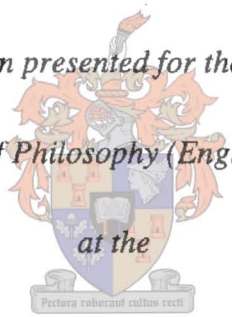
LOW EARTH ORBIT SATELLITE CONSTELLATION CONTROL USING ATMOSPHERIC DRAG

Daniel N. J. du Toit

Dissertation presented for the degree of

Doctor of Philosophy (Engineering)

at the



University of Stellenbosch.

Promoter: Prof. J.J. du Plessis

January 1997

DECLARATION

I, the undersigned, hereby declare that the work contained in this dissertation is my own original work and that I have not previously in its entirety or in part submitted it at any other university for a degree.

Signature: 

Date: 31/03/1997

SYNOPSIS

This dissertation considers the feasibility of using atmospheric drag to control constellations of micro-satellites in low Earth orbits. The constellation control requirements include an acquisition phase and a maintenance phase. Optimal strategies are designed to control the relative positions of the satellites during these two phases. It is shown that the feasibility and success of the strategies depend on many factors, including the satellite properties and orbital configuration. A nominal test constellation is presented and used as a generic example for the application of the control strategies.

The dissertation also focuses on the accurate modelling and simulation of a typical low Earth orbit satellite, moving under the influence of a variety of significant orbit perturbation forces. The simulations form an integral part of the study and are used to verify the application of all the proposed control strategies.

Keywords: atmospheric drag, constellation control, orbit propagation, special perturbations.

OPSOMMING

In hierdie verhandeling word die moontlikhede ondersoek om gebruik te maak van atmosferiese sleurkrag om mikro-satelliet konstellasies in lae aardbane te beheer. Die konstellasie-beheer vereistes sluit 'n verkrygingsfase en 'n instandhoudingsfase in. Optimale strategieë word ontwerp om die relatiewe posisies van die satelliete tydens hierdie twee fases te beheer. Die studie toon aan dat die sukses van die konsep afhang van verskeie faktore, insluitende die satelliet en wentelbaan konfigurasie. 'n Nominale konstellasie word voorgestel en gebruik as 'n generiese voorbeeld om die voorgestelde strategieë te demonstreer.

Die verhandeling fokus ook op die akkurate modellering en simulasie van 'n tipiese lae aardbaan satelliet wat onder die invloed van verskeie perturbasie-kragte beweeg. Die simulasies vorm 'n integrale deel van die studie en word gebruik om alle voorgestelde beheer-strategieë te toets.

ACKNOWLEDGEMENTS

I would like to thank Prof. J.J. du Plessis and Dr. W.H. Steyn, both of the department of Electrical & Electronic Engineering at the University of Stellenbosch, for their advice and continuous support throughout this project.

Financial support for the project was provided by the Foundation for Research and Development.

TABLE OF CONTENTS

LIST OF SYMBOLS

A	cross-sectional area (mostly used with subscripts A_{min}, A_{max}, A_1, A_2)
a	semi-major axis
$\mathbf{a_p}$	three-dimensional vector sum of all orbit perturbation accelerations
C_D	drag coefficient
d	orbit-average distance
δ	time step
Δa	change (loss) in altitude
Δh	orbit-average altitude difference
E	eccentric anomaly
e	eccentricity
γ	flight angle
h	altitude
H	Hamiltonian
i	inclination of the orbit
J_2	Earth oblateness constant
K	feedback gain matrix; also reflectivity constant for the satellite
k_1, k_2	parameters of the second-order linearised model
μ	Earth gravitational constant
m	satellite mass
M	upper and lower bounds of the control signal; also used for mean anomaly
n	mean motion
p	semilatus rectum
P	solar momentum flux
θ	true anomaly
Q, R	LQR weighting matrices
ρ	atmospheric density
r	distance from the centre of the Earth
$\mathbf{r}$	three-dimensional position vector

R_E	Earth radius
s	switching function; also Laplace variable
S_1, S_2	symbols for satellites number 1 and 2
t	time
T_0	orbital period
t_0	start time of control effort; also used for epoch time
t_f	end time of control effort
u	control signal
v	magnitude of velocity
$\mathbf{v}$	three-dimensional velocity vector
ω	argument of the perigee
Ω	right ascension of the ascending node
$\mathbf{x}$	state vector

Chapter 1

INTRODUCTION

1. OVERVIEW

1.1 MAKING USE OF ATMOSPHERIC DRAG

The theory of atmospheric drag and its influence on Earth orbiting satellites has been well analysed and studied. See King-Hele (1987, as well as earlier works in 1964 and 1969) for an in-depth treatment of the effects of atmospheric drag on Earth satellites. Many works on astrodynamics – for example Bate *et al* (1971), Battin (1987), Chobotov (1991), Danby (1962), Escobal (1965) and Roy (1988) – include a section on atmospheric drag effects. Other relevant works include Kaplan (1976), Larson *et al* (1992) and Wertz (1978). Some articles on the subject can be found in El'yasberg *et al* (1967) and Morando (1970). A thorough treatment of the physical structure of the Earth's atmosphere is given by Langton (1969). In most of the existing works, drag is considered as an *unwanted* orbit perturbation force causing the satellite to deviate from the idealised Kepler orbit.

In this study, however, the possibilities of benefiting from the presence of atmospheric drag – in other words *using* it – will be investigated. The fundamental concept that will be used is simple: if a satellite's cross-sectional area, projected on a plane perpendicular to the velocity vector can be changed, the magnitude of the atmospheric drag acting on the satellite can be controlled. The resulting control force is limited in

magnitude and application direction, but when utilised correctly, can influence the satellite motion to reach a desired effect.

The concept of utilising atmospheric drag has been investigated before, reportedly by the Orbital Sciences Corporation on the ORBCOMM project, yet no relevant publications could be found in the open literature. A complete computerised search was done on the *Inspec Electronics and Computing* abstracts compact disc (CD) database for the years 1993 to 1996, as well as the *Ei Compendex* abstracts CD database for the years 1992 to 1996 (see references).[1]

1.2 LEO CONSTELLATION CONTROL

The application of the variable drag concept will be limited to constellations of low Earth orbit (LEO) satellites. The satellites must be at these relatively low altitudes for the atmospheric drag force to be large enough. A number of LEO constellations for scientific, surveillance and communications applications are presently planned (for example Iridium, GlobalStar, Teledesic, NASA EOS, Aries, Starsys) and some are already in operation (for example Orbcomm, Transit). When considering the normal operation of a constellation of satellites, a variety of physical and other factors, especially atmospheric drag effects, tend to alter the relative positions of the satellites in the constellation. Some sort of *constellation maintenance* is usually necessary to maintain the desired constellation pattern. Atmospheric drag removes energy from the orbit and causes the satellites two slowly spiral in towards the Earth. Most constellation maintenance strategies rely on on-board propulsion for occasional boosts, each time adding energy to the orbit so that the effects of drag are counter-acted. Ross *et al* (1995) discuss the concept of thrust-drag cancellation to maintain a forced Keplerian trajectory. The disadvantages of these methods include the dependence on on-board energy and the addition of a significant component to the total mission cost. Collins *et al* (1996) presents an autonomous constellation maintenance system, specifically designed to reduce cost, but it still requires the

---

[1] Keywords for the computerised search included: "drag", "constellation", "orbcomm", "orbital sciences" and various combinations with the term "control".

additional complexity of on-board propulsion and the use of limited energy resources on the satellite.

