e\sin\theta}{G} & 0 & \frac{2e\sin\theta}{G} & 2 & 0 \\ \frac{2e\sin\theta}{G} & \frac{e\cos\theta}{G} & 0 & -2 & \frac{2e\sin\theta}{G} & 0 \\ 0 & 0 & \frac{-1}{G} & 0 & 0 & \frac{2e\sin\theta}{G} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ x' \\ y' \\ z' \end{bmatrix} + \frac{(1-e^2)^3}{G^4 n^2} \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \tag{3.24}$$

In this equation, $G = 1 + e\cos\theta$ and $()'$ implies the derivative with respect to $\theta$. The accelerations due to disturbances and controls are included in $F_x$, $F_y$ and $F_z$. Note that the out of plane motion ($z$) is decoupled from the in plane motion, just as it was for the HCW equations [26]. The homogeneous solutions presented in [26] are given below in equation 3.25 where $H(\theta)$ is defined by equation 3.26 and $E$ is defined in equation 3.27.

$$\mathbf{r}_{ref}(\theta) = \begin{bmatrix} x(\theta) \\ y(\theta) \\ z(\theta) \end{bmatrix} = \begin{bmatrix} [d_1 e + 2d_2 e^2 H(\theta)]\sin\theta - [\frac{d_2 e}{G^2} + d_3]\cos\theta \\ [d_1 + \frac{d_4}{G} + 2d_2 e H(\theta)] + [\frac{d_3}{G} + d_3]\sin\theta + [d_1 e + 2d_2 e^2 H(\theta)]\cos\theta \\ \frac{d_5}{G}\sin\theta + \frac{d_6}{G}\cos\theta \end{bmatrix} \tag{3.25}$$

$$H(\theta) = \int_{\theta_0}^{\theta} \frac{\cos\theta}{G^3} d\theta = -(1-e^2)^{-\frac{5}{2}} \times \left[\frac{3eE}{2} - (1+e^2)\sin E + \frac{e}{2}\sin E\cos E + d_H\right] \tag{3.26}$$

$$\cos E = \frac{e + \cos\theta}{G} \tag{3.27}$$

In these equations, $d_H$ is a constant of integration found by evaluating the expression $H(\theta_0) = 0$ [26]. The other constants ($d_1$, $d_2$, $d_3$, $d_4$, $d_5$, $d_6$) are selected such that when eccentricity approaches zero, the Lawden solutions become the HCW solutions [26]. To specifically model an along-track formation (such as the one desired for DelFFi), these constants must be set as: $d_1 = d_2 = d_5 = d_3 = d_6 = 0$ and $d_4 = d_{ref}$ where $d_{ref}$ is the nominal separation distance [26]. This reduces the motion of the deputy and chief to an oscillation along the y-axis in the Hill frame according to $y_{ref}(\theta) = d_{ref}/G$ [26].

### Gauss' Variational Equations

Rather than using the cartesian coordinates, several sets of equations use the orbital element sets to describe the relative motion [18]. One popular set are Gauss' Variational Equations (GVEs). GVEs relate the effect of a perturbation or control acceleration on the orbital elements [18]. The GVEs are given in terms of the classical osculating orbital elements [ $a$, $e$, $i$, $\Omega$, $\omega$, $M_0$] in equations 3.28-3.33, where $f$ is the true anomaly, $p = a(1-e^2)$ is the semilatus rectum, $h = \sqrt{\mu p}$ is the magnitude of the angular momentum vector, $r$ is the magnitude of the position vector and $\mathbf{d} = [d_r, d_\theta, d_h]^T$ is the disturbing force written in a polar frame where $d_r$ is in the radial direction, $d_\theta$ is in the direction of the instantaneous velocity and $d_h$ completes the mutually perpendicular right-handed frame [18].

$$\frac{\mathrm{d}a}{\mathrm{d}t} = 2\frac{d_r a^2 e \sin f}{h} + 2\frac{d_\theta a^2 p}{hr} \tag{3.28}$$

$$\frac{\mathrm{d}e}{\mathrm{d}t} = \frac{d_r p \sin f}{h} + \frac{d_\theta[(p+r)\cos f + re]}{h} \tag{3.29}$$

$$\frac{\mathrm{d}i}{\mathrm{d}t} = \frac{d_h r \cos(f+\omega)}{h} \tag{3.30}$$

$$\frac{\mathrm{d}\Omega}{\mathrm{d}t} = \frac{d_h r \sin(f+\omega)}{h \sin i} \tag{3.31}$$

$$\frac{\mathrm{d}\omega}{\mathrm{d}t} = -\frac{d_r p \cos f}{he} + \frac{d_\theta (p+r)\sin f}{he} - \frac{d_h r \sin(f+\omega)\cos i}{h \sin i} \tag{3.32}$$

$$\frac{\mathrm{d}M_0}{\mathrm{d}t} = d_r\left[\frac{(-2e+\cos f + e\cos^2 f)(1-e^2)}{e(1+e\cos f)na}\right] + d_\theta\left[\frac{(e^2-1)(e\cos f+2)\sin f)}{e(1+e\cos f)na}\right] \tag{3.33}$$

It is also possible to express the GVEs in terms of the mean classical orbital elements. With the mean orbital elements, and considering only to the first order $J_2$ perturbation, only the last three terms ($\Omega$, $\omega$ and $M_0$) have secular growth or long periodic growth, while the first three ($a$, $e$ and $i$) become constant [18]. This is shown in equations 3.34-3.39 where $\bar{n} = \sqrt{\frac{\mu}{\bar{a}^3}}$ and $\bar{\eta} = \sqrt{1-\bar{e}^2}$ [18].

$$\frac{\mathrm{d}\bar{a}}{\mathrm{d}t} = 0 \tag{3.34}$$

$$\frac{\mathrm{d}\bar{e}}{\mathrm{d}t} = 0 \tag{3.35}$$

$$\frac{\mathrm{d}\bar{i}}{\mathrm{d}t} = 0 \tag{3.36}$$

$$\frac{\mathrm{d}\bar{\Omega}}{\mathrm{d}t} = -\frac{3}{2}J_2\left(\frac{R_e}{\bar{p}}\right)^2 \bar{n}\cos\bar{i} \tag{3.37}$$

$$\frac{\mathrm{d}\bar{\omega}}{\mathrm{d}t} = \frac{3}{4}J_2\left(\frac{R_e}{\bar{p}}\right)^2 \bar{n}(5\cos^2\bar{i} - 1) \tag{3.38}$$

$$\frac{\mathrm{d}\bar{M}_0}{\mathrm{d}t} = \frac{3}{4}J_2\left(\frac{R_e}{\bar{p}}\right)^2 \bar{n}\bar{\eta}(3\cos^2\bar{i} - 1) \tag{3.39}$$

As was previously described, it is also possible to use a set of non-singular orbital elements, $[a, q_1, q_2, i, \Omega, \lambda]$, to describe an orbit instead of the classical orbital elements. The GVEs relating an impulsive thrust $\mathbf{\Delta V} = [\Delta V_r, \Delta V_t, \Delta V_h]^T$ in the radial, tangential and out-of-plane directions, to the variation in the non-singular orbital elements have been expressed in [22], copied here in equations 3.40-3.45, where $\gamma = \sqrt{\frac{a}{\mu}}$. In these equations all elements are the mean elements.

$$\delta i = \gamma\cos\theta\Delta V_h \tag{3.40}$$

$$\delta\Omega = (\frac{\gamma\sin\theta}{\sin i})\Delta V_h \tag{3.41}$$

$$\delta a = \frac{2}{n}\Delta V_t \tag{3.42}$$

$$\delta q_1 = \gamma\sin\theta\Delta V_r + 2\gamma\cos\theta\Delta V_t \tag{3.43}$$

$$\delta q_2 = -\gamma\cos\theta\Delta V_r + 2\gamma\sin\theta\Delta V_t \tag{3.44}$$

$$\delta\lambda = -2\gamma\Delta V_r - \gamma\sin\theta\cot i\Delta V_h \tag{3.45}$$

These equations can be used to establish an impulsive control scheme with a fixed number and location of impulsive manoeuvres, as will be detailed in Chapter 5. It has been shown that although the GVEs reference a circular orbit, it can also apply to orbits with small eccentricities [22]. Looking ahead, these GVEs form the basis for the development of the controller selected in Chapter 5.

## 3.3 Conclusions

Formation flying is a problem that requires a broad understanding of topics including GNC algorithms, orbit dynamics, and equations of relative motion. This chapter has provided background from studies into several areas from which this research will draw. The topics described here will be referenced throughout the following chapters, which will expand on the relative motion of the DelFFi formation (Chapter 4), select and characterize a control algorithm to maintain the DelFFi formation (Chapter 5) and describe the operational techniques that should or could be used by both the DelFFi team and other CubeSat Formation Flying teams in general (Chapter 6).

# Chapter 4

# Two-Line Element Based Differential Navigation

As discussed in Chapter 1, differential navigation, in which the relative state is found by subtracting the absolute states of the two satellites, is better suited to the CubeSat formation flying problem as it does not necessarily require dedicated navigation sensors. Thus, the rate and accuracy of the absolute state measurements are critical to determining the rate and accuracy of the relative state estimate. When multiple sources of data on the absolute positions are available (for example an onboard propagator estimate and a two-line element (TLE) set), it is common to use a filter to combine the data to generate a better estimate [26]. This is not considered in this study, but should be addressed in the future.

This chapter considers three aspects of the differential navigation problem. First, the characteristics of TLEs are established, with a specific emphasis on the rate and accuracy expected. The update rate of the TLEs is critical as it is necessary to receive feedback on the current state between each control input (propulsive manoeuvre). Thus, the next section addresses the passive relative motion of the two satellites to ensure the update rate of the TLEs is higher than the rate of required propulsive manoeuvres. The third section addresses the viability of the TLEs from the stand point of accuracy by performing an error analysis. Finally, conclusions are drawn in regards to the viability of TLEs in the differential navigation algorithm.

## 4.1 Two-Line Element Characteristics

TLEs are generated for each spacecraft detected by the North American Aerospace Defence Command/United States Strategic Command (NORAD/USSTRATCOM) and are made publicly available online [6], [12] at a frequency of approximately 0.6 days [64]. NORAD TLEs contain mean orbital elements which are generated using the Satellite General Perturbations 4 (SGP4) model, thus the same model must be used to estimate the osculating elements or

current position from the TLE [64]. Further information on the SGP4 algorithm is given in [11] and [65] and it is used for example in [9], [10], [12] and [75]. These authors also suggest variations to improve the accuracy of the SGP4 model, however the added complexity makes these methods undesirable unless high accuracy is required.

A sample TLE (from [11]) is shown below along with identification of each of the terms in the structure in Table 4.1.

```
#GOES 9
1 23581U 95025A 01311.43599209 -.00000094 00000-0 00000+0 0  8214
2 23581 1.1236 93.7945 0005741 214.4722 151.5103 1.00270260 23672
```

**Table 4.1:** Definition of the elements of a two-line element set (as given in [11], [65]).

| **Line 1** | | |
|---|---|---|
| Column | Value | Description |
| 01 | 1 | Line number |
| 03-07 | 23581 | Satellite number |
| 08 | U | Classification (U = Unclassified) |
| 10-11 | 95 | Last two digits of launch year |
| 12-14 | 025 | Launch number of the year |
| 15-17 | A | Piece of the launch |
| 19-20 | 01 | Epoch year (last two digits) |
| 21-32 | 311.43599209 | Epoch (day of year and portion of day) |
| 34-43 | -.00000094 | Time derivative of Mean Motion (rev/d$^2$) |
| 45-52 | 00000-0 | 2nd deriv. of Mean Motion (decimal pt. assumed) (rev/d$^3$) |
| 54-61 | 00000+0 | BSTAR drag term (decimal pt. assumed) |
| 63 | 0 | Ephemeris type |
| 65-68 | 821 | Element number |
| 69 | 4 | Checksum (modulo 10) |
| **Line 2** | | |
| Column | Value | Description |
| 01 | 2 | Line number |
| 03-07 | 23581 | Satellite number |
| 09-16 | 1.1236 | Inclination (degrees) |
| 18-25 | 93.7945 | Right Ascension of the Ascending Node (degrees) |
| 27-33 | 0005741 | Eccentricity (decimal pt. assumed) |
| 35-42 | 214.4722 | Argument of Perigee (degrees) |
| 44-51 | 151.5103 | Mean Anomaly (degrees) |
| 53-63 | 1.00270260 | Mean Motion (rev/d) |
| 64-68 | 2367 | Revolution number at epoch [revs] |
| 69 | 4 | Checksum (modulo 10) |

Note that the TLE does not contain the semi-major axis, but instead uses the mean motion (in the units revolutions per day). Also, TLEs provide a BSTAR ballistic coefficient ($B^*$)

rather than the standard ballistic coefficient B. The relationship between B* and B is given by equation 4.1, where $\rho_0$ is the TLE standard density of $2.461x10^{-8}$ kg/m$^3$ and the ballistic coefficient ($B$) was given in equation 3.5.

$$B^* = \frac{1}{2}B\rho_0 \tag{4.1}$$

These parameters can be converted into the classical mean orbital elements (or another set of mean orbital elements if desired) as discussed in the Delfi-n3Xt Onboard Navigation Function technical note [64].

Using data from CanX-2, PRISMA and PROBA-2, Kahr's team showed that TLEs for small satellites (including nanosatellites) have instantaneous absolute errors of approximately 2 km for the along-track direction and 1 km for the cross track and radial directions for up to five days of propagation using the SGP4 propagator [6]. Another team used data from the PSSCT-2 spacecraft to determine that for CubeSats (in this case a two-unit) below 350 km altitude, the position determination errors may be reduced below 1 km [12]. In this case, the team post-processed the TLE and GPS data to determine the TLE accuracy (accuracy of the orbital elements from a TLE) by propagating forward for half the time between two TLEs and propagating backward for the other half of the time [12].

Since TLEs provide the mean orbital elements, a team led by Wang looked at the accuracy of orbits propagated from TLEs [76]. They found that for low eccentricity (0.00 - 0.02) orbits with a perigee between 300 and 400 km altitude the total error was 3.50 km with one day of propagation. The majority of this error is in the along-track direction, with a relative proportion of 19:1:1 for one day of propagation of 300-400 km altitude orbits. Similarly, for orbits with a perigee between 400 km - 500 km and eccentricity between 0.00 and 0.02, the error was 2.34 km. For differential TLEs the accuracy is improved (due to the cancelling of common errors), with the along-track error expected to be below 1 km [1]. For close formations, TLEs are expected to be insufficient on their own [77]. Some teams have suggested augmenting the TLEs with sparse GPS data in order to increase the accuracy, but typically these teams select dedicated ranging sensors instead [9].

The accuracy of mean orbital elements from TLEs is severely impacted by any propulsive manoeuvres, with these impacts wearing off after five days (based on PROBA-2 data) [6]. This is attributed to the fact that the mean orbital elements are found by averaging several radar measurements taken over some number of days using the SGP4 algorithm. The number of days the averaging occurs over depends on the satellite's orbit and size, but for PROBA-2, the data points to a five day averaging since beyond that the effects of a manoeuvre no longer contaminate the averaging [6]. To put this another way, if a manoeuvre has occurred within the previous five days, the NORAD TLE will average together measurements of the satellite from the pre-manoeuvre orbit with measurements of the post-manoeuvre orbit in generating the mean orbit elements. This means that although a new TLE will still be published online every 0.6 days [64], if a manoeuvre has occurred within the previous five days of the TLEs epoch, the orbital elements of that TLE will not be a valid representation of the satellite's orbit. This issue can be addressed in two ways: either the increased error due to

averaging old positions can be accepted, or TLEs received within five days of a manoeuvre should be discarded. Since the magnitude of the error is quite large (as shown in [6]) and the error on a valid TLE is already large, the second option is preferred for the DelFFi mission.

As alluded to earlier, the SGP4 algorithm is used to generate the mean orbital elements in the TLE. In determining the mean orbital elements from the osculating orbital elements several secular, short- and long-periodic variations are removed. The effects that are removed are the secular effects of $J_2$, $J_3$ and $J_2^2$, the long-period effects of $J_3$ and the short periodic effects of $J_2$, as well as atmospheric drag [9], [64]. The atmospheric drag is modelled with a "power density function assuming a non-rotating spherical atmosphere" [64]. To reconstruct the orbit these variations must be added back in the same way, thus the SGP4 algorithm must also be used to propagate the orbit from a TLE [64].

NORAD also developed the SGP8 model which builds upon the SGP4 algorithm. Although the TLEs are built using SGP4, predictions can be made using the SGP8 algorithm [65]. Both algorithms use the same atmospheric and gravitational models, but the processing required by the SGP8 is more intensive [65], which makes it less popular onboard satellites.

To summarize, using TLEs for navigation requires no additional equipment or cost to the mission which is very beneficial. However, they must be uploaded to the satellites from the ground, have a low update rate (compared to onboard sensors) and may not provide sufficient accuracy depending on the mission requirements. It is for these reasons that the performance of TLEs is of interest and will be compared to that of the (traditional) GPS solution.

## 4.2 Passive Relative Motion

Based on the findings above, only TLEs that have been received five or more days after a manoeuvre are valid estimates of the satellite's current position. As such, the update rate of TLEs can only be considered to be once per five days for navigation, assuming manoeuvres are planned as frequently as possible to minimize their magnitudes (this will be elaborated on later in this Chapter and Chapter 6). Further, TLEs can only be transmitted (uplinked) to the satellites during a pass over Delft (the primary ground station), which will likely delay the use of the TLE by the onboard controller by up to one day. This one day is due to the fact that while passes over Delft are estimated to occur more than once per day, the passes are expected to occur sequentially (rather than spread over the day) (as per conversation with J. Guo, July 14, 2015), so if the TLE is only received at the ground station after that day's passes have completed, the upload must wait until the following day. This means that manoeuvres can be planned no more frequently than once every six days. To put this another way, the passive relative motion of the satellites must not exceed the control window for at least six days.

The control window used in this section is 10% of the nominal separation distance. This is 100 km control window on a 1000 km nominal separation distance. The formation is an along-track formation, meaning that the formation is initialized with the satellites in the same mean

orbit with an offset in the along-track direction by 1000 km. The separation distance will be used to describe the relative motion of the satellites throughout this section. This separation distance is the total distance between the satellites (in the three dimensional sense). It is calculated based on the difference in instantaneous position vectors of the chief and deputy satellites. To see only the variation in the along-track direction, the differential true anomaly or true latitude should be used instead.

This section will first describe the effects of the two most critical perturbations that affect the relative motion of the satellites: Differential Drag and Differential Gravitational Perturbations. To further understand how the ADCS affects the Autonomous Formation Flying (AFF) payload, a study of the impact of the attitude pointing error is performed. This is followed by a detailed description of the impact of the passive relative motion of the QB50 Nominal and Elliptical orbits. Then, various orbits will be modelled with these perturbations to determine how the initial orbital elements effect the passive relative motion of the satellites. Finally, the sensitivity of the system to matching the mean orbital elements at acquisition is determined. This study will allow the identification of a set of orbits that make TLE-based navigation viable.

Throughout this section, a numerical propagator has been used to model the relative motion of the satellites. This propagator integrates the instantaneous velocity and acceleration terms to determine the position and velocity at the next time step. The propagator is an existing and previously validated model from the Space Systems Engineering Department at TU Delft (as per conversation with P. Sundaramoorthy, June 2015). Several results from this propagator have been compared to the results from a propagator used by J. Chu (PhD Candidate) as well, for further validation. This method has been chosen over propagation using an analytical system of equations (such as the HCW equations from Chapter 3) as it can be used to propagate any orbit without losing accuracy due to linearization errors. For further details on the propagator see Appendix D.

### 4.2.1 Differential Gravitational Perturbation

As described in Chapter 3, due to the non-spherical nature of the Earth, the motion of a satellite is disturbed from its reference trajectory. This disturbance is relatively small compared to the differential drag in LEO, however, its impact on the relative motion increases when the satellites are more widely-spaced [69]. The gravitational potential field is modelled by dividing the Earth into zones and sectors and different levels of accuracy can be considered by considering fewer or more number of divisions. The largest contribution to the disturbance comes from the $J_2$ term and it (alone) is often used to model the gravitational disturbance on a satellite. The $J_2$ term is associated with the oblateness of the Earth (a sphere that has been slightly flattened). Another typical level is the $J_{20,20}$ which considers many more harmonics.

To determine whether the system must be modelled to a high number of geopotential terms or if $J_2$ will suffice, the following two figures show the passive relative motion of two satellites in the nominal QB50 orbit: nearly-circular ($e$ = 0.001) 380 km altitude ($a$ = 6758 km), sun-synchronous orbit with a RAAN of -15° ($\Omega$ = -15°). The true latitudes at initialization

place the chief at the equator and the deputy north of the equator by the 1000 km separation distance. They are modelled with moderate differential drag where both satellites have the same coefficient of drag (2.3 [70]), the same mass (3.64 kg [52]), the deputy satellite has a $2^\circ$ angular pointing error in the pitch direction (cross sectional area of 0.01367 $m^2$) and the chief is aligned with the velocity vector (cross sectional area of 0.01 $m^2$). Figure 4.1 shows the motion modelled with only $J_2$ and Figure 4.2 shows motion modelled with parameters up to $J_{20,20}$.

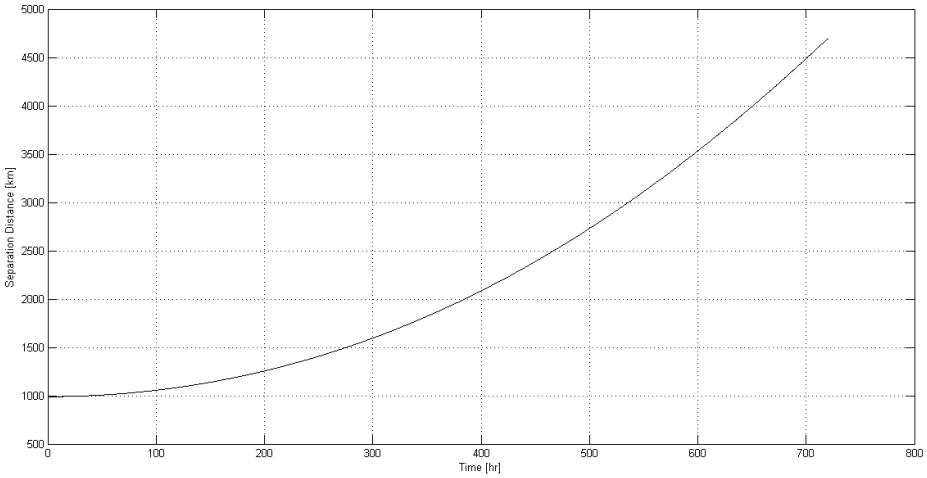



**Figure 4.1:** Passive Relative Motion of a formation subjected to moderate differential drag and the $J_2$ perturbation, initialized in a 380 km altitude circular orbit (e = 0.001) with the Chief at the equator and the Deputy ahead by the 1000 km separation distance.

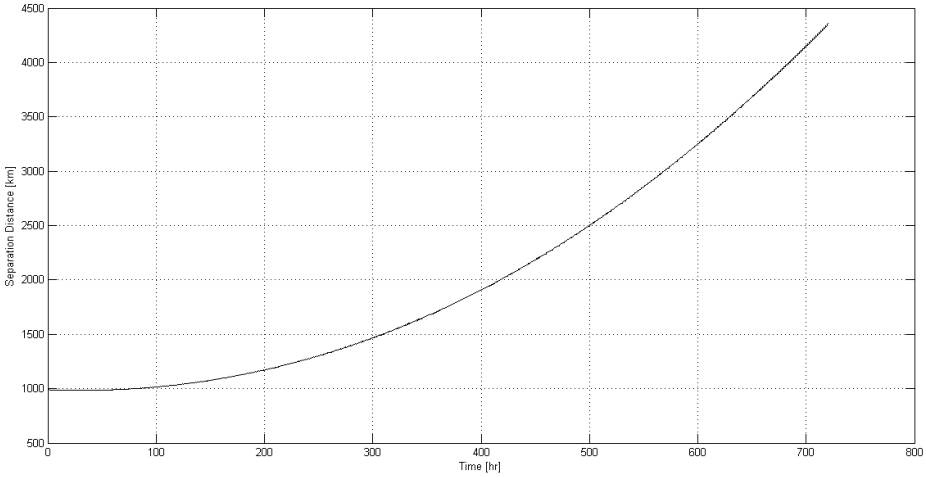


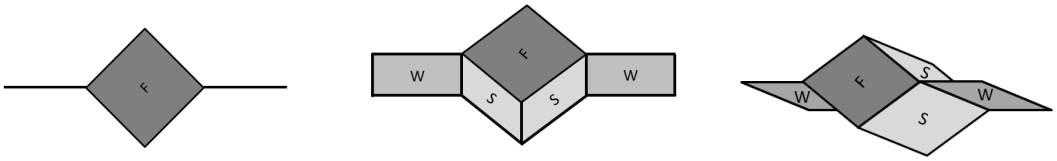

**Figure 4.2:** Passive Relative Motion of a formation subjected to moderate differential drag and the $J_{20,20}$ perturbation, initialized in a 380 km altitude circular orbit (e = 0.001) with the Chief at the equator and the Deputy ahead by the 1000 km separation distance.

In comparing these two graphs it can be seen that the time to leave the control window (further apart than 1100 km) in the two cases is 164 hours ($J_{20,20}$) and 129 hours ($J_2$). This shows that the higher order terms have significant impact on the modelling of the passive relative motion of the satellites, and they should be included in determining the time spent within the control window. Thus the following analyses will use $J_{20,20}$ unless otherwise stated.

In order to create this model, a numerical propagator was used (see Appendix D). This propagator is initialized with the osculating elements of two satellites that correspond to the same mean orbit but with a 1000 km separation between them. Thus, as can be seen in many of the plots, the initial separation distance is not exactly 1000 km since the instantaneous separation will always be different than the unperturbed separation distance. The difference between the osculating initial separation distance and the desired 1000 km separation distance is in all cases less than 10 km which is less than the control accuracy of the satellite. This mimics the effect of being unable to control the satellite to the exact desired initial conditions without losing an understanding of the relative motion trends.

### 4.2.2 Differential Drag

As described in Section 3.2.3, differential drag is especially significant for satellites in low Earth orbit. In addition to the differences in mass (due to variations in propellant consumption), there may also be a difference in frontal area between the satellites due to pointing errors of up to 10° [5]. The frontal area for a satellite in the nominal position, with a small pitch angular pointing error, and with a small pitch and a small yaw angular pointing error can be seen in Figure 4.3.




(a) Nominal cross sectional area (b) Cross sectional area with a small pitch (c) Cross sectional area with a small pitch and a small yaw

**Figure 4.3:** Cross sectional area for various satellite positions. Visible faces of the satellite are labelled 'F' for front, 'S' for side and 'W' for wing.

Clearly, the cross sectional area for a combined pitch and yaw is smaller than only a pitch. This is because as the satellite experiences a yaw, the wing area seen from the front shadows the side areas behind, reducing the front cross sectional area. Since the angular pointing error is restricted to 10° from nominal when the satellite is in ADCS's Velocity Pointing Mode, the worst case will be when the satellite is has an angular pointing error of 10° in the pitch. The Formation Flying Package (FFP) will operate while the satellite is in Thrust Vector Control Mode, which further controls the attitude pointing error to 2°. This will be considered for the moderate differential drag case. To calculate the nominal cross sectional area and the cross

sectional area for a pitched satellite, the following equations are used.

$$A_{nom} = l^2 \tag{4.2}$$

$$A_{pitch} = l^2 \cos\psi + 2lL \sin\psi + \sqrt{2}lL \sin\psi \tag{4.3}$$

In these equations the small dimension of the satellite (10 cm) is represented by $l$, the long dimension (30 cm) by $L$ and the pitch angle by $\psi$. In the pitch equation, the first term corresponds to the front face (labelled F in Figure 4.3 b), the second term corresponds to the wings (labelled W) and the third term corresponds to the side panels (labelled S). The cross sectional areas work out to $A_{nom} = 0.01$ m$^2$, $A_{10pitch} = 0.0276$ m$^2$ for a 10° pitch angle, and $A_{2pitch} = 0.01357$ m$^2$ for a 2° pitch angle.

For the DelFFi satellites, the passive relative motion including differential drag and gravitational perturbations (to $J_{20,20}$) is plotted for several cases. First, the effect of fuel balancing is addressed. Consider the moderate angular pointing error of 2° on the chief satellite, while the deputy points in the nominal direction, with the chief satellite having an empty propellant tank (mass of 3.59 kg) and the deputy having a full propellant tank (mass of 3.64 kg). This represents a case where all prior manoeuvres have been performed by one satellite and now the roles of chief and deputy are exchanged to leave the deputy to perform formation maintenance manoeuvres. This is shown in Figure 4.4.

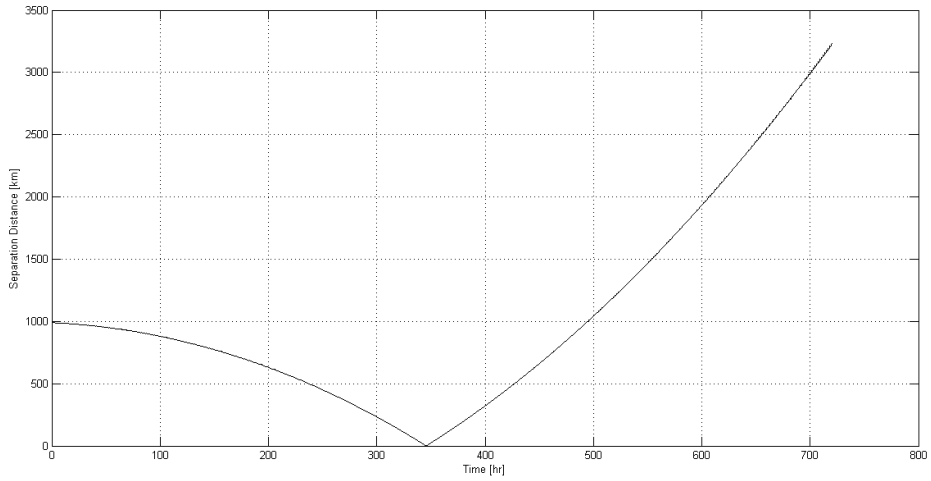



**Figure 4.4:** Passive Relative Motion of a formation subjected to $J_{20,20}$ and differential drag, initialized in a 380 km altitude circular orbit (e = 0.001) with the Chief at the equator and the Deputy ahead by the 1000 km separation distance. The Chief has a mass of 3.59 kg (empty tank), and is pointing 2° off of the velocity vector in the pitch direction. The Deputy has a mass of 3.64 kg (full tank), and is pointing along the velocity vector. Both satellites have a coefficient of drag of 2.3.

Since fuel balancing can be used to maintain a very small difference in mass between the two satellites (see Chapter 6) and Figure 4.5, which demonstrates the effect this would have on the same case as above except both satellites have a 50% full propellant tank (mass of 3.615 kg).

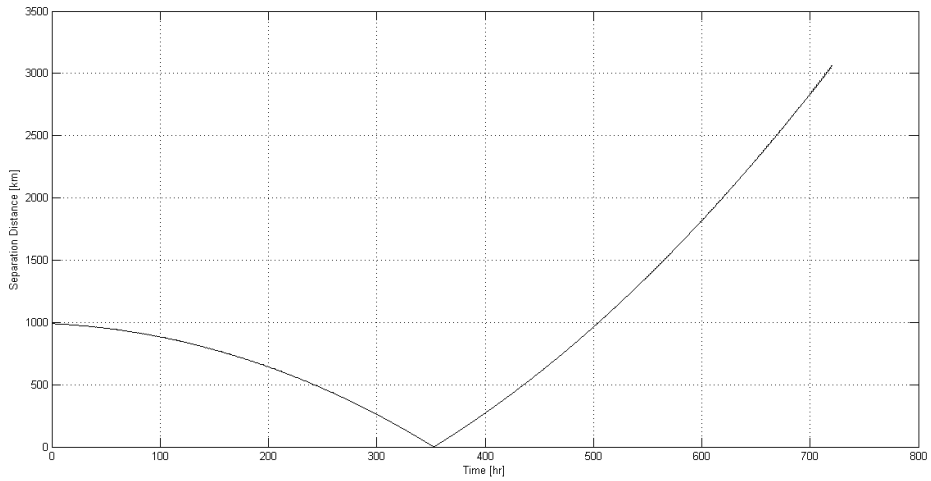



**Figure 4.5:** Passive Relative Motion of a formation subjected to $J_{20,20}$ and differential drag, initialized in a 380 km altitude circular orbit (e = 0.001) with the Chief at the equator and the Deputy ahead by the 1000 km separation distance. The Chief has a mass of 3.615 kg (half full tank), and is pointing 2° off of the velocity vector in the pitch direction. The Deputy has a mass of 3.615 kg (half full tank), and is pointing along the velocity vector. Both satellites have a coefficient of drag of 2.3.

The difference in time to leave the control window (approach closer than 900 km) is approximately one orbit (88.3 hours without fuel balancing and 89.8 hours with fuel balancing). Since the mass of the fuel is so low (50 g) compared to the total mass of the satellite, fuel balancing, while beneficial, does not have a very strong impact.

Next, the effect of the angular pointing error is considered. Figure 4.6 can be compared with Figure 4.5 as the only change from here is the angular pointing error of the chief has been increased to 10°.

In this case, the time the satellites remain in their control window is only 46.7 hours - a significant decrease from the 2° angular pointing error case. Differential drag at this low altitude is clearly very strongly impacted by the cross sectional area of the satellites, and keeping that area constant (by maintaining the pointing direction as accurately as possible) is very important to the formation flying problem.

It is also worth noting that the relative motion is impacted by which satellite incurs the attitude pointing error. If the chief is precisely controlled to the velocity vector, while the deputy is offset by a 2° angle in pitch, the relative motion can be seen in Figure 4.7.

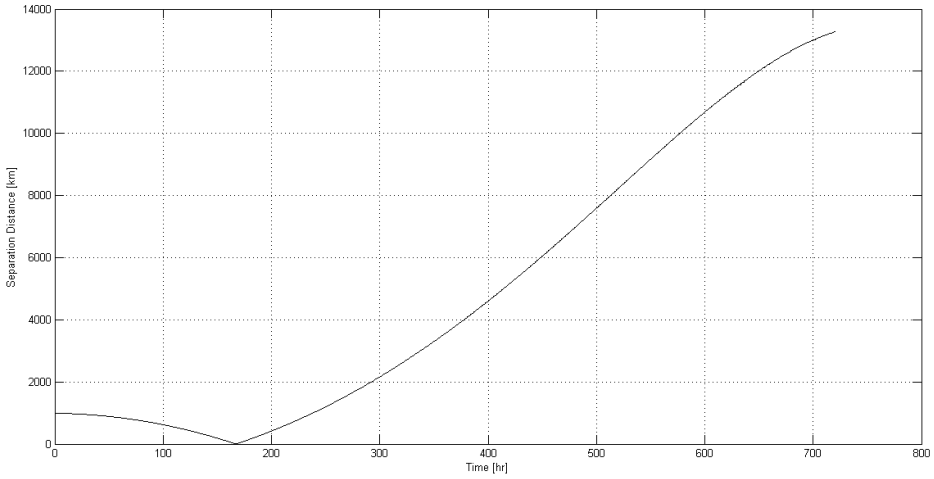



**Figure 4.6:** Passive Relative Motion of a formation subjected to $J_{20,20}$ and differential drag, initialized in a 380 km altitude circular orbit (e = 0.001) with the Chief at the equator and the Deputy ahead by the 1000 km separation distance. The Chief has a mass of 3.615 kg (half full tank), and is pointing 10° off of the velocity vector in the pitch direction. The Deputy has a mass of 3.615 kg (half full tank), and is pointing along the velocity vector. Both satellites have a coefficient of drag of 2.3.

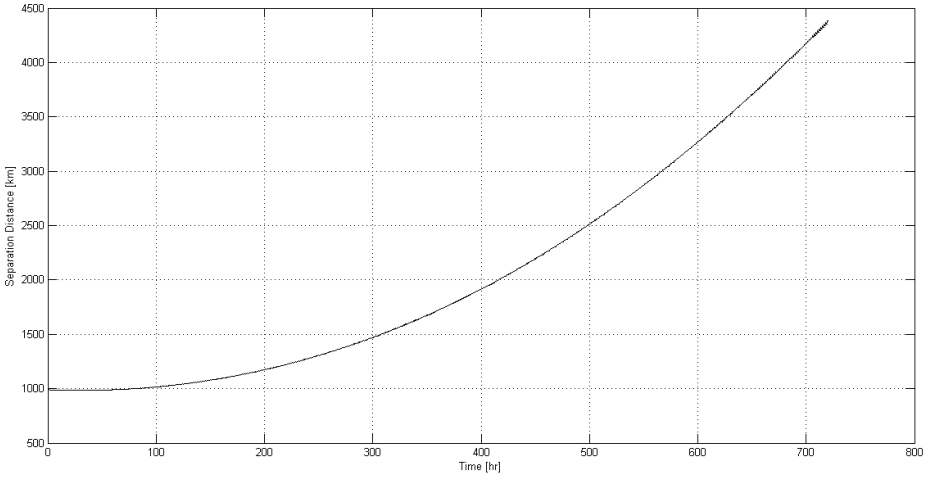



**Figure 4.7:** Passive Relative Motion of a formation subjected to $J_{20,20}$ and differential drag, initialized in a 380 km altitude circular orbit (e = 0.001) with the Chief at the equator and the Deputy ahead by the 1000 km separation distance. The Chief has a mass of 3.615 kg (half full tank), and is pointing along the velocity vector. The Deputy has a mass of 3.615 kg (half full tank), and is pointing 2° off of the velocity vector in the pitch direction. Both satellites have a coefficient of drag of 2.3.

Again comparing this figure to Figure 4.5, it can be seen that the separation distance now

grows over time and at a slower rate than in Figure 4.5. The time to leave the control window (exceed 1100 km separation) is now 164 hours from Figure 4.7. This is significantly longer than the case where the satellites converge, and demonstrates that a chief satellite with high attitude accuracy is extremely desirable. Since the chief satellite is by definition the non-active (passive) satellite of the pair, it is expected to be more stable in terms of attitude pointing as it will not be required to perform manoeuvres that cause high disturbance torques. Thus, for future analyses the case shown in Figure 4.7 will be used as the reference. This will be further discussed in Chapter 6.

The QB50 project has also considered a 400 km by 600 km elliptical orbit (see Chapter 2). If the parameters used to generate Figure 4.7 are applied to this orbit, Figure 4.8 is produced.

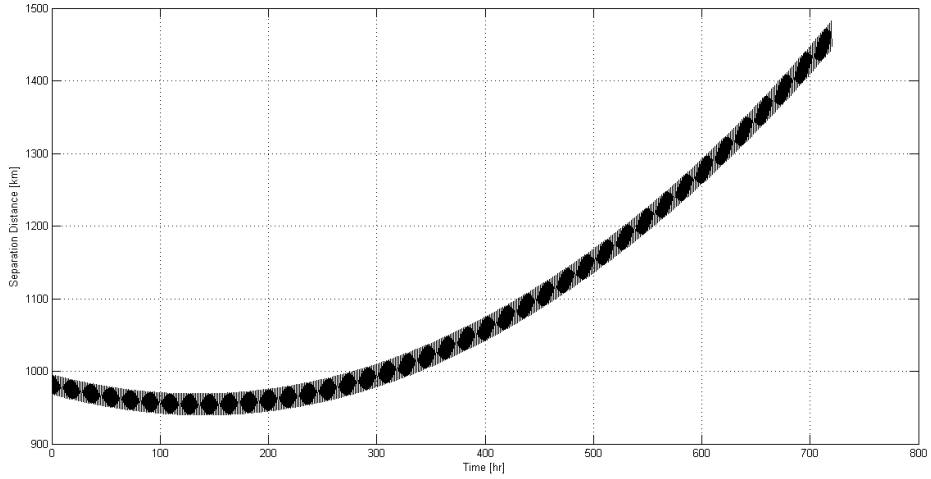



**Figure 4.8:** Passive Relative Motion of a formation subjected to $J_{20,20}$ and differential drag, initialized in a 400 km by 600 km altitude orbit with the Chief at the equator and the Deputy ahead by the 1000 km separation distance. The Chief has a mass of 3.615 kg (half full tank), and is pointing along the velocity vector. The Deputy has a mass of 3.615 kg (half full tank), and is pointing 2° off of the velocity vector in the pitch direction. Both satellites have a coefficient of drag of 2.3.

At this higher altitude, the time before the control window is breached is 430.8 hours. Note that this increase is due to the altitude not the eccentricity. If the same analysis were performed for a circular orbit with the same orbital parameters (altitude of 500 km), the time would be further increased to 543.6 hours.

This method of assessing the effect of differential drag is useful for this level of feasibility study when the actual attitude pointing error is not well known. The attitude pointing error will not be a constant value throughout the mission, rather it will vary from instant to instant on both satellites. It is more likely that the difference in attitude pointing error between the two satellites will be much smaller than the maximum variation (since generally both satellites will be offset from the nominal attitude by some amount between zero and the maximum attitude pointing error, their difference will typically be smaller than the maximum attitude

pointing error). This will decrease the impact of differential drag on the satellites allowing them to remain within the control window for longer than has been estimated here. A more detailed statistical analysis is recommended for future study.

It is clear that in some situations, differential drag can be used as an advantage, however to do so, it is necessary to control the angular pointing error using the ADCS as recommended by [45] for example. This possibility should be considered by the ADCS team, however until that has been assessed it is better to minimize the differential drag as far as possible. The most convenient way to minimize the effect of differential drag is to reduce the difference in ballistic coefficient between the satellites. This means ensuring that the propellant the satellites consumes is evenly distributed between them over their lifetime, and that the pointing error is minimized or averages out to no difference in area between the two satellites.

### 4.2.3 Impact of the Attitude Pointing Error

Since the attitude pointing error is one of the major factors that affect the time passively spent within the control window, a detailed analysis of its effect is performed here. This analysis uses the nominal QB50 orbit (380 km near circular) along with a $J_{20,20}$ model to estimate the duration spent in the control window for different combinations of attitude pointing error of the two satellites. First, the attitude pointing error of the deputy satellite (the front/lead satellite) is varied between 0° (nominal) and 10° (requirement for ADCS velocity pointing mode) for three cases of chief (rear satellite) attitude pointing error: 0° (nominal), 2° (ADCS thrust vector control mode requirement), and 10° (ADCS vector pointing mode requirement. The results are shown in Figure 4.9. This same analysis is then repeated with the chief (rear satellite) attitude pointing error varied between 0° and 10° for three cases of the deputy attitude pointing error: 0°, 2°, and 10°. This is shown in Figure 4.10.

These figures show that it is not actually optimal for the attitudes of the two satellites to match exactly. An approximate 1° offset would actually be preferred as it compensates for the gravitational perturbations. Looking at Figure 4.9 first, it can be seen that if the chief has a 2° attitude pointing error, the deputy should have between approximately 1.5° and 4.5° attitude pointing error in order to stay within its control window for 6 days (144 hours). This same offset (except in the opposite direction due to the inversion of the satellites' positions) is seen in Figure 4.10, where the chief should remain between 0 and 2.5° of the velocity vector when the deputy has an attitude pointing error of 2°. Note that the angles plotted here are pitch angles relative to the velocity vector and whether the angle is positive or negative is equivalent as the frontal area is symmetric in the pitch direction.

These plots show that the requirement on the satellites attitudes should not be absolute, but rather one satellite should be based upon the other. In other words, rather than saying that both satellites must have a pointing error under 2° relative to the velocity vector, it may be preferable to have the chief's attitude pointing error be within a 2° window around the deputy's attitude. This would allow one satellite to have a much wider allowable attitude pointing error (relative to the velocity vector) while the other has a strict attitude pointing

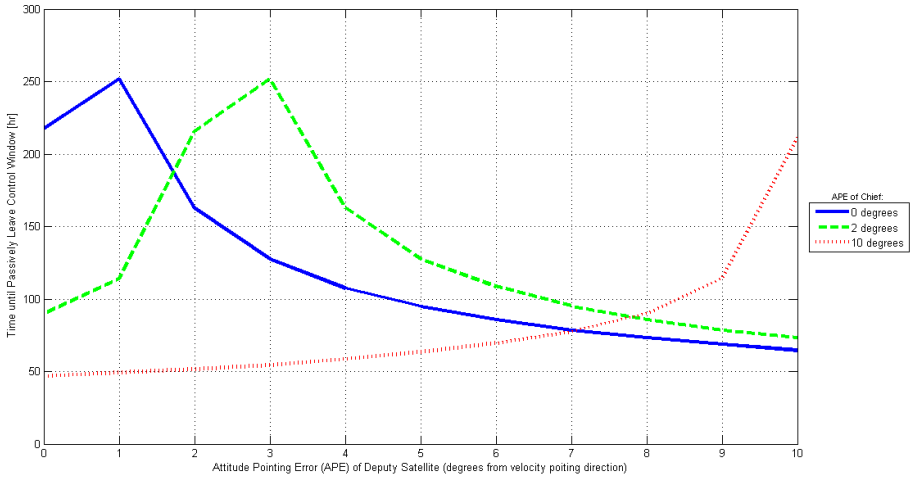



**Figure 4.9:** Effect of Attitude Pointing Error Deputy (Front Satellite) due to Passive Relative Motion of a formation subjected to $J_{20,20}$ and differential drag, initialized in a 380 km altitude circular orbit (e = 0.001) with the Chief at the equator and the Deputy ahead by the 1000 km separation distance. The Chief has a mass of 3.615 kg (half full tank), and is pointing along the velocity vector. The Deputy has a mass of 3.615 kg (half full tank) and both satellites have a coefficient of drag of 2.3.

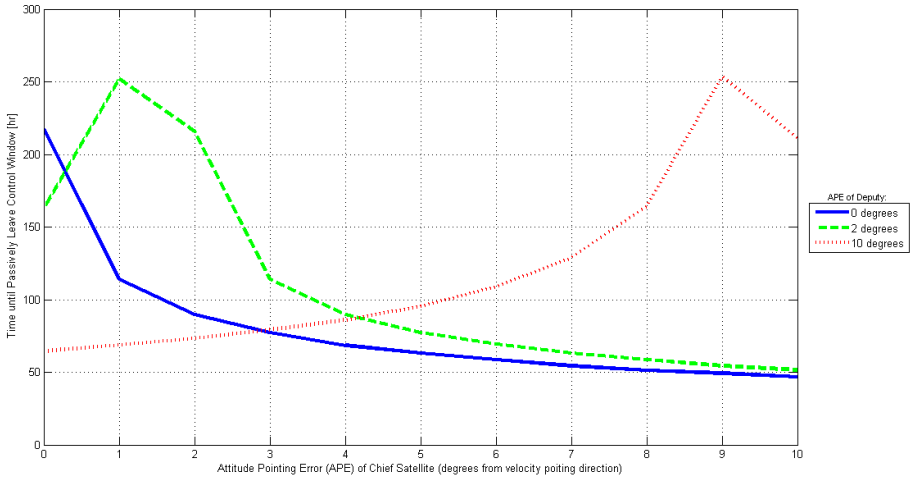



**Figure 4.10:** Effect of Attitude Pointing Error Chief (Rear Satellite) due to Passive Relative Motion of a formation subjected to $J_{20,20}$ and differential drag, initialized in a 380 km altitude circular orbit (e = 0.001) with the Chief at the equator and the Deputy ahead by the 1000 km separation distance. The Chief has a mass of 3.615 kg (half full tank), and is pointing along the velocity vector. The Deputy has a mass of 3.615 kg (half full tank) and both satellites have a coefficient of drag of 2.3.

error that is based off the measurement of the attitude from the other satellite. This is advantageous as the deputy satellite which performs the manoeuvres will experience disturbance torques that affect the attitude, while the chief can be pointed with more stability without the

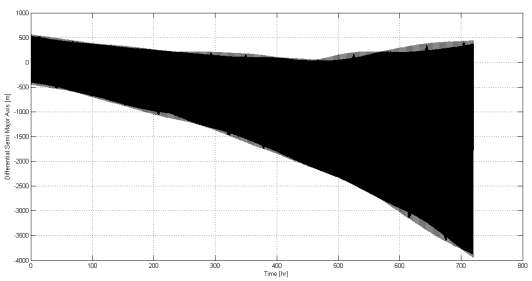
propulsive manoeuvres to account for. This would only be possible if there were feedback of the attitude between the two satellites such as through the Intersatellite Link (ISL). Feedback through the ground stations would be much too rare for this to be effective. Since the ISL has not be studied for DelFFi yet, it is not possible to rely on such an effect, and instead the attitude pointing error requirements must be formed in an absolute sense (with respect to each satellite's velocity vector). This has the further impact that the thrust vector during manoeuvres will be controlled more precisely which will be valuable as will be discussed in Chapter 5.

Since the ADCS will attempt to drive the attitude pointing error to 0° for both satellites, it is not possible to accept the 10° attitude pointing error requirement for either the chief or deputy as it would only be acceptable for maintaining the control window if the other satellite were between approximately 8° and 12° from the velocity vector. This is the same for the 2° attitude pointing error. The 2° offset for the chief allows the deputy to maintain the control window for six days only if its attitude is between approximately 1.5° and 4.5°. Noting that the width of the peaks on Figure 4.9 are -0.5° to +1.5° around the attitude pointing error of the chief, while the peaks on Figure 4.10 are between approximately -2.5° and +0.5°. In order to ensure that the peaks (portions where the time in the control window is larger than six days) on both of the graphs included the nominal pointing direction (velocity vector), the maximum attitude pointing error allowed would be 0.5°. This desired attitude pointing error requirement should be provided to the ADCS team for further analysis and iteration. Until such time, the case where the chief is aligned with the velocity vector while the deputy has a 2° offset will be used for analysis as it is the most representative of the thrust vector control mode where only the deputy is reacting to disturbance torques from the thruster.

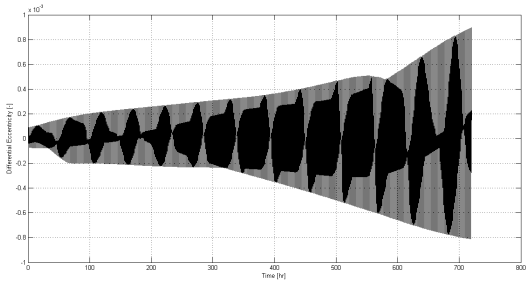
Note that when the deputy (front satellite) has the larger attitude pointing error, its semi-major axis will decrease more quickly than the chief, causing it to increase its mean motion ($n = \sqrt{\frac{\mu}{a^3}}$) and separate further from the chief. On the other hand when the chief has the larger attitude pointing error, its mean motion increases, causing it to catch up (reduce separation distance) to the deputy satellite. This effect is independent of which satellite is active. Since the chief has so far been arbitrarily labelled the rear satellite, it is tempting to claim that the rear satellite will always have a smaller attitude pointing error (no manoeuvres means fewer disturbance torques that cause attitude pointing errors), and thus the satellites will always drift apart. However, assuming a fuel balancing tactic is applied, the roles will reverse from time to time (see Chapter 6), making the front satellite the chief, with the lower attitude pointing error, causing the satellites to move closer together. Since however, when the satellite move together they will have different mean semi-major axes (as a result of the differential drag), the risk of collision remains low. These differential drag effects can be used to maintain the formation (as recommended in [45]) or even to rearrange the positions of the two satellites within the formation.

### 4.2.4 Passive Relative Motion in the QB50 Nominal Orbit

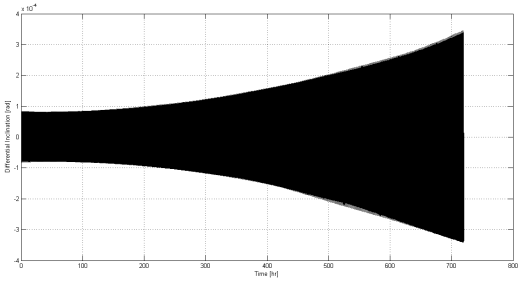
Using the scenario identified for Figure 4.7, the following six plots (Figure 4.11) show the variation of each of the differential osculating orbital elements for the formation. This scenario was the one with the QB50 Nominal orbit with both satellites with the same mass (half full propellant tank) and the deputy satellite offset from the velocity pointing direction by a 2° angular pointing error.



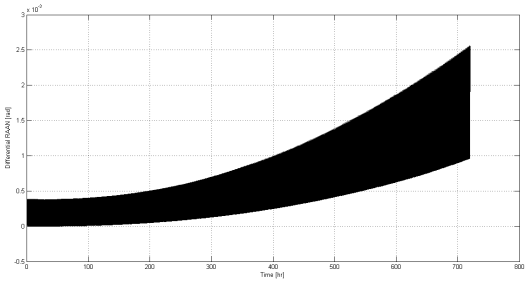
(a) Differential Semi-Major Axis



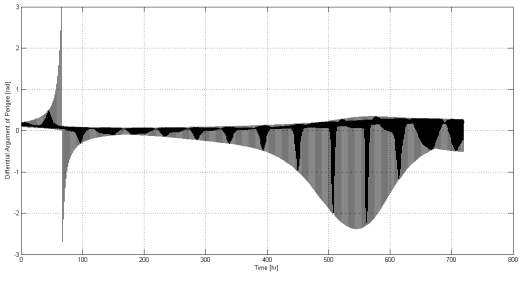
(b) Differential Eccentricity



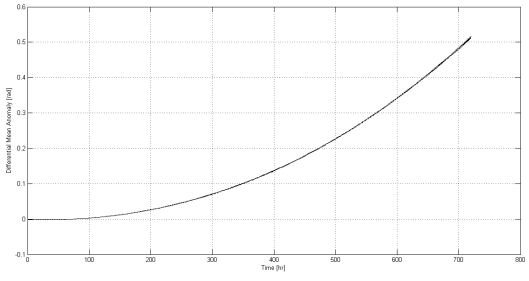
(c) Differential Inclination



(d) Differential RAAN



(e) Differential Argument of Perigee



(f) Differential Mean Anomaly

**Figure 4.11:** Passive Relative Motion of a formation subjected to $J_{20,20}$ and differential drag, initialized in a 380 km altitude circular orbit (e = 0.001) with the Chief at the equator and the Deputy ahead by the 1000 km separation distance. The Chief has a mass of 3.615 kg (half full tank), and is pointing along the velocity vector. The Deputy has a mass of 3.615 kg (half full tank), and is pointing 2° off of the velocity vector in the pitch direction. Both satellites have a coefficient of drag of 2.3.

The first and most impactful differential orbital element is the differential semi-major axis. Within six days, the satellites will have differential semi-major axes of nearly 800 m. Recall that these plots show the osculating orbital elements, not the mean, thus, the drag acceleration will be different for the two satellites when the magnitude of their semi-major axes are so different. A simplified expression for the along-track drift for a circular orbit due to differential drag ($AT_{dd}$) over a number of orbits ($N_{orbits}$) has been determined by Sundaramoorthy in [78] and is given below in Equation 4.4.

$$AT_{dd} = 3\pi * \delta a * N_{orbs} \tag{4.4}$$

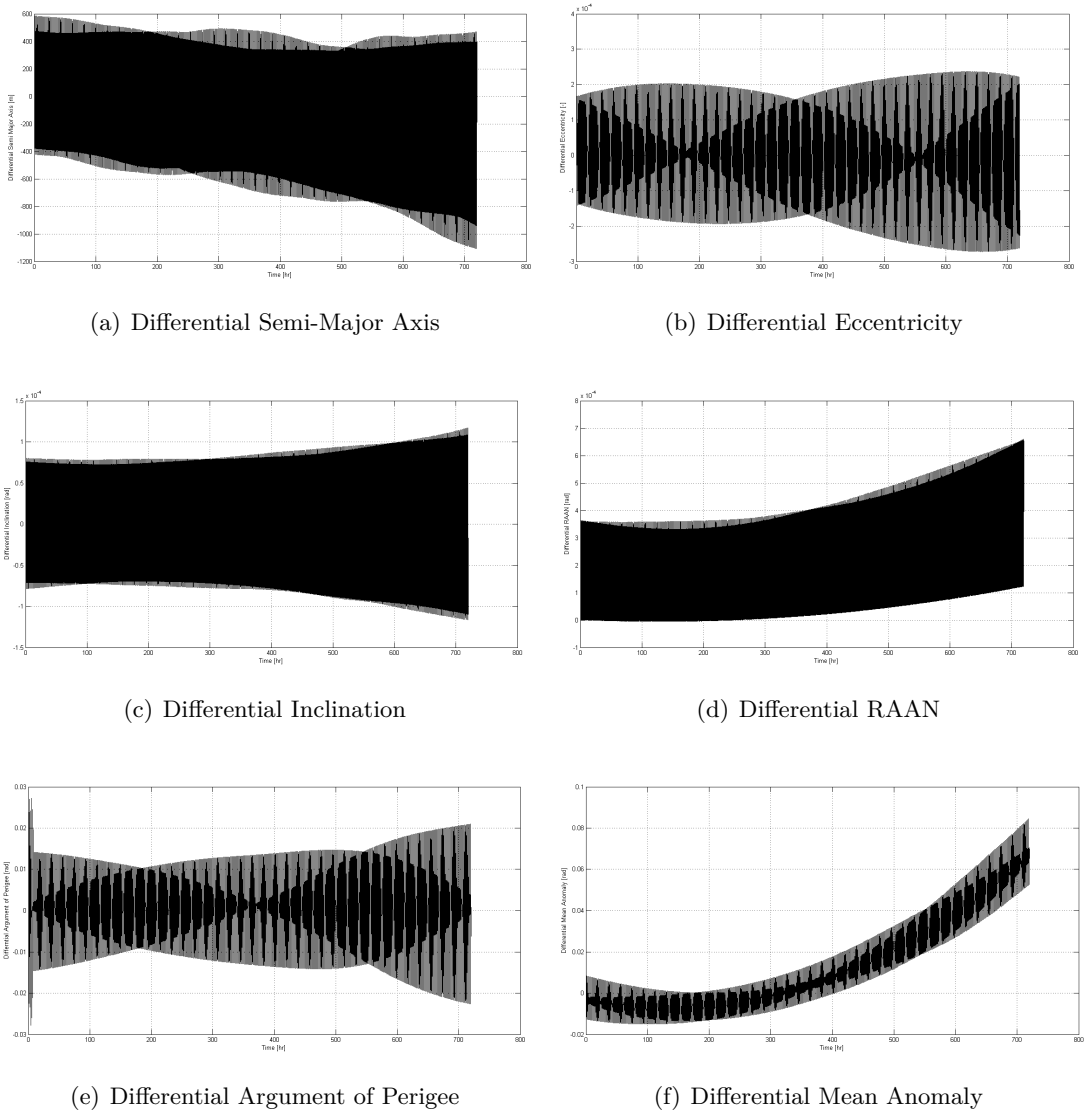
This calculation uses the mean differential mean semi-major axis $\delta a$. If the semi-major axis were to be 800 m different on average, the along-track separation would change by 7.5 km each orbit. For this orbit, the differential mean semi-major axis after six days is approximately -500 m, which corresponds to 4.7 km along-track separation per orbit. These are only rough estimation, but it makes it clear that the differential semi-major axis has a large impact on the passive relative motion.

In addition to the semi-major axis, the second figure - differential eccentricity - determines the size and shape of the orbit. Even after only six days the variation in eccentricity between the is almost 0.0003. Considering that the nominal eccentricity is 0.001, this difference is quite large (nearly a third of the nominal eccentricity magnitude). This effect is due to the differential drag acting on the satellites that causes instantaneously lower altitudes to decrease further than instantaneously higher altitudes around the orbit, increasing the elliptical quality of the orbit. It has already been established that more elliptical orbits spend less time within their control windows than circular orbits in Section 4.2.2 so the differential eccentricity cannot be ignored.

Over the six day passive relative motion requirement window, the variation in inclination (less than 0.006°) and RAAN (less than 0.03°) are small relative to their magnitudes. The variation in the argument of perigee is more noticeable. This is due to the way the argument of perigee is calculated for nearly circular orbits and is a remnant of the inaccuracies in the conversion from the propagated position and velocity vectors to orbital elements. This is also why for circular orbits, the method of calculating the sixth orbital element (mean anomaly) is to use the argument of latitude, which always measures the position of the satellite relative to the equator rather than the argument of perigee. This differential mean anomaly (calculated from the arguments of latitude) shows the same trend as the separation distance (which is calculated from the position vectors) as is expected. The plots for the inclination and RAAN show that the out of plane errors due to differential drag and gravitation perturbations are very small, and correction of them even over 30 days may not be necessary for the DelFFi mission.

### 4.2.5 Passive Relative Motion in the QB50 Elliptical Orbit

Using the scenario identified for Figure 4.8, the following six plots (Figure 4.12) show the variation of each of the differential osculating orbital elements for the formation. This scenario was the one with the QB50 Elliptical orbit with both satellites with the same mass (half full propellant tank) and the deputy satellite offset from the velocity pointing direction by a 2° angular pointing error.



(a) Differential Semi-Major Axis

(b) Differential Eccentricity

(c) Differential Inclination

(d) Differential RAAN

(e) Differential Argument of Perigee

(f) Differential Mean Anomaly

**Figure 4.12:** Passive Relative Motion of a formation subjected to $J_{20,20}$ and differential drag, initialized in a 400 km by 600 km altitude orbit with the Chief at the equator and the Deputy ahead by the 1000 km separation distance. The Chief has a mass of 3.615 kg (half full tank), and is pointing along the velocity vector. The Deputy has a mass of 3.615 kg (half full tank), and is pointing 2° off of the velocity vector in the pitch direction. Both satellites have a coefficient of drag of 2.3.

These figures show a similar trend to those of the QB50 Nominal orbit. The differential semi-major axis and differential eccentricity plots show significant increases over time, while the RAAN and inclination do not. The argument of perigee is better calculated for the elliptical case so the differential argument of perigee is small (less than 0.8° over the first six days). Finally, the Differential Mean Anomaly shows the same trend as the separation distance, as expected. One note of interest when comparing these plots to the ones from the QB50 Nominal orbit is that the differential semi-major axis is smaller for the elliptical orbit than for the circular one. However, this is not a fair comparison as the QB50 Elliptical orbit, also has a higher mean semi-major axis (500 km altitude) which is the cause of the smaller differential semi-major axis.

### 4.2.6 Initial Mean Orbital Elements

The orbital elements – in particular the altitude (semi-major axis) and eccentricity – as well as the nominal separation distance, also impact the passive relative motion. Inclination also has some impact on the passive relative motion, due to the Earth's oblateness, however the QB50 science payload is intended for a sun-synchronous orbit, which limits the choice of inclination to a single value. Assuming only an oblate Earth ($J_2$ only), the RAAN selection would have no impact due to symmetry - even modelling up to $J_{20,20}$, since $J_2$ has the largest impact the effect of RAAN is small. Thus this section will address the effects of initial mean semi-major axis, mean eccentricity and separation distance.

Two graphs have been generated that describe the passive relative motion for various orbital parameters. These graphs correspond to two different eccentricities: (1) a circular orbit with an eccentricity of 0.001, and (2) an elliptical orbit with an eccentricity of 0.015. These have been chosen to give some idea of the general trend across various eccentricities. The circular case corresponds to the eccentricity the QB50 Nominal orbit described in Section 2.1, while the Elliptical orbit's eccentricity corresponds to the QB50 Elliptical orbit from the same section. In the graphs, altitudes between 350 and 800 km (typical altitudes for LEO CubeSats, as per conversation with J. Guo March 19, 2015) are considered. Further, separation distances ranging from 500 km to 1100 km (corresponding to a widely-spaced mission with, for a 380 km altitude orbit, between 65 s and 145 s between measurements by the two satellites) are studied. These separation distances were chosen to see the variation around the 1000 km nominal separation distance. In all cases the satellites are modelled using the $J_{20,20}$ gravitational model and differential drag with the deputy pointing 2° off of the velocity vector and the chief aligned with the velocity vector. Figure 4.13 shows the time before the satellites leave their control window (where the control window is defined as ± 10% of the nominal separation distance) for the various altitudes and separation distances for circular orbits. The corresponding results for elliptical orbits are shown in Figure 4.14.

In both of these figures, orbits which remain in their control window for more than 10 days (the duration of the simulation) are plotted at 242 hours so the flat line is an artefact of this truncation. These figures show that widely spaced, circular orbits with higher altitudes spend a longer time inside their control window, with the semi-major axis having the largest impact.

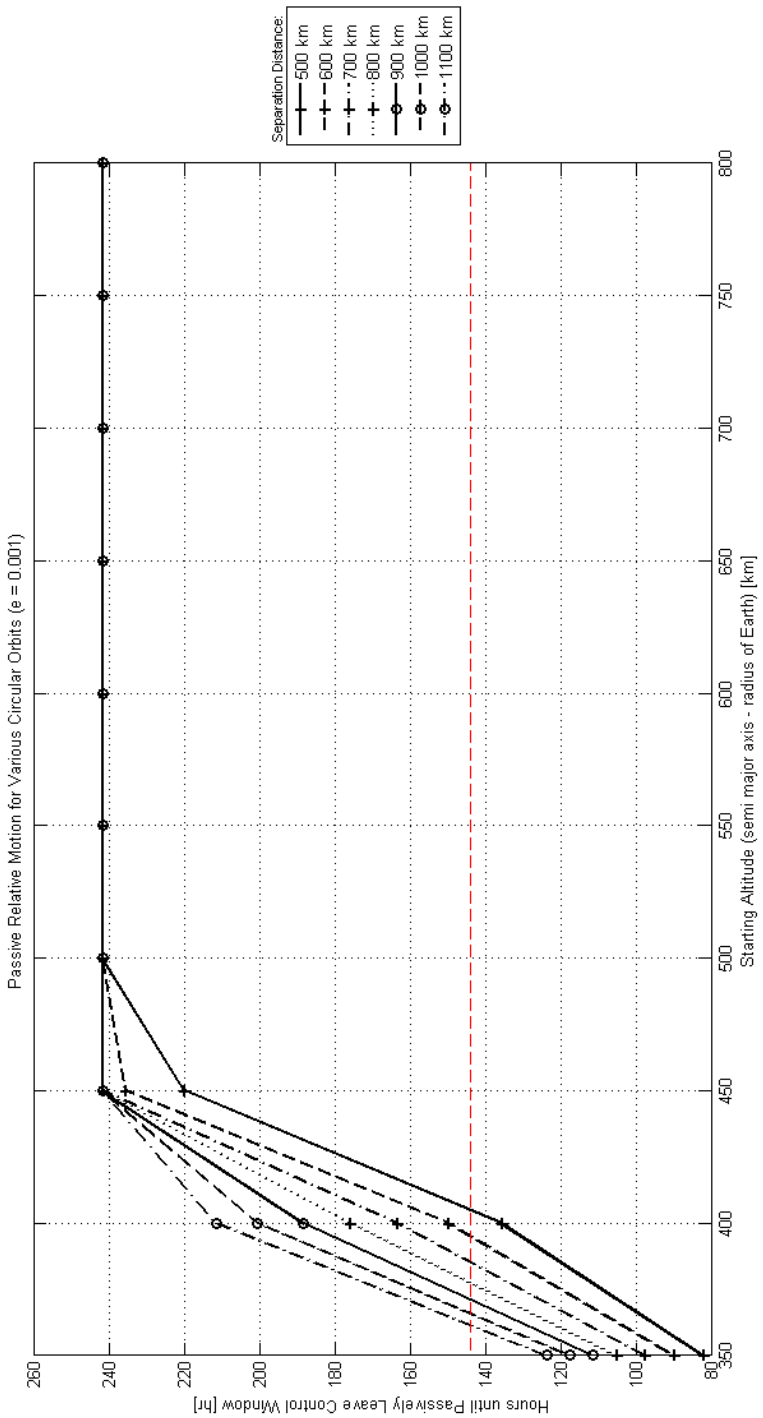



**Figure 4.13:** Duration spent within the Control Window (10% of the Separation Distance) at various altitudes and separation distances, for circular orbits (e = 0.001) where the deputy (forward satellite) is pointing 2° off from the velocity vector and chief (rear satellite) is aligned with the velocity vector.

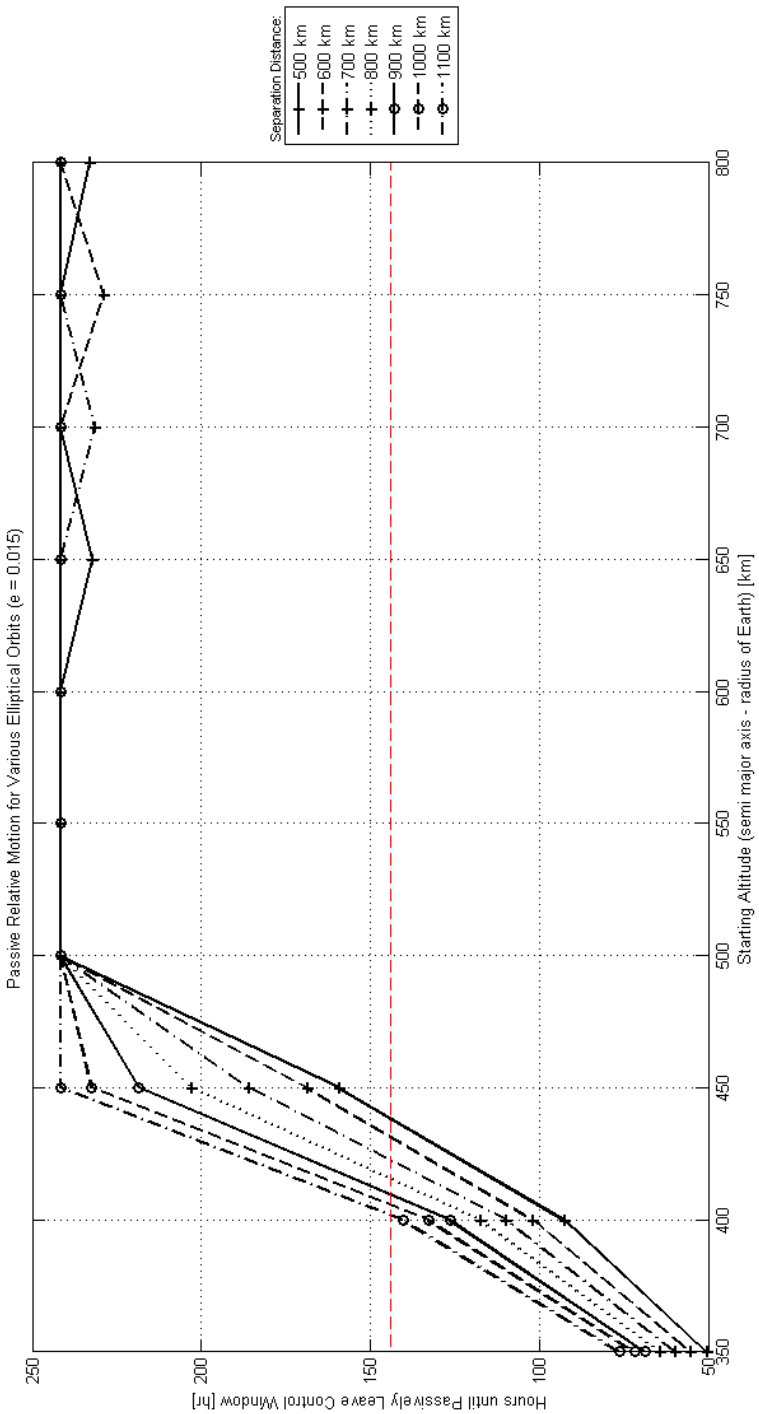



**Figure 4.14:** Duration spent within the Control Window (10% of the Separation Distance) at various altitudes and separation distances, for circular orbits (e = 0.015) where the deputy (forward satellite) is pointing 2° off from the velocity vector and chief (rear satellite) is aligned with the velocity vector.

This agrees with previous results that show that the differential drag is the largest factor and that the differential drag is smaller for higher, circular orbits. The orbits that meet the six day passive relative motion (represented by the horizontal dashed red line in Figures 4.13 and 4.14) are summarized in the list below:

1. Circular orbits: separation distance of 1000 km, the semi-major axis must be above approximately 365 km
2. Circular orbits: any modelled separation distance (500 km to 1100 km), the semi-major axis must be above approximately 410 km
3. Elliptical orbits: separation distance of 1000 km, the semi-major axis must be above approximately 405 km
4. Elliptical orbits: any modelled separation distance (500 km to 1100 km), the semi-major axis must be above approximately 440 km

It is clear from these figures that nearly all of the typical CubeSat orbit options meet the six-day passive maintenance requirement identified previously (assuming the attitude pointing error is favourably controlled), however there is a strong preference for higher altitude orbits.

### 4.2.7 Sensitivity to Matching Mean Orbital Elements

One major assumption made in this analysis is that the satellites are initialized with exactly the same mean orbital elements. If there is an error on the mean orbital elements of the deputy compared to those of the chief, there will be an increased drift. In particular, the perturbations are most influenced by the size and shape of the orbit (semi-major axis and eccentricity) so their effect will be studied here for the nominal QB50 orbit.

Two cases are considered: an error on the semi-major axis, and an error on the eccentricity. These errors will be applied to the case given in Figure 4.7 above, where the masses of the two satellites are the same (half full at 3.615 kg) and the deputy has an attitude pointing error of 2° compared to the chief which is aligned with the velocity vector. In the reference case the time to leave the control window (exceed 1100 km separation) was 164 hours.

Assuming the control accuracy is 10 km (based on Gill's Scaling Law as described in Section 4.3), it is reasonable to assume that the initial semi-major axis could vary by as much as 10 km from the desired value due to the control errors. Since lower altitudes have a larger impact of differential drag, let us consider the case where the mean semi-major axis of the deputy is 370 km (altitude) whereas that of the chief is 380 km. The following Figure 4.15 shows the effect of this error.

This figure shows that the time to leave the control window is now only 1.8 hours - a significant reduction compared to the 164 hours when the mean elements matched. Figure 4.15

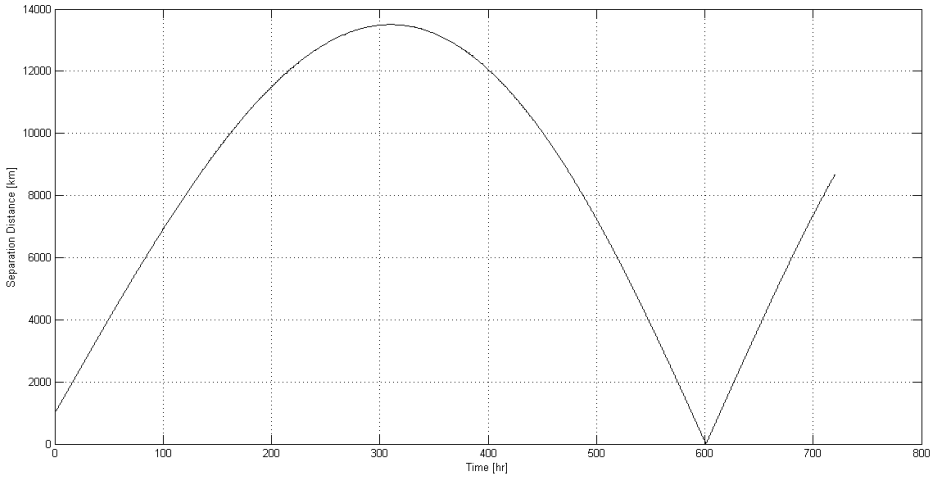



**Figure 4.15:** Passive Relative Motion of a formation subjected to $J_{20,20}$ and differential drag, where the chief is initialized in a 380 km altitude circular orbit (e = 0.001) with the Chief at the equator and the Deputy ahead by the 1000 km separation distance. The deputy satellite is initialized with mean orbital elements corresponding to the chief satellite, except the semi-major axis is reduced to 370 km. The Chief has a mass of 3.615 kg (half full tank), and is pointing along the velocity vector. The Deputy has a mass of 3.615 kg (half full tank), and is pointing 2° off of the velocity vector in the pitch direction. Both satellites have a coefficient of drag of 2.3.

shows the satellites separating so quickly that within 12 days the satellites are actually on the opposite sides of the Earth and begin to grow closer together.

In a similar way, we can assume that the perigee radius and apogee radius could have as much as a 10 km error on them. The worse case is for larger eccentricities, so if we take the apogee altitude to be 390 km and the perigee altitude to be 370 km, this gives an eccentricity of 0.0015. Figure 4.16 shows the effect of an error on eccentricity.

This figure shows that the control window is now breached in 126 hours.

From these results it is clear that matching the mean semi-major axis of the two orbits is critical to maintaining the formation - especially given the low update rate of TLEs. This means that the acquisition manoeuvre at the start of the formation flying phase as well as the correction manoeuvres throughout the flight must be extremely precise in terms of the mean semi-major axis in particular. In Chapter 5, the control accuracy will be discussed in light of this effect.

### 4.2.8 Viable orbits for TLE-based Navigation

This section showed that the two major impacts on the passive relative motion are the differential drag and the matching of the mean orbital elements at formation initialization. In

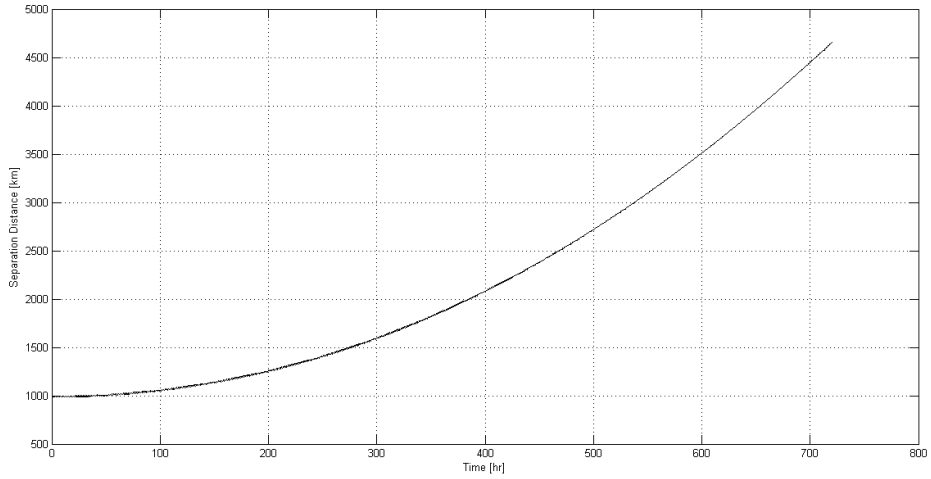



**Figure 4.16:** Passive Relative Motion of a formation subjected to $J_{20,20}$ and differential drag, where the chief is initialized in a 380 km altitude circular orbit (e = 0.001) with the Chief at the equator and the Deputy ahead by the 1000 km separation distance. The deputy satellite is initialized with mean orbital elements corresponding to the chief satellite, except the eccentricity has been increased to 0.0015. The Chief has a mass of 3.615 kg (half full tank), and is pointing along the velocity vector. The Deputy has a mass of 3.615 kg (half full tank), and is pointing 2° off of the velocity vector in the pitch direction. Both satellites have a coefficient of drag of 2.3.

terms of differential drag, this study addressed a wide variety of orbits that span the typical CubeSat orbits and found many examples of formations that are passively maintained for minimum six days. These can be summarized as:

1. Circular orbits: separation distance of 1000 km, the semi-major axis must be above approximately 365 km
2. Circular orbits: any modelled separation distance (500 km to 1100 km), the semi-major axis must be above approximately 410 km
3. Elliptical orbits: separation distance of 1000 km, the semi-major axis must be above approximately 405 km
4. Elliptical orbits: any modelled separation distance (500 km to 1100 km), the semi-major axis must be above approximately 440 km

It is always preferred to have a control window that is maintained for longer time to give more flexibility to the manoeuvre plan. Two ways to do this can be suggested assuming the orbit dimensions are fixed and that TLEs will be used as the primary source of orbital elements.

First, the control window could be enlarged. If the scientific mission can accept a larger variation on the nominal separation distance, it may be possible to choose a larger control window than Gill's Scaling Law recommends (see Section 4.3 for more details on Gill's Scaling Law). It would still be possible to know the separation distance to a the same level of accuracy as

with a smaller control window, however the value would be controlled to a less accurate level.

Secondly, the differential drag can be used to DelFFi's advantage. Using a simple Proportional-Derivative (PD) Controller, Kumar and his team have shown it is possible to maintain a formation by exploiting the differential drag [45]. If DelFFi's ADCS is able to support the attitude manoeuvres necessary, this could be used to supplement or even replace the microPropulsion system and extend the mission operations beyond the limit of the propellant budget. It is also necessary to confirm that such attitude manoeuvres would not be so large as to hinder the functioning of the science payload or to significantly reduce the power collected by the solar arrays.

With respect to the matching of the mean orbital elements, it was shown that a control accuracy of 10 km has a major impact on the passive relative motion. This analysis was only of a single case and conclusions as to the acceptable control accuracy are not rigorously established, however it is clear that the 10 km control accuracy is a maximum allowable accuracy. The following section (4.3) and the control accuracy analysis of Chapter 5 will estimate the navigation and control accuracies to determine if it will be possible to meet this control accuracy requirement.

## 4.3 Navigation Accuracy Analysis

Up to this point, Gill's scaling law has been used to establish the necessary sensor accuracy in order to perform formation flying at the specified separation distance. Gill's scaling law says, in case of no other requirements on the system, there is an order of magnitude difference between each of the characteristic parameters of a formation: sensor accuracy, control accuracy, control window dimension and separation distance [3]. For example, if a separation distance of 1000 km is desired, the control window should be 100 km wide or less, the control accuracy must be better than 10 km and the navigation accuracy must be better than 1 km. Note that the control window is a total width, but as was seen in the previous plots in this chapter, the relative motion does not cause an oscillation about the desired value but rather a consistent decrease or increase in the separation distance over time, depending on the initial parameters. Thus, although in general the control window can be said to be 1000 km $\pm$ 100 km (for a total width of 200 km), based on the initialization parameters, only half of that width is actually traversed (for DelFFi the separation distance will decrease) allowing the scaling law to be upheld while providing the largest window for the controller to operate within.

This scaling law is a useful tool for preliminary design (including top-down analyses), however it is necessary, as the design progresses, to establish the accuracy that can be expected based on the main error contributions (ie. from the bottom-up). As this thesis is focusing on the feasibility of TLEs as a navigation data source for formation flying, only the most significant contributions will be identified and estimated at this time.

To determine the various contributions to the navigation accuracy, it is reasonable to start with the raw data values and consider each activity performed until the navigation (relative state) is complete. While the accuracy of the mean orbital elements from a TLE in the along-track direction was found to be ± 2 km [6], the accuracy of the differential TLEs was identified in [1] to be ± 1 km, as discussed in Section 4.1. These TLEs are then propagated from the TLE epoch up to the current time (and possibly into the future) using the SGP4 algorithm in DelFFi's Onboard Navigation Function (ONF). In the design of the ONF it was found that the TLE error increases by between 0.1 and 3.0 km per week of propagation for satellites in LEO [64]. This specifically looked at satellites in the range of 600 - 800 km altitude, and it is assumed to not be significantly different down to 300 km altitude. It should be noted that if the numerical propagator which converts the mean orbital elements to osculating ones were used onboard, an additional error of approximately 6.8 km in the along-track (for the nominal QB50 orbit) would arise as described in Chapter 3. To avoid this contribution, the onboard propagator should use the SGP4 algorithm as described in the Onboard Navigation Function Technical Note [64]. This considers only the accuracy of the algorithm, however it is performed on a microcontroller, which means additional errors are incurred. These errors, such as truncation errors due to the limited size (number of bits) of values and bit flips due to radiation, are dominated by the clock offset between the UTC time and the time kept on the processor [64].

The size of the error due to the clock offset can be estimated using the data sheet for the Tuning Fork Crystal Unit (CM415) chosen for the DelFFi ADCS microprocessor [79]. This crystal oscillates at a nominal frequency, $f_0$, of 32.768 kHz with a Frequency Tolerance of ± 20 ppm. All of the errors on the oscillation frequency ($\epsilon$) are expressed in ppm and calculated in the following way:

$$\epsilon[ppm] = \frac{f_{Actual} - f_{Theo}}{f_{Theo}} x10^6 \tag{4.5}$$

Where $f_{Actual}$ is the actual oscillating frequency and $f_{Theo}$ is the theoretical or nominal operating frequency. Assuming the capacitance of the circuit is well matched to the crystal, the two main contributors to inaccuracies in the oscillation frequency are the operating temperature and the age of the crystal. The nominal operating temperature (also called the turnover temperature) for the crystal is 25°C ± 5°C. Outside of this temperature range the resonant frequency of the crystal varies with:

$$f = f_0 + k * (T - T_0)^2 \tag{4.6}$$

Where $k$ is the parabolic curvature constant, $T$ is the operations temperature and $T_0$ is the turnover temperature. For the CM415 crystal, the parabolic curvature constant (also called the temperature coefficient) is -0.034 ± 0.006 ppm/°C$^2$. The operations temperature of the crystal can be approximated by the temperature of the microcontroller from Delfi-n3Xt while operating on orbit. Although the data has not been published, the operational temperature of the microcontroller from Delfi-n3Xt varied between 20°C and 25°C with a few outliers during its mission (as per conversation with Nuno Baltazar dos Santos July 21, 2015). This falls within the nominal operations temperature range of the crystal and as such no error

on the frequency is incurred due to temperature. The age of the crystal causes an error of maximum ± 3 ppm in the first year of operations. As the lifetime of the nominal operations of the satellite satellites is expected to be approximately 40 days (10 days of Commissioning, 30 days of formation maintenance, as per [1], and elaborated in Chapter 6, the error will be well less than 1 ppm and is insignificant compared to the ± 20 ppm error on the nominal frequency itself. Thus for a nominal frequency of 32.768 kHz, and an error of 20 ppm, the operating frequency will be 32768 ± 0.65 Hz. This corresponds to a clock offset 1.728 s per day. To consider this as an along-track error, the velocity must be found. The orbital velocity for a satellite in a circular orbit is given in the equation below.

$$v_{orb} = \sqrt{\frac{\mu}{a}} \tag{4.7}$$

Where $\mu$ is the standard gravitational parameter (for the Earth this is 3.986x$10^5$ $km^3/s^2$) and $a$ is the semi-major axis of the orbit (in km). For example, for a circular orbit with a radius of 6758 km (380 km altitude), the orbital velocity is 7.68 km/s. Thus, over one day, the clock offset will result in an along-track error of ± 13.3 km. It is possible to minimize the effect of this error by synchronizing the clock more frequently with the ground server (which has a well synchronized time to UTC) or with the GPS constellation which also maintain extremely accurate time. As mentioned previously, in the worst case, it will only be possible to uplink to the satellite once per day and thus this is the limiting case that should be considered. It is also possible to address this error by measuring the oscillating frequency of the specific crystal on the flight microcontroller once it has been received and accounting for that error in the algorithm. The 20 ppm identified by the specification sheet is the variation that can be expected between various crystals, while the frequency for a single specific crystal will be significantly more stable. It is unclear at this time if such a test can be performed for the DelFFi mission and as such the worst case error is used in the rest of the analyses.

The errors associated with establishing the navigation accuracy for DelFFi have been summarized in Table 4.2.

**Table 4.2:** Summary Navigation Accuracy Analysis

| **Contributor** | **Accuracy [km]** | **Comment** |
|---|---|---|
| Sensor Error | ± 1 | Differential TLE Error in Along-Track [1] |
| Propagation | ± 3 /week | SGP4 Prop [64] |
| Clock Offset | ± 13.3 /day | For a 380 km circular orbit, As per values from [79] |
| Total | ± 17.3 | |

Assuming that the contributions are independent, their magnitudes can be simply added together to determine the total along-track position accuracy that can be expected by the system, in this case 17.3 km. Note that the contribution of the propagation algorithm is for one week of propagation while the clock offset contribution is for one day of propagation. This is because the TLE will be propagated for up to 6 days between updates, while the clock will be synchronized every day, so the worst case takes the total amount. This accuracy is

significantly worse than the error from the sensor alone and is mainly due to the clock offset. This implies that calibrating the crystal will be extremely critical to the proper performance of the AFF payload.

Calibrating the clock requires measuring the exact frequency of the crystal and building in a correction factor to the software. Microchip provides an application note describing calibration of crystals for watches with some information on possible methods by which this could be done [80]. In principle, the oscillator is left to run for some period of time then it is compared with a reference to determine the variation (error). If this were performed, it seems reasonable that the time could be measured to an accuracy of one second per week. This corresponds to 0.14 seconds per day, or 1.1 km/day. This would bring the total error on position based on TLE propagation to 5.1 km. This is five times the navigation error accepted by Gill's scaling law, however a detailed analysis of the control accuracy should be performed to determine if Gill's scaling law can be defied in this case.

## 4.4 Conclusions

Based on the findings from Section 4.2, both of the QB50 orbits were found to meet the six day passive relative motion requirement due to the TLE update rate. In addition, several other orbits were found to be acceptable based on the passive relative motion. They are:

1. Circular orbits: separation distance of 1000 km, the semi-major axis must be above approximately 365 km
2. Circular orbits: any modelled separation distance (500 km to 1100 km), the semi-major axis must be above approximately 410 km
3. Elliptical orbits: separation distance of 1000 km, the semi-major axis must be above approximately 405 km
4. Elliptical orbits: any modelled separation distance (500 km to 1100 km), the semi-major axis must be above approximately 440 km

For each of the listed orbits an attitude pointing error of maximum 2° was used (corresponding to the requirement for the ADCS velocity pointing mode). Since differential drag has been shown to have a very strong effect on the passive relative motion, an analysis of the acceptable attitude pointing error to maintain the control window for six days was performed for the QB50 Nominal orbit. It found that the attitude pointing error for the two satellites should be required to be less than 0.5° in order to ensure that in both arrangements (chief ahead of deputy and vice verse) the six day passive relative motion requirement is met. This finding will be provided to the ADCS team for further investigation.

These viable orbits listed above have all been determined based on the assumption that they are initialized with identical mean orbital elements. The sensitivity of the QB50 Nominal

orbit to the matching mean orbital elements was assessed and it was found that an error of 10 km (the control accuracy required for a 1000 km separation distance according to Gill's Scaling Law) on the semi-major axis, caused the satellites to drift apart and breach the control window in under two hours. The acquisition manoeuvre and each correction manoeuvre planned by the formation maintenance controller must take this into account and precisely control the differential mean semi-major axis. The limit on the acceptable control accuracy was not established in this research, but it is clear that decreasing the control accuracy (increasing the control error) beyond the recommended 10 km, is not acceptable.

Section 4.3 estimated the navigation accuracy to be 5.1 km assuming the microprocessor clocks can be calibrated precisely before launch. Starting from this 5.1 km navigation accuracy, using Gill's Scaling Law, the control accuracy would then be expected to be 51 km. This has been shown to be unacceptable for matching the mean orbital elements at initialization. Even if the control window were enlarged (to the detriment of the scientific data), the six day passive relative motion requirement would not be met. Instead, a bottom-up analysis of the control accuracy should be performed to establish if a 10 km requirement on control accuracy can be met. This will be discussed in Chapter 5.

# Chapter 5

# Formation Maintenance Control

Maintaining a formation based on Two-Line Elements (TLEs) is a difficult problem, due to the low rate and accuracy of the incoming data, as established in Chapter 4. To continue discussing the effect of TLEs on the formation maintenance problem, this chapter will discuss the development of a MATLAB Simulink Model that allows for the testing of various aspects of the formation flying software package. Next, it will address the control approaches taken by previous formation flying design teams and select a controller that is best suited to the DelFFi project considering TLE-based navigation. The characteristics of this controller will then be described and the performance in terms of propellant requirements and processing time will be assessed. Finally, this chapter will discuss the implications of formation maintenance control on the supporting subsystems of the DelFFi CubeSats - specifically, the Attitude Determination and Control System (ADCS) and the propulsion system.

## 5.1 Modelling and Implementation Approach

To model the system for both Chapters 4 and 5, a MATLAB Simulink model has been created. In Chapter 6, the Formation Flying Package (FFP) is broken down into Modes and Sub-Modes following the method established by Brauer [81]. This established three Sub-Modes for the Formation Maintenance algorithm to perform: Estimation, Planning and Monitoring. It is logical to use these Sub-Modes as a structure for the Simulink models for consistency in development of the operational plan and software. To summarize these three Modes, the main events are identified here.

When orbital elements arrive (in the form of a TLE), they are checked for validity (no manoeuvres have occurred in the previous five days and no errors in the data transfer) and the Estimation Sub-Mode is activated. The Estimation Sub-Mode propagates the state forward to determine whether a manoeuvre should be planned within the next two days (the maximum time between TLEs [64]). A manoeuvre is required if during those two days the satellites

leave the control window (either too close or too far apart). If a manoeuvre is required, the Planning Sub-Mode is activated, which determines what manoeuvre(s) (magnitude and direction) should be performed at what time(s) to return the satellites to their nominal separation distance. At this time, it is assumed that only one satellite (called the Deputy) will perform manoeuvres, though the labelling of deputy between the two satellites is arbitrary and can be varied during the mission. Once the manoeuvre has been planned, the satellite switches to the Thrust Mode and the Monitoring Sub-Mode of the FFP is activated. The Monitoring Sub-Mode tracks the progress of the satellite in meeting the manoeuvre plan using feedback from the actuator (Specifically, the thrust magnitude from the propulsion system, and if possible the pointing angles from the ADCS). If an problem is encountered during the Monitoring Sub-Mode, the manoeuvre can be stopped and mitigation procedures can begin.

The Estimation Sub-Mode will consist of an onboard propagator, such as the existing Onboard Navigation Function developed for Delfi-n3Xt, combined with a flag to decide if the satellites will exit the control window before the next TLE is expected. Since the core of this Mode (the propagator) has already been developed at TU Delft, it will not be addressed in this research. Further, the Monitoring Sub-Mode will be performed by the Propulsion System in order to remove the risk of a broken communication link ($I^2C$ drop-out, for example) between the system that decides to stop the manoeuvre and the system that actually stops it. This has the added benefit that the propulsion system can make use of the pressure sensors to estimate the thrust magnitude to improve the precision of manoeuvre monitoring. Ideally, the Monitoring algorithm would also receive feedback of the pointing angles of the two satellites, since differential drag has a large impact of the relative motion, however since the Intersatellite Link (ISL) has not been evaluated in detail for the DelFFi mission, this capability is not considered for the Monitoring Sub-Mode at this time. For this research, the main work has been towards the Planning Sub-Mode. It has been implemented in Simulink and converted to C for execution on the target processor. The details of the Planning and Monitoring Sub-Modes are discussed in the following subsections.

### 5.1.1 Planning Sub-Mode Simulink Model

When entering the Planning Sub-Mode, it is known that within two days (before the next TLE arrives) the satellite will leave the control window. As no new data sources will be available, it is preferred to plan and perform the manoeuvre(s) as soon as possible (while there is the smallest propagation error). Further, planning a manoeuvre sooner will mean that fewer of the future TLEs will be invalidated (recall that TLEs are invalid if a manoeuvre has occurred during the previous five days as NORAD averages measurements over this time to generate TLEs). With this constraint in place, the planning algorithm:

1. Requests the absolute position of both satellites (in mean classical orbital elements or position and velocity vectors) from the supporting Onboard Navigation Function (ONF),
2. Calculates the relative error (differential mean orbital elements or position and velocity depending on the chosen controller), and
3. Determines the control plan (magnitudes, directions and start times of the burns).

Since the ONF is not available to Simulink, the Simulink model not only contains the Relative Navigation and Controller blocks, it also contains a numerical orbit propagator for each of the two satellites as shown below in Figure 5.1.

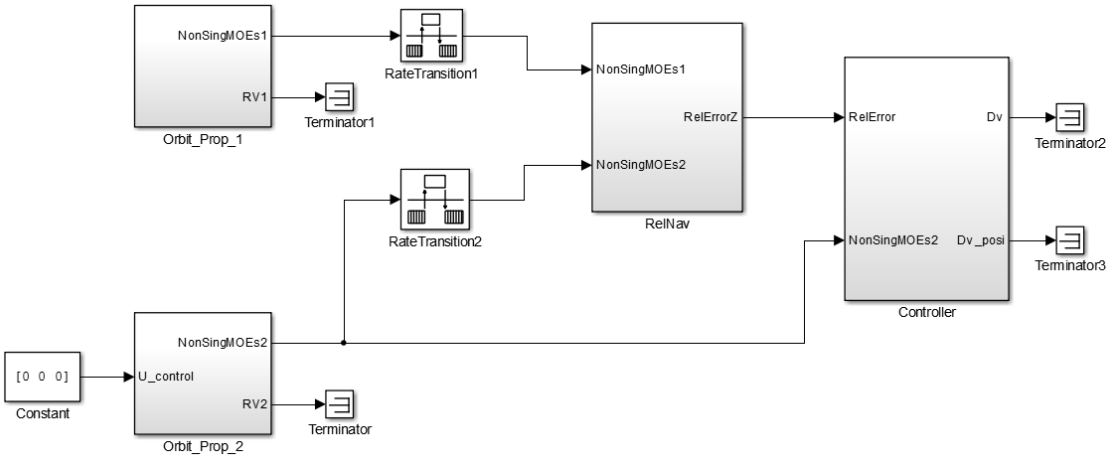



**Figure 5.1:** Outline of model implemented in Simulink for the formation Planning Sub-Mode.

The details of the orbit propagator blocks can be found in Appendix D. They are coded in MATLAB and are thus not able to converted into C using the Simulink Coder. For the flight version of this software, the data they provide will come from the ONF. The propagators allows drag, differential drag, a gravity model (currently implemented up to $J_{20,20}$), sensor noise models and sensor rate models to be included. It has been validated by comparison against two other established propagators within the space systems engineering department. The output of the orbit propagator are both the radius and velocity vectors in the ECI frame as well a set of orbital elements. In this case, due to the requirements of the controller selected in the following section (5.2) the non-singular mean orbital elements ($[a, \theta, i, q_1, q_2, \Omega]$ where $\theta$ is the true latitude) are output.

It is clear from this diagram that there is no feedback within this system. The planner accepts the position at each time step (default is 1 second) within the possible planning window and calculates the control plan that should be followed at that time. After, the system selects the best plan to forward to the actuator (propulsion system and ADCS). The possible planning window refers to the times that would be acceptable start times for the control plan. The first constraint on this time is that the manoeuvres not be too soon after the creation of the plan so that the propulsion system and ADCS can prepare. It is assumed that two orbits (approximately three hours) advance notice will be more than sufficient for this process. the second constraint is that the manoeuvres be completed before the satellites exit their control window. It is expected that no manoeuvre plan would last longer than one day, and the satellites will leave within two days, so for safety, the manoeuvres should be started within twelve hours (eight orbits) of entering the Planning Sub-Mode.

The content of the Relative Navigation block is shown below in Figure 5.2.