This study will introduce a passive control strategy, using the variable drag concept for relative constellation control, that eliminates the dependence on on-board propulsion systems. The specific mission applications, requirements and conditions under which the benefits can be obtained will be outlined and the boundaries of the advantages will be determined.

The control requirements for a LEO constellation will not only be limited to constellation maintenance, but will also include *constellation acquisition*. This refers to the control phase where a desired constellation configuration must be reached from a certain initial constellation configuration. There may only be a couple of acquisition phases during the constellation's lifetime and the time of each would normally be short in comparison to the total lifetime. The acquisition phase merely sets up the constellation for its normal operation. It may imply large relative movements between the various satellites of the constellation. The proposed variable drag control concept will also be utilised in strategies to control the relative movements of the satellites during the acquisition phase.

The following section will give an example of a concept mission where the possibility of applying variable drag control strategies during acquisition and maintenance of the constellation exists.

1.3 THE CHIPSAT MISSION

The CHIPSAT mission is currently planned as part of the NASA MTPE (Mission to Planet Earth) project and will provide accurate position information for gravity modelling and signal occultation measurements for atmospheric profiling. For these measurements, a number of satellites in a constellation will each carry a specially configured GPS receiver. To increase sensitivity to the features of the Earth's gravity field, the satellites must be flown in circular orbits at the lowest possible altitude. At these low altitudes, however, the effects of atmospheric drag on the satellites become very significant. This has two major disadvantages for the mission: firstly the lifetime

of the mission is limited and secondly the gravity field estimates coming from the precise position measurements will be contaminated with large errors due to the atmospheric drag. The first constraint simply implies that the satellites must be higher than a certain minimum altitude to ensure enough lifetime. The second problem, however, may be overcome by means of a common mode cancellation scheme. Pairs of satellites orbiting in tandem and kept a certain distance apart could provide measurements from which the effects of atmospheric drag can be cancelled and the effects of gravity variations can be deduced accurately. The distance between the satellites must be large enough to ensure sensitivity to gravity variations and small enough for the state of the atmosphere to be nearly identical at the two locations. The latter condition will ensure that the atmospheric drag on the two satellites is nearly the same. A separation distance of 500 km to 1000 km is considered to be sufficient. The additional advantage of this scheme is that the effects of other non-conservative perturbing forces, such as solar radiation pressure, are also cancelled.

This study will show that it is possible to design a control system, utilising atmospheric drag, to control the positions of LEO satellites in tandem constellations relative to each other. For the CHIPSAT mission, the necessity of this control effort becomes obvious when considering the fact that a group of satellites will be launched together and will initially be close to each other in orbit. From this initial position the constellation can be set up to meet the spacial configuration requirements of the moving measurement grid. This first phase is the acquisition phase. It will be followed by a maintenance phase to keep the constellation at the specified configuration.

2. OBJECTIVES

The primary objectives of the study can be summarised as follows:

a) Obtaining a thorough understanding of orbital theory, orbit perturbations and perturbation methods for LEO satellites.
b) The development of a high-precision orbit propagator which could provide accurate short-term solutions for the velocity and position of LEO satellites.
c) From the results of (a) and (b), to design and test optimal control strategies, using the concept of variable atmospheric drag, for
    i. constellation acquisition and
    ii. constellation maintenance

    of LEO tandem constellations.

The following section describes the layout of this document. The chapters follow the objectives logically.

3. LAYOUT

The mathematical foundation for this study is provided in chapter 2. It deals with the theory of orbits and perturbations. The emphasis will be on the mathematical modelling of orbit perturbation forces. Methods of accurately determining a satellite's position and velocity in a general perturbed orbit will also be discussed.

Chapter 3 describes the high-precision orbit propagation software that has been developed, with specific reference to the underlying structure and functionality.

The theme of chapter 4 is constellation acquisition. The simplest case of a two-satellite constellation will be considered first and the relative movement between the two satellites will be modelled. The criteria of minimum control time and minimum altitude loss during the control effort will then be used to design optimal control

strategies to take the constellation from the initial configuration to the specified final configuration. It will be shown that both criteria lead to extremal control strategies and that the minimum-time strategy has a *bang-bang* form and the minimum-altitude-loss strategy has a *bang-off-bang* form. The effects of unpredictable time variations in the atmospheric density will also be considered and adaptations to the control strategies will be introduced to maintain accuracy and robustness under these circumstances. Finally, the designs will be generalised for constellations with more than two satellites. The objective will be to maintain optimality in the minimum-time and minimum-altitude-loss sense.

Chapter 5 deals with constellation maintenance. The simplest case of a two-satellites constellation will again serve as the point of departure. A near-optimal linear quadratic regulator, to keep the satellites in the specified spacial configuration, will be designed and demonstrated. For constellations with more satellites, the problem has a multiple-input multiple-output (MIMO) nature with the potential of cross-coupling effects. A procedure to decouple the regulator will be developed.

The results will be summarised and conclusions will be drawn in chapter 6.

4. THE NOMINAL TEST CASE

The philosophy adopted for the study was to generalise problems, analyses and designs as far as possible. This allows for adaptation of the results to the widest possible range of specific cases. A nominal satellite and orbital configuration was chosen to serve as a generic and representative example, providing the platform for the application of the proposed concepts through high-precision computer simulations. The parameters of this nominal test case can easily be changed to investigate any other configuration. It is important to note that the emphasis of the thesis is not on the specific design, but rather on demonstrating the control concepts.

In the following sections, the nominal orbital configuration, co-ordinate convention and initial conditions will be explained. These were used in all analyses and

simulations, unless stated otherwise. The physical properties of the nominal satellites will be discussed in chapter 4.

4.1 ORBITAL CONFIGURATION

The proposed concepts are applied to both circular and elliptical orbits. A circular orbit was chosen to form the basis of the analyses and designs and the concepts were later extended to elliptical orbits. The nominal circular orbit has an altitude of 450 km. The nominal elliptical orbit has a perigee height of 450 km and an eccentricity (e) of 0.02. The resulting apogee height is 728 km. Both orbits have a 90° inclination, chosen to maximise the perturbation effects due to the equatorial bulge of the Earth (see chapter 3) and thus create a worst-case scenario for the application and performance evaluation of the proposed control strategies.

As mentioned before, the proposed control concepts will only be successful for satellites in the same orbital plane. This is a direct result of the fact that atmospheric drag acts in line with the satellite's velocity vector. A variety of factors, including different injection conditions and certain perturbations might result in slightly different orbital planes for different satellites of a constellation. The proposed control strategy will be unable to maintain the constellation under these conditions.

4.2 CO-ORDINATE SYSTEM CONVENTION

A Cartesian co-ordinate system will be used, with origin at the centre of the Earth. The x axis points in the direction of the vernal equinox and the z axis coincides with the Earth's spin axis, pointing in the direction of the north pole. The y axis completes the right-handed set.

Figure 1.1 (next page) shows the co-ordinate convention. The position of the vernal equinox and the orientation of the Earth's spin axis will be assumed constant so that

the co-ordinate system can be assumed inertial over the relative short time-scales of interest.[2]

A general elliptical orbit is also shown in the figure. The following orbital parameters are indicated: the inclination of the orbit (i), the argument of the perigee (ω, measured in the orbital plane) and the right ascension of the ascending node (Ω, measured in the equatorial plane).

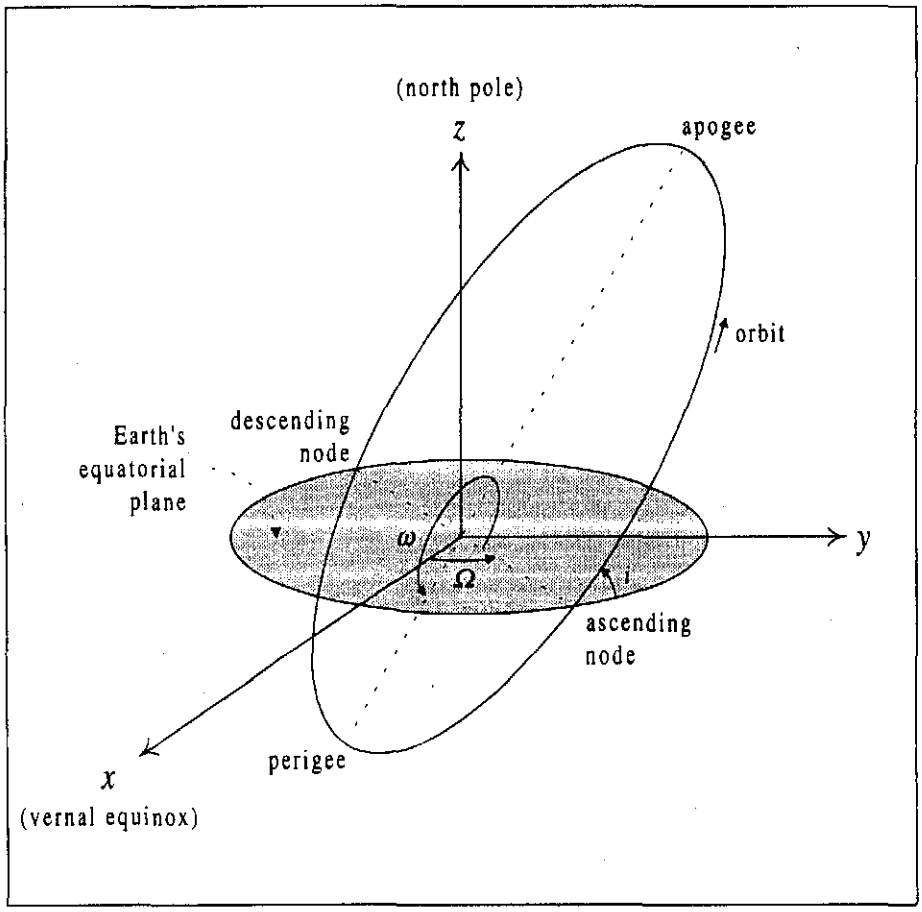


Figure 1.1: Co-ordinate system convention.

---

[2] Typical constellation acquisition control times, for a 500 km distance change between the satellites, are in the order of 100 orbits (6-7 days), which is very short in comparison with the rate of precession of the equinoxes relative to the fixed stars. This rate is approximately 50 sec of arc per year (Wertz, 1979).

4.3 INITIAL CONDITIONS

In all simulations, the satellites are initialised with a position on the positive x axis. This corresponds to a position above the equator, in the direction of the vernal equinox. The magnitude of the velocity is initialised to correspond with an altitude of 450 km in the case of the circular orbit. For the elliptical orbit, the initial velocity corresponds with a perigee height of 450 km.

Chapter 2

THE THEORY OF ORBITS AND PERTURBATIONS

1. INTRODUCTION

The study of the dynamics of bodies in interplanetary or interstellar space is in general referred to as *astrodynamics*. Within this discipline there are two major divisions. The first is the movement of the body's centre of mass, referred to as *kinematics* (also *celestial mechanics* or *orbit dynamics*). The second is the movement of the body around its centre of mass, referred to as *attitude dynamics*. This chapter deals with kinematics – specifically the movement of satellites in low Earth orbits. For the purpose of this study it is important to model the movement of such satellites as accurately as possible. The attitude dynamics are not as important here and are just mentioned briefly in appendix A.

The topic of celestial mechanics has been widely studied and is thoroughly documented in a large number of works.[1] Only a brief overview of orbital theory will be given here. Special emphasis will be placed on the modelling of orbit perturbation

---

[1] Many of the referenced works cover celestial mechanics. For a thorough treatment, see any one of the following: Bate *et al* (1971); Danby (1962); Escobal (1965) or Roy (1988).

forces, as well as methods to determine satellite trajectories in the presence of these perturbation forces.

Of particular interest is the determination of the atmospheric density, which is necessary to compute the atmospheric drag force on the LEO satellite. The spacial and temporal variations in the atmospheric density will be considered and a suitable model will be derived for application in this study.

2. OVERVIEW OF ORBITAL THEORY

2.1 KEPLER ORBITS

Consider the movement of a small satellite near the Earth. Assume that the only force acting on the satellite is the gravitational force of the Earth and that the Earth is spherical and homogenous so that it can be modelled as a point mass at its centre. Assume further that the mass of the satellite is negligible in comparison with the Earth's mass. Following these approximations, the satellite's trajectory around the Earth can be expressed analytically as an ellipse.[2] A circular orbit is included as a special case. This ideal elliptical orbit is referred to as a Kepler orbit.

Five parameters, called orbital elements, describe the size, position and shape of the orbit. The orbital parameters stay constant for a satellite in an ideal Kepler orbit. The classical set of orbital elements are:

- semi-major axis (a);
- eccentricity (e);
- inclination (i);
- argument of the perigee (ω); and
- right ascension of the ascending node (Ω).

---

[2] The derivation of this result can be found in most works on celestial mechanics.

A sixth orbital element, usually the mean anomaly (M) or true anomaly (θ), defines the position of the satellite in the orbit.

Even though the ideal Kepler orbit is in many cases a good approximation of the satellite motion, it is not sufficient for this study. It does, however, provide an essential mathematical foundation that will be needed later in this study, especially for the initialisation of simulations. The relevant theory and equations will be given in chapter 3.

2.2 ORBIT PERTURBATIONS

Orbit perturbation forces cause the satellite to deviate from the ideal Kepler orbit. These forces are caused by the physical properties of the space environment in which the LEO satellite operates. The relative significance of the various possible perturbations depends primarily on the physical properties of the satellite (shape, mass etc.) and the orbital configuration. For the typical LEO satellite, the following perturbation effects dominate and will be included in the remainder of this study:[3]

- atmospheric drag effects;
- perturbations due to the non-spherical gravitational field of the Earth;
- gravitational attractions by the Sun and the Moon; and
- solar radiation pressure.

All other perturbation effects are significantly (orders of magnitude) smaller than those mentioned above and will be neglected.[4] Some of these effects include:

- Earth-reflected solar radiation pressure;
- induced eddy currents in the satellite structure interacting with the Earth's magnetic field;
- drag due to solar wind;

---

[3] References to existing literature will be provided when the various perturbations are handled.

[4] See for example Chobotov (1991), Smith (1969), Kaplan (1979) for a discussion of these perturbations.

- gravitational effects of the Earth tides and ocean tides;
- relativity effects;
- gravitational attractions of the planets; and
- precession and nutation of the Earth's axis.

The various orbit perturbation forces can have different effects on the orbital elements.[5] These effects could fall in one of the following categories:

- *Secular*: linear variation in the orbital elements.
- *Short-period*: periodic variations in the orbital elements, on time scales shorter than the orbital period.
- *Long-period*: periodic variations in the orbital elements, on time scales longer than the orbital period.

If only long-term effects need to be determined, just taking the secular effects into account might suffice, since the periodic variations will average out. Since the accurate in-orbit location of the satellite is needed for this study, however, all three types of effects will be considered.

A detail treatment of the orbit perturbations, with the emphasis on mathematical modelling, will be given in section 4 of this chapter.