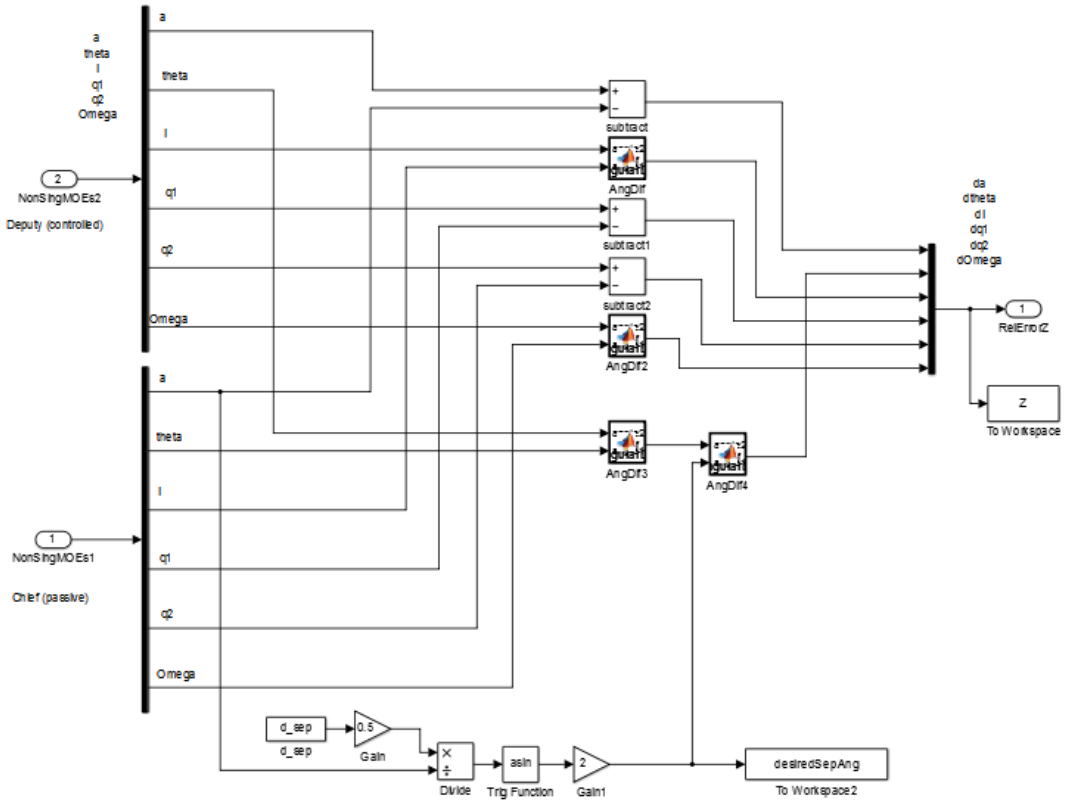



**Figure 5.2:** Internal schematic of the relative navigation block of the Planning Sub-Mode Simulink Model.

In essence, it simply subtracts the mean orbital elements of the chief from those of the deputy and returns them to the controller. The exception is the mean anomaly, which is further compared to the desired separation angle (calculated from the desired separation distance).

The Controller is a modular block that can be filled with the code relating to any controller that the team may want to test. The code of the controller selected in the following section is discussed in Appendix D. Accompanying this model are several MATLAB scripts that define the initial conditions, process and plot the data that is sent to the Workspace and determine the optimal plan to follow. These scripts are described in Appendix D.

This same model was modified to be converted into C for execution on the microcontroller. Although the Simulink model uses values from the MATLAB workplace when running in simulation, when embedded on the microcontroller these parameters must be hard-coded into C until such time as the software can be extended to access the parameter database that will exist on the ADCS microcontroller (see [59] for further detail). After setting the model configuration parameters as described in [82], the 'C' code is generated and moved into a project for Texas Instruments' Code Composer Studio. There, the code can be programmed onto the processor and the output values can be monitored to be compared with the expected values from Simulink. This processor in the loop testing will evaluate three things:

1. That the code is functioning in a way that can be embedded on the DelFFi ADCS microcontroller (when the time comes),
2. The processing time is acceptable for the ADCS microcontroller, and
3. The accuracy is within tolerance compared to the Simulink model.

The code generation method and the C code produced are included in Appendix E, and the results of execution on the target processor are discussed in Section 5.2.2.

### 5.1.2 Monitoring Sub-Mode Simulink Model

Once the Planning Sub-Mode has completed its task, the manoeuvre plan will be sent to the actuator, however it is interesting to know the effect of this manoeuvre plan on the satellite. For this reason the Monitoring Sub-Mode was modelled in Simulink. Its structure is shown below in Figure 5.3.

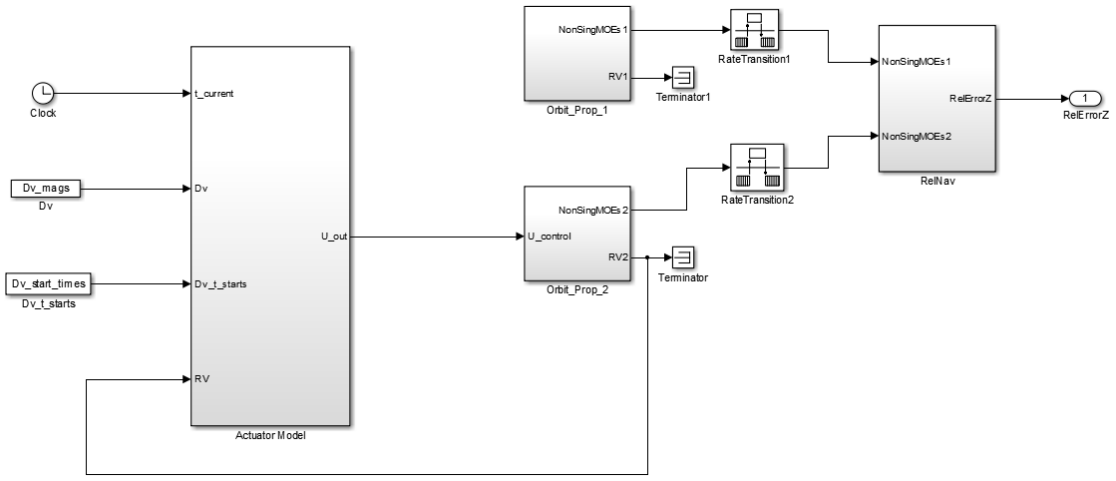



**Figure 5.3:** Outline of model implemented in Simulink for the formation Monitoring Sub-Mode.

The orbit propagators and relative navigation blocks are identical to those of the Planning Sub-Mode model. The actuator model is a new element that takes the manoeuvre plan and using theoretical models of the propulsion system, calculates the instantaneous acceleration the satellite will experience as a result of the burns. These accelerations are fed into the orbit propagator of the deputy and the resulting motion can be monitored. Note that this model uses the osculating orbital elements exclusively.

The theoretical design of the thruster was developed by Poyck [55] and the experimental design is being continued primarily by van Wees, Jansen, Zandbergen and Cervone [53]. To develop a model of the thruster, [55] has been used. Since the pressure will be measured, it is most logical to use the pressure as a reference for the instantaneous thrust magnitude. Then an error on the pressure is applied (based on the pressure sensor accuracy) in order to set an

error on the thrust magnitude. Thrust is also dependent on temperature but the temperature will be held constant using a heater and its effect will not be modelled at this time. In theory, the pressure in the tank has been shown to have the following value over time according to Poyck [55].

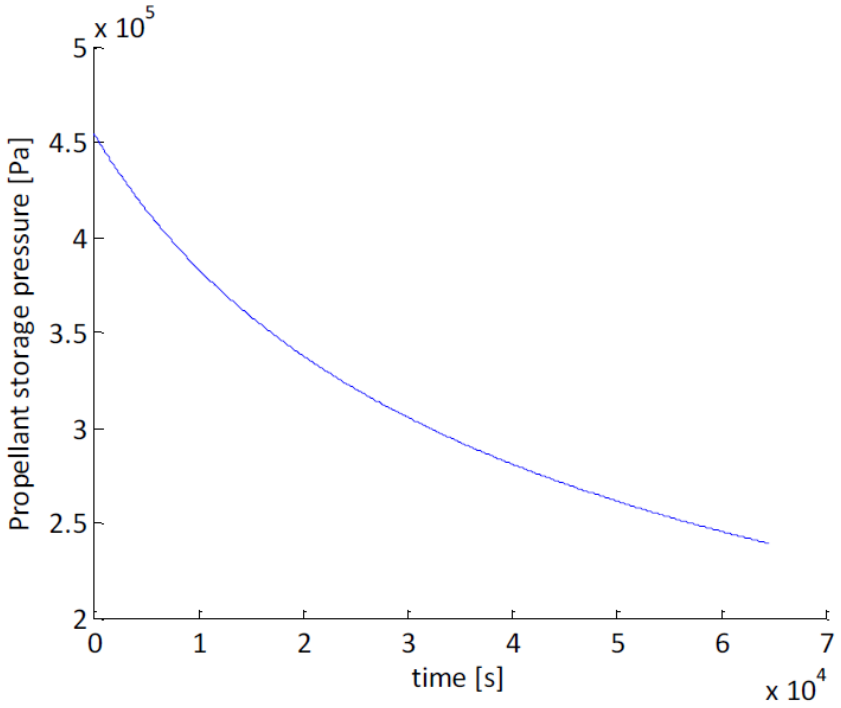



**Figure 5.4:** Model of pressure over time for the propulsion system [55].

Then, based on this pressure, the thrust profile can be determined. With an ideal rocket assumption, the thrust can be related directly to the time as shown in Figure 5.5 from Poyck [55].

The curves relating the thrust to pressure and pressure to time have been modelled as fourth order polynomials. Although it would be possible to use an analytical relationship, since the model will eventually be generated by calibration, a less processor intensive polynomial relationship is preferred. The detailed actuator design is included in Appendix D. In addition to the magnitude of the thrust, the actuator model includes a model of the mass change due to the propellant usage. The pressure measurement, also allows a calculation of the mass flow, which can be used to determine the instantaneous mass of the satellite at any time. The error placed on the thrust and mass are derived from an error on the pressure due to the inaccuracies of the pressure sensor. Although no sensor has been chosen, pressure sensors for this level typically have errors less than 0.5% (as per conversation with A. Cervone August 2015).

The second part of the actuator model is the ADCS. The ADCS is not generally considered to be actuation, however it does provide the pointing control necessary to orient the thrust in the desired direction. For the current iteration of the model, a constant attitude pointing error is applied to the thrust vector. In the future, this could be replaced with a statistical

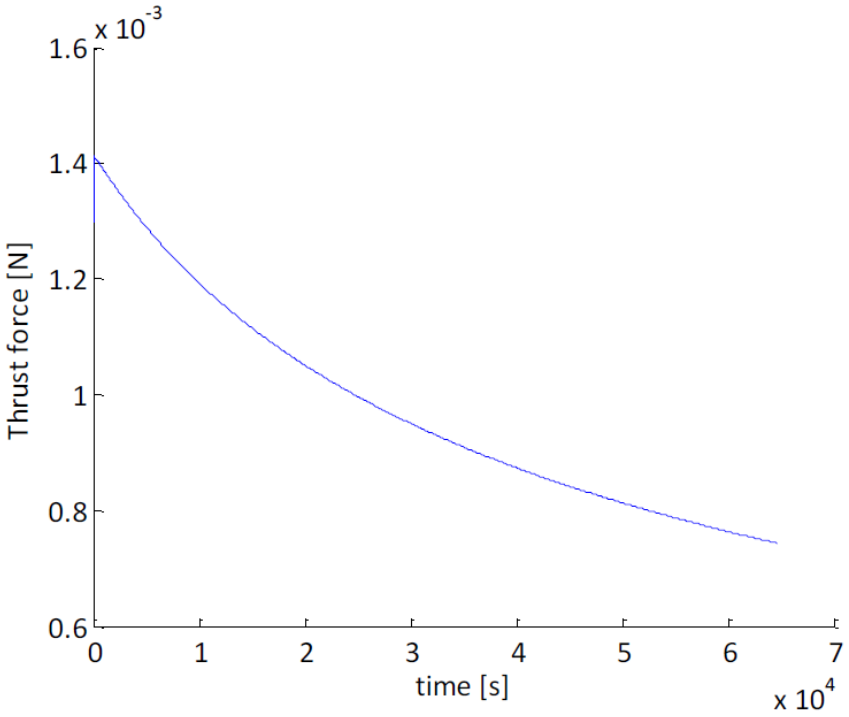



**Figure 5.5:** Model of thrust over time for the propulsion system [55].

time-varying model of the pointing error or feedback from the ADCS Simulink model if integration of the two models were performed. The actuator model can also accept time offsets between it and the rest of the Simulink model (to considered delays on the data bus) and other sources or error or noise such as thrust misalignment angles as required.

The software to monitor the manoeuvres will be executed by the propulsion system (as per conversations with B. Zandbergen April 2015), and as such will not be tested on the ADCS microprocessor. performing this Mode within the propulsion system is logical as the feedback about the thrust magnitude and the models of the thrust over time are contained within that subsystem. Further, if there is an error it is important that the propulsion system is able to end the manoeuvre without having to worry about $I^2C$ drop-outs or other communication problems in the data bus.

### 5.1.3 Use of the Simulink Models

Although the Simulink model was designed for the DelFFi mission, it has been built in such a way as to make it flexible to other formation maintenance missions. Each block could be used, tested, replaced or removed with minor changes to the rest of the system. The propagator chosen is not limited to certain types of orbits and the duration and time steps of simulations can be varied. Each element includes detailed comments on its purpose, source and usage and initialization conditions have been grouped for easy access and variation. It is hoped that this model can be built upon by future students as the DelFFi project and Delfi program develop.

## 5.2 TLE-based Controller Design

In designing a controller based on TLEs, the low accuracy and low update rate must be considered. This section will discuss the solutions of previous formation flying teams and select one that is suited to the DelFFi project considering TLE-based navigation. After this, the performance of this controller in simulation and on the microprocessor will be discussed.

### 5.2.1 Control Approaches

Formation flying algorithms use a controller to determine the magnitude, direction and start time of the actuation necessary to correct the state error calculated by the navigation algorithm. DelFFi assumes a constant desired separation distance, in other words the guidance algorithm is trivial as the only orbital element that is different for the desired orbit compared to the Chief's orbit is the true anomaly, and that offset is a constant value.

Generally, formation maintenance can be considered to be either passive or active. Passive maintenance means choosing an orbit that is invariant to the necessary error level for the desired lifetime of the formation. Since DelFFi is a secondary payload, meaning it cannot choose its orbit, and making an orbit manoeuvre to an invariant orbit would be prohibitively expensive in terms of fuel consumption, DelFFi will use an active control method. Next, controllers can be divided between impulsive and continuous. TU Delft has designed only impulsive actuators and as such an impulsive controller must be used to match.

For this work the controller architecture will be rooted in an existing controller with some small variations as necessary to best suit the DelFFi mission. This is because the main question is the suitability of TLEs to the DelFFi project, so a well-understood controller will allow the most insight into the impact of TLE-based navigation algorithms on the formation flying performance.

Amongst the existing formation maintenance controllers studied, nine controllers stand out as interesting options for the DelFFi mission.

**2-Impulse Analytical Controller** As detailed by Alfriend et. al., this controller pre-schedules two impulses separated by 180° in order to compensate for perturbations between two satellites modelled using mean orbital elements with the Hill-Clohessy-Wiltshire equations [18]. This method is characterized by low processing power requirements but non-optimal fuel consumption. It is only suitable for circular orbits or very low eccentricity orbits. The pre-scheduled thrusts mean that only the direction and magnitude of the actuation must be determined by the controller and also ensure that a consistent duration between thrust pairs is maintained. This controller has also been discussed by Vaddi et. al. [22] and is a well established, traditional control method for formations.

**Tschauner-Hempel based Linear Quadratic Regulator** This controller is very similar to the one used successfully by CanX-4/5 (see [26]) with the exception of the use of the T-H eqns instead of HCW equations to describe the relative motion [83]. This allows the incorporation of elliptical orbits of the chief satellite into the system. It has not been widely referenced by other formation flying teams, but the main structure of the controller remains the same as traditional LQR methods [18]. The controller is designed for a tight formations and thus uses a continuous thrust profile (discretized only as necessary to modulate the thrust from the propulsion system). Other sources have shown it is possible to discretize LQRs such that they can be used in an impulsive manner (refer to [26], [18]), which would make this acceptable for the DelFFi satellites.

**Hill-Clohessy-Wiltshire based Discrete Linear Quadratic Regulator** Developed for the CanX-4/5 satellites, this controller uses a discrete linear quadratic regulator to maintain a closely-spaced along-track separation (50-100 m) between the two satellites [26]. The satellites successfully demonstrated formation flying in November 2014 [84]. Since the orbits of the satellites are nearly circular and their spacing is very close, the Hill-Clohessy-Wiltshire dynamics model (circular orbits, no gravitational perturbation model) the system sufficiently well and provide a calculation rate fast enough for a continuous thrust profile [26]. The linear quadratic regulator determines a control acceleration optimized using a cost function that minimizes the propellant consumption [26].

**3-Impulse Analytical Controller** For the PRISMA mission, several controllers were used to perform the various functions. In particular, D'Amico details an analytic controller that used two impulses separated by 180° for in-plane corrections, plus a third impulse for the out of plane correction [4]. It accounts for the $J_2$ perturbation but assumes the chief satellite's orbit is circular. The solution is purely analytical, no propellant optimization is included and the impulses are timed to have long periods between them that vary depending on the accuracy required by a specific phase of the mission This controller was further analyzed by de Bruijn and Gill in their study of the influence of sensor and actuator errors of this PRISMA controller and a model predictive controller [20].

**Analytical Solution for Circular Orbits** The team that created the 'psatellite' simulation tools developed an analytical controller for circular orbits that is built off of a similar solution from Alfriend [85]. It is applicable to missions in which the time between manoeuvres is very large (compared to the manoeuvre duration) and those which do not require precision formation flying. The model is based off of the Hill-Clohessy-Wiltshire equations which do not account for $J_2$. As an analytical controller it is not very complex in terms of processing, however, it does optimize the timing of the manoeuvres to minimize propellant requirements, which requires looping through several possible combinations.

**Model Predictive Controller** Published in 2007 by Breger and How, this model predictive controller (MPC) works with a receding horizon scheme, in which a series of thrusts are planned then only the first portion of the plan is enacted before a new plan is created [86]. The dynamics model used allows for widely space formations and highly elliptical orbits such as would be encountered by a mission like MMS (the Magnetospheric Multi-Scale mission).

This MPC is widely cited by various formation flying research teams and in particular has been studied by de Bruijn and Gill in [20] as a comparison to the performance of the PRISMA analytical controller. It allows for long time spans between thrust sets, can account for thrusters with limited control acceleration levels and minimizes fuel consumption both on the individual satellites and among the entire formation [86]. It is not intended for high precision formations [86].

**Linear Programming Controller** The team behind the 'psatellite' simulation tools have also suggested a controller based on linear programming [85]. It assumes a much larger time between manoeuvres than the duration of the manoeuvres and is valid for both circular and elliptical orbits [85]. The linear programming approach determines a sequence of impulsive thrusts that occur over a fixed-duration window (ie. thrust sets are not limited to a single orbit time frame as some other analytical solutions are [85]). The dynamics model used does not account for $J_2$ [85].

**Fixed-Impulse, Single-Input Controller** Since CubeSats have limited capabilities, Ovchinnikov et. al. approached the formation flying control problem from the view of a satellite with limited attitude control. They suggest a controller that uses thrusts only in a single direction, in particular aligned with the magnetic field vector. This method considers gravitation perturbations up to $J_2$ in the relative motion model but assumes circular orbits. The controller is similar to the one intended for Magion-2 (a pair of Czechoslovakian satellites from the late 1980's), which was not able to demonstrate formation flight due to a thruster failure. The necessary correction manoeuvres can be optimized to minimize propellant (though the version described in [87] does not do so) and can be calculated whenever the satellite nears a threshold of the control window. Although published in 2009, very few researchers have cited this work.

**State Transition Matrix Receding Horizon Controller** A state transition matrix controller was developed by Yan and Gong in 2014 as a way to include both $J_2$ perturbations and an elliptical orbit without encountering the high processing complexity of a linear quadratic regulator in the same situation. This controller works with a receding horizon scheme similar to the MPC above. It is fuel optimized but expects a continuous thrusting profile.

In order to select one of these controllers upon which to base the DelFFi controller, several criteria are established.

The first, and highest weighted criteria is 'Heritage' of the controller. As discussed earlier, a well-studied and understood controller is desirable for this project so that the effect of using TLEs for navigation can be evaluated. It is also beneficial for the DelFFi project in general as it will increase the reliability of system. For this criterion, a controller that has been successfully used on-orbit will be considered to exceed requirements, a system that has been studied by a large number and variety of teams will meet requirements, established controllers that have not been widely used will be considered to have a correctable deficiency while new controllers without confirmation by other teams will be considered unacceptable.

Based on the requirements established in Chapter 2, minimizing the propellant consumption (for both each satellite individually and the formation as a whole) and reducing processing complexity should be considered in controller selection. Comparing such different controllers on these two measures is very difficult as the original authors studied different scenarios with different standards of measurement. Rather than modelling all of the controllers to establish quantitatively their propellant consumption and processing requirements for the DelFFi case, two criterion are chosen to reflect the general trend across the controller options. A binary criterion of whether or not the controller includes a term that optimizes each manoeuvre to minimize propellant will be used to gauge if the controller will have a large propellant consumption. Meeting requirements is represented by the controller optimizing for minimum propellant consumption (either formation-wide or by individual spacecraft), while not doing so is considered unacceptable. The processing power of the nine options have been compared by the DelFFi microcontroller development team to establish an approximate order from most simple (exceeding requirements) to most complex (correctable deficiencies). The team did not identify any of the controllers to be unacceptable in terms of processing complexity. These two criteria have been given an equal weighting that is below the heritage criterion but above the remaining criteria.

Establishing the accuracy of the controllers is very difficult due to the different verification cases used by the different researchers. it is expected that the most accurate controllers will include the effects of $J_2$ and eccentricity of the chief satellite's orbit. Thus these become two binary criteria with equal weighting. Including these factors would constitute meeting requirements, while not including them is a correctable deficiency.

The final criterion addresses the desire for the science payload to have long periods between sets of thrusts during which it can make measurements. Many of the controllers require a series of impulsive thrusts in different directions or at different positions over the course of one or a few orbits, followed by no thrusts for several orbits or even several days. This type of pattern will be labelled as "thrust sets separated by several orbits or days" which will be considered as meeting requirements. Other controllers are intended for high precision formations in which thrusts should be applied nearly every orbit to maintain the relative states. This is considered a continuous thrusting profile for the sake of this criterion and will be marked as a correctable deficiency since it is possible that when applied to a largely-spaced formation like DelFFi the time between thrusts will be much larger (due to the larger control window). This criterion is the most difficult to judge since the size of the control window and accuracy of formation studied by the majority of researchers is significantly different from DelFFi. As such, it is given the lowest weighting (half that of the Heritage criterion).

A graphical trade-off table has been constructed to compare the nine potential controller architectures using the criteria established above. It is shown in Figure 5.6.

Using this table it is clear that four of the controllers can be immediately removed due to the unacceptable ratings. Both the Fixed-Impulse, Single-Input Controller (1.8) and the State Transition Matrix Receding Horizon Controller (1.9) are very new and have not yet been studied in detail by other teams. Further, two controllers don't consider propellant consump-

| Factor (weight) | Heritage (10) | Well-suited for Elliptical Orbits (7) | Includes J2 in Model (7) | Minimizes Propellant (8) | Time between thrusts (sets) (5) | Processor Simplicity (8) | Score (450) |
|---|---|---|---|---|---|---|---|
| 1.1 2-Impulse Analytical Controller | studied since 2005, cited by several others G | No Y | No Y | No R | Days G | (most simple) 1 B | 270 |
| 1.2 Tschauner-Hempel based LQR | studied since 2005, cited by a limited variety of teams Y | Yes G | No Y | Yes G | Continuous Y | 5 G | 294 |
| 1.3 HCW based Discrete LQR | CanX-4/5 B | No Y | No Y | Yes G | Continuous Y | 4 G | 323 |
| 1.4 3-Impulse Analytical Controller | PRISMA B | No Y | Yes G | No R | Orbits-Days G | 2 B | 311 |
| 1.5 Analytical Solution for Circular Orbits | pre-2004, used in psatellite tool, cited widely G | No Y | No Y | Yes G | Orbits-Days, easily select B | 3 B | 344 |
| 1.6 Model Predictive Controller | studied since 2007, cited by several others G | Yes G | Yes G | Yes G | Orbits-Days G | 6 Y | 336 |
| 1.7 Linear Programming Controller | pre-2004, used in psatellite tool, cited widely G | Yes G | No Y | Yes G | Orbits-Days G | (most complex) 8 Y | 315 |
| 1.8 Fixed-Impulse, Single-Input Controller | new (2009, not widely cited by others) R | No Y | Yes G | Can be included Y | Days G | 3 G | 235 |
| 1.9 State Transistion Matrix Receeding Horizon Controller | new (2014) R | Yes G | Yes G | Yes G | Continuous Y | 7 Y | 241 |

Legend
Blue(B): Exceeds Requirements (10)
Green(G): Meets Requirements (8)
Yellow(Y): Correctable Deficiencies (5)
Red(R): Unacceptable (0)

**Figure 5.6:** Trade off table to select the formation flying controller architecture.

tion in their optimization: 2-Impulse Analytical Controller (1.1) and 3-Impulse Analytical Controller (1.4). Of the remaining five controllers, two stand out based on their overall score: the Analytical Solution for Circular Orbits (1.5) and the Model Predictive Controller (1.6). These controllers are very different. The MPC will be a significantly more accurate system that requires a high processing complexity, though not so high that it is unacceptable for a CubeSat microcontroller. On the other hand, the Analytical controller is an extremely simple algorithm that should require very little processing time and is very easily understood and modelled. It is best suited for problems where the time between manoeuvres is significantly longer than the duration of the manoeuvres and thus is poorly suited to precision formation flying control [85]. As a technology demonstration, and particularly for this feasibility study, the analytical controller is better suited to the needs of the DelFFi project. This is especially true as the strength of an MPC lies in its higher accuracy - which may not be achievable with low-accuracy data sources such as TLEs - and the receding horizon scheme - which can not be effectively implemented with low-update rate sensor such as a TLE where feedback does not occur during a manoeuvre.

This analysis described above is based solely upon the general trends seen within the trade off table. To quantify the results, a score was calculated for each of the controller option using a score of 10 for aspects that exceed requirements, 8 for those that meet requirements, 5 for those with correctable deficiencies and 0 for unacceptable ratings. Out of a highest possible score of 450, the Analytical controller received the best score of 344 with the MPC close behind at 336. The closest competitor was the HCW-based LQR, which approached the top score due to its successful implementation on the CanX-4/5 mission and its low processing complexity. However, the CanX-4/5 actuators could handle continuous thrusting patterns that the DelFFi thrusters will not be able to perform due to thermal and power constraints, which means a variation to the original HCW-based LQR would be necessary to make it suitable for DelFFi. For these reasons, the Analytical Solution for Circular Obits recommended by [85] will be selected for the DelFFi mission.

In making this selection, first a qualitative analysis of the general trends visible in the trade-off was performed, then a quantitative analysis using weights and scores was discussed. The selection of the analytical controller is mainly due to the strong desire for a simple, well understood controller that can be used on a CubeSat platform for a sufficiently long mission life. This desire was seen both int he qualitative and quantitative analyses. For the initial design of a formation flying algorithm, the performance requirements (described by the 2nd, 3rd, 4th and 6th criteria) are not exceedingly strict. If performance was the main goal, and the necessary high accuracy sensors and actuators were available, the weights of the criteria concerning elliptical orbit suitability, the inclusion of gravitational perturbations in the model, and propellant minimization would have been significantly higher. With this in mind, the selected controller would have been different - likely tending towards the Model Predictive Controller as it has shown significantly improved performance in studies by Breger [86] and de Bruijn [20]. To really make this choice, the impact of TLEs (low accuracy and update rate) on this performance of the MPC will require further study. After a preliminary design has been shown to succeed, performance improvements through different controllers can be addressed in future iterations.

On the other hand, if the processor selected for the DelFFi mission was more limited or a very fast cycle speed was needed for the ADCS computer, a higher waiting on processor simplicity would have been selected. This would have driven the selection to one of the analytical controllers described. Although DelFFi is not restricted in this way, the selection of Mueller's Analytical Solution for Circular Orbits, means that the controller could easily be used by other CubeSat teams who have more restrictive processing requirements and the FFP will have a minimal impact on the design of the rest of the satellite bus.

Further, DelFFi considers near circular orbits for both the nominal QB50 orbit (e = 0.001) and elliptical QB50 orbit (e = 0.015). Such orbits can be treated by controllers that are not suited for elliptical orbits with some loss of accuracy, however if the eccentricity were to be significantly higher, they would no longer be suitable. In this case, the MPC suggested by Breger would be well suited to the problem and could be considered in future work. Choosing an Analytical controller is certainly not the most flexible option, but it is the most transparent which is valuable for preliminary controller designs such as this project. Consulting experts who have used MPC's in the past would be highly valuable to determine the if and how an MPC could be implemented for CubeSat formation flying algorithms.

### 5.2.2 Analytical Solution for Circular Orbits Control Algorithm

The analytical solution for circular orbits developed by Mueller in [85] identifies a set of four manoeuvres (one out of plane and three in plane) to return a satellite to its desired trajectory. It is based on a previously developed solution derived from Gauss' Variational Equations, but modified to allow for multi-orbit manoeuvre durations [85]. This allows the system to minimize the propellant consumed per correction manoeuvre. The equations use the orbital elements [$a$, $u$, $i$, $q_1$, $q_2$, $\Omega$]. Where $a$ is the semi-major axis, $u$ is the true latitude, $i$ is the inclination, $q_1$ and $q_2$ replace the eccentricity and argument of perigee to avoid singularities as defined in Chapter 3, and $\Omega$ is the longitude of the ascending node (also called the Right Ascension of the Ascending Node (RAAN)).

The first manoeuvre is the out of plane manoeuvre ($\Delta$ v$_n$) which compensates for errors on the inclination and RAAN elements. For DelFFi the main concern is the along-track separation (ie. in plane errors), and, as described in Chapter 4, the passive relative motion does not produce significant errors in the inclination and RAAN elements over the course of the mission lifetime. Further, the out of plane manoeuvre is cross-coupled to the $q_1$, $q_2$ and $u$ elements, meaning if there is an error in the actuation of this manoeuvre these elements will not be their expected values in the following in plane manoeuvre sequence, leading to a potentially serious propagation of errors. For this reason, the out of plane manoeuvre will not be performed by DelFFi. If GPS were available to provide feedback on the orbital elements during and after each burn, then the application of this out of plane manoeuvre would be desirable.

The three in plane manoeuvres are defined in the equations below [85]:

$$\Delta \mathrm{v}_1 = \frac{na}{3\mathrm{N}\pi}\left[\Delta u - \frac{3}{2}\Delta\bar{a}(u_1 - u_0) - 2\Delta\tilde{q}_0\right] + \frac{na}{4}\left[\left(\frac{\mathrm{M}}{\mathrm{N}} + 1\right)\Delta q - \left(\frac{\mathrm{M}}{\mathrm{N}} - 1\right)\Delta\bar{a}\right] \tag{5.1}$$

$$\Delta \mathrm{v}_2 = \frac{na}{4}(\Delta q - \Delta\bar{a}) \tag{5.2}$$

$$\Delta \mathrm{v}_3 = -\frac{na}{3\mathrm{N}\pi}\left[\Delta u - \frac{3}{2}\Delta\bar{a}(u_1 - u_0) - 2\Delta\tilde{q}_0\right] - \frac{na}{4}\left[\frac{\mathrm{M}}{\mathrm{N}}\Delta q - \frac{\mathrm{M}}{\mathrm{N}}\Delta\bar{a}\right] \tag{5.3}$$

Where $u_0$ is the current argument of latitude, $u_1$ is the argument of latitude at the time of the first burn defined by the following equation:

$$u_1 = \tan^{-1}\left(\frac{\Delta q_2}{\Delta q_1}\right) \tag{5.4}$$

To control the manoeuvre time variables M and N allow an adjustment of the location of the second and third burns. M is an odd positive integer that is equivalent to the number of half-orbits between the first and second burn. N is an even positive integer that is equivalent to the number of half-orbits between the first and third burn. This means that N must always be greater than M with the relationship: N$\geq$ M+1. For the DelFFi mission, the power budget states that there is sufficient power for a 30 minute burn every second orbit. In other words, there must be at least one non-actuated period between each manoeuvre. Thus M$\geq$ 3 and N $\geq$M+3. The remaining parameters are defined by the following three equations:

$$\Delta q = \Delta q_1 \cos(u_1) + \Delta q_2 \sin(u_1) \tag{5.5}$$

$$\Delta\tilde{q}_0 = \Delta q_1 \sin(u_0) - \Delta q_2 \cos(u_0) \tag{5.6}$$

$$\Delta\bar{a} = \frac{\Delta a}{a} \tag{5.7}$$

In all of the above equations, $\Delta$ of one of the orbital elements corresponds to the relative orbital element (Difference between the chief value and deputy value). For example, $\Delta a = a_{deputy} - a_{chief}$.

The operator can supply a maximum value for the length of the manoeuvre set (defines the maximum value of N), then the algorithm tests each of the acceptable combinations of M and N, and compares the magnitude of the total $\Delta$ v ($\Delta$ $\mathrm{v}_{tot} = \Delta\mathrm{v}_1 + \Delta\mathrm{v}_2 + \Delta\mathrm{v}_3$) to determine the best times to perform the second and third manoeuvres. Generally, longer time periods result in lower propellant costs as less acceleration is needed when more time is allowed to accomplish the desired position change.

### 5.2.3 Controller Performance in Simulation

To demonstrate the performance of the controller, four cases were considered.

1. QB50 nominal orbit with a 100 km wide control window, where the manoeuvres are planned to begin at the first time the deputy satellite crosses the ascending node ($\theta = 0$) within the pre-defined manoeuvre period (3 hours - 12 hours after receiving a TLE on the satellite). The second manoeuvre is predefined to be three half-orbits (M= 3) after the first and the third manoeuvre is six half-orbits after the first (N= 6).
2. Same as the first case, but with a 25 km wide control window.
3. Same as the first case, but allowing the controller to optimize M and N to minimize propellant.
4. Same as the first case, except using the QB50 elliptical orbit

Recall that the QB50 nominal orbit (defined in Chapter 4) is a circular (e = 0.001) orbit with an altitude of 380 km and a separation distance of 1000 km with a control window of ± 100 km with the chief pointing along the velocity vector and the deputy 2° off of the velocity vector). Further the QB50 elliptical orbit is a 400 km by 600 km orbit with the same separation distances control window and differential drag (pointing) conditions as the QB50 nominal orbit.

#### Non-Optimized QB50 Nominal Orbit Performance

For the first case, the manoeuvre that is planned requires a total $\Delta$v of 3.43 m/s divided into three manoeuvres of 1.66 m/s, 0.065 m/s and -1.71 m/s respectively. These can be related to the time required to complete the manoeuvres using the following equation, where $F$ is the magnitude of the thrust the propulsion system generates in Newtons, $t_b$ is the manoeuvre duration in seconds, and $m_{sat}$ is the mass of the satellite at the time of the manoeuvre.

$$F \cdot t_b = m_{sat} \cdot \Delta \text{v} \tag{5.8}$$

Using the theoretical values for the propulsion system, the magnitude of the thrust at beginning of life is 1.4 mN [55], and the mass of the satellite is the initial 3.64 kg. At the end of life, the thrust is 0.8 mN and the mass is 3.59 kg. For reference, a 20 minute burn corresponds to a 0.46 m/s manoeuvre at beginning of life (0.26 m/s at end of life). The power budget (see Section 2.3 for details) allows for a 20 minute thrust every second sun-lit orbit. Since the three manoeuvres are more than one full orbit apart (by virtue of M= 3 and N= 6), each of the three manoeuvre could be 20 minutes long without violating the power budget.

Even at beginning of life, the first (1.66 m/s) and third (-1.71 m/s) manoeuvres significantly exceed the desired maximum duration of 300 s (the duration of the three manoeuvres at beginning of life are 4316 s, 169 s, 4446 s), by end of life both the first and third thrusts last

longer than a full orbit (at the end of life, the manoeuvre durations would be 7449 s, 292 s, 7674 s). Not only does this exceed the desired maximum duration as specified by the thermal budget, but it also exceeds the power budget that limits the each manoeuvre to a maximum 20 minutes. For the moment, we assume this problem could be overcome and instead consider the effects of such a set of manoeuvres on the formation.

With this high cost, the deputy satellite is nearly returned to its nominal relative position, as seen in Figure 5.7.

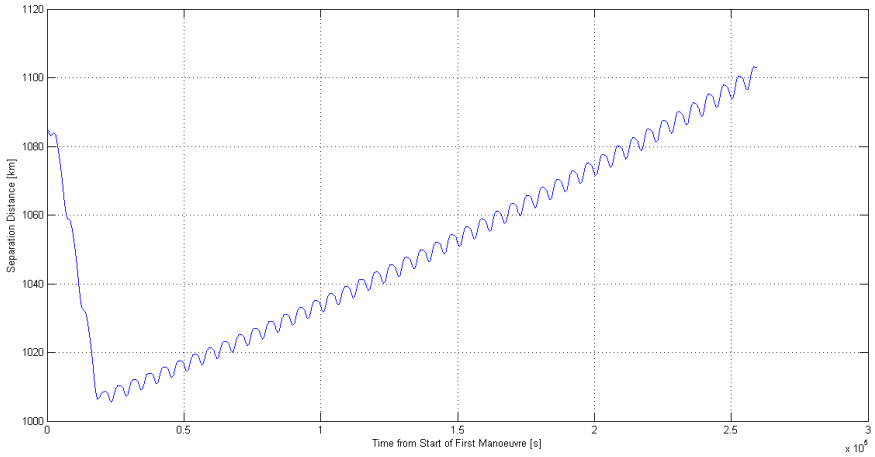



**Figure 5.7:** Non-optimized controller performance in terms of separation distance for the QB50 nominal orbit with 100 km control window width (Case 1).

Note that although the satellite is returned to nearly the nominal separation distance (closest approach is 1005.6 km), the satellites drift apart much more quickly than in the figures shown in Chapter 4. The reason for this difference is the degree to which the mean elements of the two satellites match (see Section 4.2.7 for more details). In particular, after the set of three manoeuvres are complete and the relative position has been corrected, there is still a large difference between the mean semi-major axis ($a$) of the two satellites. This is shown in Figure 5.8.

The addition to an average 235 m offset in the differential semi-major axis, there is also an oscillation that can be seen in Figure 5.8. This combination results in a much faster drift between the satellites than the Chapter 4 cases that assumed the mean orbital elements were matched precisely when the formation was initialized.

In addition to the rate of separation, Figure 5.7 showed that the nominal separation distance is only re-established with a 5.6 km accuracy. In this figure, the ideal case is shown, in which the actuator has no error (in either magnitude or direction), the thrust is applied exactly along the velocity vector and no navigation error is applied. This error on the re-initialized position is likely due to the fact that the linearization assumptions made in the development of the controller do not match the large control window dimension selected for the DelFFi

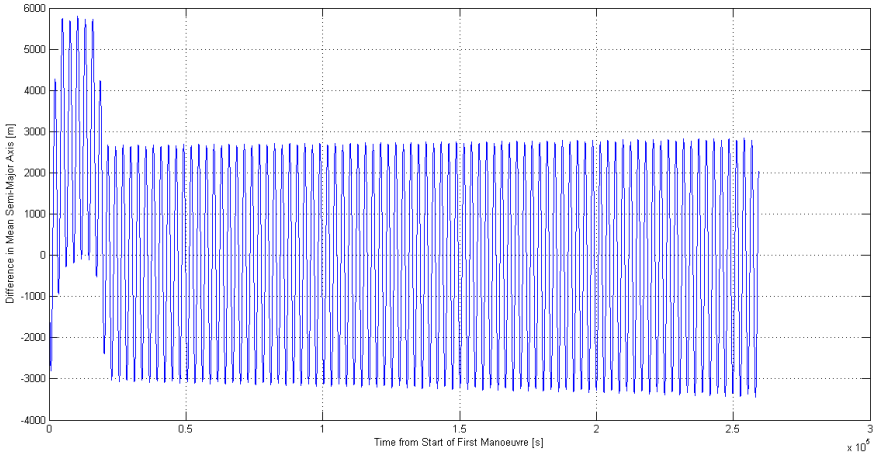



**Figure 5.8:** Non-optimized controller performance in terms of ability to match semi-major axis of the two satellites, for the QB50 nominal orbit with 100 km control window width (Case 1).

mission. Breger showed that Gauss' Variation Equations (GVEs) are only valid for relative orbital elements that equate to rectilinear distances of approximately 25 km for low Earth orbits [86]. Since Mueller's Analytical Solution for Circular Orbits is also based on GVEs, it is reasonable to expect that it, also, is only applicable to formation maintenance when the error on the separation distance is less than 25 km. Breger notes that control windows are generally between 10 m and 100 m for LEO formation flying missions, making the 25 km linearization validity limit acceptable [86], however for the DelFFi mission, a 100 km control window has been selected. Breger established the limit when the effect of an error on the orbital elements was larger than 0.01 on the norm of the control influence matrix $\mathbf{B}$ (as identified in Equation 3.8). Beyond Breger's limit, the controller does not become completely unusable, rather, the control accuracy decreases with the increasing control window size. For this reason, the second cases addresses a smaller control window of 25 km - Breger's limit for small formation maintenance error.

Another reason for the error on the corrected position is that the controller may not be suited for a long-duration low-magnitude thrust (rather than a truly impulsive thrust). In this simulation, the best case was assumed where the beginning of life (maximum) thrust was used, however the controller assumes the duration of each manoeuvre is small compared to the time between the manoeuvres - which may not be valid when the manoeuvre duration is nearly a full orbit and the time between manoeuvres is two or three days (assuming post correction manoeuvre separation rate).

### Non-Optimized QB50 Nominal Orbit with 25 km Control Window Performance

In the second case, a smaller control window is considered to determine if the size of the error on the separation distance affects the accuracy of the control algorithm. Breger showed that a 25 km error on the distance is a reasonable upper limit to avoid inaccuracies due to linearization in Gauss' Variational Equations [86]. Again using the QB50 nominal orbit, the

non-optimized version of the control algorithm (in which M= 3 and N= 6) is used to plan a set of manoeuvres that begin at the last time the deputy satellite crosses the ascending node before leaving the 25 km control window. Using the same Mean Orbital Elements to initialize the formation, the satellites passively separate to 25 km after 3.7 days. Assuming this were the first manoeuvre the satellites performed, at least one valid TLE would be available in this time, however were a manoeuvre to have occurred 3.7 days previously to initialize or acquire the formation for example, no valid TLE would be available - meaning an accurate measurement of the satellites' positions would not be available onboard. For the moment, we assume a GPS system or other position sensor is available, in order to address the linearization problem at hand.

The second case resulted in a set of three manoeuvres of 0.46 m/s, 0.039 m/s and -0.48 m/s respectively, totalling 0.98 m/s. The effect of these three manoeuvres is shown in Figure 5.9.

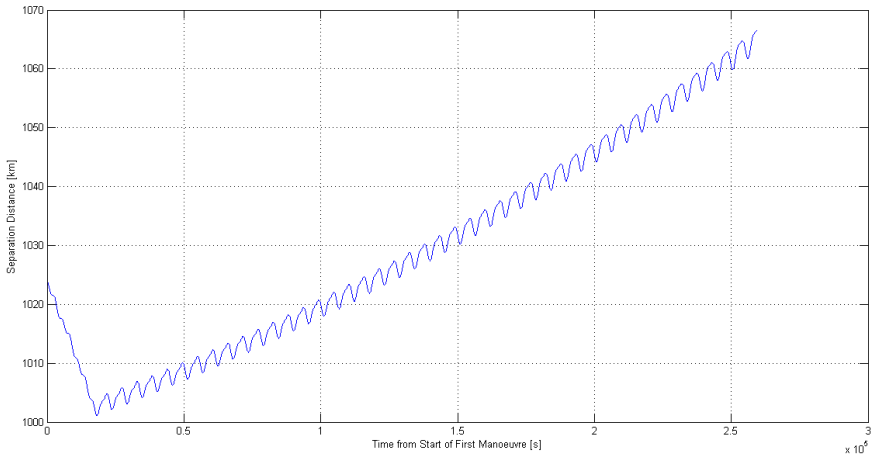



**Figure 5.9:** Non-optimized controller performance in terms of separation distance for the QB50 nominal orbit with 25 km control window width (Case 2).

In this case, the separation distance is reduced to 1001.2 km after the manoeuvre set. This confirms that the control algorithm is sensitive to the error on the separation distance that will be corrected. This means that if the control accuracy is to be improved, the control window size should be reduced. It can also be seen that the satellites again drift apart more quickly (approximately 33 hours to edge of the 25 km control window) than in the passive relative motion models of Chapter 4 (3.7 days). This is again due to the mismatched mean orbital elements - and in particular the differential mean semi-major axis which is visualized in Figure 5.10.

### Optimized QB50 Nominal Orbit Performance

The third case brings in the optimization to minimize the propellant usage. Rather than performing three manoeuvres in close succession (pre-defined one and a half orbits between

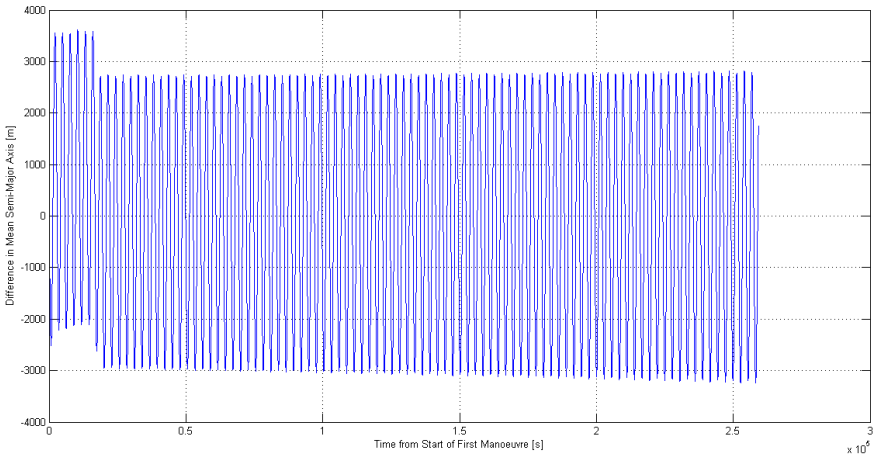



**Figure 5.10:** Non-optimized controller performance in terms of ability to match semi-major axis of the two satellites, for the QB50 nominal orbit with 25 km control window width (Case 2).

them), the timing of the second and third manoeuvres are now flexible with M between 3 and 27 and N between M+3 and 30. The initial conditions are the same as in the first case - the QB50 nominal orbit and 100 km control window. In this case, the control algorithm chooses M= 3 and N= 30 - meaning the total manoeuvre set takes nearly a full day from the start of the first manoeuvre to the end of the last manoeuvre. Further, the total $\Delta$ v required is reduced from 3.43 m/s to 0.70 m/s. The three manoeuvres are 0.30 m/s, 0.065 m/s and -0.34 m/s respectively. These correspond to 771 s, 168 s and 888 s at beginning of life (1330 s, 290 s and 1533 s at end of life). At beginning of life, all three of these manoeuvres meet the power budget requirement (20 minutes per manoeuvre), but do not meet the desired maximum duration from the thermal budget (5 minutes per manoeuvre). By end of life, the first and third manoeuvres exceed even the power budget requirement. In order to perform a 0.34 m/s $\Delta$ v in under 20 minutes, the acceleration would have to be greater than 0.283 mm/s$^2$. With the current thruster design, this could be possible for the first 11.3 hours of the burn time or the first 63% of the propellant budget (total 17 hours and 56 minutes burn time is available [55]). The control acceleration over thruster burn time is shown in Figure 5.11.

The effect the three manoeuvres have on the satellite using the beginning of life thruster model are shown in Figure 5.12 and 5.13.

Recall that the principle of the correction manoeuvre in all of these cases is the same. In order to close the separation distance between the two satellites, the deputy satellite (the one in the lead) must move into a higher orbit to allow the chief satellite to, in effect, catch up then the deputy returns to the original orbit. This is possible because, even though the orbital velocity in the higher orbits is faster, the satellite must cover a greater distance, so the relative velocity of the deputy (higher orbit) compared to the chief (lower orbit) is negative. To close a certain separation distance between the satellites, two options are possible. Either the deputy must move to a much higher orbit (than the chief) for a short time (as in the first case), or the deputy can move to a moderately higher orbit for a long time (as in this third

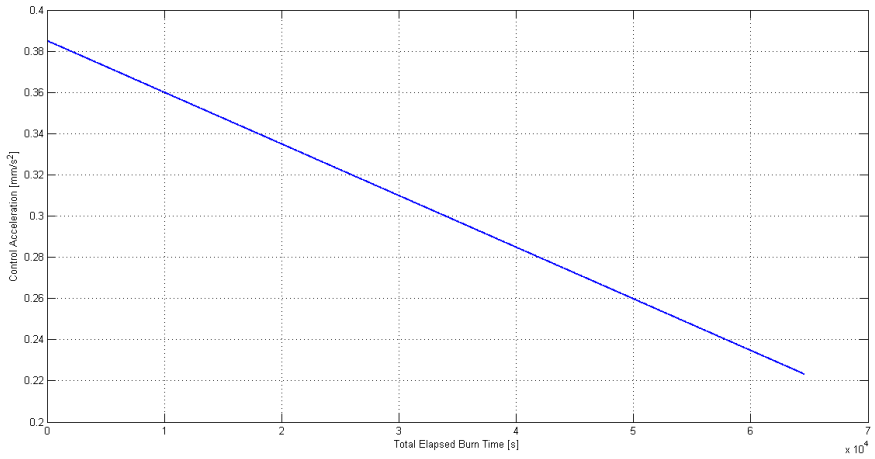



**Figure 5.11:** Magnitude of the control acceleration over time in which the thruster has been 'on' (total elapsed burn time since launch). Based on the theoretical propulsion system model from Poyck [55].

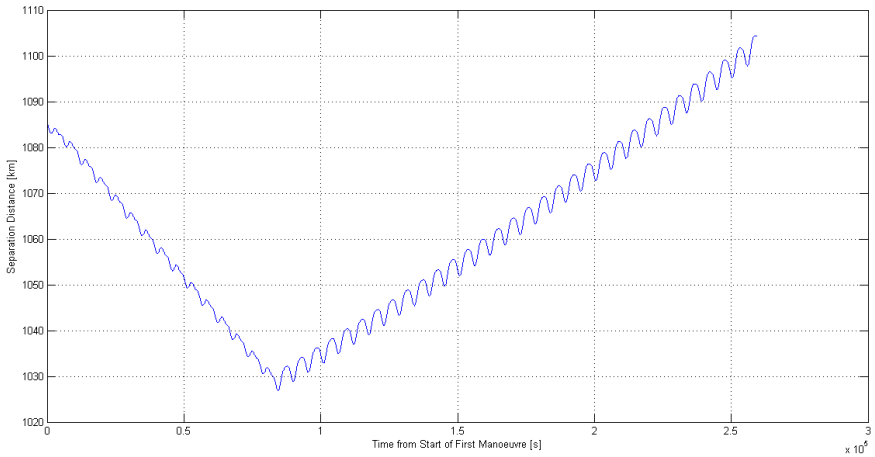



**Figure 5.12:** Optimized controller performance in terms of separation distance for the QB50 nominal orbit with 100 km control window width (Case 3). Control algorithm selected M= 3 and N= 30 to minimize propellant consumed.

case). Since the secondary orbit for the deputy is not as high in this case, the $\Delta$v required to reach it is reduced compared to the first case and the propellant is minimized. On the other hand, because longer is spent with different mean orbital elements compared to the chief, more inaccuracies are accrued by the deputy and the accuracy of the correction is reduced. Figure 5.12 shows that the closest approach of the two satellites after the correction manoeuvre is 1026.9 km - significantly worse than the 5.6 km error from the first case. Moreover, the mismatch of the mean semi-major axis is approximately as large as in the first case and the satellites breach the control window 1.95 days after the end of the third manoeuvre.