3. PERTURBATION METHODS

When a satellite moves under the influence of various orbit perturbation forces, the orbital parameters vary with time. There are no precise analytical solutions to describe the trajectory of the satellite under these circumstances. The techniques of either *special perturbations* or *general perturbations* can however be used to predict the orbit.

3.1 SPECIAL PERTURBATIONS

The methods of special perturbations rely on numerical integration techniques to solve the equations of motion. Cowell's method, Encke's method and the variation of parameters method will be discussed.

3.1.1 Cowell's method

The equations of motion of a satellite moving under the influence of the Earth's gravity field as well as other perturbation forces are given by:

$$\ddot{\mathbf{r}} + \frac{\mu}{r^3}\mathbf{r} = \mathbf{a}_p \qquad \textbf{(2.1)}$$

where $\mathbf{r}$ is the three-dimensional position vector of the satellite (with magnitude r), μ is the Earth's gravitational constant and $\mathbf{a}_p$ is the vector sum of all accelerations due to perturbation forces on the satellite. The direct numerical integration of the above set of equations is known as Cowell's method. Various numerical integration methods are available and can be used in orbit prediction. These include the Runge-Kutta, Adams-Moulton and Gauss-Jackson methods (Fox, 1984).[6]

Cowell's method provides the necessary short-term solutions to the perturbed satellite's position and will be used for orbit determination in this study. The advantages that lead to this choice include its simplicity in formulation and implementation, as well as the fact that no approximations are necessary. The one disadvantage is the relatively slow computation time.

Even though Cowell's method will be used exclusively, the other special and general perturbation methods will be discussed briefly in the following sections for the sake of comparison.

---

[5] Mathematical expressions for the changes in orbital elements due to perturbations can be found in many of the listed references. See for example Larson *et al* (1992).

[6] The numerical integration method chosen for this study will be discussed in chapter 3.

3.1.2 Encke's method

Two orbits are considered for this method: a reference orbit without perturbations and the desired orbit with perturbations. The initial position and velocity in the reference orbit equals that of the orbit with perturbations. The equations of motion in the reference orbit are:

$$\ddot{\boldsymbol{\rho}} = -\frac{\mu}{\rho^3}\boldsymbol{\rho} \tag{2.2}$$

The instantaneous Kepler trajectory (conic section) at the initial time is referred to as the *osculating* orbit. The orbit with perturbations departs from this idealised orbit and the departure can be expressed as:

$$\delta\mathbf{r} = \mathbf{r} - \boldsymbol{\rho} \tag{2.3}$$

where $\mathbf{r}$ satisfies the equation of motion of the perturbed orbit (2.1). By differentiating equation (2.3) twice and using equations (2.1) and (2.2), it follows that:

$$\delta\ddot{\mathbf{r}} = \mu\left(\frac{\boldsymbol{\rho}}{\rho^3} - \frac{\mathbf{r}}{r^3}\right) + \mathbf{a}_p \tag{2.4}$$

For Encke's method, equation (2.4) is integrated numerically to calculate the difference from the osculating orbit. Since the latter is known, the perturbed orbit can be calculated through equation (2.3). The difference $\delta\mathbf{r}$ will be small and slowly varying relative to r and ρ, because the perturbation acceleration $\mathbf{a_p}$ is small relative to the ideal two-body attraction on the satellite. For this reason the time step used for Encke's method can be much larger than that of Cowell's method.

There are certain numerical difficulties in evaluating equation (2.4). Approximations and methods to overcome these difficulties are described in more detail in Battin (1987) and Kaplan (1976). When the departure from the osculating orbit becomes too large, the approximations are no longer valid and a new osculating orbit should be

initialised with position and velocity equal to that of the perturbed orbit at the time. This procedure is called rectification.

The advantage of Encke's method is its larger time steps, thus less computer time. According to Kaplan (1976) the speed advantage of Encke's method over Cowell's method is in the order 2 or 3 to 1 for Earth satellites. Disadvantages include the complex formulation and possible truncation errors when the rectification is not done frequently enough.

3.1.3 Variation of parameters

The methods of Cowell and Encke give solutions for the velocity and position of a satellite in a perturbed orbit, but do not present any insight into the variation of orbital elements under the influence of perturbations. The *variation of parameters* approach aims to do the latter by giving analytical expressions for the rates of change of orbital elements under the influence of perturbation forces. It does not give the position and velocity of the satellite directly and was therefore not considered for orbit determination in this study.

The concept of variation of parameters is also the basis for the methods of general perturbations.

3.2 GENERAL PERTURBATIONS

The methods of general perturbations are primarily concerned with calculating the changes in orbits due to perturbation forces acting on the satellite. Analytical integration of series expansions of the perturbation accelerations are carried out to calculate these changes over long periods of time. The perturbation derivatives are obtained using the variation of parameters method.

The different methods of general perturbations can be classified in the following categories:[7]

- methods employing a reference orbit;
- methods using Keplerian elements;
- methods based on the *Vinti potential*;
- methods using a short power series in eccentricity;
- methods employing an averaging process; and
- methods using rectangular co-ordinates to handle the perturbations.

Further treatment of these methods can be found in most of the referenced works on astrodynamics. The determination of the long-term effects of perturbations on satellite orbits is for most satellite missions far more important than the determination of the short-term effects. The methods of general perturbations are thus normally covered in more detail than the methods of special perturbations.

In this study, the short-term positions of the satellites are required as accurately as possible. The methods of general perturbations do not provide these solutions directly, but rather determine the long-term changes in the shape and position of the orbit due to the various perturbations forces. General perturbation methods were thus not considered any further.

4. MODELLING ORBIT PERTURBATIONS

In order to use Cowell's method to solve the equations of motion for a satellite moving under the influence of perturbation forces, it is necessary to model the three-dimensional vector acceleration due to each perturbation force acting on the satellite.

---

[7] From Kaplan (1976).

4.1 ATMOSPHERIC DRAG

Atmospheric drag is a perturbation force caused by the Earth's upper atmosphere. The acceleration of the spacecraft due to atmospheric drag can be expressed as:

$$\mathbf{a}_{\text{drag}} = -\frac{1}{2}\rho\frac{C_D A}{m}v^2\hat{\mathbf{v}} \tag{2.5}$$

where ρ is the atmospheric density, C_D is the drag coefficient, A is the cross-sectional area of the satellite perpendicular to the velocity vector, m is the mass of the satellite, v is the velocity magnitude of the satellite relative to the atmosphere and $\hat{\mathbf{v}}$ is a unit vector in the direction of the satellite's velocity.

The negative sign indicates that the acceleration is in the opposite direction of the velocity vector. This causes energy to be removed from the orbit and results in a reduction (secular variation) in both the eccentricity and semi-major axis of the orbit. The orbit gets progressively more circular and the altitude decreases. As the altitude decreases, the atmospheric drag force increases, leading to a further decrease in altitude. The satellite eventually falls back to the Earth. The satellite lifetime is an important mission parameter which is directly influenced by the atmospheric drag.

It is interesting to consider the relationship between the velocity magnitude (v) and semi-major axis (a) for a satellite in an unperturbed circular orbit:

$$v = \sqrt{\frac{\mu}{a}} \tag{2.6}$$

The term μ is the Earth's gravitational constant. It is clear that the velocity will increase as the semi-major axis decreases. Normally a decrease in velocity would be expected when a force acts in the opposite direction of the velocity vector. This increase in the velocity magnitude due to the drag force, acting in a direction opposite to the velocity direction, is hence known as the "drag paradox". The phenomenon will be useful later to explain the relative movement between two satellites in the same orbit.

In addition to atmospheric drag, which acts in a direction opposite to the satellite's motion, there is also an aerodynamic lift force acting in a direction perpendicular to the satellite's motion. The value of the lift force depends on the orientation of the satellite. If, for example, the satellite is tumbling end-over-end as it orbits the Earth, the value of the lift will continually change sign and the resultant value would be zero.

The ratio of lift to drag depends on the shape of the satellite. Typically, a spherical shape would have a lower ratio than a disc-shape. According to King-Hele (1964) the value is usually small (< 0.1) and thus the effects of lift can be neglected, even for disc-shaped satellites.

The following paragraphs will discuss certain aspects of the evaluation of the various terms in equation (2.5). Evaluation of the cross-sectional area perpendicular to the velocity vector (A) and the drag coefficient (C_D) will be treated in chapter 4, when the nominal satellite configuration is explained in more detail.

4.1.1 Evaluating the atmospheric density

The atmospheric density depends on the physical properties of the Earth's upper atmosphere and is probably the most uncertain term in evaluating the acceleration due to atmospheric drag. The most significant spacial variation in the atmospheric density comes from the dependency on altitude. The temporal variations in density are dominated by the effects of solar activity on the structure of the Earth's upper atmosphere.

A variety of sophisticated models to predict the spacial and temporal variations in the density have been developed through the past forty years. Measured data of the upper atmosphere is insufficient to allow purely empirical models and the physical processes in the upper atmosphere are not understood well enough to allow purely theoretical models. For these reasons, the most sophisticated models are combinations of empirical and theoretical work. One of the most commonly used models is the CIRA 72 model (*COSPAR International Reference Atmosphere 1972*). It includes the altitude range from 25 km to 2500 km. The CIRA 72 model takes into account the diurnal variation in density, variations due to the 27 day Sun rotational period, annual

variations and the 11 year solar cycle. Other empirical density models include the ARDC 1959 model (static empirical model based on observations of early satellites), the US Standard Atmosphere of 1962 (static, idealised, middle-latitude, year-round mean over the range of sunspot minima to sunspot maxima), the Jacchia 1964 model (dynamic) and the empirical Mass Spectrometer Incoherent Scatter (MSIS) model.

Since the solar activity is not entirely predictable, the value for the atmospheric density remains at best a good approximation. According to Wertz (1978) the accuracy of current models is ±50%.

All the above mentioned models are complex to implement and a simplified analytical model of the atmospheric density is desirable for the purposes of this study. An important consideration towards this end is the time scale of interest. When considering the required time for which the density must be modelled, certain important simplifications can be made. These simplifications pertain to the spacial as well as temporal variations in the density.

Consider firstly the spacial dependency of the atmospheric density, dominated by the variation with altitude. It will be shown in chapter 4 that the typical altitude reduction during the constellation acquisition phase is very small (less than 1%). Consequently, over this small altitude range, the *scale height*[8] can be assumed constant. This leads to the following exponential model:

$$\rho = \rho_{ref} \exp\left[-\frac{h - h_{ref}}{H_s}\right] \tag{2.7}$$

where ρ_{ref} is the atmospheric density at the reference point. The reference point is the initial perigee point in the case of an elliptical orbit. For a circular orbit, the reference point is anywhere on the orbit at the initial altitude. The terms h_{ref} and h are the altitudes of the reference and evaluation points respectively. H_s is the constant scale height. The value of H_s is given by Wertz (1978) as 62.2 km at the nominal altitude

---

[8] See King-Hele (1987) for a definition of scale height.

of 450 km.[9] According to King-Hele (1987), the error due to the approximation of constant scale height is negligible (< 0.1%), as long as the deviation from the altitude of the initial reference point ($h - h_{ref}$) is less than 100 km.

The uncertainty in atmospheric density lies not so much with the spacial variation, but with the temporal variation. The latter can be accounted for by modelling ρ_{ref} as a time-dependent function. It is thus necessary to consider the possible time-variations in atmospheric density in detail. The variations are a result of changes in the structure of the Earth's upper atmosphere. These changes are caused by complex physical processes which are not fully understood, but, as mentioned earlier, are mostly solar-activity related.[10] The various influences can be grouped according to their characteristic time scales (see King-Hele, 1969):

- Long-term sunspot cycle: The density can vary with an order of magnitude between a minimum phase and a maximum phase. At an altitude of 500 km, a variation up to a factor 20 has been recorded. The period of the sunspot cycle is in the order of 10 years.

- Semi-annual variations: This can cause changes up to a factor 3 at a 500 km altitude. Maxima are usually during early April and late October and minima during mid-January and late July.

- The 27-day Sun rotation period: This causes periodic changes in the solar radiation which heats the upper atmosphere. Typical changes in atmospheric density are in the order of a factor 2 or 3 at an altitude of 500 km.

- Short-term solar activity (solar flares): The upper atmosphere is heated by streams of particles from the Sun. Typically, a factor 3 variation is possible over periods of 5 days for a 650 km altitude. At lower altitudes, the change is smaller.

---

[9] This value will be used throughout the study.

[10] A discussion of some physical aspects can be found in Langton (1969).

- Diurnal variations: These changes are related to the local time or the zenith angle of the Sun. The value of the atmospheric density has a peak or "day-time bulge" on the illuminated side of the Earth and a trough on the dark side. The change can be up to a factor 3 at an altitude of 450 km.

Chapter 4 will show that the typical constellation acquisition phase for the nominal satellite and orbital configuration has a duration in the order of a few days. When comparing the characteristic magnitudes and time scales of the density variations with this control time, it is clear that only the last two types of variations, namely diurnal variations and variations due to solar flares (short-term solar activity), will have a significant effect on the density during the application time of the model. As far as the 11-year, semi-annual and even the 27-day variations are concerned, the density can be assumed constant during the relatively short time of the constellation acquisition phase. Note, however, that the constellation maintenance phase would account for the largest part of the satellite's lifetime and the long-term variations will have to be considered in this case.

Consider now a model for the diurnal variation in density. Note that the day-time bulge in density is not directly underneath the Sun, but lags behind due to the Earth's rotation. The angular difference is about 30° in longitude. The latitude of the bulge equals the Sun's declination. The variation in the atmospheric density at the reference point (ρ_{ref}) can be modelled as a sinusoidal distribution, dependent on the angular distance (ϕ) from the centre of the day-time bulge:

$$\rho_{ref} = \rho_0\left(1 + F\cos(\phi)\right) \tag{2.8}$$

The term ρ_0 is the atmospheric density at a reference point at an angle of $\phi = 90°$. F is a constant factor, approximately equal to 0.7 for the nominal orbits (King-Hele, 1987). Even though this sinusoidal model is only an approximation, it still stays within 20% of the values predicted by the CIRA 72 model. It will also be shown later that the diurnal variations in density average out per orbit so that the relative positions of the satellites in a constellation are not influenced significantly. The model of equation

(2.8) does however provide better results than the case if the diurnal variation is ignored altogether.

The variations due to solar flares can be characterised, but cannot be predicted. The time scale, magnitude and form of the density variation caused by a typical solar flare will be discussed in chapter 4. The characterisation will be done through the time-dependent term $\rho_v(t)$ in the following model of the reference point density ρ_0 in (2.8):

$$\rho_0 = \rho_{nom} + \rho_v(t) \tag{2.9}$$

The term ρ_{nom} is the nominal atmospheric density. Wertz (1978) lists a mean value of 1.585×10^{-12} kg.m^{-3} at an altitude of 450 km. This value will be used throughout the study.