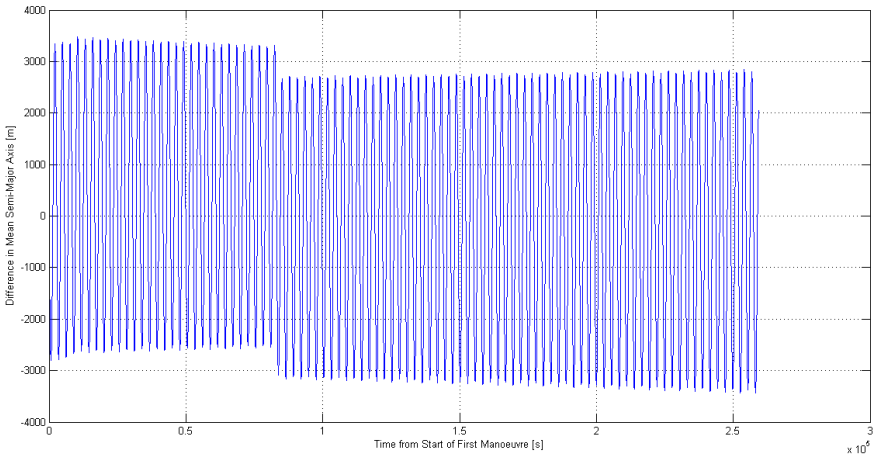



**Figure 5.13:** Optimized controller performance in terms of ability to match semi-major axis of the two satellites, for the QB50 nominal orbit with 100 km control window width (Case 3). Control algorithm selected M= 3 and N= 30 to minimize propellant consumed.

### Non-Optimized QB50 Elliptical Orbit Performance

The fourth case addresses the QB50 elliptical orbit. As in the second case, it is assumed that the manoeuvre is planned no later than three orbits before the deputy satellite exceeds the control window. This was chosen because the elliptical case remains in the control window for over seven days and a longer time between manoeuvres is desirable. The control algorithm uses the preset M= 3 and N= 6 values, rather than optimizing them, resulting in a total $\Delta$v of 4.73 m/s for the three manoeuvres (2.38 m/s, 0.024 m/s, -2.33 m/s respectively). As mentioned in Case 1, these manoeuvres exceed the power budget, however with optimization they can be reduced (in the same way as Case 3). The effect on the separtion distance and differential semi-major axis are shown in Figures 5.14 and 5.15.

Figure 5.14 shows that the three manoeuvres return the satellites to even closer than the nominal separation distance - achieving a separation distance of 959 km at the end of the third manoeuvre. After this correction (reinitialization), the satellites actually begin to drift closer together. This is because the average mean semi-major axis of the deputy is larger than that of the chief by roughly 30 m for the first two days after the end of the third manoeuvre, thus the chief will continue gaining on the deputy, albeit slowly. This large error likely due to the fact that the analytical solution is suited for circular orbits and the 400x600 km orbit is more elliptical than the solution is valid for. The range of validity of the controller was not specified by Mueller, however it is specifically listed as a solution for circular orbits [85]. It is apparent that the eccentricity of 0.015 (corresponding to a 400x600 km orbit cannot be neglected without accepting a large control error.

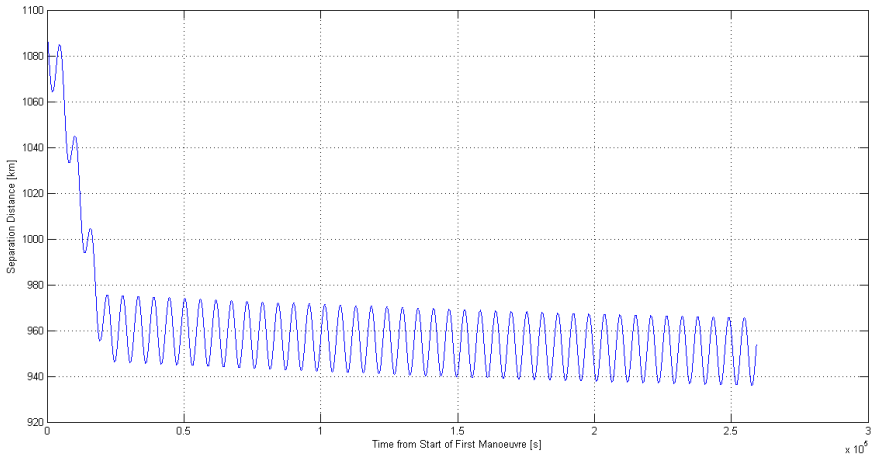



**Figure 5.14:** Non-optimized controller performance in terms of separation distance for the QB50 elliptical orbit with 100 km control window width (Case 4).

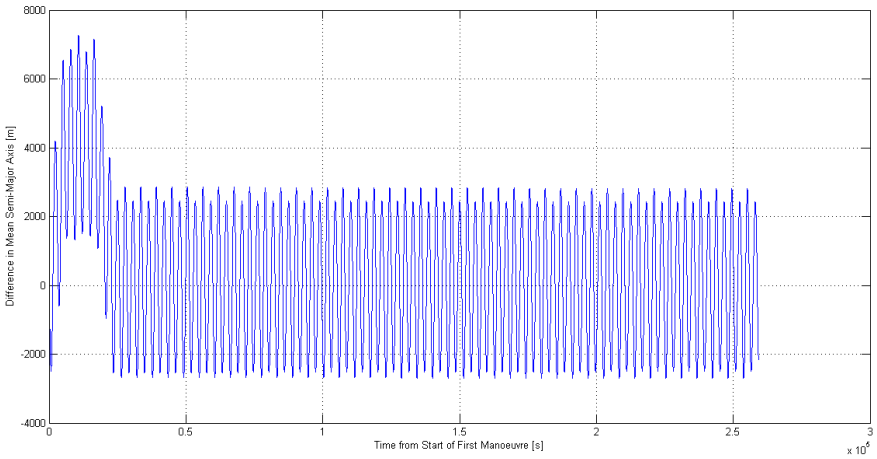



**Figure 5.15:** Non-optimized controller performance in terms of ability to match semi-major axis of the two satellites, for the QB50 elliptical orbit with 100 km control window width (Case 4).

### Simulation Performance Summary

The results of the four cases in terms of their accuracy at returning the satellite to the nominal position and the magnitude of the manoeuvres that are required are summarized in Table 5.1 below.

The above four cases showed that the chosen analytical controller is inaccurate when applied to large errors on the separation distance. This means that although the control algorithm is well suited for infrequent manoeuvres, it is poorly suited to large correction manoeuvres. Further, the optimization method of allowing additional time in the secondary orbit (between the second and third manoeuvre of the set), is effective at reducing the propellant consump-

**Table 5.1:** Analytical Controller Performance Summary

| Case | Post-Control Position [km] | $\Delta v_1$ [m/s] | $\Delta v_2$ [m/s] | $\Delta v_3$ [m/s] | $\Delta v_{Total}$ [m/s] |
|---|---|---|---|---|---|
| 1. Nominal | 1005.6 | 1.66 | 0.065 | -1.71 | 3.43 |
| 2. 25 km control window | 1001.2 | 0.46 | 0.039 | -0.48 | 0.98 |
| 3. Optimized M, N | 1026.9 | 0.30 | 0.065 | -0.34 | 0.70 |
| 4. Elliptical | 959.4 | 2.38 | 0.024 | -2.33 | 4.73 |

tion, however the consequence is larger inaccuracies in terms of the return to the nominal separation distance.

The range in values of a single $\Delta v$ ranged from 0.02 m/s to 1.71 m/s. The lower bound is within the accepted range (larger than the 0.012 m/s minimum $\Delta v$ that a 30 s burn at beginning of life would provide - ie. the worst case minimum burn). The upper bound however exceeds both the desired 300 s limit from the thermal budget (corresponds to 0.067 m/s at end of life - the worst case maximum) and the power budget's 20 minute limit (corresponds to 0.27 m/s at end of life). Even if only the optimized case is considered (Case 3) the first and third burns can only be performed in the first 63% of the thruster's total burn time.

In terms of the mission lifetime, the requirement is to maintain the formation for 20 days, with a target of 30 days. The propellant available on the two satellites totals 30 m/s (15 m/s each), with 1 m/s of that allotted for formation acquisition according to the estimate in [2]. This means up to 29 m/s are available for formation maintenance. It is assumed that the manoeuvre plan would only be required at most every six days (ie. the mean orbital elements could be made to match better than the control accuracy demonstrated in these simulations). For the non-optimized control algorithm (requiring 3.43 m/s for the QB50 nominal orbit) eight corrections (one correction is three manoeuvres) can be made - maintaining the formation for 48 days. In the optimized case, the $\Delta v$ per correction is reduced to 0.7 m/s, which means 41 corrections could be made (over eight months of formation maintenance). For the QB50 elliptical orbit, in the non-optimized case (4.73 m/s per correction, 6 corrections could be made (36 days of formation maintenance. All of these cases exceed both the required and target mission lifetimes. This analysis does assume that the satellites exchange roles (each taking their turn as the active deputy) at least once during the mission.

The most troubling finding is the large error on matching the mean orbital elements after the correction - especially in the optimized case. While a formation with precisely matched mean orbital elements would take over six days to separate beyond the control window, in the third case, it takes less than two days for the control window to be breached. This defies the six day passive relative motion requirement discussed in Chapter 4, and would mean that the next manoeuvre set would have to be planned without receiving a new TLE (ie. based only on onboard propagated values). This is not a reasonable plan, as the onboard propagator has large errors even when only considering the passive relative motion. When it also must model the control accelerations, further errors are accrued and make a reliance on the

onboard model dangerous to the formation safety (collisions or unrecoverable configurations may occur). This effect is expected to be true of most analytical controllers based on Gauss' Variational Equations due to their linearizing assumptions. To improve the performance of this calls of control algorithms, a smaller control window - requiring smaller, more frequent thrusts to maintain - should be used. To do this while meeting the six day passive relative motion requirement due to the TLEs, would mean that the effect of differential drag needs to be reduced - either by decreasing the attitude pointing error or moving to a higher orbit with less absolute drag.

### 5.2.4 Controller Performance on Target Processor

The microcontroller chosen by the ADCS team is the Texas Instruments TM4C1294NCPDT [59]. This processor has a clock frequency up to 400 MHz, however the requirements for the DelFFi project specify a minimum clock frequency of 100 MHz, so this will be used to test the control algorithm on the microcontroller. Development for this processor is enabled by the Tiva C Series Connected LaunchPad Evaluation Kit. This board includes an embedded programmer such that the processor can be programmed and debugged using Code Composer Studio. After generating the 'C' code using Simulink, it is imported into an existing project specified for the TM4C1294NCPDT processor in CCS. The code is then debugged with breakpoints to track the values of critical variables (the manoeuvre magnitudes and positions) and to determine the time required to run the control algorithm.

When the code was executed on the microcontroller, it was found that the output values (manoeuvre magnitudes and positions) were practically identical. For this test, case 3 from Table 5.1 was used. This was the QB50 nominal orbit with the optimized controller (up to a maximum M = 27 and N = 30). The largest discrepancy was in the magnitude of the first manoeuvre, which varied by $0.1367\text{x}10^{-12}$ m/s (on a magnitude of 0.2965 m/s). This is much smaller than the impulse bit of the propulsion system (0.22 m/s at end of life, see Section 5.3) and is considered acceptable. The processing time for one iteration including optimization of the M and N values up to a maximum N = 30 and a maximum M = 27, was 5.506 milliseconds (550588 clock cycles), using a clock speed of 100 MHz. This is significantly less than the loop rate on the ADCS microcontroller (1 Hz) and as such meets the requirements.

### 5.2.5 Controller Accuracy Analysis

The accuracy of the controller is difficult to generalize given the number of unknowns in the actuator design. However, it is possible to discuss the main sources of error and their expected affect on the system to assess if the controller accuracy will be sufficient for the DelFFi mission.

Chapter 4 showed that even a 10 km control accuracy (the requirement for a 1000 km separation based on Gill's scaling law), would have a significant impact on the system's ability to match the mean orbital elements sufficiently well to avoid rapid departure from the control

window due to differential drag. Thus, although the exact maximum accepted control accuracy has not been established, increasing the control accuracy beyond this level (10 km) is not acceptable. Although Gill's scaling law suggests that to meet a 10 km control accuracy a 1 km navigation accuracy is needed, it may be possible to meet the 10 km control accuracy despite the navigation accuracy being larger than 1 km if, and only if, the other contributions are sufficiently small so as to balance the one large contribution.

In Chapter 4, the navigation accuracy was estimated to be $\pm$ 5.1 km. This section will describe this and other sources of inaccuracies (thrust magnitude accuracy, pointing and misalignment, and mass estimate) and provide an estimate of their magnitude and effect. To do this, it is first noted that control errors cause problems in two (related) ways. First, they decrease the magnitude of the effective acceleration in the planned/desired direction, and second, they cause unwanted disturbance forces or torques on the satellite.

#### Error on Manoeuvre Magnitude

In order to estimate the effect of errors on the manoeuvre magnitude on the control accuracy, four main contributions must be assessed: the instantaneous magnitude of the thrust produced by the propulsion system, the pointing error due to the thruster misalignment, the ADCS attitude pointing error and the instantaneous mass estimate. From these contributions, the total error on the $\Delta$v can be estimated.

The first contribution to the control accuracy is the difference between the actual magnitude of the thrust produced and the magnitude of the thrust expected by the onboard actuator model. In order to reduce the error on the modelled thrust magnitude, the thruster will measure the tank pressure then calculate the instantaneous thrust, using the actuator model described in Section 5.1.2 of this report. This model will be calibrated using data from the flight hardware to ensure that even if the thruster does not match the theoretical model, the experimental performance will be well matched. Calibration is critical, as seen in the work of Migliaccio, where a 50% difference in performance between the theoretical and experimental thruster was found for Delfi-n3Xt. Although the DelFFi and Delfi-n3Xt systems are very different, they are both being developed by students in a university environment and must be tested thoroughly before flight. This calibrated model means that the error on the thrust magnitude will be driven by the precision of the pressure measurement. The tank pressure will be measured with a sensor with less than 0.5% error (as per conversation with A. Cervone August 19, 2015), and since the thrust magnitude is linearly related to the tank pressure, the thrust magnitude should be well known to within 0.5%. This relationship is true both at the beginning of life when the thrust is at its maximum value, and at the end of life when it is at its minimum value.

In a similar way, the propulsion system will estimate the instantaneous mass of the satellite. The mass is determined from the mass flow rate, which is calculated from the measured pressure. Since the mass flow rate and pressure are linearly related, the 0.5% error on the pressure, translates to a 0.5% error on the mass flow rate. The nominal mass flow rate is 0.775 mg/s (50 g in 17 hours and 56 min [55]), and assuming the thrust time is recorded with an

accuracy of 1 second, the error on the mass consumed will be 0.775 mg, which is 2.1x10$^{-5}$% of the total satellite mass.

The thrust magnitude in the desired direction is also influenced by the angle between the desired direction and thrust direction. This angle is formed by three sources: the misalignment of the thrust vector compared to the nozzle axis (0.1° as per [54]), the misalignment of the nozzle compared to the longitudinal axis of the satellite (3° as per [54]) and the attitude pointing error from the ADCS (2° in the Thrust Vector Control Mode [58]). As discussed in Chapter 4, the attitude pointing error has a significant impact on the system. The magnitude of the thrust is related to these pointing errors using the cosine law in the equation below.

$$F_{AT} = F_{tot} \cdot \cos(\phi) \tag{5.9}$$

In particular, at an attitude pointing error of 2°, the angle between the produced thrust and the velocity vector ($\phi$) is 5.1°. The error on the thrust is given by the following equation:

$$\epsilon_F = \frac{F_{tot} - F_{AT}}{F_{tot}} = 1 - \cos(\phi) \tag{5.10}$$

This means the error on the thrust ($\epsilon_F$) due to a 5.1° total pointing error ($\phi$) is 0.4%. The effect on the magnitude of the thrust for a range of attitude pointing errors is plotted below in Figure 5.16.

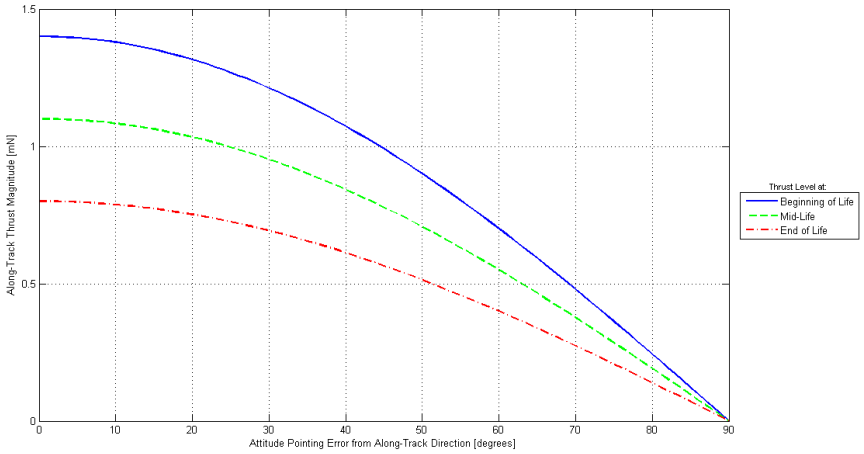



**Figure 5.16:** Magnitude of the thrust in the along-track direction (desired direction) as a result of various attitude pointing errors. The three lines correspond to the beginning, middle and end of life thrust magnitudes.

Returning to the navigation accuracy, it directly affects the $\Delta$v and can be related to the $\Delta$v by first assuming that the navigation accuracy is purely an error in the along-track position estimate. For this analysis the worst case is considered – the navigation accuracy found in Chapter 4 is taken as the absolute navigation accuracy for each satellite, and the satellites

have the worst case errors. In other words, we assume that the forward (deputy) satellite is 5.1 km further ahead in reality than the estimator expects, while the rear (chief) satellite is 5.1 km further behind. This means there is as much as a 10.2 km error on the separation distance. This could be compared to either the nominal separation distance or the maximum separation distance to determine the relative error. Here we take the maximum separation distance (1100 km) since the error we are concerned with is at the time when control is applied. A 10.2 km error on a measurement of 1100 km, corresponds to 0.93% error. This is an underestimation of the error at any time, but is suitable for a preliminary estimate of the control accuracy. To convert this error on the separation distance to an error in the magnitude of the $\Delta$v calculated by the controller, first the separation angle error must be found. For small errors, it can be assumed that the separation distance and separation angle are linearly related (by small angle theorem for the sine function), so the error on the separation angle is also estimated at 0.93%. To determine the magnitudes of the $\Delta$v using the analytical control algorithm discussed above, the estimated separation angle is compared to the desired separation angle to find $\Delta u$. Thus, this $\Delta u$ also carries the same error as the separation angle of 0.93%. Returning to the equations of the controller, we assume the entire error is in the $\Delta u$. Since, $\Delta v_1$ and $\Delta v_3$ are each linearly related to $\Delta u$ they each carry 0.93% error, while $\Delta v_2$ does not depend upon $\Delta u$ and carries no error in this simplified analysis. To find the error on the total $\Delta$v, the errors are applied to the calculated $\Delta$v's from the non-optimized QB50 nominal orbit case from the Performance in Simulation section above. The three $\Delta$v's were 1.66 m/s, 0.065 m/s and -1.71 m/s respectively, totalling 3.43 m/s. Applying a 0.93% error to the first and third manoeuvre results in a 1.06% error on the total $\Delta$v due to the navigation accuracy of 5.1 km.

The effects of these various contributions are summarized in Table 5.2.

**Table 5.2:** Summary of contributions to control error.

| Contribution | Value | Comment |
|---|---|---|
| Applied to Thrust Magnitude | | |
| Propulsion System Estimate | 0.5% | Based on pressure measurement |
| Pointing Error | 0.4% | 2° attitude pointing error and 3.1° thruster misalignment |
| Applied to Mass | | |
| Propellant Consumed Estimate | $2.1\text{x}10^{-5}$% | Based on pressure measurement |
| Applied to $\Delta$v Directly | | |
| Navigation Error | 1.06% | Considering 5.1 km navigation accuracy |

To combine these contributions, the thrust and mass errors should be converted to an error on the $\Delta$v produced. First, the errors that affect the thrust magnitude should be combined by adding them together since they are independent contributions. This means the error on the thrust magnitude is 0.9%. To combine this with the error on the mass estimate recall Equation 5.8 which can be rewritten as:

$$\Delta \text{v} = \frac{F}{m_{sat}} t_b \tag{5.11}$$

Since the thrust and mass are related by division to the $\Delta$v, their errors should be combined by adding the sum of the squares as shown in the following equation:

$$\epsilon_{\Delta \mathrm{v},F,m} = \sqrt{(\epsilon_F^2 + \epsilon_{m_{sat}}^2} \tag{5.12}$$

For the errors in the Table above, the error on $\Delta$v due to the thrust and mass errors is 0.9%. Since the error on the mass is so much less than that of the thrust, its effect is not noticed. This error can now be combined with the error on $\Delta$v due to the navigation error by simple addition since the contributions are independent, for a total error on $\Delta$v of 1.96%.

To translate this into an error on the separation distance to compare with the 10 km control accuracy requirement would require a detailed analysis of the various ways the $\Delta$v affects the return to the nominal position. A simplified way to estimate this effect is to imagine the simplest correction manoeuvre set that can be used to return the deputy from a position too far ahead of the chief. First the deputy is moved to a higher orbit by a propulsive manoeuvre, it is then allowed to travel at the lower orbital speed for some time (allowing the chief to effectively catch up), then the deputy is returned to the nominal altitude by a final burn. The along-track distance that the satellite is corrected by ($\Delta d_{AT}$) (based on the time spent in the higher orbit ($\Delta t_{ho}$), is at most the control window dimension (100 km for DelFFi). The distance that is travelled can be related to the $\Delta$v by the following relationship:

$$\Delta d_{AT} = \Delta \mathrm{v}_{AT} \cdot \Delta t_{ho} \tag{5.13}$$

This shows that the along-track distance correction manoeuvre is linearly related to the magnitude of the $\Delta$v, and so the error on the along-track distance correction is 1.96% as well. For a desired 100 km correction (maximum) the control error due to the actuator (propulsion system and ADCS) is 1.96 km.

In addition to the control accuracy due to the actuator errors (1.96 km), the navigation accuracy (10.2 km on the separation distance in the worst case) and the intrinsic control algorithm accuracy operate as independent contributions to the magnitude of the control accuracy. In terms of the controller itself, Mueller acknowledges that the analytical controller is suited for coarse formation maintenance only, though the accuracy is highly dependent on the specific case under consideration [85]. For the QB50 nominal orbit it was shown that the (non-optimized) control algorithm could return the formation to its nominal separation distance with an error of 5.6 km. In total, the control accuracy is thus 17.76 km in the along-track direction. As shown in Table 5.3.

#### Other Effects due to Control Errors

Besides the decrease of the magnitude of $\Delta$v in the desired direction the control errors also manifest in the discretization of the thrust magnitude due to the minimum impulse bit of the system, disturbance forces along the cross track and radial directions and disturbance torques that cause the satellite to rotate and increase its attitude pointing error. These contributions

**Table 5.3:** Summary of the contributions to the total control accuracy.

| Contribution | Value | Comment |
|---|---|---|
| Actuator Errors | 1.96 km | Thrust, mass estimates |
| Navigation Accuracy | 10.2 km | Error on separation distance |
| Control Algorithm | 5.6 km | Non-optimized, QB50 nominal orbit |
| Total | 17.76 km | Main contributions only, No safety factor |

are discussed qualitatively below.

In addition to the thrust magnitude being mis-estimated, the minimum impulse bit of the thruster can also determine whether the system will be able to produce the desired $\Delta$v with sufficient accuracy. An impulse bit is the minimum resolution that the impulse (in this case represented as the $\Delta$v) can be controlled to. For example, an impulse bit corresponding to 5 s would mean that a 40 s burn or 45 s burn would be possible but a 42 s burn would be rounded to 40 s causing an error on the $\Delta$v provided. This is considered a quantization error. The impulse bit has not been identified for the DelFFi propulsion system, but can be assumed to be well under the loop rate of the propulsion system microprocessor. This loop rate has not been selected either, but in the worst case will be the same as the ADCS loop rate (the ADCS algorithms require significantly more processing resources and its loop can be considered a conservative estimate of the propulsion system loop speed), which is 1 Hz. At the beginning of the mission, the impulse bit will be 0.39 mm/s and will reduce to 0.22 mm/s at the end of life. Since this is well below the magnitudes identified in section 5.2.2, the effect of the impulse bit on the system will be very low.

Thrusting in the non-desired direction (where the desired is the velocity vector direction) would cause a disturbing acceleration in either the radial or cross track direction. Its effect would be significantly smaller than a disturbance in the along-track direction, because disturbances in the along-track grow secularly, while those in the cross-track and radial result in periodic disturbances that have a smaller dimension.

Misalignment of the thrust vector also causes a disturbance torque about the satellite's centre of mass. The disturbance torque generated by the maximum 2° misalignment required for the thruster, [54], has been taken into account with the design of the reaction wheel. If the reaction wheels do not function, or stop functioning after some time in orbit, this compensation will not be possible and the duration of thrusts will have to be limited to ensure the pointing error does not become too large. In both the thrust vector control mode and the vector pointing mode (which uses magnetorquers but not reaction wheels to maintain the pointing direction), the rotation rate must be controlled to below 1°/s. Considering this level, if a 30 second burn were applied (the minimum recommended for efficient functioning of the propulsion system according to T. van Wees in April 2015), this would cause an angular pointing error of 30 degrees, which, first, as discussed in Chapter 4 would be extremely problematic to the differential drag and second would decrease the magnitude of the thrust in the desired direction even further. This could result in the system switching to the Safe Mode, where

the attitude pointing vector is not controlled (or is controlled to a lower accuracy). In this case, it is possible that the differential drag will cause the separation distance of the satellites to change dramatically and make re-acquisition of the formation difficult. This is a serious consequence that must be addressed in case the reaction wheel development is unsuccessful.

### Control Accuracy Summary

The control accuracy analysis performed in this section showed that due to the errors in the magnitude of the thrust (both from the thrust generation and the pointing error), the mass estimate and the navigation accuracy, the control accuracy will be larger than 17.76 km. Further, the effects of disturbance torques from misalignment of the thruster were shown to be serious and must be compensated for (using reaction wheels, for example) in order for the formation maintenance algorithm to function properly.

Several conclusions with respect to the control accuracy are clear.

1. The navigation accuracy due to the TLEs is too large (5.1 km) to meet the 10 km control accuracy according to Gill's Scaling Law

2. Considering only the main contributions, the total control accuracy is estimated to be 17.76 km, which is larger than the required 10 km accuracy, meaning the current subsystems and navigation accuracy are insufficient to support formation maintenance for DelFFi

3. The attitude pointing error produces a large effect on the control accuracy, and further influences the differential drag, and disturbance torques that the system must compensate for

4. The use of reaction wheels to mitigate propulsive disturbance torques is necessary, especially if the thrusts will last longer than a few seconds

5. Calibrating the propulsion system is absolutely critical to maintaining a reasonable onboard position estimate

It is clear that without (1) an improvement in attitude pointing error, (2) experimental verification of the reaction wheel system and (3) a proper thruster calibration model, the contributions of manoeuvres to the control accuracy will be too large to allow the TLE navigation accuracy to be accepted by the DelFFi team. With the current level of development of the DelFFi satellites, and considering the expected development of the current in-progress iterations, it is expected that the control accuracy will not meet the requirements using TLEs. Despite this fact, it is possible to consider how the satellite development should proceed, such that Formation Maintenance could be used in the future when the error contribution from the navigation accuracy is decreased.

## 5.3 Actuator Considerations

Based on the results of testing the controller in the context of the QB50 nominal orbit, some recommendations can be made for the actuator design. For the DelFFi mission, the actuator consists of two separate subsystems: the propulsion system (responsible for the magnitude of the thrusts) and the attitude determination and control subsystem (ADCS) (responsible for the direction of the thrusts).

The propulsion system should be able to provide thrust magnitudes in the range of 0.02 m/s to 1.71 m/s at a minimum based on the results of the Controller performance in simulation for the QB50 nominal and elliptical orbits. Some additional range on these values would be beneficial. Based on the current propulsion design, the propulsion team has recommended manoeuvres be limited to between 30 s and 300 s - corresponding at beginning of life (1.4 mN thrust) to a $\Delta$v range of 0.0115 m/s to 0.1154 m/s (as per conversation with T. van Wees April 2015). The range was selected based on the thermal model of the satellite to ensure first that the system does not overheat, and second that the energy spent pre-heating the system is not as large as the energy required to perform the manoeuvre. It is not wide enough to meet the requirements of either the non-optimized or optimized case for neither the nominal nor the elliptical QB50 orbits. This range was preliminary and will be further detailed after the breadboard design has been tested. Depending on the results of these tests, further iteration between the control algorithm's demands and the propulsion system's capabilities may be required. The power budget however is more strict in limiting the power available to the thruster to 20 minutes every second sunlit orbit. A twenty minute thrust corresponds to a $\Delta$v of 0.46 m/s at beginning of life and 0.27 m/s at end of life. As discussed previously, these values are sufficient for the optimized control algorithm, but only for the first 63% of the total available burn time. The original requirements on the propulsion system indicate that a maximum thrust level of 9.5 mN was considered [54]. Were this to be achieved, the propulsion system would be more than capable of providing the necessary $\Delta$v even in the non-optimized cases (where the accuracy is better). This feedback will be provided to the propulsion team to be considered in future design iterations.

On the side of the ADCS, the angular pointing error will combine with the thrust misalignment angular error to cause the desired thrust to be in the wrong direction. This will mean a smaller acceleration than desired is delivered and an additional perturbing acceleration is felt by the satellite. This perturbing effect has not been modelled at this time, but should be once the DelFFi ADCS design, and in particular the reaction wheel development, has been finalized. More critical than the pointing error, is the rotational rate error. As the manoeuvre is performed, the misalignment of the thruster and any angular pointing error will result in a torque that will cause the satellite to begin tumbling. This will then increase the magnitude of the pointing error, leading to an even greater increase in the rotational rate. This rate is generally controlled using a reaction wheel to "absorb" the rotational disturbances to later be off-loaded using the magnetorquers [26], [88]. Based on the experience of the CanX-4/5 team, it is possible to control the rotational rate to an acceptable level, however the analysis for the dimension of reaction wheel necessary for the DelFFi case has not been performed since there are too many unknowns for the propulsion system and expected angular pointing errors. Currently, the ADCS is required to maintain a rotational rate below $1^\circ$/s. For a 300

s long manoeuvre, this corresponds to nearly a full rotation of the spacecraft, this is clearly not tolerable. For a comparison, arbitrarily assuming that the thrust in the desired direction should be no more than 15% less than the expected (desired) magnitude, which corresponds to a 30° angular pointing error, a 300 s thrust would require the rotational rate to maintained below 0.1°/s. This is a significant difference from the nominal requirement (1°/s) and should be studied by the ADCS team to determine the ramifications. Similarly, a 15% reduction in thrust magnitude and a 30° attitude pointing error is significant and it has already been shown that attitude pointing errors of this magnitude are not acceptable for the passive relative motion requirement of six days. If the system switched away from TLEs to GPS for example, this attitude pointing error may be tolerable. A detailed analysis of the effect of the disturbance torques and rotation rates on the controllability and passive relative motion of the system should be performed in the future.

The ADCS team must also consider the fact that the manoeuvre plans developed by this analytical controller require two thrusts in the positive velocity pointing (along track) direction, followed by one in the opposite direction (still in the along-track but in the negative velocity vector direction). This means that between the second and third manoeuvre a 180° attitude rotation is needed. At this time, the effects of such a manoeuvre on the pointing stability are not quantified, but should be in the future.

## 5.4 Conclusions

In this chapter, a MATLAB Simulink model was developed to allow testing of various aspects of the formation flying payload. It was divided into two aspects: a Planning Sub-Mode model that received TLEs and developed a series of manoeuvres to return the formation to its nominal relative positions, and a Monitoring Sub-Mode model that received the manoeuvre plan and propagated the relative state of the satellites to ensure the proper execution of the manoeuvre plan. This model was modular to allow various blocks to be replaced to test different sensors, actuators, controllers and navigation algorithms.

After this, the design of the TLE-based controller was developed. After reviewing several existing formation maintenance controllers, two potential controllers were identified: an analytical controller for circular orbits [85] and a Model Predictive Controller [86]. Since the purpose of this study has been to assess the feasibility of TLE-based navigation for the FFP, the less complex analytical controller was chosen to be implemented. For each of the QB50 orbits (nominal and elliptical), a sequence of three manoeuvres was determined to return the formation to its nominal state, requiring a total $\Delta$v of 3.43 m/s and 4.73 m/s, respectively, in the non-optimized control algorithm cases. These total $\Delta$v magnitudes were divided unevenly over three burns with a range of values between 0.02 m/s and 1.71 m/s. This is a larger range than the thermal and power budgets for the propulsion system allows, thus further optimization of the control algorithm and relaxing of the requirements from the propulsion system should be considered. The Planning Sub-Mode was converted to C to be executed on the target microprocessor, which showed practically no difference in performance compared to the Simulink results. The run time for the Planning Sub-Mode was 5.51 ms, which is well

under the required 500 ms from the ADCS processor.

In assessing the ability of the controller to return the satellites to their nominal position, it was shown that the calculated manoeuvre plan was able to return the deputy satellite to the nominal separation distance (within 5.6 km for the QB50 nominal orbit with the non-optimized control algorithm), however the mean orbital elements were poorly matched to those of the chief causing rapid separation (two days after the correction manoeuvres, the control window was breached). The inherent accuracy of the control algorithm is attributed to the linearization validity of the controller as it is based on GVEs. Allowing the satellites to operate within a control window as wide as 100 km results in a contribution to the control accuracy intrinsic to the controller design of 5.6 km for the QB50 nominal orbit. The control accuracy and manoeuvre magnitude ($\Delta$v) were provided for (1) the QB50 nominal orbit, (2) the QB50 nominal orbit but with only a 25 km wide control window, (3) the QB50 nominal orbit with the optimization of M and N to minimize propellant cost and (4) the QB50 elliptical orbit. It was shown that although the optimization technique of the analytical controller allows the propellant cost to be reduced from 3.43 m/s to 0.70 m/s, the control accuracy was negatively impacted - increasing from 5.6 km for the non-optimized algorithm to 26.9 km for the optimized algorithm.

The control accuracy was then estimated based on the effects of the propulsion system thrust magnitude error, the misalignment error, the attitude pointing error, the error in mass estimate and the navigation accuracy. These contributions resulted in an error of 1.96% on the $\Delta$v. Using Gauss' Variational Equations, and assuming a circular orbit, this can be compared to an error on the along-track correction distance (at most the control window dimension of 100 km), which corresponds to an error of 1.96 km. In total The control accuracy is thus 17.76 km, which is too large to meet the control accuracy requirement of 10 km established by Gill's Scaling Law and discussed in Chapter 4. This value was found using the major contributions that could be quantified with the currently level of development for DelFFi and requires several assumptions. However, it is clear that the majority of the error is due to the navigation accuracy, which must be reduced in order for such a formation maintenance algorithm to be successful. Other effects related to inaccurate control (disturbance torques, control forces in undesired directions, and impulse bit size) were discussed qualitatively, with the conclusion that the attitude pointing error and should be minimized as far as possible, and the rotation rate should be compensated for, especially during a propulsive manoeuvre. In the future, these other effects should be assessed using a statistical model (such as a Monte Carlo analysis) to determine their impact on the control accuracy. This accuracy analysis has been performed with the QB50 nominal orbit in mind and is only valid for that case. It is expected that other orbits would have similar results for many aspects, however, the control accuracy cannot be directly generalized to any CubeSat orbit.

Finally, based on the accuracy analysis and the results found in Chapter 4, some recommendations for the development of the propulsion system and ADCS as supporting subsystems for a formation maintenance payload are made. First, the propulsion system should ensure a calibrated model of the thrust based on the measured pressure is included in the onboard manoeuvre monitoring software. Using feedback from the pressure sensor can ensure that the error on the estimated thrust (and thereby the error on the estimated $\Delta$v) is minimized. For

the ADCS, a reduction in the attitude pointing error at least at the times when manoeuvres will be performed is desirable, however the limitations are more strict on the attitude pointing error for the passive relative motion in Chapter 4 (0.5°) than those discussed in the control accuracy analysis (2°). The more important requirement on the ADCS from the control accuracy analysis is the need to reduce the rotation rate to at the absolute most 0.1°/s during manoeuvres (assumes a 30° angular pointing error builds up during a 300 s long manoeuvre where the rotation rate is constant throughout). This capability will be critical to ensuring the satellite does not enter Safe Mode during a manoeuvre as a result of the disturbance torques the propulsion system will generate. Entering Safe Mode will interrupt the formation maintenance process and could cause the satellites to drift to an unrecoverable arrangement if not addressed quickly due to the extreme effects of differential drag. Analysis and Testing of the reaction wheel in light of the disturbance torque is critical to the continuation of the FFP. These requirements to the supporting subsystems will not guarantee that TLE-based formation maintenance on CubeSats is possible. However, they will be a significant step towards making CubeSat formation maintenance feasible.

# Chapter 6

# Operational Considerations

Now that Two-Line Elements (TLEs) have been shown as a viable navigation data source, it is necessary to determine what the effect of this choice will be on the operations of the CubeSats. This chapter will first address the characteristics of the mission from the perspective of a distributed system, then discuss formation flying operations in each of the mission phases.

## 6.1 Distributed System Philosophy

Consisting of two satellites, DelFFi will be the first mission of the Delfi programme that can benefit from the advantages of a distributed system. A distributed system can be described as a set of independent elements, often known as agents, that coordinate using messages to achieve a common goal [1]. A distributed system architecture allows for greater flexibility in the mission as it is possible to specialize individual agents or to use one agent to compensate for the failure of another agent, increasing the reliability of the system overall through redundancy. In general, a distributed system can be described by Figure 6.1 below [1].

The Mission Planner, shown in Figure 6.1, autonomously decomposes a high-level goal into a set of cooperative tasks, while the Allocator autonomously assigns tasks to the individual agents, then the local controllers are used to achieve the tasks. In the case of DelFFi, there are two agents - the two satellites, Delta and Phi - and the Wireless Network they communicate through is either the Intersatellite Link (ISL) or the ground communication link. This thesis has mainly focused on the Local Controllers for the two satellites, which in this case will be identical, although they may be in different states during operation. At this time, the Mission Planner and Allocator will be handled by the ground operations team as they will not be closely related to the short term operations of the satellites and maintaining an operator's oversight will mitigate risks of improper operations. Once the local controllers have been successfully demonstrated, the transition to full onboard autonomy can be made.

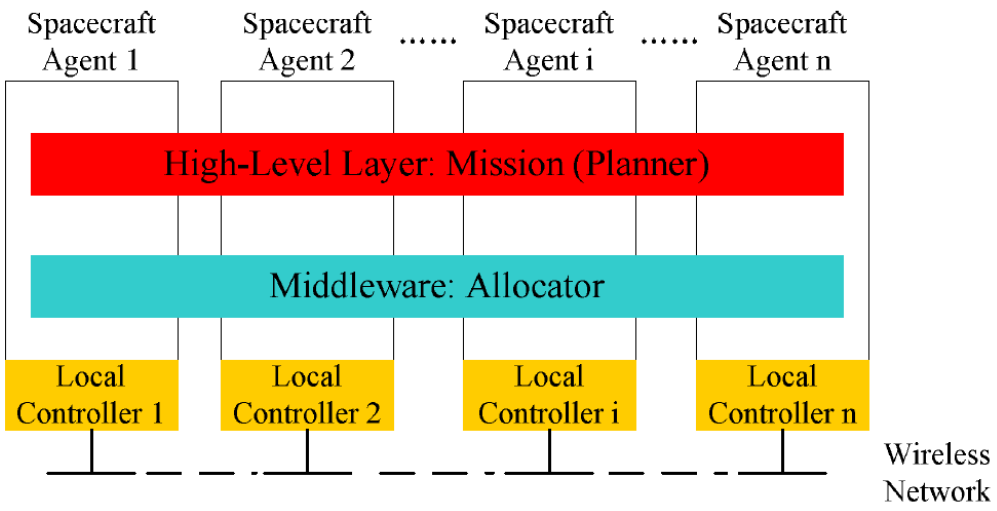



**Figure 6.1:** General architecture of an autonomous distributed system [1].

As was mentioned previously, a Chief-Deputy arrangement has been chosen for DelFFi, in which the deputy actively controls its position relative to the passive chief. If the entire mission was performed where one of the two satellites (for instance Delta) was the chief, while the other (Phi) was the deputy, only one satellite (Phi) would consume its propellant. This is undesirable, first, because it would result in a waste of fuel of the chief satellite (Delta), but also because the difference in ballistic coefficient between the two satellites would grow over time. Instead, the role of chief and deputy should be interchanged to ensure a balanced fuel consumption between the satellites. This has been suggested by several teams in the past as documented in the textbook from Alfriend et. al. [18]. This exchange should happen as frequently as is reasonable to minimize variation in ballistic coefficient (differential drag). Chapters 4 and 5 described a control structure in which the satellites are allowed to passively drift until they near the boundary of their control window at which point a set of three propulsive burns are planned and enacted. The time between manoeuvre sets is between six and 30 days depending on the initial conditions. Exchanging the role of chief and deputy between every manoeuvre is not prohibitive, as even in the shortest case of six days, there would be at most ten manoeuvres over the formation flying mission phase, corresponding to ten role exchanges. This exchange can occur upon request of the deputy to become the chief after it has completed its propulsive manoeuvre set. The added benefit of this arrangement is that if a significant error was found in the "corrected" position, the more stable (originally chief/non-active) satellite could be used to adjust the relative state. This analysis is only valid in the case of a TLE-based formation flying algorithm, as GPS-based formation flying would call for more frequent propulsive burns (to minimize the propellant cost and duration of each manoeuvre) and it would not be reasonable to exchange roles between these more frequent burns.

Another consideration that should be taken into account is whether to perform the correction manoeuvres as soon as new position data is available (many smaller manoeuvres) or to wait until the satellites near their control window boundary then perform the necessary manoeuvres (fewer larger manoeuvres). In the case of the QB50 Nominal and Elliptical orbits using

TLEs as navigation sources makes this a moot point since the difference between the corrections as soon as possible or just before leaving would be six days or seven days respectively. In general, for other CubeSat missions, which may be in higher altitude orbits or exhibit less differential drag, this trade-off should be made. On one hand, the control accuracy has been shown to be better for smaller manoeuvres. However, the navigation accuracy could be improved by having more data points, to combine to get the best position and velocity estimate possible. Further, long manoeuvres have repercussions on the attitude stability, which could be dealt with better if there were more time between the manoeuvres. This decision is highly dependent upon the orbit chosen, the specific control algorithm used, and the tolerable navigation and control accuracies, thus the operational plan concerning manoeuvre scheduling should be made after these have been defined.

In Chapter 3, eight guidance, navigation and control scenarios were identified for the Autonomous Formation Flying (AFF) Payload with varying levels of autonomy. One of the differences between these scenarios was the communication link used to coordinate between the satellites. Scenarios 1 - 5 (Basic, Onboard, Distributed, Distributed onboard, and Distributed coordinated) all include both satellites receiving the state of both satellites from the ground. Meanwhile, Scenarios 6 and 7 (Onboard ISL-based relay and Distributed coordinated ISL-based), demonstrate each satellite receiving its own state from the ground by uplink then exchanging their state with the other satellite using the ISL. The final scenario (Full autonomous formation flying) severs the link to the ground within the formation flying algorithm and instead uses an onboard sensor - in this case GPS - to determine its state which is then exchanged with the other satellite using the ISL. As this thesis is only addressing the TLE-based problem, the eighth scenario will be disregarded here.

This means there are, in general, two different communication structures: either the ground station will uplink the state information (TLEs) of both satellites to both satellites, or the ground station will provide only the satellite's own TLE and then leave the satellites to exchange data via the ISL. This second option is not really interesting from an operational stand point as the ground station will have both TLEs available so there is no reason to require an extra communication link (the ISL) before navigation can proceed. However, from a technology demonstration perspective it is interesting to demonstrate such a link as it could allow further capabilities in future missions. These capabilities could include, for example, the full autonomous formation flying scenario described above, the comparison of calculations between the two satellites for reliability in case of bit flips or other errors on one of the two spacecraft, or even an improved allocation algorithm that uses the current orientations of the satellites to determine which is more suited to being the deputy (ie. actively correct the relative position). In addition to increasing the operational capabilities, using an ISL will allow more frequent clock synchronizations between the two satellites which, as discussed in Chapter 4, will improve the accuracy of the navigation algorithm. These advantages do come at a cost of complexity, as the use of the ISL will require further development to confirm the pointing accuracy is sufficient to establish and maintain the link between the spacecraft over time - especially considering the position inaccuracies established in Chapter 4. Establishing communication protocols and synchronization plans over the ISL is critical to ensuring that it functions as desired.

As DelFFi is TU Delft's first distributed system, thorough testing procedures for the inter-satellite communication link - whether the link is through the ground station or established using an S-band ISL - should be developed and performed. The decision to use several scenarios to gradually build up to full onboard autonomy can allow the performance of each element (the actuator, navigation algorithm, communication link and guidance algorithm) to be confirmed in sequence, minimizing the risk to the mission. Since the mission duration is nominally only 20-30 days (see the mission duration requirement of Table 2.3), it may not be feasible to perform all eight of the scenarios in that time. Because of this, it is recommended that at a minimum the following Scenarios are performed: 1, 4, 5, and 6. Scenario 1 (Basic) corresponds to a test of the actuator without the rest of the formation flying algorithm being performed onboard. Two steps are taken in advancing to Scenario 4 (Distributed Onboard): the guidance (which in this case refers to determining the relative state and determining the best course of action) is moved onboard, and both satellites will be involved (ie. allocated a role of either chief or deputy and performing as necessary). Scenario 5 (Distributed Coordinated) adds some interaction within the system. The satellites each receive their own TLE and they must exchange it with the other satellite. This allows the testing of the communication protocol while the ground is still kept in the loop. To test the addition of the ISL, Scenario 6 (Onboard ISL-based relay) must be performed. At this time, the supporting software for Scenarios 7 and 8 (which test the improved allocation algorithms and switch to the use of GPS instead of TLEs) have not been developed or tested on the ground, so they are not included. These would be best suited to be tested only if the initial scenarios are successful and there is sufficient mission lifetime to test the additional capabilities. Skipping over some of the lower-complexity scenarios adds risk as it is unknown if all capabilities will be functioning properly. For example, switching immediately to the full autonomous formation flying scenario at the beginning of the mission would effectively cut the ground out of the loop, potentially leading to an irrecoverable relative position if there is an error in either the Guidance, Navigation and Control (GNC) algorithms or the actuator itself. This may not prevent the mission from continuing but could end the formation flying experiment.

## 6.2 Phases and Modes

In discussing the effect of formation flying on the mission operations, it is convenient to look at the effects in each phase of the mission. Then the modes associated with the formation maintenance mission phase can be identified and described. This process was started for the DelFFi mission by Brauer in [81]. The methodology suggested in that thesis will be followed here.

### 6.2.1 Mission Phase Considerations

In [81], three Phases are identified: Pre-mission, Mission and Post-mission. Of specific interest here is the final Sub-Phase of the Pre-mission: Launch and Early Operations (LEOP), and the first Sub-Phase of the Mission Phase: Nominal Operations. This section will summarize the activities of the LEOP and Nominal Operations Sub-Phases based on [81] and detail the tasks associated with the AFF payload.

### Launch and Early Operations

The LEOP Sub-Phase includes launch, deployment, detumbling and commissioning of the satellites. The satellites will be launched along with the other satellites from the QB50 project and sequentially deployed from their individual P-PODs (see Section 2.2). Although they will be deployed from the same position, due to the deployment and the orbital perturbations, the satellites will have a non-zero initial relative velocity causing them to drift apart. In order to initialize the formation, a drift stop manoeuvre must be performed to counter the differential velocity. This manoeuvre can also be used to establish the desired relative position if properly timed. This is called Formation Acquisition. There are many unknowns that significantly affect when the formation acquisition or drift stop manoeuvre must occur such that the Nominal intersatellite distance is not over-shot. For a 300 km circular orbit with an initial relative velocity of 1 m/s, the time to achieve the desired separation distance is estimated to be between 3 and 12 days by Brauer [81] depending on the solar cycle at launch, with an expected value of 12 days based on the current QB50 launch schedule.

Planning of the formation acquisition manoeuvre using TLEs will be a challenge as when there are many small bodies that have only recently been inserted, it is not always possible to know whether the TLE belongs to the assigned satellite [64]. For this reason, the formation acquisition has not been studied as an onboard algorithm, instead it will be planned and uploaded to the satellite from the ground station.