The later philosophy will be to design control strategies that are robust to the changing operating conditions as far as possible, using a real-time density estimation scheme. When the limits of robustness are exceeded, the possible errors will be quantified. The same density-estimation scheme will be used during the long-term constellation maintenance phase where all temporal variations must be taken into account. These long-term variations in density will also be characterised through the function $\rho_v(t)$ in equation (2.9).

To summarise: the density model used for the remainder of this study has an exponential altitude dependence (2.7), with the density at the reference point (ρ_{ref}) given by the combination of (2.8) and (2.9). The resulting model is:

$$\rho = \left(\rho_{nom} + \rho_v(t)\right)\left(1 + F\cos(\phi)\right)\exp\left[-\frac{h - h_{ref}}{H_s}\right] \tag{2.10}$$

All temporal variations in the reference density, over and above the diurnal variation, will be characterised through the function $\rho_v(t)$. When $\rho_v(t) = 0$, only the diurnal variation is included.

4.1.2 Evaluating the relative speed

The velocity of the satellite relative to the atmosphere is the vector difference between the velocity of the satellite relative to the Earth's centre and the velocity of the atmosphere relative to the Earth's centre. The movement of the atmosphere relative to the Earth is dominated by the west-to-east rotation, which greatly exceeds the meridional (north-south) winds. Even so, the east-west rotation remains a small effect in comparison with the other perturbation forces and will be ignored for this study. This approximation is also justified since the effects of atmospheric rotation would be nearly identical for any two satellites in the typical tandem constellation and the influence on their relative positions would be minimal.

4.2 NON-SPHERICAL GRAVITATIONAL FIELD OF THE EARTH

In deriving the ideal elliptical Keplerian orbit, it is assumed that the Earth is spherical and homogeneous. For the purposes of accurate orbit determination, this assumption is no more valid. The major deviations from this idealised representation are a bulge at the equator, a flattening at the poles and a slight pear-shape. This asymmetrical mass distribution of the Earth causes its gravitational field to deviate from the ideal spherical model and it causes periodic variations in all the orbital elements. Additionally, the right ascension of the ascending node and the argument of the perigee undergo secular variations. These secular changes dominate the variations in the orbital elements and are caused by the oblate shape of the Earth.

The most convenient way to account for the deviation of the Earth's gravitational field from the perfect sphere is to model the Earth's gravitational potential as a spherical harmonics expansion (Battin, 1987):

$$U = \frac{\mu}{r}\left\{1+\sum_{n=2}^{\infty}\left[\left(\frac{R_E}{r}\right)^n J_n P_{n0}\cos\Phi + \sum_{m=1}^{n}\left(\frac{R_E}{r}\right)^n \left(C_{nm}\cos m\Lambda + S_{nm}\sin m\Lambda\right)P_{nm}(\cos\Phi)\right]\right\} \tag{2.11}$$

In equation (2.11), U is the gravitational potential at a distance r from the centre of the Earth and Φ and Λ designates the latitude and longitude respectively. P_{nm} are Legendre polynomials. J_n, C_{nm} and S_{nm} are functions of the mass distribution of the Earth. The J_n terms are the *zonal* harmonic coefficients, reflecting the mass distribution independently of longitude. C_{nm} and S_{nm} are the *tesseral* harmonic coefficients for $n \neq m$ and *sectoral* harmonic coefficients for $n=m$.

The value of the J_2 coefficient is about three orders of magnitude larger than all the other coefficients and thus dominates the gravitational perturbation influences of the Earth. It represents the equatorial bulge (oblateness) of the Earth. Neglecting all but this term and taking the gradient of the scalar potential function yields the vector perturbation acceleration on the satellite:

$$\mathbf{a}_{J_2} = \frac{-3\mu R_E^{\,2} J_2}{2r^7}\begin{bmatrix} r_x\left(r_x^{\,2} + r_y^{\,2} - 4r_z^{\,2}\right) \\ r_y\left(r_x^{\,2} + r_y^{\,2} - 4r_z^{\,2}\right) \\ r_z\left(3r_x^{\,2} + 3r_y^{\,2} - 2r_z^{\,2}\right) \end{bmatrix} \qquad \textbf{(2.12)}$$

where r_x, r_y, and r_z are the components of the satellite position $\mathbf{r}$ along the x, y and z axes respectively. The perturbations due to the higher order terms in equation (2.11) are also small relative to the other perturbation forces that were included for the nominal satellite and orbital configuration. The error due to their omission is insignificant over the duration of the typical constellation acquisition phase.

4.3 THIRD BODY ATTRACTIONS

The "term third" body refers to any other body in space, besides the Earth, which could have a gravitational influence on the satellite. These third body gravitational attractions cause periodic variations in all the orbital elements, as well as secular variations in the right ascension of the ascending node, argument of the perigee and mean anomaly.

The perturbation acceleration due to the gravitational attraction of a third body can be calculated as follows (Battin, 1987):

$$\mathbf{a_d} = -\frac{\mu_d}{r_{ds}^3}\left(\mathbf{r_s} + f(q)\mathbf{r_d}\right) \tag{2.13}$$

The term μ_d is the gravitational parameter of the third body and:

$$q = \frac{\mathbf{r_s}\cdot\left(\mathbf{r_s} - 2\mathbf{r_d}\right)}{\mathbf{r_d}\cdot\mathbf{r_d}}$$
$$f(q) = q\frac{3 + 3q + q^2}{1 + \left(1+q\right)^{\frac{3}{2}}} \tag{2.14}$$

The definitions of the vectors are given in figure 2.1.

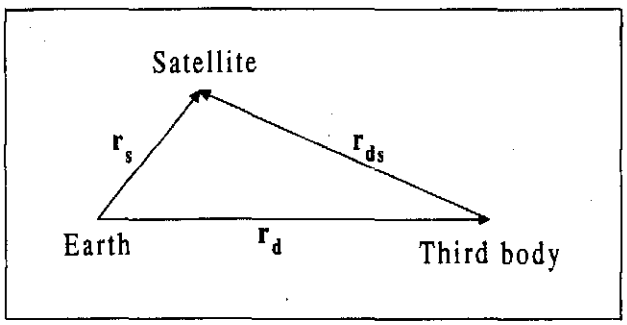


Figure 2.1: Vector definitions for third body attractions.

It is clear that the magnitude of the third body attraction depends on its gravitational parameter and distance from the satellite. The most significant influences for the LEO satellite come from the Sun and the Moon, exceeding the influence of the largest planet (Jupiter) by approximately six orders of magnitude.[11] Even so, these influences are still very small relative to the oblateness and drag perturbations at low altitudes (accelerations in the order of 10^{-6} m.s^{-2} at an altitude of 450 km). The effects do however increase for higher altitudes and will be included in all simulations.

---

[11] Fortescue *et al* (1991).

For simulations with third body perturbations included, it is necessary to have the position of the Sun and the Moon. The Sun is modelled as an Earth satellite in a circular orbit with a radius of one astronomical unit and a period of 365.26 days. The inclination of the orbit is 23.439°. The Moon is modelled as an unperturbed Earth satellite in an orbit with an eccentricity of 0.055, semi-major axis of 384000 km and inclination of 23°.

4.4 SOLAR RADIATION PRESSURE

Solar radiation pressure is a force on the satellite due to the momentum flux from the Sun and is thus only present when the Sun is visible from the satellite (it could be obscured by the Earth). The perturbation force causes periodic variations in all the orbital elements. The magnitude of the acceleration due to solar radiation pressure on the satellite is given by:

$$a_{Solar} = KP\frac{A_s}{m} \tag{2.15}$$

where K is a dimensionless constant between 1 and 2 (K=1: surface perfectly absorbent; K=2: surface reflects all light), P is the momentum flux from the Sun, A_s is the cross-sectional area of the satellite perpendicular to the sun-line and m is the mass of the satellite. The mean value of P is approximately 4.4×10^{-6} kg·m^{-1}·s^{-2} at the distance of the Earth from the Sun (Wertz, 1978). The direction of the acceleration is radially away from the Sun.

The solar radiation pressure is a very small force relative to the drag and J_2 perturbations and will be assumed constant during the times that the satellite is illuminated by the Sun. This constant value will correspond to the worst possible case where the cross-sectional area perpendicular to the sun-line (A_s) equals the maximum possible cross-sectional area of the satellite (A_{max}).

Chapter 3

HIGH-PRECISION ORBIT GENERATION

1. INTRODUCTION

This chapter describes the orbit propagation software that has been developed to provide accurate solutions to the equations of motion of a LEO satellite, moving under the influence of the various perturbation forces. The orbit propagation software is an essential part of this study, but has been designed and implemented in such a way that it can also be used outside the current scope.

Although the source code will not be discussed in detail here, the underlying data structures and functionality should become clear from the following sections. Commented listings of the orbit propagation source code is provided in appendix B.

2. SOFTWARE DESIGN CONSIDERATIONS

2.1 PROGRAMMING LANGUAGE

The criteria used to choose a programming environment for the orbit propagator were mainly the execution speed and the ease of implementation. The programming

environment also had to be flexible, allowing for quick modifications to the simulations, especially for the design and testing of control strategies.

With these criteria in mind, three different types of computer packages were considered to implement the orbit propagator:

- a programming language compiler (Borland Pascal[1]);
- a mathematical interpreter (MATLAB[2]); and
- block diagram simulators.

Each different type of implementation has its own set of advantages and disadvantages. The Borland Pascal (version 7.0) programming language was chosen for the following advantages over the other types of implementation:

- Execution speed (MATLAB is relatively slow).
- Easy implementation of complex mathematical equations (difficult in block diagram simulators).

Borland Pascal also allows for quick modifications to the simulations.

2.2 NUMERICAL INTEGRATION

2.2.1 Integration method

Various integration methods have been implemented for the solution to differential equations governing the movement of bodies in the solar system. Fox (1984) presents results of a comparison between some newly developed methods and the more traditional methods. The standard fourth-order Runge-Kutta method performs satisfactorily with a sufficiently small time step. It is also easy to implement and was hence chosen for this study. The error propagation is not a serious problem in this context, since no long-term solutions are needed.

---

[1] Copyright © (1983, 1992) by Borland International, Inc.

[2] Copyright © (1984 - 1994) by the MathWorks, Inc.

Consider the first-order differential equation:

$$\frac{dx}{dt} = f(x,t) \tag{3.1}$$

where $x = x_0$ at $t = t_0$. The Runge-Kutta method estimates the value of x at $t = t_1 = t_0+\delta$, denoted as x_1, as follows:

$$x_1 = x_0 + \frac{\delta}{6}[A+2B+2C+D] \tag{3.2}$$

where δ denotes the fixed time step. The values of A, B, C and D are given by:

$$\begin{aligned} A &= f(x_0,t_0) \\ B &= f\left(x_0+\tfrac{1}{2}A, t_0+\tfrac{1}{2}\delta\right) \\ C &= f\left(x_0+\tfrac{1}{2}B, t_0+\tfrac{1}{2}\delta\right) \\ D &= f(x_0+C, t_0+\delta) \end{aligned} \tag{3.3}$$

For the perturbed satellite, the equations of motion are given by (from chapter 2):

$$\ddot{\mathbf{r}} + \frac{\mu}{r^3}\mathbf{r} = \mathbf{a}_p \tag{3.4}$$

To implement the Runge-Kutta methods, the equations of (3.4) are reduced to the following set of first order differential equations:

$$\begin{aligned} \dot{\mathbf{r}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= -\frac{\mu}{r^3}\mathbf{r} + \mathbf{a}_p \end{aligned} \tag{3.5}$$

Note that both the velocity and position are three-dimensional vectors so that the total number of first-order differential equations is 6.

2.2.2 Time step choice

The smallest characteristic time scale in the total system comes from the Earth's oblateness effects – periodic variations at 4 times the orbital rate. Since only short-term solutions are of interest, the time step can be chosen orders of magnitude smaller than this characteristic time scale to allow for great accuracy, while the simulation time should remain practical. Choosing a factor 50 leads to 200 steps per orbit. This time step (approximately 28 seconds for the nominal circular orbit and 29 seconds for the nominal elliptical orbit) has been used for all simulations.

To verify the accuracy with the chosen time step, the results of the numerical solutions with different time step choices were compared. Each time, the trajectory of a single perturbed satellite was propagated over 100 orbits, which is the typical duration of the constellation acquisition phase. The error propagation can be measured by inspecting the position of the satellite at the end of each simulation. Figure 3.1 shows the results in the x-z plane. The figure "N" indicates the number of step per orbit, thus a larger N implies a smaller time step.

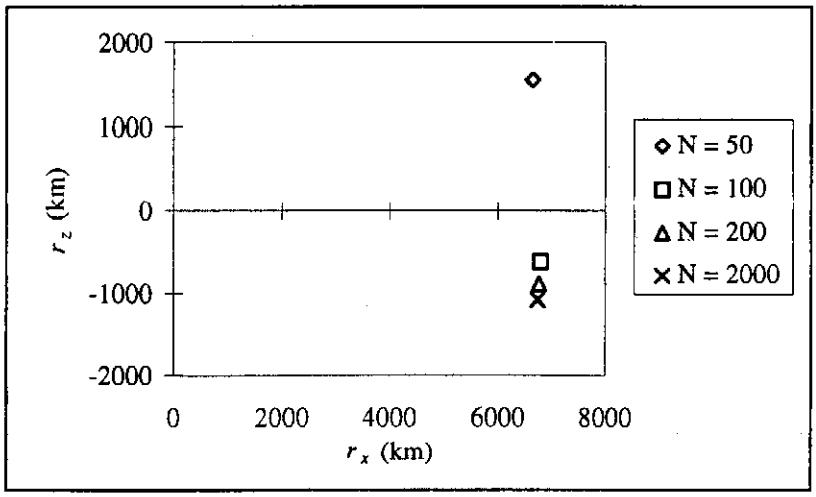


Figure 3.1: Runge-Kutta 4 time step evaluation.

It is clear that the final position does not change significantly if N increases ten-fold, from 200 to 2000. If N is halved to 100, the difference is still relatively small, but for smaller values of N, the final position starts to vary drastically. It can be deduced that a value of N = 200 is sufficiently, but not unnecessarily small. The existing difference in the final position between the N=200 and N=2000 cases is tolerable, since the simulations will be done primarily to determine control dynamics and not for the sake

of propagation studies. If orbit propagation studies are the goal, N can be increased by orders of magnitude to ensure high accuracy.

For the chosen time step, the typical simulation time on a pentium-60 processor, for a single satellite with all the significant perturbations (from chapter 2) included, is in the order of 80 seconds per 100 orbits.

3. SOFTWARE STRUCTURE AND FUNCTIONALITY

The orbit propagator involves a main program (*ORBIT.PAS*) and a number of Pascal *units*. The unit structure is displayed in figure 3.2. It does not show all the inter-connections between the different units, but should serve to explain the general hierarchy and structure.