### Nominal Operations

The Nominal Operations Sub-Phase includes operating both payloads: the FIPEX and the AFF Payload. This phase lasts as long as the formation flying mission is of interest to the science team or until the propellant is depleted. The maximum duration of the Nominal Operations Sub-Phase - determined by the propellant budget - is very sensitive to the initial orbital elements. To estimate the mission lifetime, we first assume that acquisition will require 1 m/s total between the two satellites (divided between them based on the initial differential velocity on insertion). Further, each satellite has a propellant budget of 15 m/s, for a total of 29 m/s available for formation maintenance, assuming both satellites perform manoeuvres at various times during the mission. For the QB50 Nominal orbit, which requires a manoeuvre set of 3.43 m/s (non-optimized control) every six days, the 29 m/s propellant budget allows for 48 days of Nominal Operations. This exceeds the 20 day mission lifetime required in AFF-M-02, as well as the optional 30 day mission lifetime requirement, allowing the possibility to extend the nominal mission lifetime if it is of interest to the science team. After this, the satellite will transition into the extended operations sub-phase which ends the formation flying experiment and only operates the science payload (FIPEX).

Brauer defines three Modes within the Nominal Operations Sub-Phase: Main, Thrust and Safe. Safe is accessed in case of faults to allow the ground team time to recover the satellite. The Main Mode includes all activities relating to the operation of the satellite except for thrusting, which is handled specifically in the Thrust Mode. In relation to the formation flying experiment, the Main Mode includes attitude determination and control, communication with

the ground (including receipt of TLEs, updated parameters, and new guidance instructions), communication with the other satellite (either through the ground or using the ISL), and execution of the formation maintenance algorithms described previously in Chapters 4 and 5. When a set of manoeuvres has been planned in the Main Mode, the system transitions to the Thrust Mode, in which the attitude determination and control system (ADCS) drives the satellite to the desired attitude and the micropropulsion system delivers the desired thrust. During this time the relative state is still computed based on an onboard propagator and feedback from the ADCS (attitude estimate) and micropropulsion system (estimated thrust based on temperature and pressure) and checks are performed to ensure the manoeuvre is progressing as planned. If there is an error, the system will switch to Safe Mode. After the planned manoeuvre(s) are complete, the system will return to the Main Mode.

### 6.2.2 Formation Maintenance Modes Identification

The next step after identifying the Mission Modes, is to identify the Sub-Modes belonging to the Formation Maintenance Algorithm within the Nominal Operations Sub-Phase. This is done, according to [81], by identifying the use cases within the Mode, grouping similar use cases into Sub-Modes and describing the tasks, and entrance and exit criteria for each Sub-Mode. This is significantly simpler for the formation maintenance algorithm than for the satellite as a whole. There are two main use cases for the formation maintenance algorithm: (1) Estimation of the relative state and (2) Planning a set of manoeuvres to correct the relative state. Estimation of the relative state is performed at every time step of the algorithm, while Planning is only performed when the system is approaching the boundary of the control window. Thus when the satellite moves into the Nominal Operations Sub-Phase and is in the Main Mode, the Estimation Sub-Mode will be entered. In this mode, TLEs will be received, checked for validity (for example, no manoeuvres in previous five days), the absolute state of both satellites will be propagated to the current time and the relative state will be calculated. The calculated relative state will be compared with the desired relative state to determine if it is necessary to plan a manoeuvre set. If it is, the system will transition to the Planning Sub-Mode within the Main Mode and the manoeuvre set will be planned. After planning the system will move to the Thrust Mode to perform the manoeuvres, after which the system returns to the Main Mode and the Estimation Sub-Mode. Within the Thrust Mode, estimation should continue but now including real-time feedback from the ADCS and micropropulsion systems. This Sub-Mode will be called Monitoring and is the only one within the Thrust Mode relating to formation flying. These transitions between Sub-Modes are shown in the Figure 6.2 below.

Using these Sub-Modes it is possible to separate the development of the Formation Flying Package (FFP) into separate elements that can be individually addressed in the future. This structure has already been used in the development of the Simulink models discussed in Chapter 5.

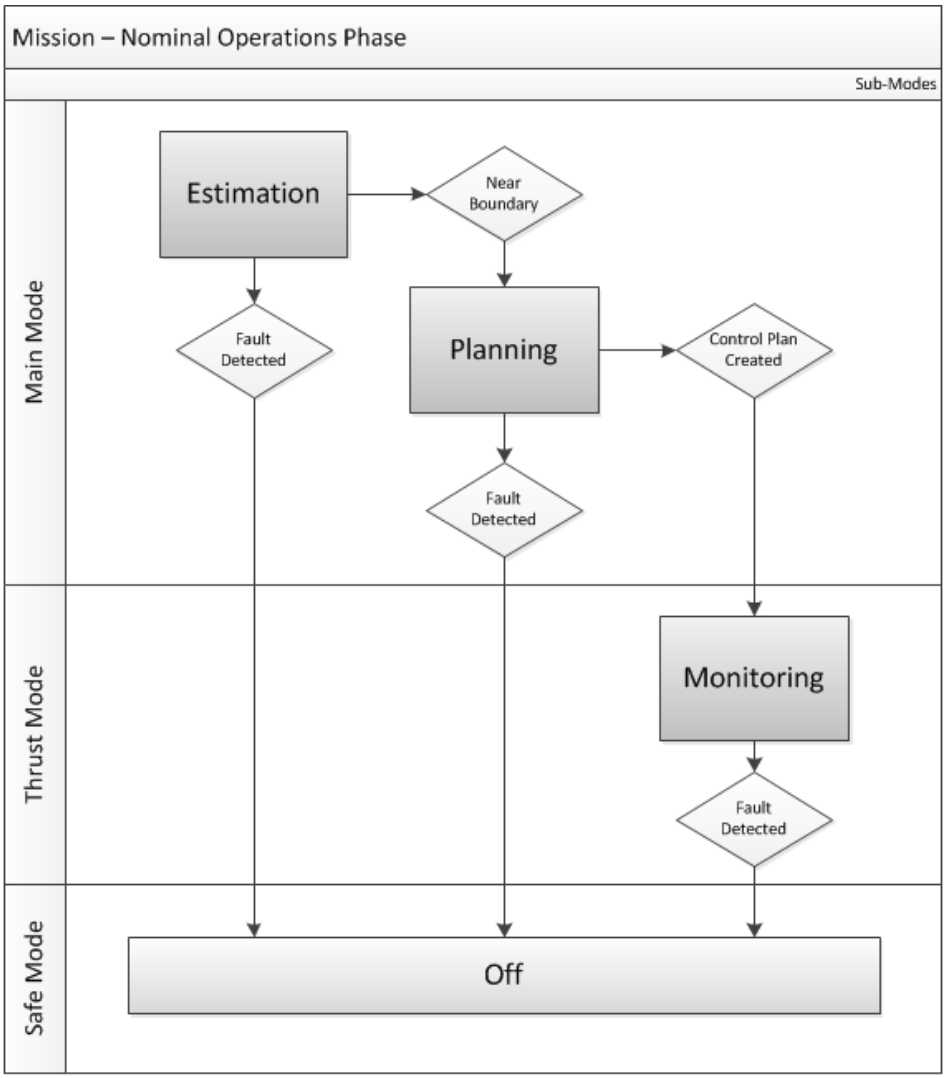



**Figure 6.2:** Identification and interfaces between Sub-Modes for the formation maintenance algorithm of the Formation Flying Package (FFP) of the DelFFi mission.

## 6.3 Support from the Satellite Bus for the Formation Flying Package

In considering the mission operations, the interfaces within the bus and the combined operational plan should be addressed. Up until this point, the main focus has been on the propulsion system and ADCS, which function as the actuators for the formation maintenance payload. This section will address (1) the onboard computer and data bus, (2) the transfer to and from Safe Mode while the FFP is active and (3) manoeuvre timing with respect to the sunlit/eclipse periods and the ADCS stability pattern.

### 6.3.1 Onboard Computer and Data Bus

The command and data handling subsystem for DelFFi was discussed most recently by Brauer in [81]. The onboard computer (OBC), which supervises the satellite's activities is connected to the other subsystems using an $I^2C$ bus - this includes the connection to the propulsion system thruster and the ADCS microcontroller. The problem with the $I^2C$ bus, is its tendency to experience frequent 'drop-outs' that stop communication between the various subsystems. One concern is that an $I^2C$ dropout could occur during a burn, meaning that the thruster would continue firing without any monitoring of the attitude. Since the magnitudes of the $\Delta$v's are so small, the fear that the position could be changed to an irrecoverable one is small, however it is possible that the combination of the thrust in non-desired directions and the disturbance torques this generates could place the satellite in an unknown orbit that cannot be estimated until the next valid TLE is received (a minimum of five days later) or that the satellite could spin up to an extremely large rate requiring the satellite to return to a Detumble Mode.

The communication drop-out frequency and duration should be assessed such that a recovery plan can be developed that ensures the thruster shuts off when necessary. One solution would be to control the thruster through the ADCS microcontroller rather than the onboard computer - eliminating the $I^2C$ link. However this means that any thrust commands sent from the ground and any housekeeping data relating to the propulsion system would have to be passed through the OBC to the ADCS, rather than having a direct link to the propulsion system.

### 6.3.2 Safe Mode Transitions

There are many different faults which would cause the satellite to switch into Safe Mode, however the reaction from the FFP would always be the same: switch to the 'Off' Mode and wait to be turned back on. If possible, the system should store the status of the FFP and, if applicable, the control plan that was in the process of being executed, so that the data can be transferred to the ground to be analyzed. When the system leaves Safe Mode - for example by command from the ground station - the system will restart to Nominal Operations and eventually activate the Main Mode which will start with the FFP in the Estimation Sub-Mode as if from a clean restart.

One method of Safe Mode activation that is of particular interest is if the rotation rate becomes extremely high in a short period of time, the satellite may either switch to Safe Mode or Detumbling Mode to compensate. In either case, the FFP will stop and switch to 'Off' Mode no matter where it is - even if it is in the middle of a burn. This means that it is possible that the satellite will have a different set of mean orbital elements than the chief (nominal/desired), causing significant drift. Although it may be possible to return to the formation maintenance modes rather quickly, resuming the same control plan is not valid because of the drift that will have occurred in the interim. Since the only form of data bout the satellite's position would come from TLEs, the satellite would have to wait six days before the new manoeuvre plan could be created - a duration where no information on the position

and velocity is available and the onboard estimation has not been maintained.

The transition in and out of Safe Mode is problematic, it may be desirable to change the activation criteria for Safe Mode that are effective while in the Thrust Mode. The most obvious of these would be to allow a larger rotation rate than in other modes, however a consequence of this is further control inaccuracies. The effects of manoeuvres on the attitude should be assessed in detail, particularly in context with the reaction wheel performance to ensure that Safe Mode is only activated when there is a fault that cannot be tolerated by the Nominal Operations Modes.

### 6.3.3 Manoeuvre Scheduling

In scheduling the manoeuvres, it has been assumed up until now that the manoeuvres could occur at any time assuming their duration did not exceed the power budget's limit. However, the power budget only allows the propulsion system to be active in sunlight. Further, the ADCS analysis by Haghayegh showed that the best pointing accuracy and rotation rate stability was in the second half of the sunlit portion of the orbit [58]. The analytical controller selected and assessed in Chapter 5 requires the first and third manoeuvres to begin at the crossing of the ascending node, while the second 180 ° opposite at the descending node. To determine if these three manoeuvres will occur in sunlight, recall that the QB50 orbit is a sun-synchronous orbit with a RAAN of -15° (corresponds to an LTAN of 11:00 pm). This means that the ascending node will be always be in eclipse, while the descending node will always be in sunlight. Even if the RAAN were an 11:00 am orbit (for example) there would still be at least one of the three manoeuvres that would need to occur in eclipse. In this case however, the second manoeuvre (the smallest one) would occur in eclipse. The ability of the electrical power system to store and provide sufficient power while only on batter power for the propulsion system for a short manoeuvre in eclipse should be considered before this is ruled out.

## 6.4 Conclusions

The choice of a distributed system provides several capabilities to CubeSats, however it also complicates the operational activities for the mission. The timeliness of communication and coordination between satellites becomes a significant consideration that can affect the overall success of the mission. Dividing the mission into Phases, Sub-Phases, Modes and Sub-Modes allows for a more complete understanding of the problem at hand. This chapter identified two Sub-Modes relevant to formation flying within the Mission Phase, Nominal Operations Sub-Phase, Main Mode, namely: Estimation and Planning, and one Sub-Mode within the Mission Phase, Nominal Operations Sub-Phase, Thrust Mode, namely: Monitoring. The specific states relating to these three Sub-Modes have not yet been identified as they will depend on the detailed micropropulsion system and ADCS functions and interfaces.

The choice of TLEs was intended to result in a system that was less operationally complex than other navigation methods. While it has been shown that it is possible to plan operations

to allow for formation flying based on TLEs, these operations will be more complex than for a GPS-based formation flying algorithm, due to the upload and timing of the provision of the TLEs to the satellites. As such, it may be preferable, in some cases, to choose a more complicated algorithm (GPS-based) over the complex operational plan described above (TLE-based). This is particularly true when using unproven ADCS hardware, which may trigger a transition to Safe Mode (caused large rotation rates) during manoeuvres. This could prove disastrous as the satellite is left in a state that is poorly estimated, with no position measurements for up to six days. This risk must be addressed and mitigated by the DelFFi team as a whole to ensure the mission is not lost as a result of the first propulsive manoeuvre.

# Chapter 7

# Conclusions

This research began with the objective of making recommendations for the implementation of formation maintenance on the DelFFi satellites by characterizing the performance of a Two-Line Element (TLE) based controller within the context of a CubeSat mission. This would have two main benefits, first the DelFFi Autonomous Formation Flying (AFF) Payload's development would be furthered, and second the general suitability of TLE-based navigation for formation maintenance would be assessed. To accomplish this three questions were posed:

1. What are the capabilities and limitations of a two-line element based navigation algorithm for CubeSats?
2. What impact will CubeSat technologies have on the implementation of a formation maintenance controller for the DelFFi mission?
3. In what ways are the operations of the satellite affected by the Autonomous Formation Flying Payload?

These questions were answered in Chapters 4, 5 and 6 respectively, using background information from Chapters 1, 2 and 3. The conclusions arrived at in those chapters are summarized in Sections 7.1 and 7.2 to provide a complete response to the research questions. Following this a summary of the requirement verification is provided in Section 7.3. Recommendations based on this research are provided in Section 7.4. While answering these questions some interesting areas were identified that did not fit into the scope of this thesis. These subjects are reserved for future research projects and are summarized in Section 7.5. Finally, this chapter concludes with an outlook, in Section 7.6 of the how the DelFFi project and TU Delft can best contribute to and take advantage of developments in the field for formation flying.

## 7.1 TLE-based Formation Flight

The research of [6] showed that TLEs are generated by averaging measurements from the previous five days. This means that if a propulsive manoeuvre has been performed within

the previous five days, the TLE is not valid, as it will include measurements of the previous orbit. Since it may take up to a day to uplink the TLEs to the satellites after it has been received, the satellites must maintain their control window passively for a minimum of six days. Chapter 4 showed that this could be achieved by many of the orbits studied (studied orbits had altitudes ranging from 350 to 800 km, with eccentricities of 0.001 (circular) and 0.015 (elliptical) and the satellites were separated by between 500 km and 1100 km) assuming a control window dimension that is 10% of the separation distance. These viable orbits were:

1. Circular orbits: separation distance of 1000 km, the semi-major axis must be above approximately 365 km
2. Circular orbits: any modelled separation distance (500 km to 1100 km), the semi-major axis must be above approximately 410 km
3. Elliptical orbits: separation distance of 1000 km, the semi-major axis must be above approximately 405 km
4. Elliptical orbits: any modelled separation distance (500 km to 1100 km), the semi-major axis must be above approximately 440 km

Note that this analysis assumed a 2° attitude pointing error on the deputy (forward) satellite, while the chief (rear) satellite was oriented along the velocity vector. Since the attitude pointing error had such a large effect on the passive relative motion, a study of the allowable attitude pointing error was performed. It found that the attitude pointing error for the two satellites should be required to be less than 0.5° from nominal (velocity vector) in order to ensure that in both arrangements (chief ahead of deputy and vice versa) the six day passive relative motion requirement will be met. This finding will be provided to the Attitude Determination and Control System (ADCS) team for further investigation.

The sensitivity of the QB50 nominal orbit to the matching mean orbital elements was also assessed and it was found that an error of 10 km (the control accuracy required for a 1000 km separation distance according to Gill's Scaling Law) on the semi-major axis, caused the satellites to drift apart and breach the control window in under 2 hours. The acquisition manoeuvre and each correction manoeuvre planned by the formation maintenance controller must take this into account and precisely control the differential mean semi-major axis. The acceptable control accuracy was not established in this research, but it is clear that increasing the control accuracy beyond the recommended 10 km is not acceptable.

The next step was to address the navigation accuracy that the TLEs could provide. By assessing the accuracy of TLEs themselves (± 1 km), the accuracy after propagation of TLEs using the SGP4 algorithm for one week (± 3 km) and the clock drift (± 1.1 km if calibrated), the navigation accuracy is estimated to be ± 5.1 km. This is larger than the 1 km required by Gill's Scaling Law for a 1000 km separation distance, so a control accuracy analysis was performed to determine the effect of this navigation accuracy on the ability to maintain the separation distance within the control window.

To determine if the separation distance could be maintained by the DelFFi actuator, a controller was selected and implemented in a MATLAB Simulink environment. An analytical

controller for circular orbits from [85] was chosen, which plans a set of three manoeuvres to return the formation to its nominal positions. It is applicable for circular reference orbits, and is not suited to precision formations due to the assumption that the time between thrusts is much larger than the duration of each thrust. One major advantage of this controller is the ability to adjust the manoeuvre schedule to reduce the propellant cost. This reduced propellant cost is achieved by providing a large period of time over which the manoeuvres can occur. Allowing the three manoeuvres to occur over the course of 15 orbits (just under one day), reduces the propellant cost to 0.70 m/s (from 3.43 m/s) for the QB50 nominal orbit with a 2° attitude pointing error on the deputy satellite. Assuming the propulsion system is functioning at the beginning of life thrust of 1.4 mN and the mass of the satellite is the initial 3.64 kg, the three manoeuvres (in the non-optimized case), which have magnitudes 1.66 m/s, 0.065 m/s and -1.71 m/s (corresponding to 4316 s, 169 s, and 4446 s respectively). Considering the optimized case and the QB50 elliptical orbit case, the propulsion system must (in particular its power and thermal budgets) accommodate $\Delta$v's from approximately 0.02 m/s to 1.7 m/s. It also requires the ADCS to be able to control the rotational rate during manoeuvres to a much lower rotational rate. It is recommended that the impact of requiring a rotational rate below 0.1°/s (corresponding to a maximum 15% variation in thrust magnitude, or 30° over the maximum 10 minute burn) be assessed by the ADCS team. Based on the experiences of the CanX-4/5 team, meeting these requirements should be possible, even with existing nanosatellite technology.

Finally, the control accuracy of the formation maintenance controller was assessed. Considering the contributions of the magnitude of the thrust produced by the propulsion system, the misalignment of the thruster mounting the attitude pointing error of the satellite, the intrinsic error of the controller and the navigation accuracy, the control accuracy is estimated to be approximately 17.76 km. This is larger than the 10 km limit established in Chapter 4, meaning formation maintenance based on TLEs does not meet the DelFFi mission requirements according to the current design and assumptions. Further, the analytical controller failed to match the mean orbital elements - specifically the mean semi-major axis - of the deputy satellite to the nominal mean orbital elements. The differential mean semi-major axis caused the satellites to drift apart extremely quickly, exceeding the control window in under two days in the case of the QB50 nominal orbit. This inability to properly match the mean orbital elements means that the six day passive relative motion requirement established in Chapter 4 cannot be met throughout the mission, using this control algorithm. If the navigation accuracy could be improved (mainly by increasing the propagation accuracy of the orbital elements from the TLEs), and if the intrinsic error of the controller could be reduced (by choosing a controller better suited to DelFFi's large control window), it may be possible to use TLEs as a basis for formation maintenance, however the simplicity of the TLE-based algorithms compared to GPS-based algorithms, will no longer be as pronounced after these developments.

## 7.2 Formation Flying with CubeSats

Chapter 6 divided the system modes identified in [81] into three Sub-Modes: Estimation (Main Mode), Planning (Main Mode) and Monitoring (Thrust Mode). This division will allow future analysis to focus on each of these Sub-Modes both individually and collectively. A closer look at the communications architecture between the satellites, and with the ground (in order to receive the TLEs), showed that, operationally, choosing to use TLE-based formation maintenance adds complexity compared to a more traditional GPS-based formation maintenance algorithm (such as CanX-4/5 [26]. Especially without an intersatellite link, the transmission of messages, such that the formation can behave as a distributed system, requires development of an automated processing algorithm for the ground station that receives data from the first satellite and repackages it to return to the second satellite in the same pass over the ground station. As a technology demonstration the Onboard Basic scenario described in Chapter 6, in which one satellite (the Deputy) receives TLEs for both satellites, then calculates and performs a maintenance manoeuvre with no input from the Chief satellite (in this case the Chief functions solely as a reference state) can be performed with low complexity, however this does not allow for the advantages of formation flying with CubeSats to be seen. Only by allowing communication between the satellites can the distributed system, which optimizes fuel consumption globally (both to minimize consumption and balance between the satellites), be implemented effectively. Without establishing a distributed system architecture, formation maintenance algorithms have limited difference compared to station keeping algorithms.

## 7.3 DelFFi Autonomous Formation Flying Payload Requirement Verification

The requirements developed for the DelFFi Formation Flying Package (FFP) in Chapter 2 have been verified based on the results of this thesis. The compliancy of the design described in this thesis is discussed in Appendix B and is summarized in Tables 7.1 to 7.5.

**Table 7.1:** Formation Flying Package Mission Requirement Verification

| ID | Requirement | Compliancy | | | |
|---|---|---|---|---|---|
| | | Yes | Partial | No | N/A |
| AFF-M-01 | In the formation keeping phase, the formation flying package shall maintain a nominal inter-satellite distance (in the along-track direction). | | x | | |
| AFF-M-02 | The formation keeping phase of the mission shall last no less than 20 days [OPTIONAL: 30 days]. | x | | | |

In the case of many of the requirements, their compliancy is based off of several assumptions since it was not possible to perform full verification at this stage in the design. The details

of these assumptions and the future verification that should be performed is in Appendix B. Some highlights are provided below:

- AFF-F-24 (100 km Control Window) is not compliant at this time. This is because the control accuracy was shown to be 17.76 km and the rate of correction to accomplish this would be too high to use the TLE-based navigation assumed in this design.
- AFF-F-27 (30 s to 300 s Burn Time) is not compliant, since the magnitudes of the manoeuvres have been shown to be up to 1.71 m/s (corresponding to 128 minutes).
- AFF-F-29 (Only Burn in Second Half of Sunlit Periods) is not compliant due to the need to perform one of the manoeuvres on the opposite side of the orbit from the other two, thus at least one is in eclipse.
- AFF-D-42 (Comply with QB50 orbits) has been assessed as partially compliant due to the compatibility only in certain cases - for example, if the attitude pointing error is decreased, the reaction wheels are functional and the propulsion system increases the thrust magnitude.
- AFF-D-43 and AFF-D-44 (Intersatellite link and GPS compatibility) have not been assessed in this thesis and are not currently verified.
- AFF-O-60 and AFF-O-61 (GNC Architectures and Autonomous Operations) have been discussed in this thesis in that they are used as a starting point for other developments. In this way they are assumed to be met, however no formal verification has been performed relating to them, so they have been labelled partially compliant.

## 7.4 Recommendations

Based on these results, although the TLE-based navigation algorithm is significantly simplified compared to GPS-based navigation algorithms, the low accuracy and added operational complexity reduce enthusiasm towards using TLE-based formation maintenance. In particular, the navigation accuracy of the TLE ($\pm$ 5.1 km) has been shown to have too large an effect on the control accuracy for the differential mean orbital elements to be properly initiated (matched to the reference/Chief satellite) after a correction manoeuvre. Improving the navigation accuracy from TLEs, and increasing the maturity of the ADCS and propulsion system, is likely to be no less complicated than developing the navigation algorithms for a GPS-based formation maintenance algorithm. It is highly recommended that GPS-based algorithms are investigated and developed. This includes putting additional effort towards the acquisition of a GPS receiver in the near future, such that the expertise can be developed within the department. Considering the existing system, reducing the differential drag would have the largest affect on the system. In particular, this means decreasing the attitude pointing error, providing attitude feedback to the onboard orbit models or increasing the orbital altitude.

## 7.5 Future Work

Future work related to formation flying for the DelFFi project falls into three categories: assessment of assumption validity, topics to investigate to provide better understanding of the system and methods to increase the systems capabilities.

Several assumptions and simplifications were made during this research, the following list identifies several questions or capabilities that should be addressed to further ensure those assumptions are valid:

- How often do the satellites pass over the Uplink station (Delft)? (Assumed at least once per day)
- What is the propagation error for 300 km altitude orbits? (Assumed the same as 600 km - 800 km orbits)
- How well does the capacitance of the circuit match the crystal? (Assumed sufficiently well to provide negligible error on the clock time)
- To what degree can the clock be calibrated? (Assumed sufficiently well to provide negligible error on the clock time)
- What is the maximum allowable control window the for the science payload to benefit from formation flying? (Assumed separation distance accuracy is more critical than precision/dimension of control window)
- Do all controllers based on Gauss' Variational Equations exhibit the same linearization errors for large corrections?
- What is the navigation accuracy using differential orbital elements from TLEs considering the effects of propagation, and microprocessor errors (clock synchronization, truncation, latency etc.)?

To better understand the system the following topics are recommended for detailed investigation:

- What are the mean orbital elements after the formation has been acquired (at the start of the formation maintenance phase)? How much do they vary during the LEOP phase?
- What are the extreme minimum and maximum manoeuvre durations, considering the power and thermal budgets?
- Is a one day limit on the duration of the total manoeuvre set the most effective an operational perspective, or should a longer or shorter period be used?
- What is the limit on rotational rate control that the ADCS can provide based on the current design?
- What is the maximum rotational rate tolerated by the formation maintenance controller?

- What is the passive relative motion of non-sun-synchronous orbits?
- How do other inclinations and LTANs affect the passive relative motion and TLE-based formation maintenance performance?
- Is the navigation and/or control accuracy improved by making calculations immediately after receiving TLE data/after contact with a ground station compared to waiting after receiving TLEs to make a control plan?

Finally, several topics have been identified that may be of interest to the DelFFi mission but did not fall within the scope of this thesis:

- Filtering of sensor data before the navigation algorithm
- Fine tuning of manoeuvres to control initialization true latitude
- Differential Mean Orbital Element algorithms (see [89])
- Differential drag control
- Model Predictive Controllers (How much fuel savings is possible? At what cost to the processing time?)
- Access to invariant orbits from QB50 nominal orbit
- Communication protocols for ISL and ground station (including automation of data pass through on the ground)
- Formation acquisition algorithms
- Fuel balancing (for example, role exchange between Chief and Deputy)
- Subsystem state identification (for each sub-mode)
- Applicability of Gill's Scaling Law to long baseline formations (is an order of magnitude difference between the navigation accuracy and the control accuracy reasonable?)
- Inclusion of attitude feedback in onboard propagator
- Ability to request shorter averaging period from NORAD (at the cost of reduced accuracy)
- The possibility of rejecting pre-manoeuvre data from the averaged mean orbital elements provided in TLEs

In addition, significant work is required to implement these algorithms on the microcontroller. This work will require an embedded programmer who can transfer the Simulink models and C code onto the flight hardware in order for the interfaces to parameters and other subsystems to be realized. The challenges and opportunities associated with this work have not been addressed in detail in this research.

Recently, a combined relative orbit and attitude propagator has been under study at the TU Delft by Adolfo Chaves Jimenez (as per conversation, September 2015). It defines the relative

position, velocity and attitude of the deputy satellite with respect to the chief based only on initial conditions, position and velocity data of the chief and attitude data of the chief and deputy. Once this propagator has been developed and verified, it may be possible to use TLEs as a navigation source for the Chief (passive so updates would occur every 0.6 days), while the onboard propagator could estimate the state of the deputy based on attitude feedback from the ADCS (magnetometer and sun-sensor data for example) and thrust feedback from the propulsion system. This would allow for an estimate of the deputy's state long before a valid TLE was received, which would increase the operator's knowledge of the system and further ensure safe operations. This development shows promise for formation maintenance for CubeSat systems and opportunities for integration to DelFFi should be investigated.

## 7.6 Outlook

The formation flying field has recently diverged along two different directions. Some researchers are looking at high precision formation flying algorithms, while others are focused on low complexity mission with slightly less strict performances. From the point of view of CubeSats, the latter is certainly more applicable - especially given the current capabilities of CubeSat actuators. Over time there has been an significant increase in the abilities of CubeSat actuators and processors [2], which may eventually allow for high precision formations on the CubeSat platform. This is exemplified by the success of the CanX-4/5 mission which uses a non-CubeSat standard nanosatellite bus.

For DelFFi to be able to follow this progression, three major developments areas must be addressed: differential drag reduction (through the attitude pointing error), intersatellite communication link architectures and reaction wheel momentum storage and off-loading methods. The propulsion systems under development at TU Delft are well on their way to being able to provide the necessary performance, however rigorous testing for calibration purposes is also necessary. In addition, a relative motion model that considers both the orbit and attitude dynamics is under development. This will allow a better onboard state estimate to be generated that may be able to provide some data between TLE updates, which would improve the formation maintenance performance. Formation flying can provide significant scientific benefits to a mission, however having a mature satellite bus, with highly capable subsystems, is the first, and most critical, step.

**Table 7.2:** Formation Flying Package Functional Requirement Verification

| ID | Requirement | Compliancy | | | |
|---|---|---|---|---|---|
| | | Yes | Partial | No | N/A |
| AFF-F-20 | The AFF payload shall implement relative guidance and navigation using ground and/or onboard information. | x | | | |
| AFF-F-21 | The AFF payload shall generate and implement relative control commands. | | x | | |
| AFF-F-22 | The satellite shall determine its position to within 10 km [OPTIONAL: 1 km] accuracy (TBC). | | x | | |
| AFF-F-23 | The nominal intersatellite distance shall be 1000 km. | x | | | |
| AFF-F-24 | The nominal intersatellite distance shall be maintained to within 10% of its value - ie. 1000 km $\pm$ 100 km. | | | x | |
| AFF-F-25 | The attitude vector (the desired control acceleration direction) shall be requested no less than 15 minutes [5 minutes if reaction wheel is active] (TBC) in advance of a propulsive manoeuvre. | x | | | |
| AFF-F-26 | The AFF payload shall require no more than 15 m/s delta-v during the mission lifetime to maintain the required control window. | x | | | |
| AFF-F-27 | The burn time required from the thrusters shall be no less than 30 seconds (TBC) and no more than 300 seconds (TBC). | | | x | |
| AFF-F-28 | The thruster activation shall be requested no less than 300 (TBC) seconds in advance of the thrust start time. | x | | | |
| AFF-F-29 | [OPTIONAL] The propulsive manoeuvres shall be timed to occur during the second half of sun-lit periods only. | | | x | |
| AFF-F-30 | [OPTIONAL] The propulsive manoeuvres shall be separated by at least one non-thrusting sun-lit period (ie. 2 orbital periods). | x | | | |
| AFF-F-31 | [OPTIONAL] The AFF payload should maximize the time available for scientific observation (non-firing, in-track pointing). | | | | x |
| AFF-F-32 | The AFF payload shall provide housekeeping parameters (to be detailed) at a rate of 1 Hz (TBC) to the Onboard Computer (OBC). | | | | x |
| AFF-F-33 | [OPTIONAL] The formation flying package shall require no more than 500 ms per cycle when executed on the ADCS microcontroller. | | x | | |

**Table 7.3:** Formation Flying Package Design Requirement Verification

| ID | Requirement | Compliancy | | | |
|---|---|---|---|---|---|
| | | Yes | Partial | No | N/A |
| AFF-D-40 | The formation flying package will be identical on both satellites. | x | | | |
| AFF-D-41 | The AFF payload shall be compatible with the nominal QB50 orbit. | | x | | |
| AFF-D-42 | [OPTIONAL] The AFF payload should be compatible with a generic near-circular low Earth orbit with the ranges of parameters (as specified in Appendix B and above in Table 2.1). | | x | | |
| AFF-D-43 | The formation flying package shall remain functional despite loss of the intersatellite link. | | | | x |
| AFF-D-44 | [OPTIONAL] The formation flying package should remain functional despite loss of GPS fixes on one or both satellites. | | | | x |

**Table 7.4:** Formation Flying Package Operational Requirement Verification

| ID | Requirement | Compliancy | | | |
|---|---|---|---|---|---|
| | | Yes | Partial | No | N/A |
| AFF-O-60 | [OPTIONAL] The AFF payload should allow the eight GNC Architectures, as defined in the Phase A Study. | | x | | |
| AFF-O-61 | The AFF payload shall facilitate autonomous operation (without human interaction). | | x | | |

**Table 7.5:** Formation Flying Package Interface Requirement Verification

| ID | Requirement | Compliancy | | | |
|---|---|---|---|---|---|
| | | Yes | Partial | No | N/A |
| AFF-I-80 | The Formation Flying Package shall be compatible with the ADCS microcontroller. | | x | | |
| AFF-I-81 | The AFF payload shall be compatible with a clock signal which uses Coordinated Universal time (UTC) as the time reference. | | | | x |

# Appendix A

# Survey of Formation Flying Missions

This appendix provides further background on several recent formation flying missions.

**Morning Constellation** The Morning Constellation includes two satellites which are travelling in formation: Landsat-7 and EO-1 [3]. The two satellites are in 700 km sun-synchronous polar orbits with a small inclination offset. The formation is autonomous, thus each satellite is equipped with GPS for navigation and maintenance manoeuvres are executed without human intervention. No intersatellite link (ISL) is available on Landsat-7, thus its state vector transferred via the ground to EO-1, which then performs the formation maintenance [90]. The enhanced formation flying experiment on EO-1 was required to keep a one minute separation (within 10 seconds) from Landsat-7 [90].

**GRACE** The Gravity Recovery and Climate Experiment (GRACE) satellites were launched in 2002 to study time-varying gravity models of the Earth [7]. The satellites were placed in the same nominally circular 500 km orbit and are required to maintain an intersatellite distance between 170 km and 270 km as measured by a K-Band relative distance sensor [10]. The time between formation maintenance manoeuvres is required to be no less than 30 days [10]. GRACE successfully demonstrated formation control with the ground in the loop [16].

**A-train Constellation** Also known as the Afternoon Constellation, the A-train satellites follow one another in a 700 km circular orbit. In particular, CloudSat and CALIPSO (previously called Picasso/Cena) follow one another with a spacing of 112 km within a 38 km wide control window [91]. This corresponds to, at most, a 15 second time difference between measurements on the two satellites [92]. For the scientific objectives of this mission, the most critical aspect of the formation control is the cross-track error, which must be kept below 1 km [92]. The relative position is calculated on the ground using GPS data from the two satellites, and maintenance manoeuvres are executed by CloudSat (once a week on average) by telecommand [92].

**TerraSAR-X/TanDEM-X** TanDEM-X joined TerraSAR-X on orbit in 2010 to form a single-pass synthetic aperture radar interferometer [7]. These large spacecraft (each with a mass of about 1200 kg [2]) follow each other at 500 km altitude [91] separated by 5 km with an accuracy of 200 m [16]. This small control window requires daily maintenance manoeuvres [91]. While both satellites perform the same absolute orbit maintenance manoeuvres, TanDEM-X performs addition formation maintenance corrections using a dedicated cold gas propulsion system [16]. Formation control for acquisition and reconfiguration is controlled by the ground segment, however maintenance (in-plane manoeuvres) is completed autonomously on-board TanDEM-X using GPS data for TerraSAR-X received via S-band intersatellite link [16].

**PRISMA** The two satellites (Mango and Tango) that make up PRISMA (the Prototype Research Instruments and Space Mission technology Advancement) were separated from one another after being inserted into their orbit to minimize variation between their orbits at formation initialization [3]. Launched in 2010, Mango and Tango are microsatellites (145 kg and 45 kg respectively [2]) that occupy a sun-synchronous orbit at around 700 km altitude [93]. Both satellites are 3-axis stabilized, but only Mango is equipped with manoeuvring thrusters (full 3D actuation independent of attitude) [7]. One of several experiments on PRISMA is the Spaceborne Autonomous Formation Flying Experiment (SAFE), which uses GPS-based navigation to autonomously maintain a separation typically closer than 1 km with an accuracy better than 30 m (3D, RMS) [7]. SAFE uses the relative eccentricity/inclination vector separation control methods originally used for geostationary satellites, which has been shown to both minimize the number of thrust activations and be robust [7].

**CanX-4/5** This mission is the first (and, as of March 2015, only) to demonstrate sub-meter autonomous formation flight on a pair of nanosatellites (each satellite is a 20 cm cube) [94]. Four different formations have been performed by the satellites: two along track orbits (spaced at 1000 m and 500 m) and two projected circular orbits (spaced at 100 m and 50 m) [94]. Although either satellite can be the master/deputy, manoeuvres are only performed by the deputy [26]. Each satellite uses GPS to determine its state and the data is exchanged using the intersatellite link [94]. Position control was required to be better than 1 m and the position estimation was required to be better than 10 cm [94].

**Swarm** The Swarm constellation consists of three satellites in near-polar low Earth orbits that were launched in 2013 to study the geomagnetic field. Swarm-A and Swarm-B fly close together side-by-side with 1.5 degree longitudinal separation, while Swarm-C orbits 80 km above them (530 km apogee) [95]. The three satellites are each approximately 320 kg and are launched together requiring a separation manoeuvre to acquire their relative orbits [96]. Maintenance manoeuvres are also necessary for Swarm-A/B pair, due to their low altitude, which conceptually occur with an interval longer than 14 days [96]. To perform the maintenance, the Swarm satellites will vary their ballistic coefficient through attitude adjustments to avoid propulsion (which will be used to compensate for orbit decay) [96].

**MMS** Consisting of four satellites arranged in a tetrahedral formation, the Magnetospheric Multi-Scale (MMS) mission was launched in 2015 and will eventually occupy a highly elliptical orbit above the earth with apogees up to 25 times the radius of the Earth [97]. Separation

distances between the satellites in the tetrahedron vary between the four mission phases. In Phase one, separation is increased from 1000 km to 2000 km, Phase 2 varies the separation between to 1000 km to a few thousand kilometers, Phase 3 will move the satellites into a string of pearls (along-track) configuration with separation distances of a few times the radius of the Earth and finally Phase 4 will space the satellites along track with separations of a few thousand kilometers at apogee [98]. The satellites' absolute states are estimated using GPS and the requirement on the relative position error is 1% of the separation distance or 100 m (whichever is larger) [97]. Design of the reference trajectories for the four satellites is determined by optimizing a quality factor, which assesses the size and shape of the tetrahedron, with the fuel budget of each satellite and of the total reconfiguration manoeuvre [98] [27]. The formation control uses the differential mean non-singular orbital elements with a curvilinear-corrected local-vertical-local-horizontal reference frame and the Gim-Alfriend state transition matrix [27].

# Appendix B

# DelFFi Formation Flying Package Requirements

The following Technical Note (DFF-TUD-TN-1157) for the DelFFi project defines the Formation Flying requirements as well as their verification method and current compliancy status.

# Formation Flying Package Requirements for the DelFFi Satellites

**Description:** This document includes the requirements for the Autonomous Formation Flying payload of the DelFFi satellites.

**Subsystem(s) involved:**

| ADCS | CDHS | COMMS | EPS | MechS | STS | TCS | FIPEX | µPS+ | ISL | AFF | GSE | GSN | Launch |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | X | | | |

## Revision Record and Authorization

| Issue | Date | Author / Editor | Reviewer checked | PM approved | Affected Section(s) | Description of change |
|---|---|---|---|---|---|---|
| 0.1 | 17-06-2015 | A. Deeb | | | ALL | Preliminary Draft |
| 1.0 | 22-07-2015 | A. Deeb | x | x | 2.5 | As per comment from JG, removed microcontroller detail (AFF-I-80) |
| 1.1 | 30-08-2015 | A. Deeb | | | ALL | Added Compliancy, Updated AFF-F-28 to be 300 seconds, updated AFF-D-41 with QB50 nominal orbit and elliptical orbit, updated AFF-D-42 with updated variation parameters, Updated SLRs |
| | | | | | | |
| | | | | | | |

## Action Items

| TBW | TBD | TBC | Applicable Section(s) | Description of action item |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |

## List of Used References

| SLR code | Version | Data/Variable |
|---|---|---|
| 1158 | 1.0 | DelFFi CubeSat Design Overview Report – Critical Design Overview TUD_v2 |
| 1111 | 1.0 | DelFFi Phase A Study on Formation Flight within QB50 (DFF-TUD-RP-1111) |
| 1159 | 1.0 | ADCS_Processor Selection (Jaan Viru) |
| 1160 | 6 Mar 2009 | ECSS-E-ST-10-06C |
| 1161 | 6 Mar 2009 | ECSS-E-ST-10-02C |
| 1127 | 6.0 | DFF-TUD-PROP-REQ |
| | | |

# Table of Ciontents

# 1 Introduction

This document defines the requirements for the Autonomous Formation Flying (AFF) payload of the DelFFi mission. In doing so, this document also defines the verification methods and test cases to be used during the test campaign.

The goal of the AFF payload is to maintain an relative orbit between the two DelFFi satellites. In particular, it should control the along-track separation distance between the two satellites to a certain nominal distance within an accepted control window. To accomplish this, the AFF payload consists of several aspects including a propulsion system, a 3-axis stabilized attitude determination and control system (ADCS), an inter-satellite link, a global positioning system (GPS) receiver and the formation flying package (FFP) of algorithms. It also makes use of the on-board computer, the ADCS microcontroller, the communication system (including the main Telemetry and Telecommand (TTC) antenna) and the ground system as shown in the context diagram below.

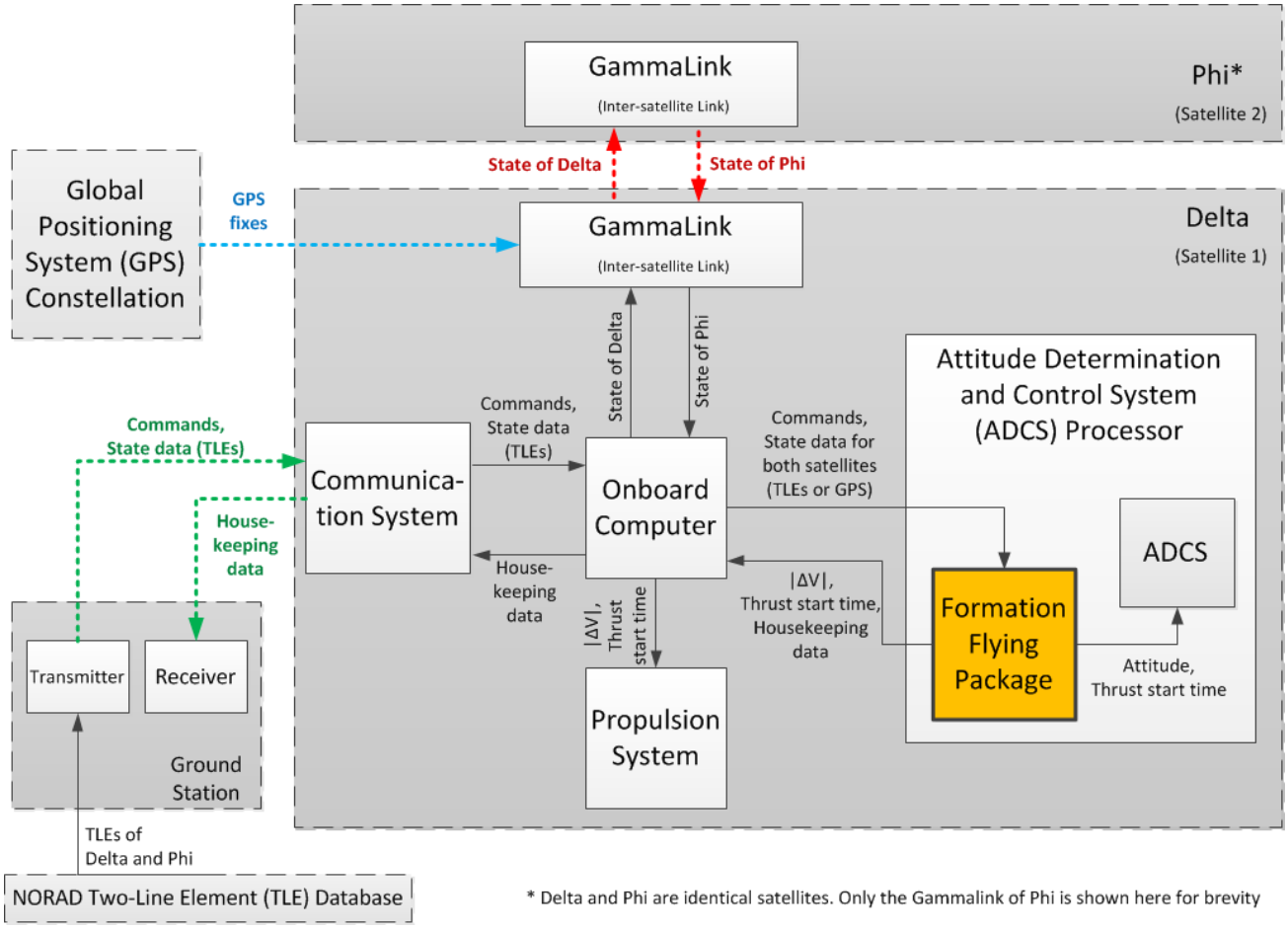



**Figure 1: Context Diagram**

In the context diagram above, The GammaLink (inter-satellite link) and GPS constellation are shown as integral to the system, however due to procurement risks, it is likely that these will not be available. Thus, the system should be shown to work without these elements if at all possible.

Of specific interest in this document are the requirements pertaining to the FFP. The FFP is a software package consisting of relative guidance, navigation and control algorithms. The preliminary concept locates the guidance algorithms in the ground segment and, as such, they are not directly addressed in this list.

Initially, the FFP will be analyzed in a MATLAB Simulink Environment that will be validated based on data from previous formation flying missions. It will then be integrated and executed on a target microcontroller that is representative of the ADCS microcontroller for processor-in-the-loop tests. If it becomes available (currently it is not) a formation flying test bench would be valuable as a hardware-in-the-loop test platform for the FFP. During the integration phase, the FFP will be integrated onto the flight ADCS microcontroller and the interfaces to the other elements of the AFF and the rest of the CubeSat will be tested.

The requirements have been developed using the standards recommended in ECSS-E-ST-10-06C as far as possible. Some of the requirements have been taken from higher levels directly and have not been modified to suit this standard. As per the ECSS standard, five types of requirements have been used for classification: Mission, Functional, Design, Operational and Interface. The requirements have been numbered for ease of reference and traceability in a structure as shown below in the table. For example, the third functional requirement would be numbered AFF-F-22.

**Table 1: Definition of Requirement Identification Codes**

| First Clause | | Second Clause | | Requirement Number |
|---|---|---|---|---|
| AFF | - | Mission (M) | - | 01 – 19 |
| | - | Functional (F) | - | 20 – 39 |
| | - | Design (D) | - | 40 – 59 |
| | - | Operational (O) | - | 60 – 79 |
| | - | Interface (I) | - | 80 – 99 |

In addition to the identifier and classification, each requirement is accompanied by a rationale, which details any higher-level requirements and the spirit behind the requirement. This should help to clarify any terms that may be ambiguous without the proper background.

Finally, a verification method and description is provided with each requirement. These follow the definitions set in the ECSS-E-ST-10-02C standard but do not provide as much detail as recommended in that document. Further detail will be provided in the test case definitions in the future FFP verification plan document. Four methods can be used to test this package: Test, Analysis, Review of Design and Inspection (as defined in the ECSS standard). A summary of the requirements, the verification method and the current compliancy status is provided in a requirement verification matrix at the end of this document. This table also indicates which Test Case will be used to check the compliancy for each requirement.

This document consists of 3 sections. After this introduction, section 2 defines the requirements for the formation flying package then section 3 provides the requirement verification matrix as a summary.

# 2 Requirement Definitions

## 2.1 Mission Requirements

**AFF-M-01**
In the formation keeping phase, the formation flying package shall maintain a nominal inter-satellite distance (in the along-track direction).

**Rationale:** Based on mission requirement MIS-F-07 "The mission shall maintain at least one certain inter-satellite distance for scientific observation" and phase definition from the Phase-A Study.

**Verification:** Analysis.
Simulations using a MATLAB Simulink Environment with a representative processor–in–the–loop set up will be run with extreme values for the environmental variables to confirm performance. The MATLAB Simulink Environment will be validated by similarity (demonstration of the same performance of known (on-orbit) data in the simulated environment). If test equipment is available, hardware-in-the-loop tests may be run to further demonstrate performance using individual agents and a formation flying test bench that has been validated for its similarity to the orbit environment

**Compliancy:** Partial.
The nominal intersatellite distance maintenance has been shown to work in certain specific cases but has not been generalized to the entire DelFFi mission. Further the simulated results are based on validated models but have not be verified on hardware, beyond confirm functionality of the code on the target microprocessor.

**AFF-M-02**
The formation keeping phase of the mission shall last no less than 20 days [OPTIONAL: 30 days].

**Rationale:** Based on Phase A Study where a 20-30 day mission was identified and analysed.

**Verification:** Analysis.
Simulations using a validated MATLAB Simulink Environment to confirm all requirements are met throughout the lifetime.

**Compliancy:** Yes.
Based on a 6-day correction frequency, and correction magnitudes below 4 m/s in all cases, the 15 m/s propellant budget x 2 satellites – 1 m/s for acquisition will allow well above 30 days of operations.

## 2.2 Functional Requirements

**AFF-F-20**
The AFF payloads shall implement relative guidance and navigation using ground and/or onboard information.

**Rationale:** As per mission requirement SAT.1-F-01.

**Verification:** Review of Design.
Design documents of the AFF package's architecture will show relative guidance and navigation algorithms and their information sources from the ground and/or onboard sensors.

**Compliancy:** Yes.
The architecture used for the development of the formation flying package software works based on orbital elements from two line elements (TLEs) that would be provided from the ground.

**AFF-F-21**
The AFF payloads shall generate and implement relative control commands.

**Rationale:** As per mission requirement SAT.1-F-02.

**Verification:** Review of Design.
Design documents will detail the controller design used by the AFF payload.

**Compliancy:** Partial.
In the Planning Mode Simulink Model the Controller (Analytical Controller Function) generates relative control commands. Transferring these commands into the Monitoring Mode Simulink Model shows the theoretical implementation response. The implementation will be handled by other aspects of the AFF payload – the propulsion system, attitude determination and control system and the ADCS microcontroller in particular.

**AFF-F-22**
The satellite shall determine its position to within 10 km [OPTIONAL: 1 km] accuracy (TBC).

**Rationale:** The 10 km requirement is from mission requirement SAT.2.2-?-??. Gill's scaling law indicates a 100 km control window requires control accuracy of 10 km and thus a position estimate accuracy of 1 km (one order of magnitude for each level).

**Verification:** Analysis.
Simulations in a validated MATLAB Simulink Environment will show the accuracy of the position estimation algorithm for the satellite.

**Compliancy:** Partial.
Based on expected error contributions, the navigation accuracy has been shown to be better than 5.1 km. This has not yet been tested using real TLEs from NORAD for two CubeSats that are close together, nor has an analysis of the improved accuracy of the differential TLE compared to absolute TLE been performed.

**AFF-F-23**
The nominal inter-satellite distance shall be 1000 km.

**Rationale:** Based on mission analysis from the Phase-A Study for desired separation distance for relevant science use.

**Verification:** Review of Design.
Design documents will show that the nominal inter-satellite distance is used in all analyses.

**Compliancy:** Yes.
For all test cases the nominal intersatellite distance has been 1000 km.

**AFF-F-24**
The nominal inter-satellite distance shall be maintained to within 10% of its value – ie. 1000 km ± 100 km.

**Rationale:** Based on Phase A Study. Provides a limit that maintains the value of the synchronized science measurements, the linearity of the relative motion equations of the controller and provides enough margin to reduce frequency and magnitude of the propulsive manoeuvres.

**Verification:** Analysis.
Simulations in a validated MATLAB Simulink Environment will show the accuracy of the relative control algorithm for the satellites.

**Compliancy:** No.

A 100 km control window has not yet been shown to be successfully maintained for an entire mission. It is possible to maintain it for the six days between TLEs in some cases, and it is possible to correct from a 100 km error back to the nominal 1000 km however this correction is not sufficiently accurate to allow the six day maintenance to be met after the manoeuvre.

**AFF-F-25**
The attitude vector (the desired control acceleration direction) shall be requested no less than 15 minutes [5 minutes if reaction wheel is active] (TBC) in advance of a propulsive manoeuvre.

**Rationale:** The magnetorquers require as much as 15 minutes to change and re-stabilize the attitude of the CubeSat. If the reaction wheels are active, this can be reduced to 5 minutes.

**Verification:** Analysis.
Simulations in a validated MATLAB Simulink Environment will show that the time between calculation of the required thrust direction and the time that thrust should be applied is larger than the required delay.

**Compliancy:** Yes.
The Planning Mode is activated when a TLE is received that shows the satellites will exit the control window in the following two days. The Planner will determine a plan that starts between 3 and 12 hours after the TLE has been received. Since generation of the plan takes less than one clock cycle, the ADCS will have at least three hours' notice before the correction manoeuvre is set to begin.

**AFF-F-26**
The AFF payload shall require no more than 15 m/s delta-v during the mission lifetime to maintain the required control window.

**Rationale:** Only a certain volume of propellant is available on-board which limits the total manoeuvre capability of the satellite. As per the propulsion team this is sized at 15 m/s including some margin (see PROP-PERF-100).

**Verification:** Analysis.
Simulations in a validated MATLAB Simulink Environment will show the total propellant used over the mission lifetime is less than the maximum available.

**Compliancy:** Yes.
Analysis shows that assuming corrections every 6 days, with total magnitude well under 4 m/s (the worst, non-optimized case) and a 1 m/s acquisition manoeuvre (as per preliminary DelFFi analyses), 21 m/s total is required, divided over two satellites is 10.5 m/s. Including some margin this makes the 15 m/s feasible.

**AFF-F-27**
The burn time required from the thrusters shall be no less than 30 seconds (TBC) and no more than 300 seconds (TBC).

**Rationale:** The minimum is due to inefficiencies of short manoeuvres and maximum due to over-heating of the system.

**Verification:** Analysis.
Simulations in a validated MATLAB Simulink Environment will show the magnitudes of thrusts (measured here in burn time) do not exceed either the minimum or maximum values allowed by the propellant system.

**Compliancy:** No.
The analysis showed control manoeuvres that would be required that were significantly larger than what a 300 second burn can provide in most cases. This is considering both beginning and end of life thrust levels.

**AFF-F-28**
The thruster activation shall be requested no less than 300 seconds (TBC) in advance of the thrust start time.

**Rationale:** The thruster requires a heater which must be turned on in advance of the thrust start time.

**Verification:** Analysis.
Simulations in a validated MATLAB Simulink Environment will show that the time between calculation of the required thrust magnitude and the time that thrust should be applied is larger than the required delay. May be over-ridden by verification of AFF-F-25.

**Compliancy:** Yes.
See AFF-F-25.

**AFF-F-29**
[OPTIONAL] The propulsive manoeuvres shall be timed to occur during the second half of sun-lit periods only.

**Rationale:** The propulsion system requires significant power which may be difficult to provide during eclipse. If only a small thrust is requested it may be possible to perform during eclipse. The attitude of the satellite is only stabilized well enough to perform a thrust after approximately 30 minutes in the sun period (since the attitude control algorithm depends upon the sun vector).

**Verification:** Analysis.
Simulations in a validated MATLAB Simulink Environment will show the propellant manoeuvres do not occur outside of the allowed periods.

**Compliancy:** No.
The analytical controller selected requires at least one manoeuvre at the ascending node and at least one at the descending node. At least one of these will be in eclipse unless the orbit is a dusk dawn orbit and both will be near the terminator. This is not the case for the QB50 orbits.

**AFF-F-30**
[OPTIONAL] The propulsive manoeuvres shall be separated by at least one non-thrusting sun-lit period (ie. 2 orbital periods).

**Rationale:** Due to the high power demands, a full sun-lit period may be required to recharge the batteries before the next propulsive manoeuvre can occur. If only small thrusts are requested this time between manoeuvres may be reduced.

**Verification:** Analysis.
Simulations in a validated MATLAB Simulink Environment will show the propellant manoeuvres do not occur outside of the allowed periods. May be combined with verification of AFF-F-29.

**Compliancy:** Yes.
The analytical controller uses two variables (M and N) to define the number of half-orbits between manoeuvres. Their choice has been deliberate to ensure at least one non-thrusting orbit in between each manoeuvre.

**AFF-F-31**
[OPTIONAL] The AFF payload should maximize the time available for scientific observation (non-firing, in-track pointing).

**Rationale:** As part of QB50, the primary goal is scientific so the formation flying package should not hinder the functionality of the science payload.

**Verification:** Analysis.
Simulations in a validated MATLAB Simulink Environment will assess the average time between propulsive manoeuvres (including time used to realign the satellite in the thrust-direction).

**Compliancy:** Not Assessed.
Optional requirement not assessed at this time. As it is, 6 days between manoeuvres followed by 1 day of manoeuvres may be considered long duration between manoeuvres.

**AFF-F-32**
The AFF payload shall provide housekeeping parameters (to be detailed) at a rate of 1 Hz (TBC) to the On-Board Computer (OBC).

**Rationale:** In case of errors the OBC should be able to track critical parameters so they can be transmitted to the ground for troubleshooting.

**Verification:** Review of Design.
Design documents including description of the AFF architecture will demonstrate the rate of housekeeping data generation to match the required rate from the on-board computer.

**Compliancy:** Not Assessed.
No OBC or Ground operations team to coordinate with at this time. Unclear which parameters are interesting to the ground at every time step. Variation to the controller required for successful onboard implementation. Not assessed at this time.

**AFF-F-33**
[OPTIONAL] The formation flying package shall require no more than 500 ms per cycle when executed on the ADCS microcontroller.

**Rationale:** At each clock cycle, the ADCS microprocessor will first run the ADCS algorithm then the formation flying algorithm. The clock cycle of 1 second (corresponding to 1 Hz frequency) will be divided evenly between the ADCS and the formation flying algorithms as their relative requirements are not yet known. Based on preliminary simulations the ADCS is able to execute much more quickly than 500 ms so this limit is reasonable for the formation flying algorithms as well.

**Verification:** Test.
Simulations in a validated MATLAB Simulink Environment will assess the predicted maximum cycle time for the AFF package algorithms on a simulated processor (Preliminary Analysis). A validated processor-in-the-loop environment will be used to measure the maximum cycle time for the AFF algorithms on a representative target microprocessor.

**Compliancy:** Partial.
The Planning Mode was converted to C for testing on the ADCS microcontroller. It showed a run time of 5 ms on a 100 MHz clock. However, this implementation used hardcoded values rather than requesting them from the service layer and was not optimized for embedding. Also, flight hardware was not used – only a prototype testing board with debug features which may not be complete representative. Further, the other modes of the formation flying package have not been implemented on hardware.

## 2.3 Design Requirements

**AFF-D-40**
The formation flying package will be identical on both satellites.

**Rationale:** Based on mission requirement MIS-F-04 "The mission shall facilitate two identical satellites."

**Verification:** Review of Design.
Design documents will show the development of one formation flying package that is capable of being used on either (both) the chief or the deputy spacecraft.

**Compliancy:** Yes.
Only one formation flying package has been developed. For the chief (passive) satellite the FFP will be Inactive, for the deputy (active) satellite the FFP will be active.

**AFF-D-41**
The AFF payload shall be compatible with the nominal QB50 orbit:

Semi-major axis: 6758 km (380 km altitude)
Eccentricity: 0
Inclination: 96.6 degrees
Argument of perigee: 0 degrees
Right Ascension of the Ascending Node: -15 degrees
Mean Anomaly: 0 degrees

**Rationale:** The primary design point is the QB50 orbit which is (nominally) defined above as per DelFFi CubeSat Design Overview Report. This has been used for both the thermal design and ADCS design.

**Verification:** Analysis.
Simulations in a validated MATLAB Simulink Environment will show the compatibility of the AFF payload to the nominal orbit.

**Compliancy:** Partial.
The analysis showed that the FFP is compatible with this orbit for certain scenarios – specifically when the ballistic coefficients and the mean orbital elements of the satellites matched sufficiently well. Further analysis including statistical models with feedback from the ADCS will be required to confirm the FFP is successful in all cases.

**AFF-D-42**
[OPTIONAL] The AFF payload shall be compatible with a generic near-circular low earth orbit with the following ranges of parameters:

- Altitude: 300 km – 800 km
- Eccentricity: 0 – 0.015
- Inclination, Right Ascension of the Ascending Node, Ballistic Coefficient: up to TBD deviation between the two satellites

**Rationale:** A large range of orbits would be desirable to make the software multi-purpose for future use or in case of launch changes (since CubeSats are secondary payloads, orbit selection is not generally possible).

**Verification:** Analysis.
Simulations in a validated MATLAB Simulink Environment will show the compatibility of the AFF payload to various other orbits of interest.

**Compliancy:** Partial.
Analysis showed some of these orbits are viable in some situations – well matched ballistic coefficients and mean orbital elements – while a few are not viable in most cases due to excessive differential drag. From a control perspective, the controller selected is best suited to circular orbits; however it has been shown to be usable for several near circular orbits with eccentricities up to 0.015.

**AFF-D-43**
The formation flying package shall remain functional despite loss of the inter-satellite link.

**Rationale:** Due to procurement issues and ADCS pointing error levels, there is a risk that an inter-satellite link may not be included on the platform or that it may not provide a consistent connection between the satellites, thus the algorithm shall be designed such that its performance is non-critical.

**Verification:** Analysis.
Simulations in a validated MATLAB Simulink Environment will show that functionality is maintained with and without the inter-satellite link for the same input conditions.

**Compliancy:** Not Assessed.
Design has been developed assuming no intersatellite link is available. Development with an intersatellite link is expected to result in better performance, though this has not been assessed.

**AFF-D-44**
[OPTIONAL] The formation flying package shall remain functional despite loss of GPS fixes on one or both satellites.

**Rationale:** Due to procurement issues and known limitation of GPS systems on CubeSats it may not be possible to get consistent GPS fixes. For this reason, other methods of navigation must be assessed for their feasibility in the DelFFi mission.

**Verification:** Analysis.
Simulations in a validated MATLAB Simulink Environment will show that functionality is maintained with and without the global positioning system for the same input conditions.

**Compliancy:** Not Assessed.
Analysis has focused entirely on the TLE-based formation flying. Since TLE-based formation flying does not display optimum performance, GPS is recommended. The ability to use GPS as the primary with a TLE back up remains to be assessed.

## 2.4 Operational Requirements

**AFF-O-60**
[OPTIONAL] The AFF payload should allow the eight GNC Architectures, as defined in the Phase A Study.

**Rationale:** As per mission requirement SAT.1-F-03. See Phase A Study for detail of the various GNC Architectures that should be allowed.

**Verification:** Analysis.
Simulations in a validated MATLAB Simulink Environment will show the performance of the system in each of the eight architectures for the (otherwise) same input conditions.

**Compliancy:** Partial.
The FFP has been developed with the 4$^{th}$ scenario in mind. Other scenarios have not been assessed. The structure of the MATLAB Simulink environment can be made to accept these other scenarios.

**AFF-O-61**
The AFF payload shall facilitate autonomous operation (without human interaction).

**Rationale:** As per mission requirement SAT.1.-F-04.

**Verification:** Review of Design.
Design documents will describe the AFF package architecture that will not include humans in the loop.

**Compliancy:** Partial.
Formation Maintenance has been designed to be autonomous with the exception of the upload of TLEs to the satellites. Formation Acquisition however has not been assessed.

## 2.5 Interface Requirements

**AFF-I-80**
The Formation Flying Package shall be compatible with the ADCS microcontroller.

**Rationale:** As per Phase A Study, the formation flying package algorithms will be compiled on the ADCS microcontroller with an architecture and processing capability as defined in ADCS_Processor from Jaan Viru.

**Verification:** Test.
A validated processor-in-the-loop test environment will confirm the performance of the AFF package is maintained between the simulation environment and the target ADCS microcontroller.

**Compliancy:** Partial.
The formation flying package has been successfully tested with the processor in the loop using the Tiva C Series TM4C1294 Evaluation Kit. This kit was selected by the ADCS processor team as it most closely resembles the selected ADCS processor. However, the tests were completed in isolation with no other service layer or applications running making the test not completely representative of the onboard environment.

**AFF-I-81**
The AFF payload shall be compatible with a clock signal which uses Coordinated Universal time (UTC) as time reference.

**Rationale:** As per mission requirement SAT.2.6-?-?? "Any computer clock used on the CubeSat and on the ground segment shall exclusively use UTC as time reference." Hence the ADCS microprocessor shall provide time to the AFF payload in this format.

**Verification:** Review of Design.
Design documents will describe the AFF package architecture that will use the specified time reference.

**Compliancy:** Not Assessed.
The portion of the formation flying package that was executed on the microcontroller did not require a clock signal, though other portions – such as the onboard navigation function that will provide the interpreted and propagated orbital elements from the TLEs – will need a clock signal.

# 3 Requirement Verification Matrix

The above requirements are summarized in a requirement verification matrix as shown below in Table 2. The current compliancy standard is indicated along with the test case that will be used to assess the compliancy in the method indicated.

**Table 2: Requirement Verification Matrix**

| ID | Requirement | Method | | | | Compliancy | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Test | Analysis | Review of Design | Inspection | Yes | Partial | No | Not yet assessed |
| **AFF-M-01** | In the formation keeping phase, the formation flying package shall maintain a nominal inter-satellite distance (in the along-track direction). | | x | | | | x | | |
| **AFF-M-02** | The formation keeping phase of the mission shall last no less than 20 days [OPTIONAL: 30 days]. | | x | | | x | | | |
| **AFF-F-20** | The AFF payloads shall implement relative guidance and navigation using ground and/or onboard information. | | | x | | x | | | |
| **AFF-F-21** | The AFF payloads shall generate and implement relative control commands. | | | x | | | x | | |
| **AFF-F-22** | The satellite shall determine its position to within 10 km [OPTIONAL: 1 km] accuracy (TBC). | | x | | | | x | | |
| **AFF-F-23** | The nominal inter-satellite distance shall be 1000 km. | | | x | | x | | | |
| **AFF-F-24** | The nominal inter-satellite distance shall be maintained to within 10% of its value – ie. 1000 km ± 100 km. | | x | | | | | x | |
| **AFF-F-25** | The attitude vector (the desired control acceleration direction) shall be requested no less than 15 minutes [5 minutes if reaction wheel is active] (TBC) in advance of a propulsive manoeuvre. | | x | | | x | | | |
| **AFF-F-26** | The AFF payload shall require no more than 15 m/s delta-v during the mission lifetime to maintain the required control window. | | x | | | x | | | |
| **AFF-F-27** | The burn time required from the thrusters shall be no less than 30 seconds (TBC) and no more than 300 seconds (TBC). | | x | | | | | x | |
| **AFF-F-28** | The thruster activation shall be requested no less than 300 seconds (TBC) in advance of the thrust start time. | | x | | | x | | | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **AFF-F-29** | [OPTIONAL] The propulsive manoeuvres shall be timed to occur during the second half of sun-lit periods only. | | x | | | | | x | |
| **AFF-F-30** | [OPTIONAL] The propulsive manoeuvres shall be separated by at least one non-thrusting sun-lit period (ie. 2 orbital periods). | | x | | | x | | | |
| **AFF-F-31** | [OPTIONAL] The AFF payload should maximize the time available for scientific observation (non-firing, in-track pointing). | | x | | | | | | x |
| **AFF-F-32** | The AFF payload shall provide housekeeping parameters (to be detailed) at a rate of 1 Hz (TBC) to the On-Board Computer (OBC). | | | x | | | | | x |
| **AFF-F-33** | [OPTIONAL] The formation flying package shall require no more than 500 ms per cycle when executed on the ADCS microcontroller. | x | | | | | x | | |
| **AFF-D-40** | The formation flying package will be identical on both satellites. | | | x | | x | | | |
| **AFF-D-41** | The AFF payload shall be compatible with the nominal QB50 orbit. | | x | | | | x | | |
| **AFF-D-42** | [OPTIONAL] The AFF payload shall be compatible with a generic near-circular low earth orbit with the ranges of parameters (as specified above). | | x | | | | x | | |
| **AFF-D-43** | The formation flying package shall remain functional despite loss of the inter-satellite link. | | x | | | x | | | |
| **AFF-D-44** | [OPTIONAL] The formation flying package shall remain functional despite loss of GPS fixes on one or both satellites. | | x | | | | | | x |
| **AFF-O-60** | [OPTIONAL] The AFF payload should allow the eight GNC Architectures, as defined in the Phase A Study. | | x | | | | x | | |
| **AFF-O-61** | The AFF payload shall facilitate autonomous operation (without human interaction). | | | x | | | x | | |
| **AFF-I-80** | The Formation Flying Package shall be compatible with the ADCS microcontroller. | x | | | | | x | | |
| **AFF-I-81** | The AFF payload shall be compatible with a clock signal which uses Coordinated Universal time (UTC) as time reference. | | | x | | | | | x |

# Appendix C

# DelFFi Formation Flying Package Risk Assessment

The following Technical Note (DFF-TUD-TN-1163) for the DelFFi project includes the risk assessment and brief recommendations for risk mitigation for the Formation Flying Package.

# Technical risk assessment: Formation Flying Package

**Description:** This document contains the risk management of Formation Flying Package of DelFFi

**Subsystem(s) involved:**

| ADCS | CDHS | COMMS | EPS | MechS | STS | TCS | FIPEX | µPS+ | ISL | AFF | GSE | GSN | Launch |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | x | | | |

## Revision Record and Authorization

| Issue | Date | Author / Editor | Reviewer checked | PM approved | Affected Section(s) | Description of change |
|---|---|---|---|---|---|---|
| 0.9 | 15-09-2015 | A. Deeb | | | 1, 2, 6 | Added Reference to Amy's Thesis work, Updated TRL post-thesis, updated conclusion with results of thesis, added correct SLRs |
| 0.8 | 11-06-2015 | A. Deeb | | | 3, 4, 5 | Added Risks #18, 19. Changed consequence levels of risks #1 and #4. Reword of Risk #15. |
| 0.7 | 10-06-2015 | S. van Kuijk | | | 2 | Changed component tree |
| 0.6 | 03-06-2015 | A. Deeb | x | | 2, 4 | Added Component Tree, added full risk identifier to risk map table |
| 0.5 | 03-06-2015 | A. Deeb | | | 2 | Updated Context Diagram |
| 0.4 | 01-06-2015 | A. Deeb | | | 3, 4, 5 | Added Risk #17, reformatted table to comply with new template |
| 0.3 | 23-05-2015 | A. Deeb | | | 3, 4, 5 | Added additional risks related to processor/data flow |
| 0.2 | 21-5-2015 | A. Deeb | | | All | First Draft with FFP info |
| 0.1 | 19-3-2015 | S van Kuijk | | | All | Template design |
| | | | | | | |

## Action Items

| TBW | TBD | TBC | Applicable Section(s) | Description of action item |
|---|---|---|---|---|
| | x | | 3. Risk Identification, 5. Mitigation | RI.FFP.11, RI.FFP.13 must be assigned to someone and the corresponding mitigation plan must be developed |
| | | | | |

# List of Used References

| SLR code | Version | Data/Variable |
|---|---|---|
| 1162 | 1.0 | Technical Risk Management |
| 1111 | 1.0 | DelFFi Phase A Study on Formation Flight within QB50 (DFF-TUD-RP-1111) |
| TBD | 1.0 | Maintenance of a Long Baseline Along-Track Formation for the DelFFi Mission (MSc Thesis) |
| | | |

# Table of Contents

# 1 Introduction

This document will assess the technical risks pertaining to the formation flying package (FFP) payload for the DelFFi project. It follows the methodology provided in the risk management document SLR 1162. This is a living document that will be updated as the design of the FFP continues.

The DelFFi mission statement, as per SLR 1111, is as follows:

> *"The DelFFi mission shall demonstrate autonomous formation flying and provide enhanced scientific return within QB50 from 2015 onwards, by utilizing two identical triple-unit Cubesats of TU Delft which further advance the Delfi-n3Xt platform."*

Although the objective is to perform formation flying in an entirely autonomous, on-board way, the threshold for success will be the demonstration of formation maintenance without human interference. In other words, although it would be preferable to use only on-board sensors for navigation and an intersatellite link (ISL) for team communication, communication through the ground station and the use of ground-based sensors (specifically two-line elements (TLEs)) for navigation is considered acceptable for mission success.

The work relating to the Formation Flying Package has recently (January – August 2015) been progressed in the MSc Thesis entitled *Maintenance of an Along-Track Formation for the DelFFi Mission*. Based on this thesis, the TRLs have been updated in Section 2. Conclusions from this thesis work are discussed in Section 6 along with further recommendations for the FFP.

# 2 System description

The formation flying package is a software package that will maintain a relative state between the two DelFFi satellites. A context diagram is provided below to describe the interfaces to the rest of the DelFFi project.

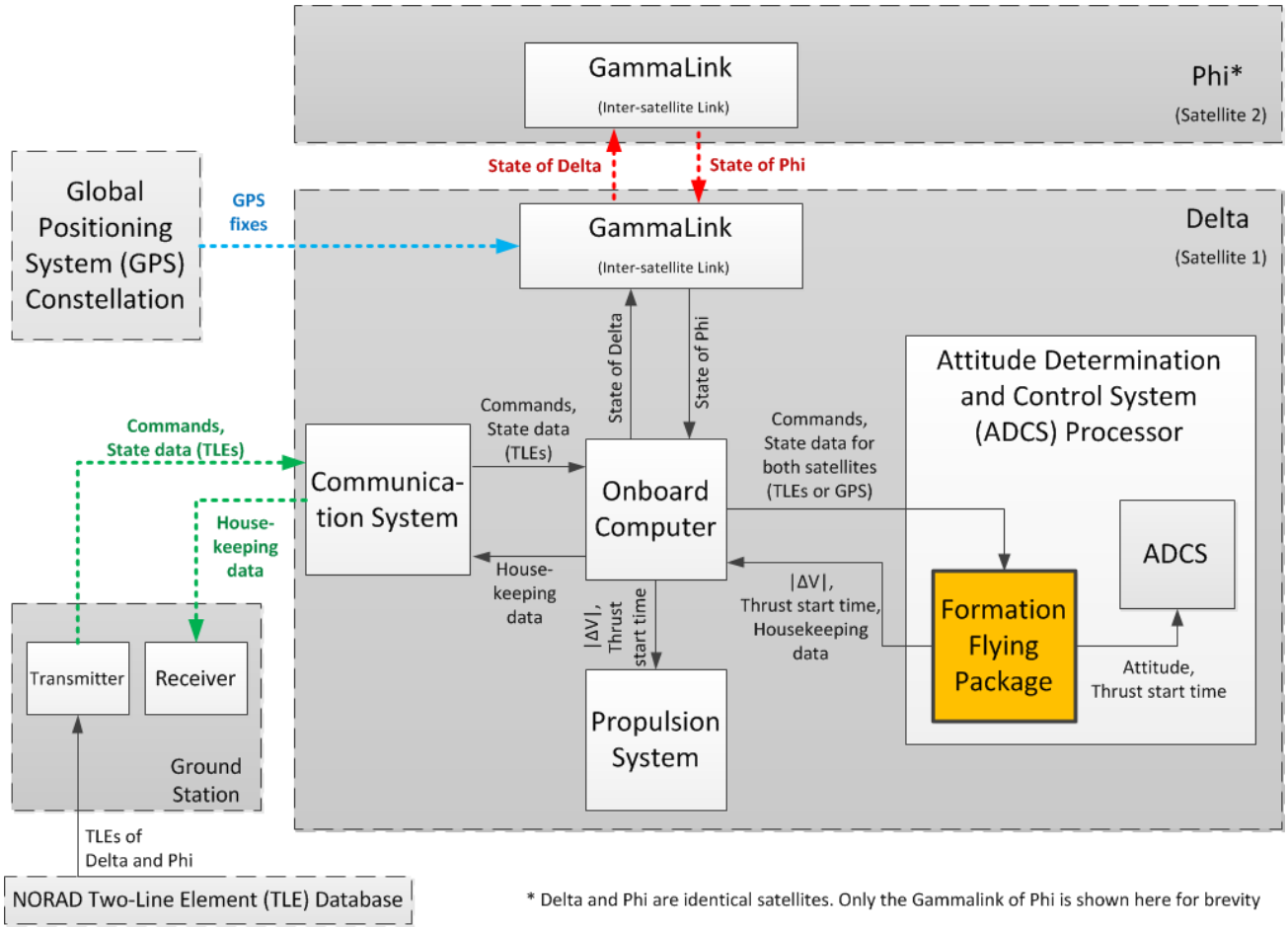



**Figure 2-1: Context Diagram**

The formation flying package will be developed in several pieces that are then integrated together to form the overall package. These individual algorithms are identified below in Table 2-1: Components. The table also contains a corresponding TRL (as defined in SLR 1162) for each element. To better visualize this table, a component tree diagram is provided in Figure 2-2.

**Table 2-1: Components**

| **Software** | | |
|---|---|---|
| **ID** | **Description** | **TRL** |
| FFP.s1. | Relative Navigation Algorithm | 4 |
| FFP.s2. | Formation Maintenance Controller | 3 |
| FFP.s3. | Team Management | 3 |
| FFP.s4. | Thruster Interface | 3 |
| FFP.s5. | ADCS Interface | 3 |
| FFP.s6. | OBC Interface | 3 |
| **Total system** | | |
| **ID** | **Description** | **TRL** |
| FFP | Formation Flying Package | 3 |

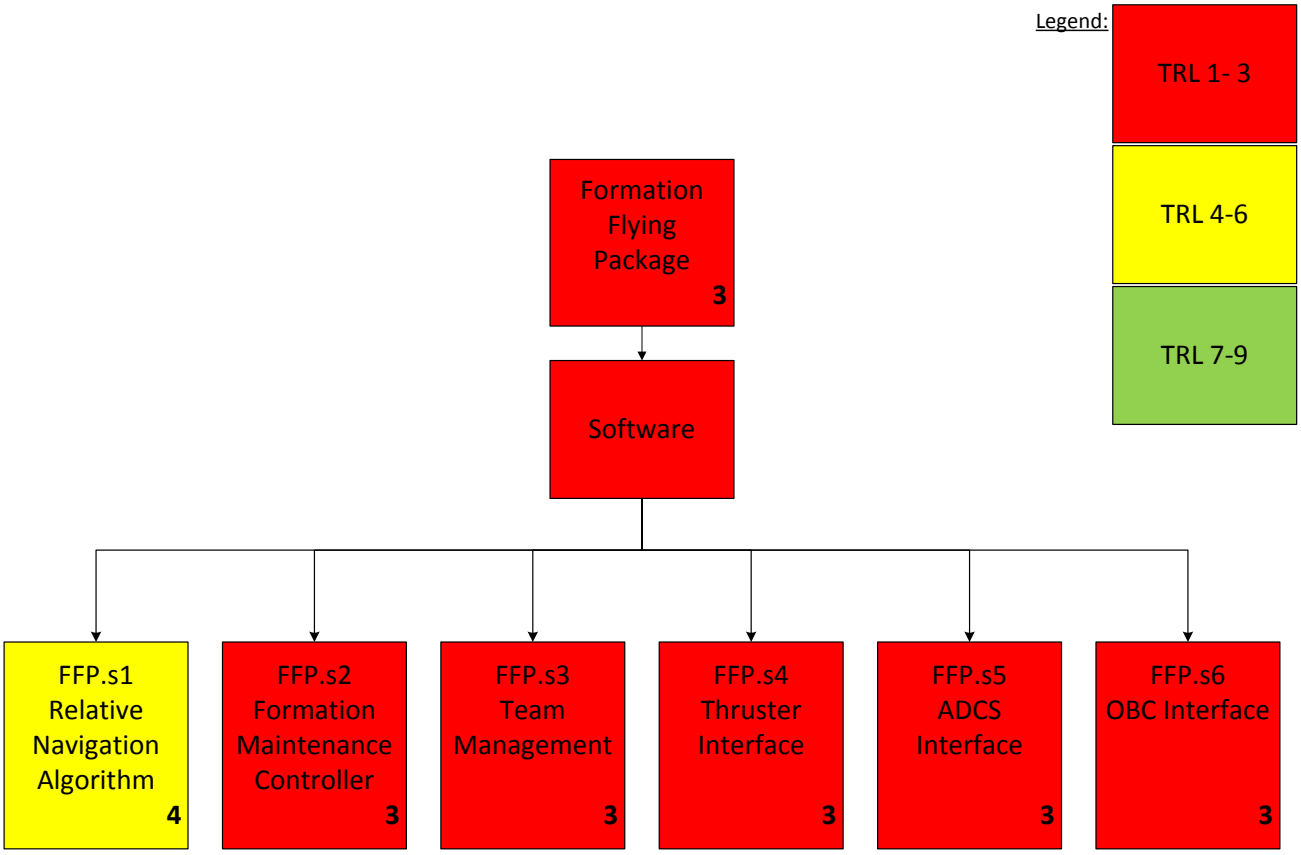



**Figure 2-2: Component Tree Diagram**

As this is a living document, the following table is included to provide descriptions of changes to these components and their TRLs.

**Table 2-2: Changelog component table**

| Changelog date | Component ID | Description |
|---|---|---|
| 30/08/2015 | FFP.s1. | TRL increased from 3 to 4 as a result of Thesis work |
| 30/08/2015 | FFP.s2. | TRL increased from 2 to 3 as a result of Thesis work |
| 30/08/2015 | FFP.s3. | TRL increased from 2 to 3 as a result of Thesis work |
| 30/08/2015 | FFP.s5. | TRL increased from 2 to 3 as a result of Thesis work |
| 30/08/2015 | FFP | TRL increased from 1 to 3 as a result of Thesis work |
| | | |

# 3 Risk identification

Several risks have been identified for the Formation Flying Package. Risk identification has been limited to top level technical and operational risks that will affect the functioning of the formation flying package in particular. Risks that affect the entire satellite (such as OBC failures or launch risks etc) have not been included.

**Table 3-1: Risk identification**

| ID | | | | Owner |
|---|---|---|---|---|
| RI.FFP.1. | **Description** | Intersatellite link hardware not available/functional | | A. Deeb |
| | **Likelihood** | 4 | High procurement uncertainty for the GammaLink system | |
| | **Consequence** | 2 | Can continue mission through ground station, no longer autonomous but meets threshold for mission success, should add automated ground station pass-through | |
| | **Status** | Mitigation planned (ground station automation) | | |
| | | | | |
| RI.FFP.2. | **Description** | Both navigation data sources unavailable simultaneously | | A. Deeb |
| | **Likelihood** | 2 | The following causes for one of the two possible navigation sources being unavailable have been identified (likelihood of each is in brackets):<br>- The GPS receiver integrated in the GammaLink system has high procurement uncertainty (4)<br>- Pointing the GPS receiver to receive sufficient signals may not be possible/continuous (3)<br>- TLEs not available at all for long duration (such as not able to detect CubeSats reliably) (2)<br>- TLEs at much lower rate than expected (3)<br>- TLEs incorrect (ex. Mixing up the two CubeSats) (2)<br>The combination of both GPS and TLEs unavailable at the same time however is considered low likelihood | |
| | **Consequence** | 4 | FFP cannot function without navigation data, thus mission will need to be redefined | |
| | **Status** | Mitigation planned | | |
| | | | | |
| RI.FFP.3. | **Description** | Pointing control not functional | | M. Haghayegh |
| | **Likelihood** | 3 | Experience from Delfi-n3Xt indicates moderate risk of failure early in operational lifetime – potentially before formation flying will begin. | |
| | **Consequence** | 3 | Formation flying requires active control with defined attitude angles. If the tumble rate is very low and the thruster is well aligned (limited disturbance torques due to thrusting), it may still be possible to perform manoeuvres but unlikely to be possible | |
| | **Status** | Mitigation planned | | |
| | | | | |
| RI.FFP.4. | **Description** | Reaction wheel not available/functional | | M. Haghayegh (support of W. Wu, A. Deeb) |
| | **Likelihood** | 5 | The reaction wheel lifetime is unknown at this time, but based on previous hardware it is expected to be less than other ADCS components and possibly less than the mission lifetime (3 months) so a failure at some point is guaranteed | |
| | **Consequence** | 2 | The reaction wheel is necessary to ensuring a stable | |

| | | | | |
|---|---|---|---|---|
| | | | pointing direction during thrusts. Without keeping a stable direction the thrust vector may change rapidly making the system uncontrollable and requiring the end of the formation flying phase of the mission | |
| | **Status** | Mitigation planned | | |
| | | | | |
| RI.FFP.5. | **Description** | Thruster unavailable/fails before mission complete | | T. van Wees |
| | **Likelihood** | 3 | Experience from Delfi-n3Xt indicates moderate risk of failure early in operational lifetime – potentially before formation flying will begin. | |
| | **Consequence** | 4 | Formation flying requires active control using the thruster | |
| | **Status** | Mitigation planned | | |
| | | | | |
| RI.FFP.6. | **Description** | Insertion difference outside tolerance | | A. Deeb |
| | **Likelihood** | 1 | The two satellites will be inserted from the same ISI-POD, thus the risk of them being inserted into very different orbits is quite low, however it is possible that a unexpectedly long time delay between the two releases or other large disturbance could cause the difference between the injections to be too large | |
| | **Consequence** | 4 | Due to low propellant, an orbit manoeuvre to move the two satellites back to the same orbit may be too extreme and without being in nearly the same orbit the FFP will not be able to function | |
| | **Status** | Should be accepted | | |
| | | | | |
| RI.FFP.7. | **Description** | Duration of LEOP too long | | A. Deeb (to be replaced w/ operations manager) |
| | **Likelihood** | 3 | LEOP can have some unexpected delays as various subsystems are initialized and tested | |
| | **Consequence** | 3 | During LEOP the satellites will drift apart. If the time before the drift stop manoeuvre is too large the satellites may drift apart so far as to not be able to return them to the 1000 km nominal separation distance. A different distance would be acceptable if it is near this value, however, the propellant requirement may make it not possible to establish the formation in the first place. | |
| | **Status** | Should be mitigated | | |
| | | | | |
| RI.FFP.8. | **Description** | Magnitude of thrust in single impulse lower than required | | T. van Wees |
| | **Likelihood** | 3 | Based on the T3uPS the difference between the expected thrust magnitude and the experimental magnitude was 50%. It is also possible that the maximum thrusting time (which leads to less magnitude) is reduced due to thermal limitations or power budgets. | |
| | **Consequence** | 2 | If the impulse is too low, many thrusts will be necessary which will require a variation to the FFP but it is not expected that the thrust level will be so low as to stop the FFP from functioning at all (this case would be considered as the same as RI.FFP.5). | |
| | **Status** | Mitigation planned | | |
| | | | | |
| RI.FFP.9. | **Description** | Relative dynamics model not suitable to real environment | | A. Deeb |
| | **Likelihood** | 2 | Model will be compared with several other research | |

| | | | | |
|---|---|---|---|---|
| | | | teams to ensure it is representative | |
| | **Consequence** | 4 | FFP will be unstable if dynamics are incorrect | |
| | **Status** | Mitigation planned | | |
| | | | | |
| RI.FFP.10. | **Description** | Error when loading software onto ADCS Microcontroller | | A. Deeb |
| | **Likelihood** | 2 | The FFP will be verified using Processor-in-the-loop testing to ensure it is compatible with the ADCS microcontroller. It will already be packaged and tested as a single unit which should reduce the likelihood of errors in transferring the software onto the flight hardware | |
| | **Consequence** | 2 | If there is an error in the parameters the control may become unstable. However, parameters can be changed by the ground in order to override errors (the most likely error in the software loading). | |
| | **Status** | Mitigation planned | | |
| | | | | |
| RI.FFP.11. | **Description** | Clock Offset between ADCS, OBC, Actuator and/or GND | | unknown |
| | **Likelihood** | 2 | OBC synchronizes the various clocks frequently (every second, TBC) | |
| | **Consequence** | 1 | Clock offsets will lead to larger state estimation errors (due to the on board propagator requiring the time compared to epoch to interpret the TLEs). However, unless the asynchronization becomes larger than a few seconds, the magnitude of this error will be low. | |
| | **Status** | Mitigation planned | | |
| | | | | |
| RI.FFP.12. | **Description** | Error in storage of data | | A. Deeb |
| | **Likelihood** | 2 | Radiation effects can cause bit-flips in the data such that the incorrect parameters are accessed by the algorithm, this is expected to happen very rarely | |
| | **Consequence** | 2 | Reading the wrong parameters the control may become unstable, however parameters can be overridden by the ground so the larger mission is not affected. | |
| | **Status** | Mitigation planned | | |
| | | | | |
| RI.FFP.13. | **Description** | Error in transfer of data between ADCS and OBC | | unknown |
| | **Likelihood** | 4 | I2C lock up, (also, bit flips in transfer) are very likely to occur during the mission (see for example the experience with Delfi-n3Xt). The frequency of these lock ups is expected to be moderately-high. | |
| | **Consequence** | 3 | Depending on the frequency of these errors, it is possible that they are in no way intrusive to the larger mission. However, it is possible that they occur so often that the functioning of the FFP is inhibited as the connection to the thruster is effectively severed. | |
| | **Status** | Should be mitigated | | |
| | | | | |
| RI.FFP.14. | **Description** | Invalid (error in) Telecommand | | A. Deeb (and operations manager)) |
| | **Likelihood** | 5 | As humans are part of the telecommand loop it is extremely likely that at some point there will be an error in the command. | |
| | **Consequence** | 4 | If the error affects solely the performance of the FFP it may cause a fatal error that would require the shutdown | |

| | | | | |
|---|---|---|---|---|
| | | | of the FFP system. | |
| | **Status** | Mitigation planned | | |
| | | | | |
| RI.FFP.15. | **Description** | Incorrect propulsive manoeuvre instruction | | A. Deeb |
| | **Likelihood** | 2 | Due to the maximum on-time of the thrusters and limited power, it is very unlikely that the manoeuvre instruction (either calculated on ground or on-orbit) will be so large as to cause a significant error (there is an forced limit placed on the magnitude due to the hardware). | |
| | **Consequence** | 5 | If this occurs early in the mission life it could cause a collision or lead to relative motion that is too large to be recovered with the limited propellant budget | |
| | **Status** | Mitigation planned | | |
| | | | | |
| RI.FFP.16. | **Description** | Propellant used before end of mission life | | A. Deeb |
| | **Likelihood** | 2 | An error in the algorithm could cause larger or more frequent actuation than necessary thus expelling propellant. However, this is unlikely due to the testing procedures for the model | |
| | **Consequence** | 4 | If there is no propellant it is no longer possible to perform formation flying manoeuvres and the active orbit control will stop. | |
| | **Status** | Mitigation planned | | |
| | | | | |
| RI.FFP.17. | **Description** | Development exceeds time estimate | | A. Deeb |
| | **Likelihood** | 3 | As this is a student project it is moderately likely to run over schedule. Margin has been included in the original planning. | |
| | **Consequence** | 2 | A delay of 8 weeks overall is expected at the worst for the development of the satellite as a whole not including margin. Further delay would be caused by the inability to use the design as currently proposed which will require a new student and nearly a year for them to get up to speed and continue/complete development. | |
| | **Status** | Mitigation planned | | |
| | | | | |
| RI.FFP.18. | **Description** | Performance of selected controller with TLEs does not meet requirements | | A. Deeb |
| | **Likelihood** | 3 | Literature indicates that TLEs are not suitable for Formation flying algorithms, however the requirements of DelFFi and the experience of the TU Delft department (Prof. Gill in particular) indicates that in certain circumstances (such as the DelFFi mission) this should be possible | |
| | **Consequence** | 5 | The selected controller was chosen to give the best possible chance for a TLE-based formation algorithm, if it does not function it is very unlikely that it is possible to use TLE-based formation flying. Instead, GPS will need to be available or other methods of relative navigation will need to be assessed – both requiring over 6 months of additional work. | |
| | **Status** | Should be planned | | |
| | | | | |

| | | | | |
|---|---|---|---|---|
| RI.FFP.19. | **Description** | Model for TLE availability/accuracy does not match reality | | A. Deeb |
| | **Likelihood** | 2 | There is a large amount of experience with TLEs based on other CubeSat missions, as well as Delfi-n3Xt and Delfi-C3 locally at the TU Delft making the likelihood of not matching low. It is possible that the update frequency or measurement method of TLEs changes as it is under control of NORAD and not the TU. | |
| | **Consequence** | 2 | Variation unlikely to be large so mission success should still be possible | |
| | **Status** | Should be accepted | | |
| | | | | |
| RI.FFP.20. | **Description** | | | |
| | **Likelihood** | | | |
| | **Consequence** | | | |
| | **Status** | | | |
| | | | | |
| | | | | |

As the FFP is still under development there is a scheduling risk associated with this development that was not included in the list of technical risks above. If the preliminary design of the FFP fails to meet the requirements, either the mission must be redefined to remove the FFP, or more time (and possibly a new student) must be added. This would have a moderate to high impact. As far as it is possible to estimate, the likelihood of this risk occurring is low. Time has been allotted for testing, optimization and small iterations of the design which helps to mitigate this risk, however as it is a student project this risk must be considered.

# 4 Risk map

The risks identified in section 3 are visualized below on a risk map.

**Table 4-1: Risk map**

| Likelihood >> | | | | | |
|---|---|---|---|---|---|
| | | | | RI.FFP.14 | |
| | | RI.FFP.1 | RI.FFP.13 | | |
| | | RI.FFP.17 | RI.FFP.3, RI.FFP.7 | RI.FFP.5 | RI.FFP.18 |
| | RI.FFP.11 | RI.FFP.10, RI.FFP.12, RI.FFP.19 | RI.FFP.8 | RI.FFP.2, RI.FFP.4, RI.FFP.9, RI.FFP.16 | RI.FFP.15 |
| | | | | RI.FFP.6 | |
| | Consequence >> | | | | |

| |
|---|
| Very high |
| High |
| Moderate |
| Low |
| Very low |

# 5 Mitigation

The mitigation planned for each of the risks identified in section 3 is summarized below in Table 5-1. Where a formal mitigation plan has been documented, references to those documents are provided.

**Table 5-1: Risk mitigation table**

| ID | Mitigation plan |
|---|---|
| RI.FFP.1. | Moderate; Risk of no ISL must be accepted (programmatically could look for other hardware to fill this gap but risk low enough to not require this). Automation of the ground station to allow pass through of data from one satellite to the other should be developed to compensate for this loss. |
| RI.FFP.2. | Moderate Risk; The availability of the GPS is not able to be mitigated from a technical standpoint, however the FFP should be designed in such a way to make the GPS non-critical assuming TLEs are available at a reasonable rate (likelihood of unavailable TLEs is much lower than GPS unavailability). |
| RI.FFP.3. | Moderate Risk; The lack of accurate pointing control will be mitigated through thorough testing of the ADCS algorithms as completed by the ADCS design engineering. From the FFP perspective the FFP controller should be tested to determine the absolute limits on the acceptable pointing error and attitude rates. |
| RI.FFP.4. | Moderate; Thorough testing will increase reliability of reaction wheel. For the FFP constraints on thrust maximum levels should be established such that it is possible to perform actuation without reaction wheels. |
| RI.FFP.5. | High; Thruster testing will be completed over the coming weeks to decrease this risk. May want to determine if other systems are available to replace the system. |
| RI.FFP.6. | Low; Risk must be accepted (no control to external launch provided to address insertion) |
| RI.FFP.7. | Moderate; A plan for LEOP should be established to ensure that the time before the drift stop manoeuvre is sufficiently short to mitigate this risk. The FFP should determine what the expected max/min duration of LEOP is acceptable and determine how much longer can be compensated for with the current propellant limitations. |
| RI.FFP.8. | Moderate; The thruster will be tested over the coming weeks to determine the impulse magnitudes. The FFP should be tested to determine the limit on the minimum impulse magnitude that would be acceptable. |
| RI.FFP.9. | Moderate; The relative dynamics model will be tested to ensure it is comparable to those used by other successful formation flying satellites (for ex. CanX-4/5 and PRISMA) |
| RI.FFP.10. | Low; The FFP software should be loaded onto the flight ADCS microcontroller as a single unit and should be tested using processor-in-the-loop tests to confirm its performance. The ability to change parameters on-orbit will be highly valuable and all critical parameters should be checked for their ability to be over-written by the ground. |
| RI.FFP.11. | Very low; The clock synchronization should happen frequently enough that the state estimation error is tolerable. |
| RI.FFP.12. | Low; All critical parameters should be able to over-written by the ground. |
| RI.FFP.13. | High; Mitigation plan to be determined |
| RI.FFP.14. | Very high; Telecommands that affect the FFP should be designed such that they do not permanently stop the functioning of the FFP. No command should allow the variation of a parameter in a way that makes it impossible to re-alter at a later time. |
| RI.FFP.15. | Moderate; Safety procedures should be put into place to prevent actuation that would create extreme orbital manoeuvres or potential collisions. |
| RI.FFP.16. | Moderate; Ground procedures to confirm the proper functioning of the FFP and specifically the rationing of propellant should be developed. |
| RI.FFP.17. | Low; Schedule should be built with margin for development time in case of delays (this has been done). Additional support from PhDs, experts/professors for the most risky aspects should be acquired. New students to continue development should be recruited so the |

| | |
|---|---|
| | transition is as smooth as possible. |
| RI.FFP.18. | High; Data on the performance of the TLE-based algorithm should be gathered as soon as possible to be able to assess the ability to proceed in this development line. |
| RI.FFP.19. | Low; TLE standards should be checked carefully to ensure they do not change during the development/operation of DelFFi. Extreme values for availability (rate) and accuracy should be used in all simulations. |

# 6 Conclusion and recommendations

Based on the Technical risks identified and assessed in this report, the following recommendations are made to mitigate the risks:

- Automation of the ground station to allow pass through of data from one satellite to the other should be developed to compensate for the lack of ISL.
- The FFP should be designed in such a way to make the GPS non-critical (since the likelihood of unavailable TLEs is much lower than GPS unavailability).
- The FFP controller should be tested to determine the absolute limits on the acceptable pointing error and attitude rates
- The FFP should establish constraints on thrust maximum levels, such that it is possible to perform actuation without reaction wheels
- The FFP should determine what the expected max/min time is needed to establish the formation and determine what absolute maximum time before the drift stop manoeuvre would be acceptable (given propellant limits)
- The FFP should be tested to determine the limit on the minimum impulse magnitude that would be acceptable
- The relative dynamics model will be tested to ensure it is comparable to those used by other successful formation flying satellites (for ex. CanX-4/5 and PRISMA)
- The FFP software should be loaded onto the flight ADCS microcontroller as a single unit and should be tested using processor-in-the-loop tests to confirm its performance
- All (critical) parameters should be able to be over-written by the ground
- The largest tolerable state estimation error should be determined for the FFP such that the clock synchronization rate is sufficiently high
- Telecommands that affect the FFP should be designed such that they do not permanently stop the functioning of the FFP
- No telecommand should allow the variation of a parameter in a way that makes it impossible to re-alter at a later time
- Safety procedures should be put into place to prevent actuation (either instructed from the ground or calculated on-board) that would create extreme orbital manoeuvres or potential collisions
- Ground procedures to confirm the proper functioning of the FFP and specifically the rationing of propellant should be developed.
- Additional support from PhDs, experts/professors for the most risky aspects (the controller) should be acquired.
- New students to continue development should be recruited so the transition is as smooth as possible.
- Data on the performance of the TLE-based algorithm should be gathered as soon as possible to be able to assess the ability to proceed in this development line.
- Extreme values for availability (rate) and accuracy of TLEs should be used in all simulations.

This Risk assessment was used to help direct the Formation Flying Payload Thesis work. As a result of the thesis the TRL of the FFP was increased from 1 to 3. It was shown in that thesis work that TLE-based navigation/formation flight is not recommended for DelFFi, especially if TLEs are the sole source of navigation data. Since the use of TLEs was a primary assumption in the development of this risk assessment, it should be further updated in the future to address the use of other navigation sources (presumably GPS). The developments with respect to GPS-based formation flying will result in a need for a new student to perform the GPS development, integration and testing. This will require at least one MSc student (7 months plus lead-in time) to complete and does cause a scheduling problem for the DelFFi mission as discussed in Risk RI.FFP.18. Further analysis of what this means on a system level must be performed.

# Appendix D

# Matlab Scripts and Simulink Models

The following Technical Note (DFF-TUD-TN-1166) for the DelFFi project describes the Simulink models created for the Formation Flying Package. The Software has been placed on the Delfi drive in the Working Directory of the author.

# Formation Flying Payload MATLAB Simulink Software

**Description:** Software developed for the FFP was created in MATLAB Simulink and is described here.

| Subsystem(s) involved: | ADCS | CDHS | COMMS | EPS | MechS | STS | TCS | FIPEX | µPS+ | ISL | AFF | GSE | GSN | Launch |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | x | | | |

## Revision Record and Authorization

| Issue | Date | Author / Editor | Reviewer checked | PM approved | Affected Section(s) | Description of change |
|---|---|---|---|---|---|---|
| 0.1 | 15-Sep-15 | A. Deeb | | | All | First Draft |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

## Action Items

| TBW | TBD | TBC | Applicable Section(s) | Description of action item |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

## List of Used References

| SLR code | Version | Data/Variable |
|---|---|---|
| 1165 | 0.1 | Formation Flying Payload Microcontroller Software |
| | | |
| | | |
| | | |
| | | |

# Table of Contents

# 1 Introduction

This technical note will describe the software developed for the formation flying package (FFP). The FFP identified several modes, two of which were developed in Simulink: Planning Mode and Monitoring Mode. The functions and scripts developed for these modes are described in Sections 2 and 3 respectively. The code itself can be found on the DelFFi drive in the working directory of the author (Amy Deeb).

# 2 Planning Mode

The Formation flying package Planning Mode takes the incoming satellite states and determines the timing and magnitude(s) of the correction manoeuvre(s) required to return the satellites to the nominal separation distance. For simulation in MATLAB Simulink, orbit propagators are used to produce the satellite's states then these are fed into the relative navigation and control blocks as shown in Figure 2.1.

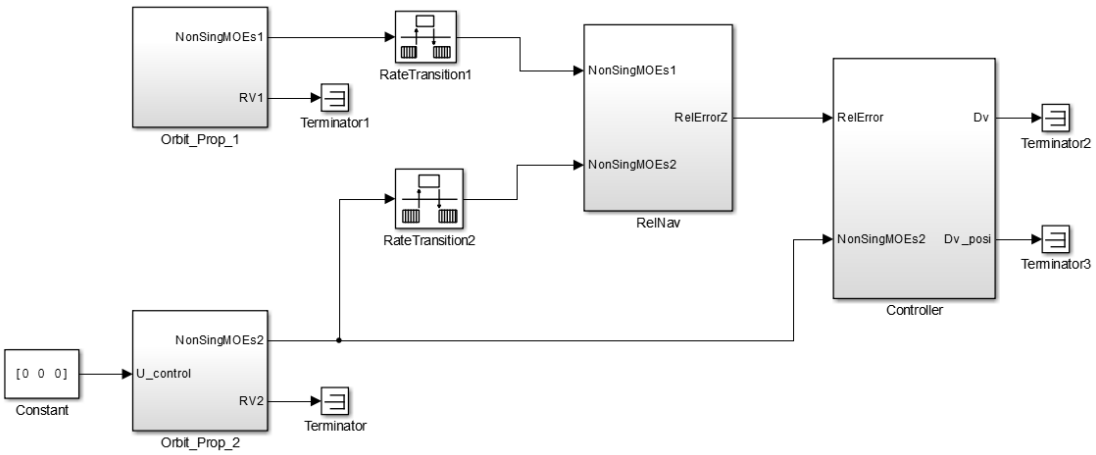



**Figure 2.1: Planning Mode Simulink Model**

There are three main aspects the orbit propagators (identical for the two satellites), the relative navigation function and the controller. The internals of these blocks are shown in Figures 2.2, 2.3 and 2.4. The orbit propagator is a numerical propagator developed and validated by Prem Sundaramoorthy. It integrates the accelerations due to gravity (up to J20,20 model), drag and control accelerations (if applicable) to generate the positions and velocities in the Earth Centred Inertial frame (ECI). The original code is on the Delfi Drive in the Working Directory of Amy Deeb. In addition to the code from the orbit propagator that can be found in the Planning Mode MATLAB folder, the following files have been added: CalcEA.m, mean2osculating.m, nuFromM.m, Run_FFP.m and uC_Inputs_Outputs.m. The complete list of files in the Planning Mode folder is in Appendix A.1.

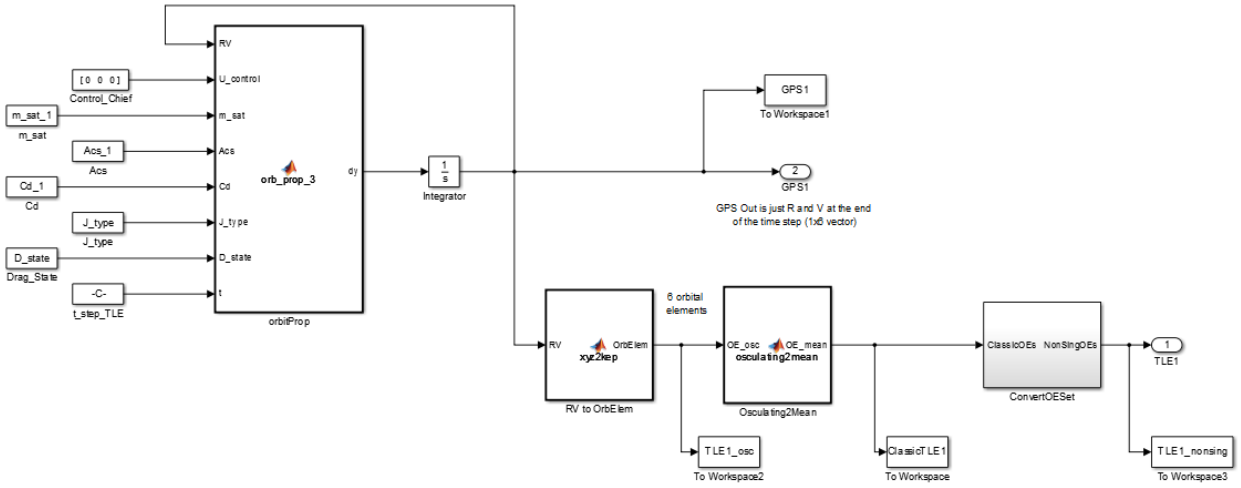



**Figure 2.2: Planning Mode Simulink Model: Orbit Propagator (Chief)**

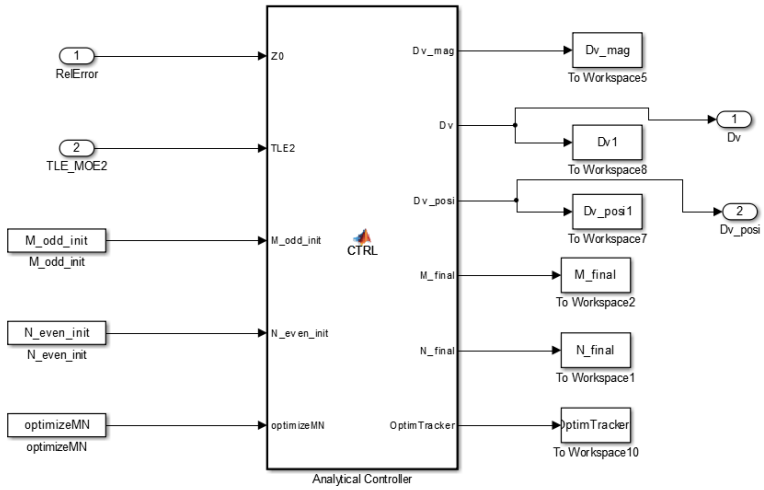



**Figure 2.3: Planning Mode Simulink Model: Controller**

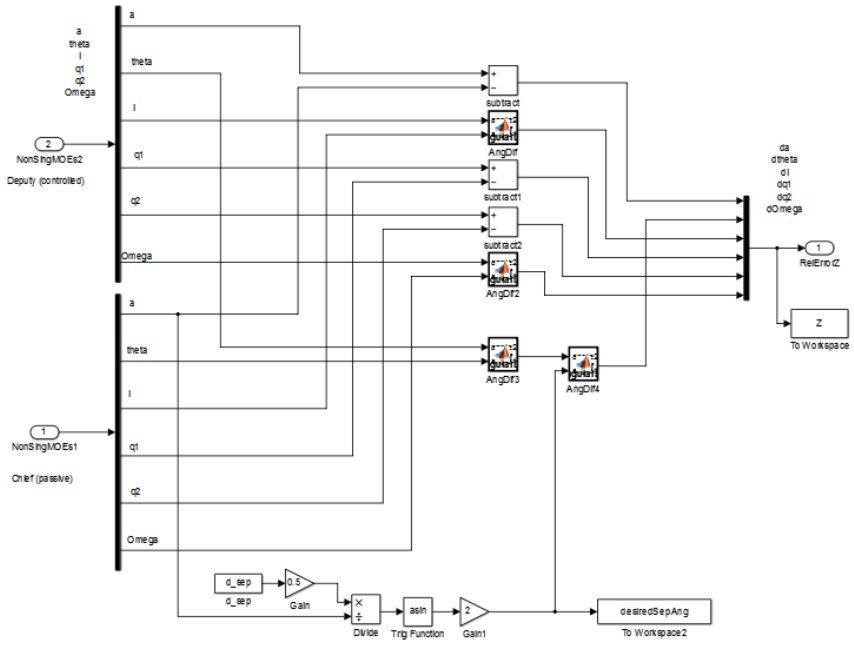



**Figure 2.4: Planning Mode Simulink Model: RelNav**

CalcEA.m and nuFromM.m convert from mean anomaly (M) to eccentric anomaly (EA) to true anomaly (nu). These files were created by Richard Rieber for an orbital mechanics library for mathworks (http://nl.mathworks.com/matlabcentral/fileexchange/13439-orbital-mechanics-library/content/) and have been validated against course notes from the TUDelft. Mean2osculating.m maps a set of mean classical orbital elements (such as those from TLEs) to the corresponding set of osculating classical orbital elements. The equations were provided in "Analytical Mechanics of Space Systems" by Schaub and Junkins. Run_FFP.m provides a single interface to initialize the relevant variables and then run the Simulink model (FFP_v1) that corresponds to the planning mode. It then processes the data and plots certain variables and saves the entire workspace.

This Simulink model was later converted to 'C' using Simulink Coder for execution on the target microcontroller. This process and the results are described in SLR 1165.

# 3 Monitoring Mode

The formation flying monitoring mode takes the planned manoeuvres from the Planning mode and propagates the state forward with these control accelerations planned. To do this, an actuator model is combined with the orbit propagators and relative navigation functions from the Planning Mode Simulink Model as shown in Figure 3.1.

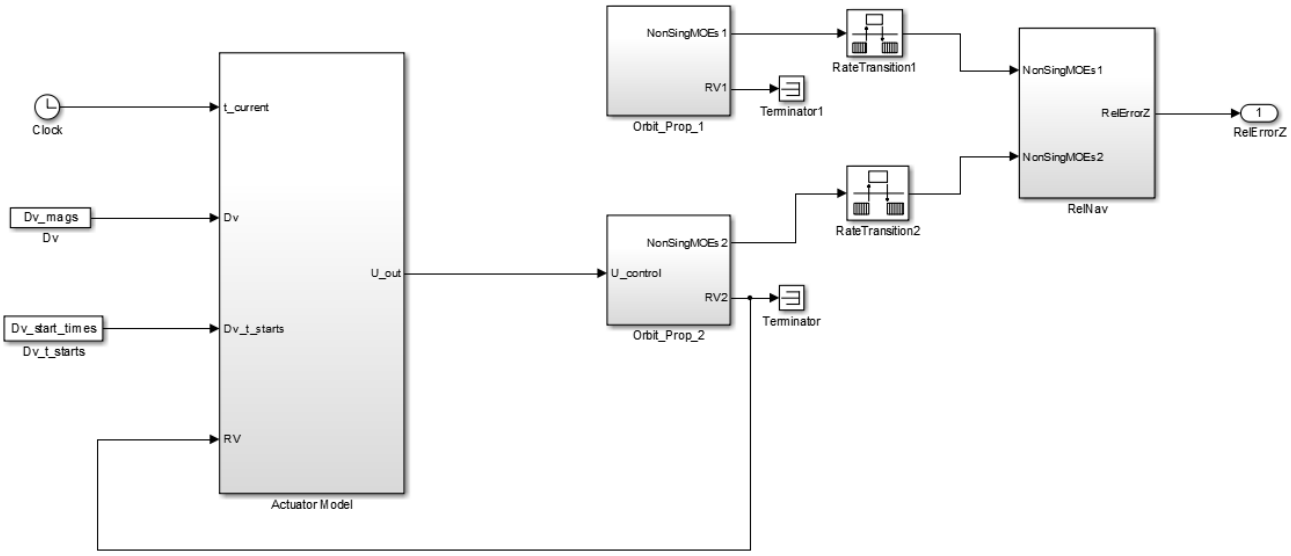



**Figure 3.1: Monitoring Mode Simulink Model**

The contents of the Orbit Propagators and Relative Navigation Function are not varied from the Planning Mode described above. However the Actuator Model is a new addition and will be detailed in Section 3.1. The MATLAB folder for the Monitoring Mode contains only the files described for the Planning Mode – see Appendix A.3 for the list of files.

The monitoring mode will be executed by the propulsion system to minimize the interface between the system performing the manoeuvre (propulsion) and the algorithm confirming the manoeuvre is being executed properly (monitoring mode software). Ideally, it would have live feedback from the ADCS to ensure the angular pointing error is properly accounted for as it has a large impact both on the thrust delivered and the differential drag.

## 3.1 Actuator Model

The actuator model attempts to model both the propulsion system and the attitude determination and control system. Neither of these systems have completed their baseline design, so theoretical designs and general expected values have been used in the model. The internals of the Actuator model are shown in Figure 3.2. Not that the gain in the top right hand corner of this figure was used for debug purposes to turn on (Gain = 1) and off (gain = 0) the control acceleration in the orbit propagator, and is now obsolete.

Within the Actuator Model Function three steps are taken:

(1) Determine if the thruster should be on at the current time
(2) Determine the magnitude of the thrust that would be generated given the total time the thruster has been on
(3) Determine the direction the thrust will be applied

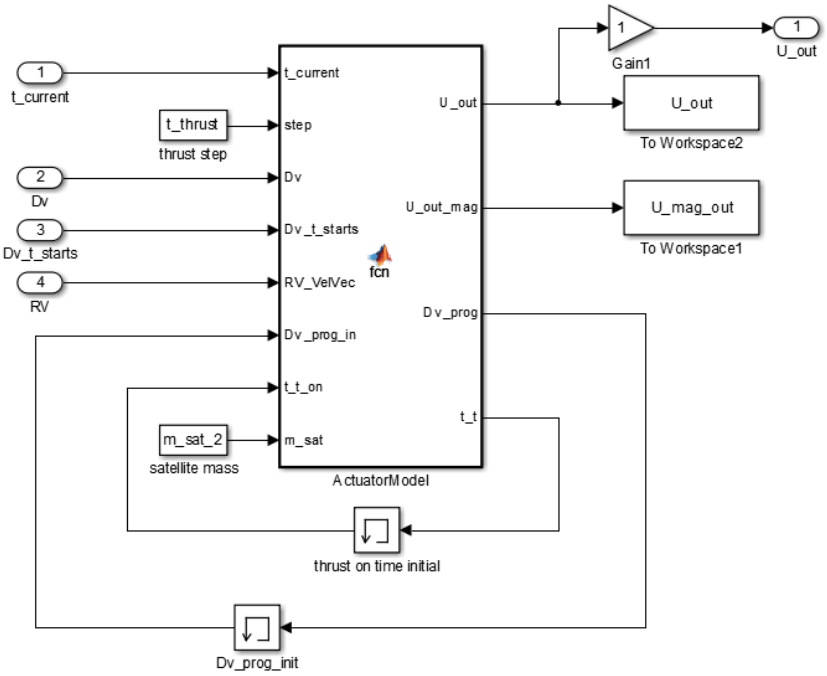



**Figure 3.2: Monitoring Mode Simulink Model: Actuator Model**

To determine if the thruster should be on at a certain time, the start times provided by the Planning Mode are used as start triggers. The system then takes the thrust generated in that time step, converts it to a Delta-v (by multiplying the thrust by the time step then dividing by the satellite's mass) and adds it to the ongoing total of thrust for that manoeuvre. The system stops when the difference between the desired thrust and the provided thrust is less than some tolerance. The tolerance is set at the amount of Delta-v provided in 1 second (the time step) at the beginning of life.

Determining the thrust magnitude without either a thruster model or an ADCS model is more difficult. Eventually, the thruster should provide feedback of the tank pressure at each time step. This can then be correlated to the thrust generated by calibrating the system. Since there is no pressure feedback, the thrust is related directly to the time the propulsion system has been firing using the theoretical model developed in Rob Poyck's thesis. This resulted in a linear relationship between the thrust and time. At the beginning of life, thrust is 1.4 mN and at the end of life the thrust is 0.8 mN. The estimated lifetime is 17 hours and 56 minutes. This generates the linear relationship used to calculate the expected thrust magnitude.

This method calculates the thrust the propulsion system produces, but says nothing about the direction in which it is applied. To determine that, the ADCS must supply an attitude pointing error with reference to the velocity vector. Modelling this is difficult at the current development level, so preliminary analyses have been performed with no attitude pointing error – ie. the thrust is directed along the velocity vector (the ideal case). A place to use the attitude pointing error has been reserved but is currently set to all zeros.

Multiplying the thrust magnitude by a unit vector in the thrust direction (velocity vector) and the time step (1 second) generates the control acceleration that is passed to the orbit propagator for the deputy.

This model is preliminary only, but creates the spaces necessary for the future calibration models to fit into.

# 4 Conclusions and Recommendations

This document highlighted several aspects of the MATLAB Simulink software developed for the formation flying package for DelFFi. In particular, the structures of the Planning Mode and Actuator Mode have been defined with a look to what is included within. Further, the actuator model has been discussed and the reasoning behinds its current design is provided. The Software (functions and scripts) have been included on the DelFFi drive in the working directory of the author (Amy Deeb). These should be referred to for further information in the form of comments.

This software is by no means a complete project. It is intended to provide the necessary structure and variables to be able to expand the system to better reflect the formation flying package design for DelFFi. Some specific recommendations include, but are not limited to:

- Consider using a different orbit propagator that directly uses the orbital elements
- Include other perturbations besides gravity and drag in the propagation
- Add noise to the orbit propagator
- Consider using multiple sources of navigation data (ex. GPS and TLEs) and combine them using a filter (ex. Kalman Filters)
- Interface this software to the Onboard Navigation Function (especially since this will eventually be done for the onboard/embedded software).
- Develop/Design other controllers and compare their performance against the same set of inputs.
- Integrate a pressure sensor model into the actuator model
- Model the attitude pointing error in a representative statistical model based on the results from the ADCS design
- Determine how to "remember" how long the thruster has been on/how much impulse the thruster has provided so far in a manoeuvre on the embedded system
- Add a model of the satellite's mass over time as propellant is consumed
- Combine the Planning and Monitoring Modes into one larger formation flying package model that can account of various sources of noise/errors as well as different rates of sensor inputs.
- Create structure to handle the transition between deputy and chief that will happen at least once during the mission (to link these independent models into a larger formation flying model)

Future development of these Simulink Models will help DelFFi to better understand the autonomous formation flying payload and test the interfaces between the formation flying software and the supporting subsystems.

# A Contents of MATLAB Folders

This appendix lists the contents of each of the MATLAB Folders that have been added to the DelFFi drive. Only contents related to the functioning of the MATLAB Simulink models are listed. Results and Obsolete functions that have been included on the drive for completeness have not been listed here.

## A.1 Planning Mode

**Folder name:** FF_v4_PlanningMode

Contents:

slprj <DIR>
AccelDrag.m
AccelHarmonic_vec1b0.m
AccelHarmonic_vec20b20.m
AccelHarmonic_vec2b0.m
Accel_aux_var.mat
CalcEA.m
Density_HP.m
EqnEquinox.m
FFP_microController_sfun.mexw64
FFP_v1.slx
FFP_v1.slx.original
FFP_v1_acc.mexw64
FFP_v1_sfun.mexw64
Frac.m
GAST.m
GHAMatrix.m
GMST.m
kep2xyz.m
mean2osculating.m
MeanObliquity.m
nuFromM.m
NutAngles_vec.m
NutMatrix.m
PrecMatrix.m
Run_FFP.m
sumx.m
Sun_pos.m
uC_Inputs_Outputs.m
UU.m
VW_value.mat
xMJD.m
xyz2kep.m
xyz2llh.m

## A.2 Code Generation

**Folder name:** FF_v4_CodeGeneration

Contents:

slprj <DIR>
FFP_microController_ert_rtw <DIR>
AccelDrag.m
AccelHarmonic_vec1b0.m
AccelHarmonic_vec20b20.m
AccelHarmonic_vec2b0.m
Accel_aux_var.mat
AngDiff.m
angDiff_fn.m
Density_HP.m
EqnEquinox.m
FFP_microController.slx
FFP_microController_sfun.mexw64
Frac.m
GAST.m
GHAMatrix.m
GMST.m
kep2xyz.m
MeanObliquity.m
NutAngles_vec.m
NutMatrix.m
PrecMatrix.m
sumx.m
Sun_pos.m
UU.m
VW_value.mat
xMJD.m
xyz2llh.m

## A.3 Monitoring Mode

**Folder name:** FF_v4_MonitoringMode

Contents:

slprj <DIR>
AccelDrag.m
AccelHarmonic_vec1b0.m
AccelHarmonic_vec20b20.m
AccelHarmonic_vec2b0.m
Accel_aux_var.mat
AngDiff.m
angDiff_fn.m
Density_HP.m
EqnEquinox.m
FFP_v1.slx
FFP_v1.slx.original
FFP_v1_acc.mexw64
FFP_v1_sfun.mexw64
Frac.m
GAST.m
GHAMatrix.m
GMST.m
kep2xyz.m
MeanObliquity.m
NutAngles_vec.m
NutMatrix.m
PrecMatrix.m
Run_FFP.m
sumx.m
Sun_pos.m
UU.m
VW_value.mat
xMJD.m
xyz2kep.m
xyz2llh.m

# Appendix E

# Software Generation for Target Processor Execution

The following Technical Note (DFF-TUD-TN-1165) for the DelFFi project describes the process and results of converting the Simulink models to 'C' code for execution on the target microcontroller using Simulink Coder and Texas Instrument's Code Composer Studio.

# Formation Flying Payload Microcontroller Software

**Description:** Description of the method of generating 'C' code to be executed on the microcontroller from the formation flying payload Simulink models.

**Subsystem(s) involved:**

| ADCS | CDHS | COMMS | EPS | MechS | STS | TCS | FIPEX | µPS+ | ISL | AFF | GSE | GSN | Launch |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | x | | | |

## Revision Record and Authorization

| Issue | Date | Author / Editor | Reviewer checked | PM approved | Affected Section(s) | Description of change |
|---|---|---|---|---|---|---|
| 0.1 | 15-09-2015 | A. Deeb | | | All | First Draft |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

## Action Items

| TBW | TBD | TBC | Applicable Section(s) | Description of action item |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

## List of Used References

| SLR code | Version | Data/Variable |
|---|---|---|
| - | | Mueller, J. "A Multiple-Team Organization for Decentralized Guidance and Control of Formation Flying Spacecraft". Princeton Satellite Systems, 2004. |
| 1159 | | Jaan Viru's Doc |
| 1164 | | Simulink Code Generation Doc from Fraticelli |
| | | |
| | | |

# Table of Contents

# 1 Introduction

The formation flying package consists of several elements, one of which is a set of algorithms to perform formation maintenance. Different algorithms are used in different phases of the mission. Three phases have been identified for formation maintenance: Estimation, Planning and Monitoring.

Estimation mode is a submode to the DelFFi system's Nominal Operations Phase, and consists of (1) receiving navigation data (Two-line Elements) and extracting the mean orbital elements of the two satellites (2) estimating the position, velocity and mean orbital elements of the satellites at set time increments for two days (3) determining the differential position and differential orbital elements of the deputy satellite with respect to the chief and (4) deciding if a correction manoeuvre should be performed within the two days following the TLE reception. If a correction is required, the satellite determines when the first manoeuvre should begin. This is set to be at minimum 3 hours (2 orbits) after the TLEs were received, at the time the deputy satellite will pass the ascending node (cross the equator). When the start time for the corrections has been selected, the start time as well as the mean orbital elements for the two satellites at that time are passed to the Planning mode.

In Planning mode, the satellite calculates the magnitude of the necessary correction manoeuvre(s). If there are multiple manoeuvres required (as there are using the analytical controller selected for the DelFFi formation maintenance algorithm), the times of the subsequent manoeuvres are also determined. For DelFFi, a three-impulse analytical controller for circular orbits designed by Mueller in (Mueller2004) was selected. This controller allows the optimization of propellant by adjusting when the second and third manoeuvres occur by two parameters $M$ (the number of half orbits between the first and second manoeuvres) and $N$ (the number of half orbits between the first and third manoeuvres). For the DelFFi mission, the maximum value for $M$ has been set to 27 and the maximum value for $N$ is set to 30, where $N$ must always be at least 3 larger than $M$. When the magnitudes and start times of the manoeuvres have been planned, the DelFFi system switches to Thrust mode and the Monitoring mode is activated

The Monitoring mode uses feedback from the attitude determination and control subsystem (ADCS) and propulsion system to confirm that he manoeuvre(s) are being performed correctly by maintaining an estimate of the relative state of the formation. This mode will be performed on the microcontroller of the propulsion system.

At this time, confirmation of the Planning mode's performance on the microcontroller is the most important. This is because the Estimation mode uses several elements of the onboard navigation function (ONF) from the Delfi-n3Xt project, and the Monitoring mode is executed on a different microcontroller. The Planning mode will be executed on the ADCS microcontroller which is selected and characterized by Viru in SLR ###1. For testing, the Tiva C Series TM4C1294XL Connected LaunchPad Evaluation Kit is used along with Texas Instrument's Code Composer Studio (CCS).

This technical note describes the necessary changes to the Simulink Planning Mode Model to allow auto code generation and the transition and execution of the generated code onto the microcontroller.

# 2 Simulink Model Settings

In addition to following general good programming practices (some of which are highlighted in SLR 1164), there are several settings within Simulink in order to use Simulink's AutoCoder feature. The structure of the model and the settings that had to be changed are highlighted in this section. For this project, MATLAB R2012b (Student License) has been used.

## 2.1 Simulink Model

The development of this model began with the Simulink Model used for simulation testing of the on-board Planning Mode Software. This original planning model is described in Section 2.1.1. Following this, the variations to the model in order for it to be compatible with autocoding are discussed in Sections 2.1.2 and 2.1.3.

### 2.1.1 Original Planning Mode Model

The Simulink Model for the Planning Mode is shown below in Figure 2.1. It consists of orbit propagators for the two satellites, a relative navigation function and the analytical controller.

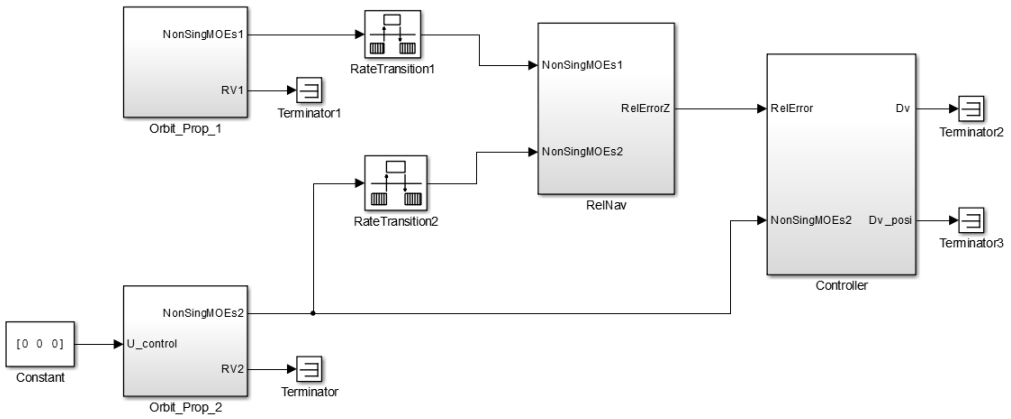



**Figure 2.1: Original Planning Mode Simulink Model**

For the on-board software the orbit propagation will be provided by the onboard navigation function application when requested by the formation flying algorithm, thus these blocks do not need to be included in the tests. Instead only the Relative Navigation Function (labelled RelNav in Figure 2.1) and the Controller need to be modified for conversion to 'C'.

The Controller block in Figure 2.1 consists of only a User-defined MATLAB function (see Appendix A.1) with the control code inside and some additional constant inputs as shown in Figure 2.2. For the on-board version, a slimmed-down version with fewer monitored parameters will be used (remove parameters meant for development/debug purposes).

The RelNav block (Figure 2.3) performs the differential navigation calculations on the mean orbital elements from the two satellites. There are three direct subtraction blocks and four angular difference blocks (subtraction while being careful of the 2π-normalized angles). Further, the differential true latitude (θ) is further compared to the desired separation angle to find the error on the separation angle. The desired separation angle is calculated from the desired separation distance and the current semi major axis of the chief satellite. In this block the orbital elements are passed in as a 6x1 vector and the relative error (Z) is returned as a vector.

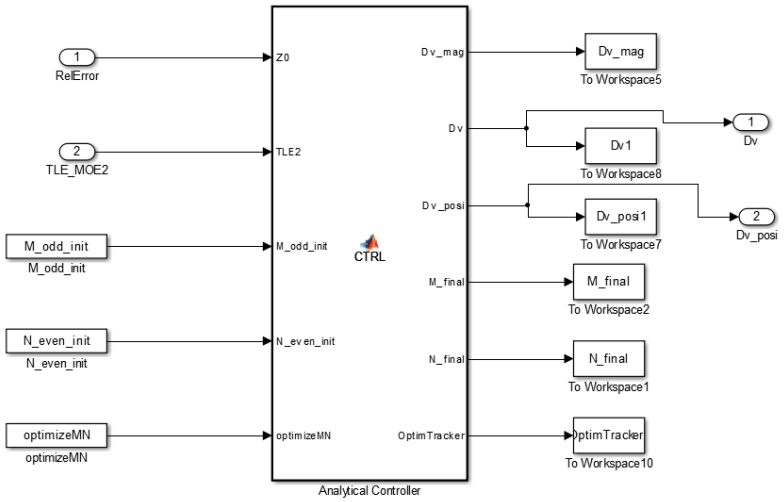



**Figure 2.2: Original Planning Mode Simulink Model: Controller Block**

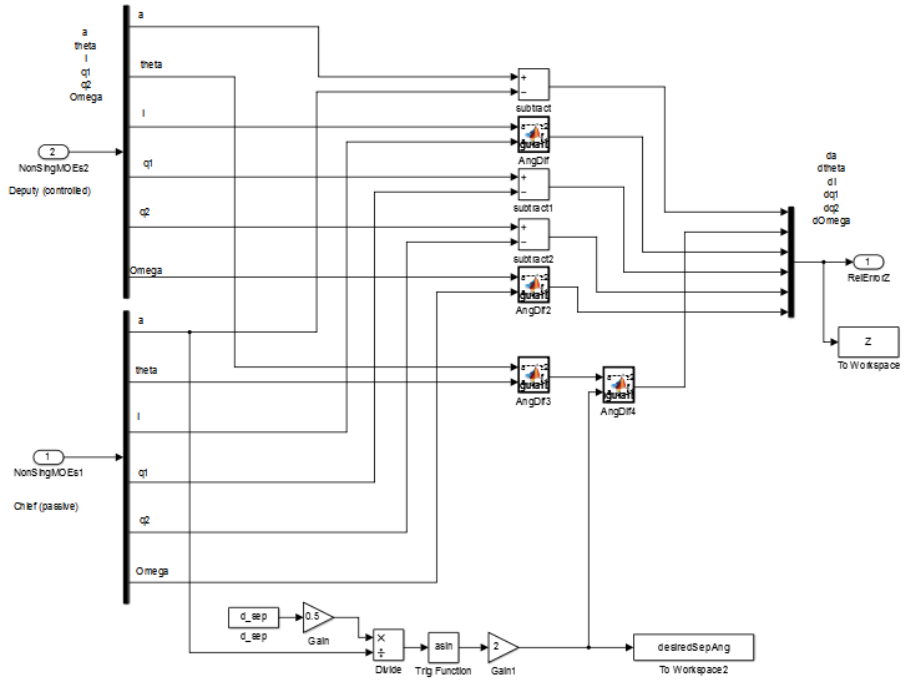



**Figure 2.3: Original Planning Mode Simulink Model: RelNav Block**

These blocks are the basis from which the blocks intended for code generation are created.

### 2.1.2 Variation of Model for Code Generation

In order to execute the code on the microcontroller, all inputs had to be hardcoded into the Simulink structure since the data bases that will eventually be available to the formation flying software have not yet been developed. In addition, vector inputs/outputs needed to be separated into individual items that can be passed through the way that individual variables will be requested when executed onboard. The results of these changes to the Simulink blocks are shown in the following figures.

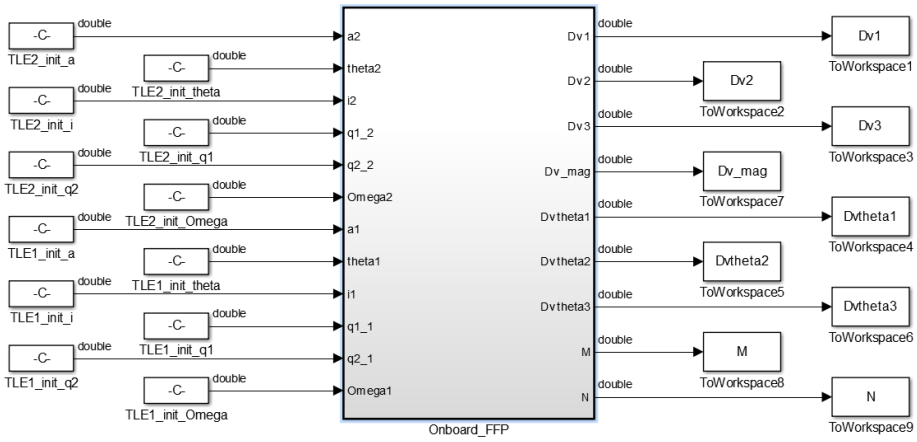



**Figure 2.4: Auto-Code-able Planning Mode Simulink Model: Overview**

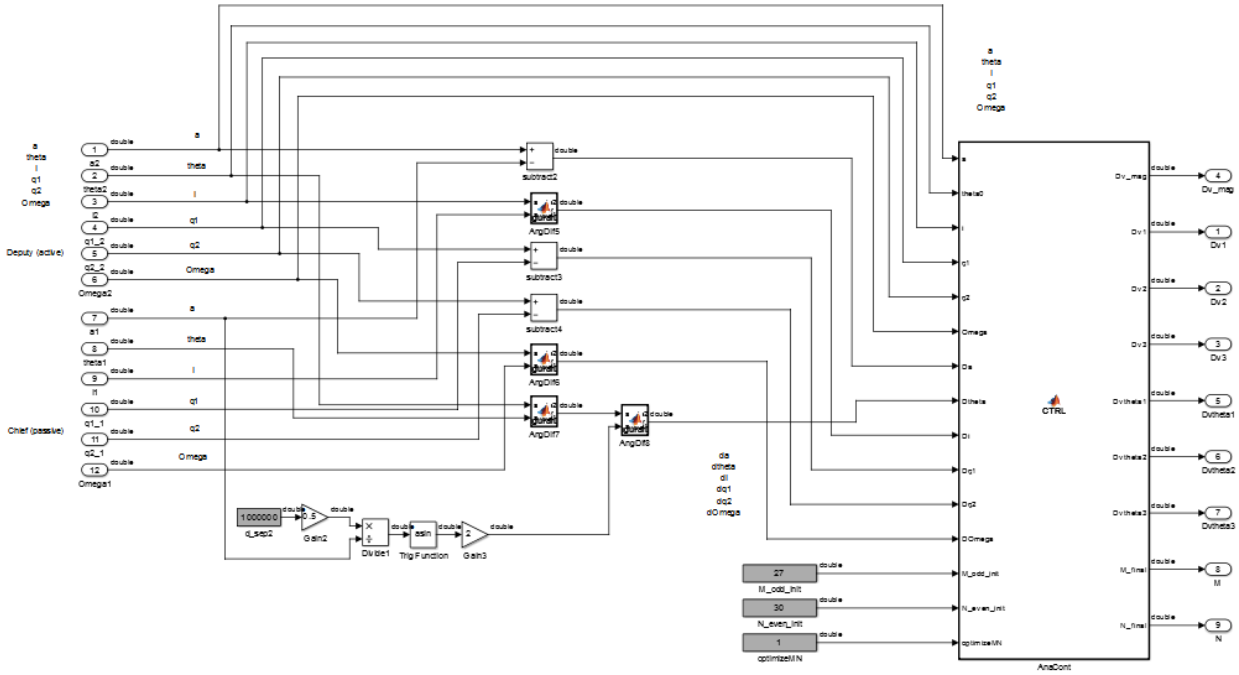



**Figure 2.5: Auto-Code-able Planning Mode Simulink Model: Internal**

The majority of the code has remained the same, just expanded out into a flat structure – rather than being several layers deep. This is not necessarily an efficient mode of coding, however it does make traceability after auto-coding more convenient. The code from the AnaControl and AngDiff blocks are given in Appendix A.2 and A.3.

### 2.1.3 Subsystem and Function Settings

In addition to varying the structure of the Simulink Model, the subsystems and functions had to be slightly adjusted. First, the User Defined MATLAB Functions had to be set as Reusable Functions and the naming options for Code Generation had to be made explicitly. This was done by right-clicking on each block

selecting 'Block Parameters (Subsystem)' then choosing the second tab. This process is shown in SLR 1164. Further, it was necessary to ensure that the main block of code that would be AutoCoded was an Atomic Subsystem by checking the relevant box in the 'Block Parameters (Subsystem)' Menu. Doing this changes the outline of the block from a thin line to a thick (heavy) black line). This is also discussed in SLR 1164. Finally the naming of all elements and variables should be set to follow acceptable naming conventions for the 'C' language – as listed in SLR 1164.

### 2.1.4 Model Configuration Parameters

The final step to set up the Simulink Model is to configure the model parameters correctly. The steps to do this are detailed in SLR 1164 and are not repeated here for brevity. Certain screenshots of the Model Configuration window are shown below to provide clarification to the settings used for the formation flying software.

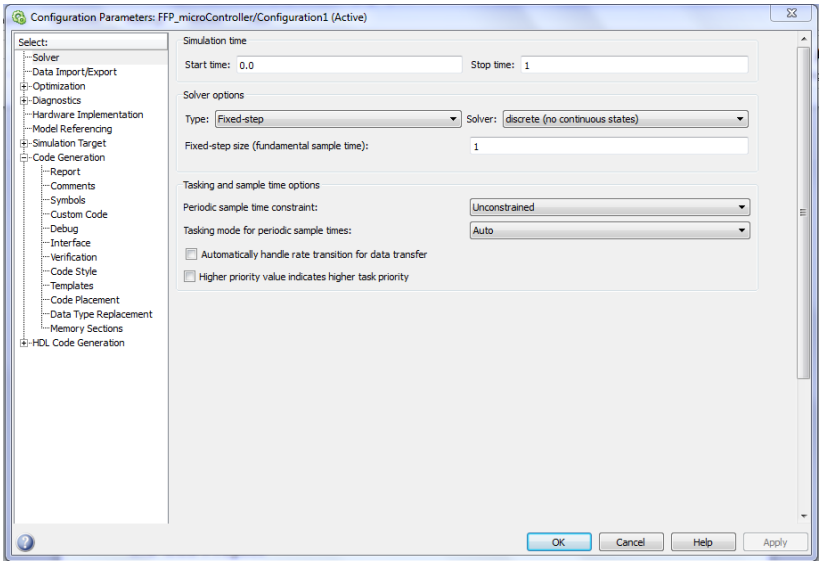



**Figure 2.6: Model Configuration Parameters - Solver**

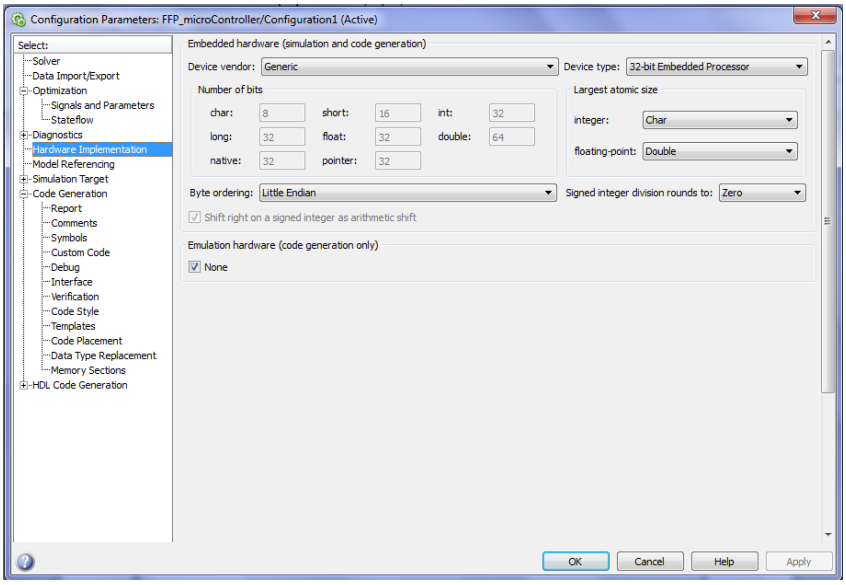



**Figure 2.7: Model Configuration Parameters – Hardware Implementation**

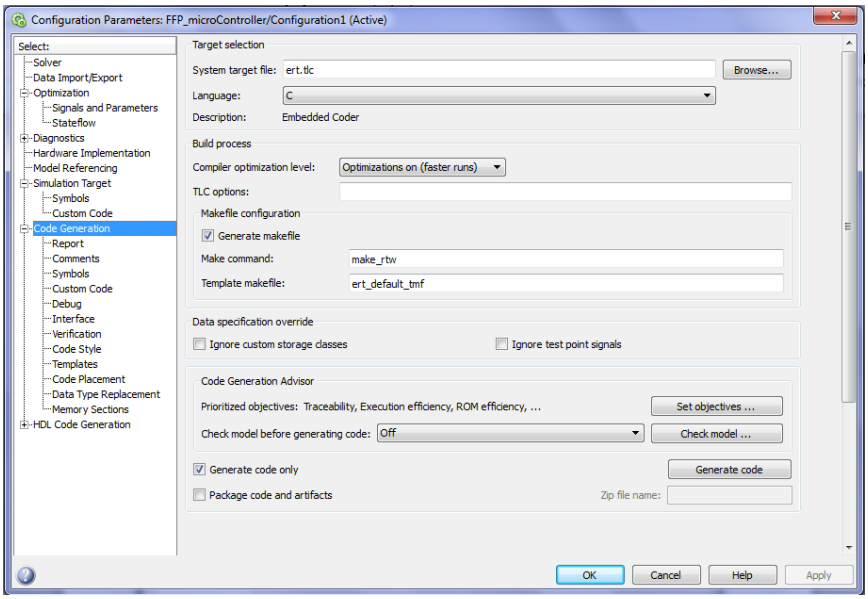



**Figure 2.8: Model Configuration Parameters – Code Generation**

## 2.2 Code Generation

Generation of the 'C' Code is done in the Model Configuration Parameters Window on the Code Generation Tab. The procedure for this is detailed in SLR 1164 but the general method is described here.

### 2.2.1 Code Generation Procedure

The first step is to run the Simulink Model (using the run( ); command or the green play button) and confirm that the inputs (still) generate the expected output and to record the results to later be compared with the microcontroller results.

The next step is to open the Model Configuration Parameters window and navigate to the Code Generation Tab. Near the bottom there is an option labelled "Check model before generating code: " with a drop down menu. Choose "On (stop for warnings)", then click "Apply". Next, click the "Check Model" button. This will generate a window with a series of tests that have either been passed, failed or have generated a warning (see SLR 1164 for details). Tests that have failed or have warnings should be addressed. Failures should be corrected (read the message and follow the instructions – most are pretty straight forward and generally require renaming a parameter or perhaps a "Treat as Atomic" was not selected for one of the Subsystem blocks. Warnings should be dealt with as if they were failures however possible, however some will not be able to be removed. For the formation flying software three warnings remained as shown in the figure below. These were not critical to the functionality of the code and were accepted for this stage of development. Once the rest of the software has been developed, the embedded programmer responsible for the system may be able to address them in detail and remove their causes.

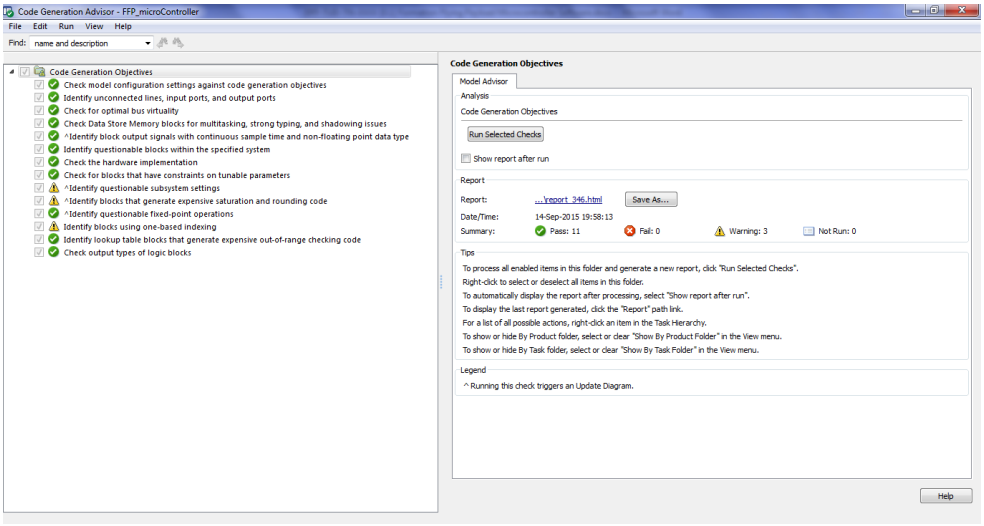



**Figure 2.9: Check Model Results**

The final step is to change the drop down menu for "Check model before generating code: " back to "Off", click "Apply" and then "Generate Code". This process will result in a Code Generation Report as shown in Figure 2.10 below.

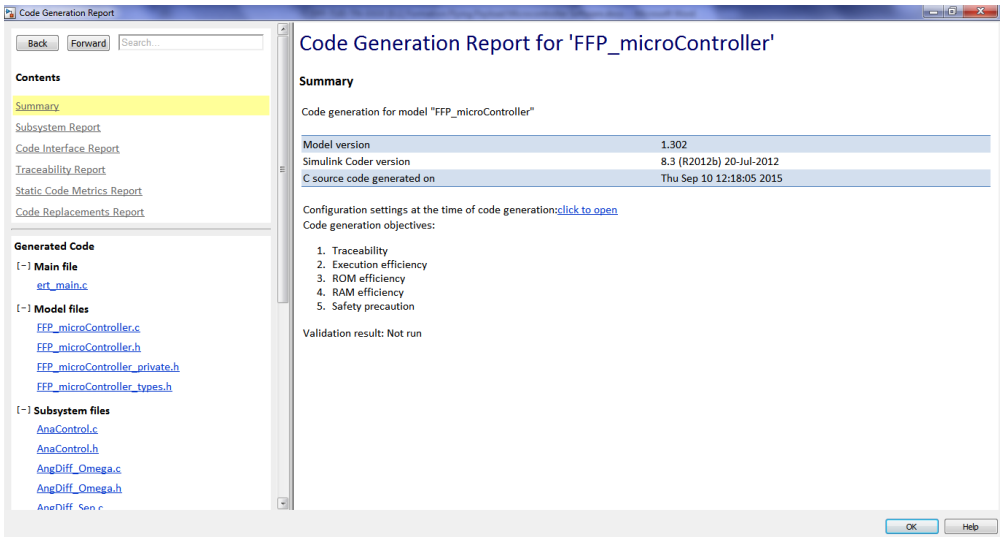



**Figure 2.10: Code Generation Report**

Included in this report is a list of each piece of code that was generated. For the Formation flying software this list is shown in Figure 2.11. Be sure to check that this list includes a .c and a .h file for all blocks that used User Defined MATLAB functions in the Simulink Model under the "Subsystem files" header. Once these files have been generated, be sure to locate the directory in which they have been placed for easy access in the next step.

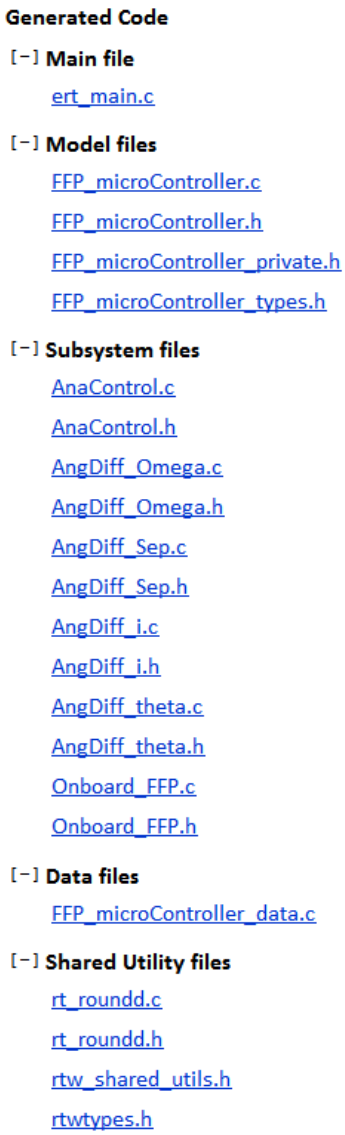



**Figure 2.11: Code Generation Report**

# 3 Code Composer Studio Software

Once the 'C' Code has been generated, development switches to Code Composer Studio (CCS) where it can be integrated into a CCS project for execution on the microcontroller in Debug mode. The most convenient method to do this is to use an existing CCS project that functions on the target microcontroller and modify it using the generated files from Simulink. This is not the most elegant solution, nor does it produce code that can be directly used on the flight hardware, however it does allow confirmation that the software can function on the target microcontroller. This section will detail the process of incorporating the Simuilnk-generated code into a CCS project and how to measure values and processing time using CCS.

## 3.1 CCS Project

This document assumes that CCS has been installed and that the user has a basic familiarity with CCS and the 'C' programming environment. Texas Instruments has many tutorials online to assist with learning how to use CCS and are highly recommended. Once installed, the first step is to ensure that an example project works properly on the target microcontroller. To do this, go to the "Getting Started" screen and selected "Browse Examples".

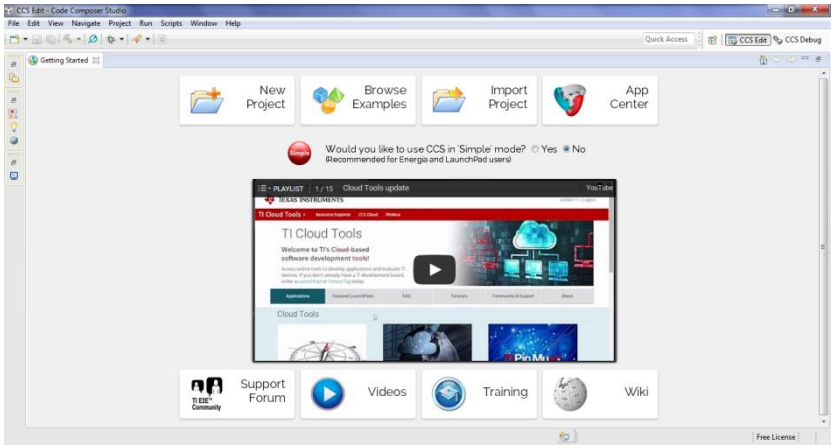



**Figure 3.1: CCS Getting Started**

When the list of examples appears, scroll to the board that is being used, then select the example called "hello". Choose to import the example project and wait while it is imported. Once it is imported, Build the project (the little hammer at the top). After it has finished building (it takes a while the first time), choose Debug (the little bug). If there are no errors, the project is being executed on the microcontroller. Lights will flash on the microcontroller and you should be able to read values on the screen by hovering over variable names with the pointer. Now it is time to begin integrating the Simulink Code.

## 3.2 Integration of Auto-Generated Code

The instructions in SLR 1164 provide the method to integrate the code using the Dev C++ compiler. An analogous process is followed here. All of the .c and .h files generated by Simulink are copied (drag and drop into the project explorer on the left hand side of CCS) into the "hello" project ***except*** the ert_main.c file. These files can be found in two places – the one named after the Simulink Model (FFP_microController_ert_rtw here) and the "\slprj\ert_sharedutils" directory. The final list of files in the project are shown in Figure 3.2.

Now open the ert_main.c file in an external text editor (Recommend: Notepad++). Also, open hello.c (the main file of the hello project) in CCS. To integrate the code, we copy the contents of ert_main.c into hello.c in a strategic way. The includes are copied to the start of the file, the "rt_OneStep" function is copied just after the "ConfigureUART" function, the "FFP_initialize" call is added to the "main" function and the rt_OneStep() call is added to the while(1) loop. These changes are shown in the modified hello.c included in Appendix A.4. A further change is to set the clock frequency to 100 MHz (to match the target processor requirements) in the "main" function.

In addition to these changes, two print statements (in the while loop of the main function of hello.c) and one delay statement (in AnaControl.c as per Appendix A.5), have been added. These act as place holders in which Breakpoints can be added during debugging to confirm the code is functioning properly and to monitor the parameter values and the processing duration.

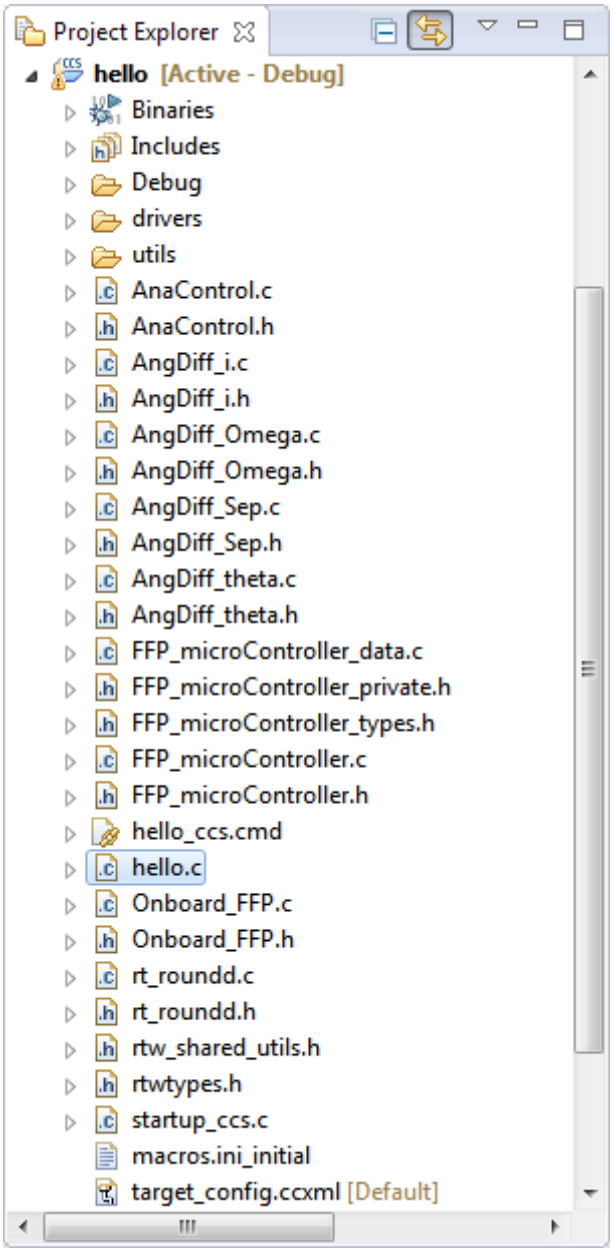



**Figure 3.2: Project Explorer – List of Files**

## 3.3 Debug and Parameter Monitoring

Once these changes have been made, save the project, Build it and then Debug it. This will start it running on the microcontroller.

To determine if the software is performing properly on the microcontroller, the parameters (in this case Dv1, Dv2, Dv3, Dvtheta1, Dvtheta2, Dvtheta3, M and N) can be checked. To do this, place a breakpoint at the end of the AnaControl.c file on the delay line added in the previous section. Then step the process forward until it reaches that breakpoint. It is then possible to add watch expressions to the desired parameters or to simply hover the pointer over the parameter of interest and see its value at the time the code reaches the breakpoint.

To determine the run time of the code – in this case how long rt_OneStep takes to execute – breakpoints are added before and after the rt_OneStep call in the while loop of the main function of the hello.c file. Note: be sure to remove the delay from AnaControl.c and remove any other breakpoints before measuring the run time. To determine the run time, first the number of clock cycles are counted, then, using the clock frequency, the duration is calculated. The clock cycles between two breakpoints are counted using a Count Event. After inserting the two breakpoints, generate a new Count Event by choosing the drop down arrow next to the new Breakpoint icon and choose Count Event – as in Figure 3.4

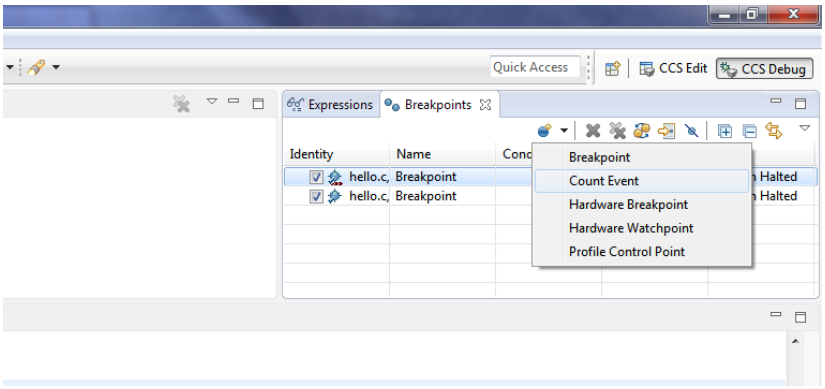



**Figure 3.3: Breakpoint Count Event**

Since the desired output of the counter is the number of counts between the two breakpoints, right click on the Count Event and Choose Breakpoint Properties. Under Debugger Response, select **true** for the Reset Count on Run. This means when the debugger steps from the breakpoint before the rt_OneStep to the breakpoint after rt_OneStep, the counter will reset. This is shown in Figure 3.4.

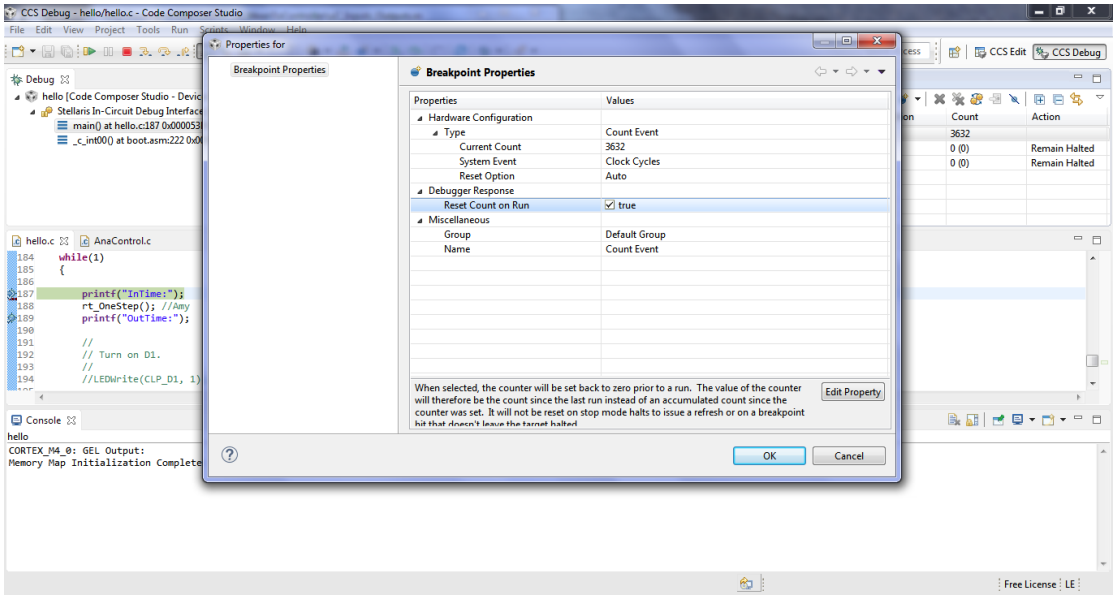



**Figure 3.4: Count Event Properties**

This should complete the process necessary to test the formation flying code on the target microcontroller. The results of this process for the formation flying software package are summarized in Appendix B.

# 4 Conclusions and Recommendations

This technical note described the procedure to convert a Simulink Model into 'C' code using SimulinkCoder for execution on the target microcontroller. The procedure is general in that it can be applied to any Simuilnk Model, however details have been provided specifically for the formation flying payload's planning mode Simulink model. The method followed here is sufficient for preliminary testing but is not suitable for developing the final onboard software as it is inefficient and is not arranged to connect to the other modules of the software architecture on the Attitude Determination and Control Architecture. For this to be completed a specialist in embedded programming should be enlisted who can efficiently and effectively bridge the gap between algorithm (in Simulink and roughly in 'C') to embedded software.

# A Select Code Excerpts

This appendix provides segments of code that are interesting or important for understanding the Code Generation process and results.

## A.1 Analytical Controller Simulink Block (Original Model)

```
function [Dv_mag, Dv, Dv_posi, M_final, N_final, OptimTracker] = CTRL(Z0, TLE2, M_odd_init,
N_even_init, optimizeMN)
%#codegen

%% Constants
GM_Earth = 398600.4415e+9;      % [m^3/s^2]; JGM3
R_JGM3 = 6378.11363e3;


%% Check input is valid
% these should be the mean orbital elements so that J2 is considered
a = TLE2(1);
theta0 = TLE2(2); % current argument of latitude
i = TLE2(3);
q1 = TLE2(4); %ecos(w)
q2 = TLE2(5); %esin(w)
Omega = TLE2(6);

% Differential OE's
Da = Z0(1);
Dtheta = Z0(2); %difference in true latitude from the desired sep ang
Di = Z0(3);
Dq1 = Z0(4);
Dq2 = Z0(5);
DOmega = Z0(6);

n = sqrt(GM_Earth/(a^3)); % mean orbital rate
Da_bar = Da/a;
theta1 = atan2(Dq2, Dq1); % should be near 0 (circular truly = 0)
Dq = Dq1*cos(theta1) + Dq2*sin(theta1);
Dq_0 = Dq1*sin(theta0) - Dq2*cos(theta0);

% Adjust when the burns will occur - Now from the initialization file
%M_odd = 1; %# half orbits between 1st and 2nd burn
%N_even = 2; %# half orbits between 1st and 3rd burn

% Check M and N are proper
if M_odd_init < 3
    disp('M too small')
    M_odd_init = 3;
    disp(M_odd_init)
end

if N_even_init < 6
    disp('N too small')
    N_even_init = 6;
    disp(N_even_init)
end

M_final = 0;
N_final = 0;
Dv_mag_final = 0;
Dv1_final = 0;
Dv2_final = 0;
Dv3_final = 0;
OptimTracker = zeros(100, 3);


if optimizeMN == 1
    %%
    % Calculate Delta-V Sequence
    Dv1 = 100; % initialize Delta V's for comparison to a very high number
    Dv2 = 100;
```

```
    Dv3 = 100;
    Dv_mag_final = abs(Dv1) + abs(Dv2) + abs(Dv3);
    Dv_mag = 0;
    k = 1;

    for M_odd = [3:2:M_odd_init]
        for N_even = [M_odd+3:2:N_even_init]
            Dv1  =  (n*a/(3*N_even*pi))*(Dtheta  -  (3/2)*Da_bar*(theta1  -  theta0)  -  2*Dq_0)  +
(n*a/4)*(((M_odd/N_even)+1)*Dq - ((M_odd/N_even) - 1)*Da_bar);
            Dv2 = (n*a/4)*(Dq - Da_bar);
            Dv3  =  -(n*a/(3*N_even*pi))*(Dtheta  -  (3/2)*Da_bar*(theta1  -  theta0)  -  2*Dq_0)  -
(n*a/4)*((M_odd/N_even)*Dq - (M_odd/N_even)*Da_bar);
            Dv_mag_iter = abs(Dv1) + abs(Dv2) + abs(Dv3);
            OptimTracker(k, 1) = M_odd;
            OptimTracker(k, 2) = N_even;
            OptimTracker(k, 3) = Dv_mag_iter;
            k = k+1;
            if Dv_mag_iter <= Dv_mag_final
                M_final = M_odd;
                N_final = N_even;
                Dv1_final = Dv1;
                Dv2_final = Dv2;
                Dv3_final = Dv3;
                Dv_mag_final = abs(Dv1) + abs(Dv2) + abs(Dv3);
            end
        end
    end
    OptimTracker(k, 1) = 0;
    OptimTracker(k,2)= 0;
    OptimTracker (k,3) = 0;
    k = k+1;
    OptimTracker(k,1) = M_final;
    OptimTracker(k,2) = N_final;
    OptimTracker(k,3) = Dv_mag_final;
else
    N_even = N_even_init;
    M_odd = M_odd_init;
    Dv1_final  =  (n*a/(3*N_even*pi))*(Dtheta  -  (3/2)*Da_bar*(theta1  -  theta0)  -  2*Dq_0)  +
(n*a/4)*(((M_odd/N_even)+1)*Dq - ((M_odd/N_even) - 1)*Da_bar);
    Dv2_final = (n*a/4)*(Dq - Da_bar);
    Dv3_final  =  -(n*a/(3*N_even*pi))*(Dtheta  -  (3/2)*Da_bar*(theta1  -  theta0)  -  2*Dq_0)  -
(n*a/4)*((M_odd/N_even)*Dq - (M_odd/N_even)*Da_bar);
    M_final = M_odd;
    N_final = N_even;
end

% Calculate locations (true anomaly) of 2nd and 3rd thrust
theta2 = theta1 + pi*M_final;
theta3 = theta1 + pi*N_final;

%%
% Confirm output is valid
Dv = [Dv1_final;Dv2_final;Dv3_final];
Dv_mag = abs(Dv1_final)+abs(Dv2_final)+abs(Dv3_final);
Dv_posi = [theta1; theta2; theta3];
```

## A.2 AngDiff Simulink Block (Original and Modified Models)

```
function dif = AngularDiff(angle2, angle1)
%#codegen
% subtract angle 1 from angle2

a2 = mod(angle2, 2*pi);
a1 = mod(angle1, 2*pi);
dif = mod((angle2-angle1)+pi, 2*pi) - pi;
```

## A.3 AnaControl Simulink Block (Modified Model)

```
function [Dv_mag, Dv1, Dv2, Dv3, Dvtheta1, Dvtheta2, Dvtheta3, M_final, N_final] = CTRL(a, theta0, i,
q1, q2, Omega, Da, Dtheta, Di, Dq1, Dq2, DOmega, M_odd_init, N_even_init, optimizeMN)
%#codegen

%% Constants
GM_Earth = 398600.4415e+9;      % [m^3/s^2]; JGM3
R_JGM3 = 6378.11363e3;


%% Check input is valid

n = sqrt(GM_Earth/(a^3)); % mean orbital rate
Da_bar = Da/a;
theta1 = atan(Dq2/Dq1); % will be 0
Dq = Dq1*cos(theta1) + Dq2*sin(theta1);
Dq_0 = Dq1*sin(theta0) - Dq2*cos(theta0);

% Adjust when the burns will occur - Now from the initialization file
%M_odd = 1; %# half orbits between 1st and 2nd burn
%N_even = 2; %# half orbits between 1st and 3rd burn

% Check M and N are proper
if M_odd_init < 3
    disp('M too small')
    M_odd_init = 3;
    disp(M_odd_init)
end

if N_even_init < 6
    disp('N too small')
    N_even_init = 6;
    disp(N_even_init)
end

M_final = 0;
N_final = 0;
Dv_mag_final = 0;
Dv1_final = 0;
Dv2_final = 0;
Dv3_final = 0;
%OptimTracker = zeros(100, 3);


if optimizeMN == 1
    %%
    % Calculate Delta-V Sequence
    Dv1 = 100; % initialize Delta V's for comparison to a very high number
    Dv2 = 100;
    Dv3 = 100;
    Dv_mag_final = abs(Dv1) + abs(Dv2) + abs(Dv3);
    Dv_mag = 0;
    k = 1;

    for M_odd = [3:2:M_odd_init]
        for N_even = [M_odd+3:2:N_even_init]
            Dv1 = (n*a/(3*N_even*pi))*(Dtheta - (3/2)*Da_bar*(theta1 - theta0) - 2*Dq_0) +
(n*a/4)*(((M_odd/N_even)+1)*Dq - ((M_odd/N_even) - 1)*Da_bar);
            Dv2 = (n*a/4)*(Dq - Da_bar);
            Dv3 = -(n*a/(3*N_even*pi))*(Dtheta - (3/2)*Da_bar*(theta1 - theta0) - 2*Dq_0) -
(n*a/4)*((M_odd/N_even)*Dq - (M_odd/N_even)*Da_bar);
            Dv_mag_iter = abs(Dv1) + abs(Dv2) + abs(Dv3);
            %OptimTracker(k, 1) = M_odd;
            %OptimTracker(k, 2) = N_even;
            %OptimTracker(k, 3) = Dv_mag_iter;
            k = k+1;
            if Dv_mag_iter <= Dv_mag_final
                M_final = M_odd;
                N_final = N_even;
                Dv1_final = Dv1;
```

```
                Dv2_final = Dv2;
                Dv3_final = Dv3;
                Dv_mag_final = abs(Dv1) + abs(Dv2) + abs(Dv3);
            end
        end
    end
    %OptimTracker(k, 1) = 0;
    %OptimTracker(k,2)= 0;
    %OptimTracker (k,3) = 0;
    k = k+1;
    %OptimTracker(k,1) = M_final;
    %OptimTracker(k,2) = N_final;
    %OptimTracker(k,3) = Dv_mag_final;
else
    N_even = N_even_init;
    M_odd = M_odd_init;
    Dv1_final = (n*a/(3*N_even*pi))*(Dtheta - (3/2)*Da_bar*(theta1 - theta0) - 2*Dq_0) +
(n*a/4)*(((M_odd/N_even)+1)*Dq - ((M_odd/N_even) - 1)*Da_bar);
    Dv2_final = (n*a/4)*(Dq - Da_bar);
    Dv3_final = -(n*a/(3*N_even*pi))*(Dtheta - (3/2)*Da_bar*(theta1 - theta0) - 2*Dq_0) -
(n*a/4)*((M_odd/N_even)*Dq - (M_odd/N_even)*Da_bar);
    M_final = M_odd;
    N_final = N_even;
end

% Calculate locations (true anomaly) of 2nd and 3rd thrust
theta2 = theta1 + pi*M_final;
theta3 = theta1 + pi*N_final;

%%
% Confirm output is valid
%Dv = [Dv1_final;Dv2_final;Dv3_final];
Dv1 = Dv1_final;
Dv2 = Dv2_final;
Dv3 = Dv3_final;
Dv_mag = abs(Dv1_final)+abs(Dv2_final)+abs(Dv3_final);
%Dv_posi = [theta1; theta2; theta3];
Dvtheta1 = theta1;
Dvtheta2 = theta2;
Dvtheta3 = theta3;
```

## A.4 Modified hello.c

```
//*****************************************************************************
//
// hello.c - Simple hello world example.
//
// Copyright (c) 2013-2015 Texas Instruments Incorporated.  All rights reserved.
// Software License Agreement
//
// Texas Instruments (TI) is supplying this software for use solely and
// exclusively on TI's microcontroller products. The software is owned by
// TI and/or its suppliers, and is protected under applicable copyright
// laws. You may not combine this software with "viral" open-source
// software in order to form a larger program.
//
// THIS SOFTWARE IS PROVIDED "AS IS" AND WITH ALL FAULTS.
// NO WARRANTIES, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT
// NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE APPLY TO THIS SOFTWARE. TI SHALL NOT, UNDER ANY
// CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL, OR CONSEQUENTIAL
// DAMAGES, FOR ANY REASON WHATSOEVER.
//
// This is part of revision 2.1.1.71 of the EK-TM4C1294XL Firmware Package.
//
//*****************************************************************************
```

```
#include <stdint.h>
#include <stdbool.h>
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "driverlib/gpio.h"
#include "drivers/pinout.h"
#include "driverlib/pin_map.h"
#include "driverlib/rom.h"
#include "driverlib/rom_map.h"
#include "driverlib/sysctl.h"
#include "driverlib/uart.h"
#include "utils/uartstdio.h"

// Added for FFP uC Testing
#include <stdio.h>                     /* This ert_main.c example uses printf/fflush */
#include "FFP_microController.h"       /* Model's header file */
#include "rtwtypes.h"                  /* MathWorks types */

static D_Work_FFP_microController FFP_microController_DWork;/* Observable states */
// end of addition


//*****************************************************************************
//
//! \addtogroup example_list
//! <h1>Hello World (hello)</h1>
//!
//! A very simple ``hello world'' example.  It simply displays ``Hello World!''
//! on the UART and is a starting point for more complicated applications.
//!
//! Open a terminal with 115,200 8-N-1 to see the output for this demo.
//
//*****************************************************************************

//*****************************************************************************
//
// System clock rate in Hz.
//
//*****************************************************************************
uint32_t g_ui32SysClock;

//*****************************************************************************
//
// The error routine that is called if the driver library encounters an error.
//
//*****************************************************************************
#ifdef DEBUG
void
__error__(char *pcFilename, uint32_t ui32Line)
{
}
#endif

//*****************************************************************************
//
// Configure the UART and its pins.  This must be called before UARTprintf().
//
//*****************************************************************************
void
ConfigureUART(void)
{
    //
    // Enable the GPIO Peripheral used by the UART.
    //
    ROM_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);

    //
    // Enable UART0
    //
    ROM_SysCtlPeripheralEnable(SYSCTL_PERIPH_UART0);
```

```
    //
    // Configure GPIO Pins for UART mode.
    //
    ROM_GPIOPinConfigure(GPIO_PA0_U0RX);
    ROM_GPIOPinConfigure(GPIO_PA1_U0TX);
    ROM_GPIOPinTypeUART(GPIO_PORTA_BASE, GPIO_PIN_0 | GPIO_PIN_1);

    //
    // Initialize the UART for console I/O.
    //
    UARTStdioConfig(0, 115200, g_ui32SysClock);
}

// Added for FFP uC Testing
void rt_OneStep(void);
void rt_OneStep(void)
{
  static boolean_T OverrunFlag = 0;

  /* Disable interrupts here */

  /* Check for overrun */
  if (OverrunFlag) {
    rtmSetErrorStatus(0, "Overrun");
    return;
  }

  OverrunFlag = TRUE;

  /* Save FPU context here (if necessary) */
  /* Re-enable timer or interrupt here */
  /* Set model inputs here */

  /* Step the model */
  FFP_microController_step(&FFP_microController_DWork);

  /* Get model outputs here */

  /* Indicate task complete */
  OverrunFlag = FALSE;

  /* Disable interrupts here */
  /* Restore FPU context here (if necessary) */
  /* Enable interrupts here */
}
// end of addition

//*****************************************************************************
//
// Print "Hello World!" to the UART on the Intelligent UART Module.
//
//*****************************************************************************
int
main(void)
{
    //
    // Run from the PLL at 120 MHz.
    //
    g_ui32SysClock = MAP_SysCtlClockFreqSet((SYSCTL_XTAL_25MHZ |
                SYSCTL_OSC_MAIN | SYSCTL_USE_PLL |
                SYSCTL_CFG_VCO_480), 100000000); //Changed to 100 MHz

    //
    // Configure the device pins.
    //
    PinoutSet(false, false);

    //
    // Enable the GPIO pins for the LED D1 (PN1).
    //
    ROM_GPIOPinTypeGPIOOutput(GPIO_PORTN_BASE, GPIO_PIN_1);
```

```
    //
    // Initialize the UART.
    //
    ConfigureUART();

    // Added for FFP uC Testing
    FFP_microController_initialize();

    //
    // Hello!
    //
    UARTprintf("Hello, world!\n");

    //
    // We are finished.  Hang around flashing D1.
    //
    while(1)
    {

        printf("InTime:");
        rt_OneStep(); //Amy
        printf("OutTime:");
    }
}
```

## A.5 Modified AnaControl.c

Only one line was added to AnaControl.c. The very last line before the final semicolon: a short wait instruction such that a breakpoint could be added during debugging. This is shown in the figure below.

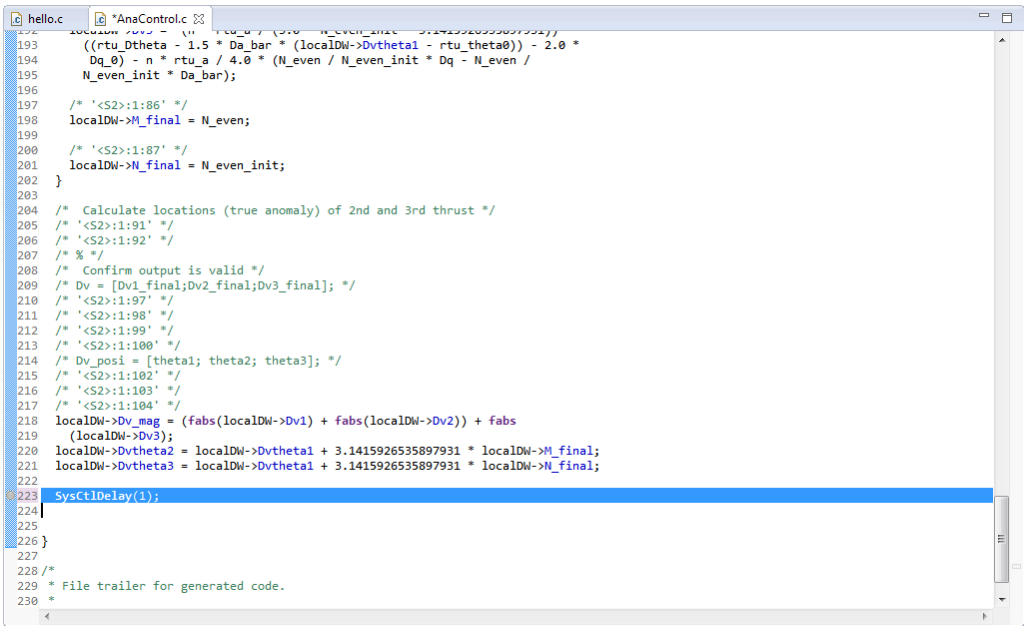



**Figure A.1: Modified Line of AnaControl.c**

# B Microcontroller Performance Summary

The following script summarizes the inputs, expected results and actual results of the Simulink Code when executed on the microcontroller.

```
% Preparation for Running Code on Microcontroller
% Data from: FFP_J20_D_3.732419e+02x3.867581e+02_1000_A_OptimMN_2015-9-7_23-35.mat
% at Time Step: 8857 [min]
%%
%Inputs

%TLE1_nonsing_code
[6.757411221422149e+06,
    -0.132008331002335,
    1.689734718108285,
    2.040784224372669e-04,
    -0.001484090442093,
    -0.158273083834361];


%TLE2_nonsing_code
[6.757218129943740e+06,
    0.028446881378213,
    1.689728514669987,
    2.091751874558519e-04,
    -0.001483847372420,
    -0.158242319133329];


%%
%Expected Results

Dv = [0.296468243701444,
    0.064665567732272,
    -0.341538920109528];

Positions_code = [0.047654863819695,
    9.472432824589074,
    94.295434471513490];

M_final = 3;

N_final = 30;

%% Actual Results

Dv1 = 0.2964682437013073;
Dv2 = 0.0646655677322395;
Dv3 = -0.3415389201094243;
M = 3.0;
N = 30.0;
theta1 = 0.04765486374969757;
theta2 = 9.472432824519077;
theta3 = 94.2954344714435;

Counts = 550588; % @ 100 MHz -> 5.506 ms
```

# Bibliography


[1] J. Guo, J. Bouwmeester, and E. Gill, "From Single to Formation Flying CubeSats: An Update of the Delfi Programme," in *27th Annual AIAA/USU Conference on Small Satellites*, 2013.

[2] E. Gill, P. Sundaramoorthy, J. Bouwmeester, B. Zandbergen, and R. Reinhard, "Formation flying within a constellation of nano-satellites: The QB50 mission," *Acta Astronautica*, vol. 82, no. 1, pp. 110–117, 2013.

[3] D. C. Maessen, *Autonomous Relative Navigation for Small Spacecraft.* PhD thesis, 2014.

[4] S. D'Amico, *Autonomous formation flying in low earth orbit.* PhD thesis, TU Delft, 2010.

[5] J. Guo, "Phase-B Study on Formation Flight within QB50," tech. rep., TU Delft, 2013.

[6] E. Kahr, O. Montenbruck, and K. P. G. OKeefe, "Estimation and Analysis of Two-Line Elements for Small Satellites," *Journal of Spacecraft and Rockets*, vol. 50, pp. 433–439, Mar. 2013.

[7] S. D'Amico, J. S. Ardaens, and S. De Florio, "Autonomous formation flying based on GPS - PRISMA flight results," *Acta Astronautica*, vol. 82, pp. 69–79, Jan. 2013.

[8] J. R. Carpenter and K. T. Alfriend, "Navigation Guidelines for Orbital Formation Flying Missions," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2003.

[9] M. R. Greene and R. E. Zee, "Increasing the Accuracy of Orbital Position Information from NORAD SGP4 Using Intermittent GPS Readings," in *Small Satellite Conference*, 2009.

[10] M. Kirschner, O. Montenbruck, and S. Bettadpur, "Flight dynamics aspects of the GRACE formation flying," *The 2nd International Workshop on Satellite Constellations and Formation Flying Haifa Israel*, pp. 1–8, 2001.

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

[12] B. G. Coffee, K. Cahoy, and R. L. Bishop, "Propagation of CubeSats in LEO using NORAD Two Line Element Sets : Accuracy and Update Frequency," in *AIAA Guidance, Navigation and Control Conference*, (Boston, MA), pp. 1–15, AIAA, 2013.

[13] D. P. Scharf, F. Y. Hadaegh, and S. R. Ploen, "A survey of spacecraft formation flying guidance and control. Part II: control," in *Proceedings of the 2004 American Control Conference*, vol. 4, JPL, AACC, 2004.

[14] D. Scharf, F. Y. Hadaegh, and S. R. Ploen, "A Survey of Spacecraft Formation Flying Guidance and Control (Part I): Guidance," in *American Control Conference*, 2003.

[15] H. Zhang and P. Gurfil, "Satellite cluster flight using on-off cyclic control," *Acta Astronautica*, vol. 106, pp. 1–12, 2015.

[16] J.-S. Ardaens and S. D'Amico, "Spaceborne Autonomous Relative Control System for Dual Satellite Formations," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 6, pp. 1859–1870, 2009.

[17] A. Guerman, M. Ovchinnikov, G. Smirnov, and S. Trofimov, "Closed relative trajectories for formation flying with single-input control," *Mathematical Problems in Engineering*, 2012.

[18] K. T. Alfriend, S. R. Vadali, P. Gurfil, J. P. How, and L. S. Breger, "Spacecraft Formation Flying," in *Spacecraft Formation Flying*, pp. 305–328, Elsevier, 2010.

[19] A. Guerman, M. Ovchinnikov, and G. Smirnov, "High-precision single-input control of relative motion in spacecraft formation," *Acta Astronautica*, vol. 94, no. 1, pp. 375–382, 2014.

[20] F. J. De Bruijn and E. Gill, "Influence of sensor and actuator errors on impulsive satellite formation control methods," *Acta Astronautica*, vol. 94, no. 2, pp. 608–618, 2014.

[21] W. Huang, "Optimal Multi-Impulse Orbit Transfer Using Nonlinear Relative Motion Dynamics," *Journal of the Astronautical Sciences*, vol. 59, no. January-June, pp. 243–264, 2012.

[22] S. S. Vaddi, K. T. Alfriend, S. R. Vadali, and P. Sengupta, "Formation Establishment and Reconfiguration Using Impulsive Control," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 2, pp. 262–268, 2005.

[23] D. J. Irvin, "Linear vs. Nonlinear Control Techniques For the Reconfiguration of Satellite Formations," *AIAA Guidance, Navigation and Control Conference and Exhibit*, no. August, 2001.

[24] Godard, K. Dev Kumar, and A. Zou, "Robust stationkeeping and reconfiguration of underactuated spacecraft formations," *Acta Astronautica*, vol. 105, no. 2, pp. 495–510, 2014.

[25] J.-F. Hamel and J. D. Lafontaine, "Neighboring Optimum Feedback Control Law for Earth-Orbiting Formation-Flying Spacecraft," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 1, pp. 290–299, 2009.

[26] J. K. Eyer, *A Dynamics and Control Algorithm for Low Earth Orbit Precision Formation Flying Satellites*. Phd, 2009.

[27] C. W. T. Roscoe, S. R. Vadali, K. T. Alfriend, and U. P. Desai, "Satellite formation design in orbits of high eccentricity with performance constraints specified over a region of interest: MMS phase II," *Acta Astronautica*, vol. 82, no. 1, pp. 16–24, 2013.

[28] M. Navabi and M. Barati, "A Comparative Study of Dynamics Models and a Control Strategy for Satellite Formation Flying," in *IAA Conference on Dynamics and Control of Space Systems*, (Porto), pp. 1–13, 2012.

[29] R. Kristiansen and P. J. Nicklasson, "Spacecraft formation flying: A review and new results on state feedback control," *Acta Astronautica*, vol. 65, no. 11-12, pp. 1537–1552, 2009.

[30] Y. Choi, S. Mok, and H. Bang, "Impulsive formation control using orbital energy and angular momentum vector," *Acta Astronautica*, vol. 67, no. 5-6, pp. 613–622, 2010.

[31] X. Hou, Y. Zhao, and L. Liu, "Formation flying in elliptic orbits with the J2 perturbation," *Research in Astronomy and Astrophysics*, vol. 12, no. 11, pp. 1563–1575, 2012.

[32] C. Wei, S.-Y. Park, and C. Park, "Optimal H-infinity robust output feedback control for satellite formation in arbitrary elliptical reference orbits," *Advances in Space Research*, vol. 54, no. 6, pp. 969–989, 2014.

[33] J. Li and X.-N. Xi, "Fuel-Optimal Low-Thrust Reconfiguration of Formation- Flying Satellites via Homotopic Approach," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 6, pp. 1709–1717, 2012.

[34] L. Cao and X. Chen, "Input-output linearization minimum sliding-mode error feedback control for spacecraft formation with large perturbations," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 229, no. 2, pp. 352–368, 2014.

[35] L. Cao, X. Chen, and A. K. Misra, "Minimum sliding mode error feedback control for fault tolerant reconfigurable satellite formations with J2 perturbations," *Acta Astronautica*, vol. 96, no. 1, pp. 201–216, 2014.

[36] K. D. Kumar, A. K. Misra, S. Varma, T. Reid, and F. Bellefeuille, "Maintenance of satellite formations using environmental forces," *Acta Astronautica*, vol. 102, pp. 341–354, 2014.

[37] H. E. Park, S. Y. Park, and K. H. Choi, "Satellite formation reconfiguration and station-keeping using state-dependent Riccati equation technique," *Aerospace Science and Technology*, vol. 15, no. 6, pp. 440–452, 2011.

[38] G. Di Mauro, P. Di Lizia, R. Armellin, and M. Lavagna, "Nonlinear control of leader-follower formation flying," *Advances in the Astronautical Sciences*, vol. 145, pp. 215–230, 2012.

[39] M. Massari, F. Bernelli-Zazzera, and S. Canavesi, "Nonlinear Control of Formation Flying with State Constraints," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 6, pp. 1919–1925, 2012.

[40] D. Morgan and S. Chung, "Model Predictive Control of Swarms of Spacecraft Using Sequential Convex Programming," vol. 37, no. 6, pp. 1–25, 2014.

[41] M. S. De Queiroz, V. Kapila, and Q. Yan, "Adaptive Nonlinear Control of Multiple Spacecraft Formation Flying," *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 3, pp. 385–390, 2000.

[42] A. De Ruiter, "Adaptive spacecraft formation flying with actuator saturation," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 224, no. 4, pp. 373–385, 2010.

[43] K. W. Lee and S. N. Singh, "Variable-Structure Model Reference Adaptive Formation Control of Spacecraft," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 1, pp. 104–115, 2012.

[44] Y. Zhao, Y. Wang, and C. Zhang, "The state-space model identification-based adaptive collaborative multi-satellite simulation system," in *Fourth World Congress on Software Engineering*, pp. 293–297, 2013.

[45] B. S. Kumar, A. Ng, K. Yoshihara, and A. De Ruiter, "Differential drag as a means of spacecraft formation control," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 2, pp. 1125–1135, 2011.

[46] T. Williams and Z.-s. Wang, "Uses of solar radiation pressure for satellite formation flight," *Int. J. Robust Nonlinear Control*, vol. 12, pp. 163–183, 2002.

[47] D. W. Miller, U. Ahsun, and J. L. Ramirez-Riberos, "Control of Electromagnetic Satellite Formations in Near-Earth Orbits," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 6, pp. 1883–1891, 2010.

[48] D. R. Jones and H. Schaub, "Periodic relative orbits of two spacecraft subject to differential gravity and electrostatic forcing," *Acta Astronautica*, vol. 89, pp. 21–30, 2013.

[49] D. Jones and H. Schaub, "Collinear Three-Craft Coulomb Formation Stability Analysis and Control," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 1, pp. 224–232, 2014.

[50] H. Huang, Y. W. Zhu, L. P. Yang, and Y. W. Zhang, "Stability and shape analysis of relative equilibrium for three-spacecraft electromagnetic formation," *Acta Astronautica*, vol. 94, no. 1, pp. 116–131, 2014.

[51] J. Guo, "CubeSat Design Overview Report," tech. rep., TU Delft, 2014.

[52] J. Guo, "DelFFi Mass Budget," tech. rep., TU Delft, 2015.

[53] A. S. D. Cervone, B. S. D. Zandbergen, J. S. D. Guo, E. S. D. Gill, W. T. Wieling, F. T. Tata Nardini, and C. T. Schuurbiers, "Application of an Advanced Micro-Propulsion system to the DelFFi Formation-Flying Demonstration Within the QB50 Mission," in *63rd International Astronautical Congress*, (Naples, Italy), IAC, 2012.

[54] A. Cervone, E. Janssen, and T. van Wees, "Propulsion Subsystem Requirements for the DelFFi Satellites," tech. rep., TU Delft, 2015.

[55] R. M. A. Poyck, *Design, manufacturing and characterisation of a water fed CubeSat micro-resistojet.* Master if science, TU Delft, 2014.

[56] A. Migliaccio, B. Zandbergen, F. Nardini, and M. C. Louwerse, "Vacuum Testing of a Micropropulsion System Based on Solid Propellant Cool Gas Generators," in *61st International Astronautical Congress*, TU Delft, 2010.

[57] J. Guo, "DelFFi Power Budget," tech. rep., TU Delft, 2015.

[58] M. Haghayegh, *Design, Implementation and Verification of the Attitude Determination and Control Algorithm for the DelFFi Satellites.* Master of science, TU Delft, 2015.

[59] J. Viru, "ADCS Processor Technical Note," tech. rep., TU Delft, 2015.

[60] ECSS, "ECSS-E-ST-10-06C Space engineering: Technical requirements specification," Tech. Rep. March, European Cooperation for Space Standardization, 2009.

[61] ECSS, "ECSS-E-ST-10-02C Space engineering: Verification," Tech. Rep. March, European Cooperation for Space Standardization, 2009.

[62] A. Deeb, "Formation Maintenance for the DelFFi Mission: Literature Study," tech. rep., TU Delft, 2015.

[63] J. Guo, "Phase-A Study on Formation Flight within QB50," tech. rep., TU Delft, 2012.

[64] E. Gill, "Onboard Navigation Function," tech. rep., TU Delft, 2012.

[65] F. R. Hoots and R. L. Roehrich, "Spacetrack Report Number 3 - Models for propagation of NORAD element sets," *U.S. Air Force Aerospace Defense Command Report*, no. 3, pp. 1–87, 1980.

[66] S. R. Vadali, S. S. Vaddi, and K. T. Alfriend, "An Intelligent Control Concept for Formation Flying Satellites," *Int. J. Robust Nonlinear Control*, vol. 12, pp. 97–115, 2002.

[67] H. Schaub, H. Schaub, S. R. Vadali, S. R. Vadali, J. L. Junkins, J. L. Junkins, K. T. Alfriend, and K. T. Alfriend, "Spacecraft Formation Flying Control Using Mean Orbit Elements," *Journal of the Astronautical Sciences*, vol. 48, no. 1, pp. 69–87, 2000.

[68] H. Schaub and J. L. Junkins, *Analytical Mechanics of Space Systems.* Reston, VA: American Institute of Aeronautics and Astronautics, 2003.

[69] C. W. T. Roscoe, J. D. Griesbach, J. J. Westphal, D. R. Hawes, and J. P. Carrico, Jr., "Force Modeling and State Propagation for Navigation and Maneuver Planning for CubeSat Rendezvous, Proximity Operations, and Docking," *Advances in the Astronautical Sciences*, vol. 150, pp. 573–590, 2014.

[70] T. Scholz, C. O. Asma, and A. Aruliah, "Recommended Set of Models and Input Parameters for the Simulations of Orbital Dynamics of the Qb50 Cubesats," no. 1.0, pp. 1–8, 2012.

[71] E. Gill, P. Sundaramoorthy, J. Bouwmeester, and B. Sanders, "Formation flying to enhance the QB50 space network," in *Small Satellite Systems and Services Symposium (4S)*, ESA.

[72] B. D. Tapley, M. M. Watkins, J. C. Ries, G. W. Davis, R. J. Eanes, S. R. Poole, H. J. Rim, B. E. Schutz, C. K. Shum, R. S. Nerem, F. J. Lerch, J. a. Marshall, S. M. Klosko, N. K. Pavlis, and R. G. Williamson, "The Joint Gravity Model 3," *Journal of Geophysical Research*, 1996.

[73] N. H. Roth, *Navigation and Control Design for the CanX-4/-5 Satellite Formation Flying Mission.* Master of applied science, 2010.

[74] G. Inalhan, M. Tillerson, and J. P. How, "Relative Dynamics and Control of Spacecraft Formations in Eccentric Orbits," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 1, pp. 48–59, 2002.

[75] R. Abay, *KAM Torus Orbit Prediction from Two Line Element Sets.* PhD thesis, 2014.

[76] R. Wang, J. Liu, and Q. M. Zhang, "Propagation errors analysis of TLE data," *Advances in Space Research*, vol. 43, no. 7, pp. 1065–1069, 2009.

[77] M. Marszalek, O. Kurz, M. Drentschew, M. Schmidt, and K. Schilling, "Intersatellite links and relative navigation: Pre-conditions for formation flights with pico- and nanosatellites," in *18th IFAC World Congress*, vol. 18, pp. 3027–3032, International Federation of Automatic Control (IFAC), 2011.

[78] P. Sundaramoorthy, E. Gill, C. Verhoeven, and J. Bouwmeester, "Two CubeSats with Micro-Propulsion in the QB50 Satellite Network," in *24th Annual AIAA/USU Conference on Small Satellites*, pp. 1–11, TU Delft, AIAA, 2010.

[79] "Tuning Fork Crystal Unit CM415," tech. rep., CITIZEN, 2015.

[80] K. Kudapali, "Run-Time Calibration of Watch Crystals (AN1155)," tech. rep., Microchip Technology Inc., 2008.

[81] F. Bräuer, *System Architecture Definition of the DelFFi Command and Data Handling Subsystem.* Master science, TU Delft, 2015.

[82] J. C. M. Fraticelli, "Simulink Code Generation," tech. rep., NASA Marshall Space Flight Center, 2012.

[83] P. M. Bainum, A. Strong, Z. Tan, and X. Duan, "Techniques for Station Keeping Elliptically Orbiting Constellations in Along-track Formation," *International Journal of Solids and Structures*, vol. 42, pp. 5683–5691, 2005.

[84] UTIAS-SFL, "CanX-4&5 Formation Flying Mission Accomplished," *Available at: http://utias-sfl.net/?p=2191*, 2015.

[85] J. B. Mueller, "A multiple-team organization for decentralized guidance and control of formation flying spacecraft," *American Institute of Aeronautics and Astronautics (AIAA)*, 2004.

[86] L. Breger and J. P. How, "Gauss's Variational Equation-Based Dynamics and Control for Formation Flying Spacecraft," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 2, pp. 437–448, 2007.

[87] M. Ovchinnikov, G. V. Smirnov, and I. Zaramenskikh, "Orbital corrections by a single-input impulsive control applied along the geomagnetic field," *Acta Astronautica*, vol. 65, no. 11-12, pp. 1826–1830, 2009.

[88] W. Wu, J. Guo, and D. Florijn, "DelFFi ADCS - Control Torques," tech. rep., TU Delft, 2014.

[89] K. Wang, T. Chen, and S. Xu, "A Comparison of Algorithms for Computing Differential Mean Orbital Elements in Formation Flying Missions," in *AIAA Guidance, Navigation, and Control (GNC) Conference*, pp. 1–21, AIAA, 2013.

[90] F. Bauer, J. Bristow, D. Folta, K. Hartman, D. Quinn, J. How, F. Bauer, J. Bristow, D. Folta, K. Hartman, D. Quinn, and J. How, "Satellite formation flying using an innovative autonomous control system (AutoCon) environment," in *Guidance, Navigation, and Control Conference*, American Institute of Aeronautics and Astronautics, Aug. 1997.

[91] E. Gill, S. D'Amico, and O. Montenbruck, "Autonomous Formation Flying for the PRISMA Mission," *Journal of Spacecraft and Rockets*, vol. 44, pp. 671–681, May 2007.

[92] D. E. Keenan, "A formation flying strategy for CloudSat/Picasso-Cena," *2001 IEEE Aerospace Conference Proceedings (Cat. No.01TH8542)*, vol. 2, pp. 535–552, 2001.

[93] S. Djalal, M. Delpech, P. Guidotti, and J. Berges, "Relative Guidance of Formation Flying Mission Spacecraft: from Design to Flight Tests in FFIORD Mission," in *22nd International Symposium on Space Flight Dynamics*, no. 1, (Sao Jose dos Campos, Brazil), CNES, 2011.

[94] G. Bonin, N. Orr, S. Armitage, N. Roth, B. Risi, and R. E. Zee, "The CanX-4&5 Mission: Achieving Precise Formation Flight at the Nanosatellite Scale," in *64th International Astronautical Congress*, pp. 1–6, 2013.

[95] N. Olsen and E. Friis-Christensen, "Swarm End-to-End Mission Performance Simulator Study Final Report," tech. rep., Danish Space Research Institute, 2005.

[96] H. Lübberstedt, M. Kassebom, and C. Tobehn, "SWARM Constellation Deployment & Maintenance," in *5th International Symposium of the International Academy of Astronautics*, (Berlin), 2005.

[97] C. Olson, A. Long, and J. R. Carpenter, "Sensitivity of Magnetospheric Multi-Scale (MMS) mission navigation accuracy to major error sources," *Advances in the Astronautical Sciences*, vol. 140, pp. 1495–1512, 2011.

[98] J. Guzman and A. Edery, "Mission design for the MMS tetrahedron formation," *2004 IEEE Aerospace Conference Proceedings*, vol. 1, pp. 533–540, 2004.