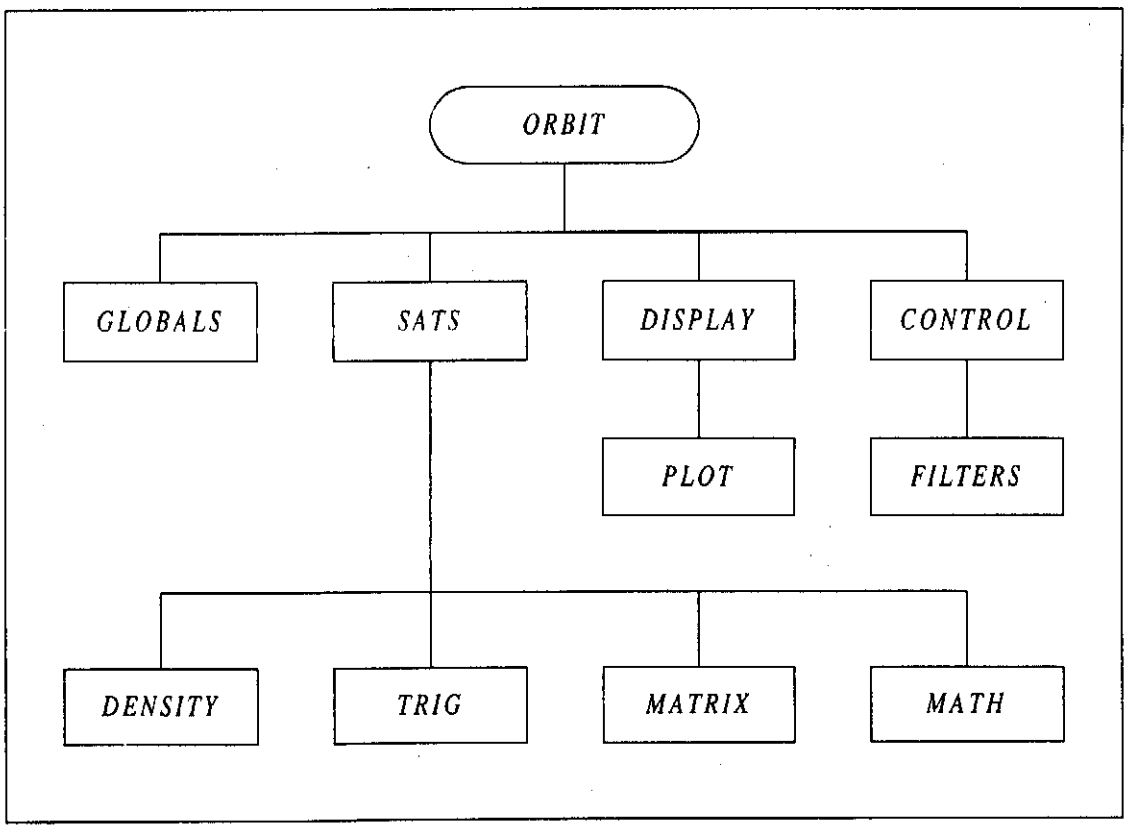


Figure 3.2: Software structure (units).

The various components in figure 3.2 will now be treated.

3.1 THE MAIN PROGRAM (PROGRAM *ORBIT*)

The main program forms the centre of each simulation. Figure 3.3 gives a top-level flowchart.

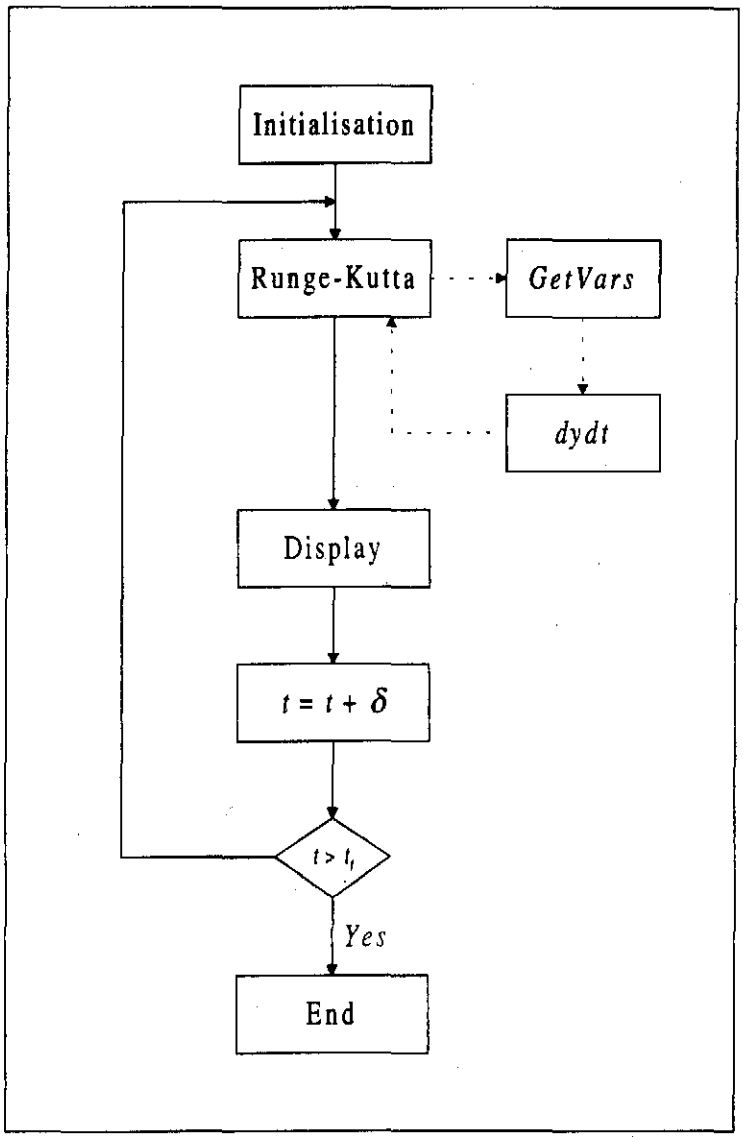


Figure 3.3: Simulation flow chart.

The initialisation routine sets up the simulation and calls the relevant initialisation routines from the other units. The orbital period (T_0) is calculated during the initialisation phase from the satellite's mean motion (n):

$$n = \sqrt{\frac{\mu}{a^3}}$$
$$T_0 = \frac{2\pi}{n} \qquad \textbf{(3.6)}$$

After the initialisation, the main simulation loop follows. From this loop, all other relevant functions and procedures are called. The *Runge-Kutta* block in figure 3.3 refers to the numerical integration process, taking place in the *SATS* unit. This process makes repeated calls to the procedures *GetVars* and *dydt*, to be discussed in section 4.2.2.

The *Display* block represents the routines that generate the graphical display of results. It will be discussed in section 3.4. If constellation control (acquisition or maintenance) is done, a call to the control routine(s) will also be included in the main loop.

Finally, the time variable (t) is incremented by the time step (δ) and the loop is terminated when the final time (t_f) is reached.

3.2 GLOBAL CONSTANTS AND VARIABLES (UNIT *GLOBALS*)

The *GLOBALS* unit is accessible to most other units and defines a number of global constants and variables. The constants include:

- nominal satellite properties (mass, minimum and maximum cross-sectional areas, drag coefficient etc.);
- nominal orbit properties (Kepler parameters) for the initialisation of satellites;
- Sun, Moon and Earth constants (gravitational constants, solar flux constant, orbit parameters etc.); and
- simulation constants (for example the number of steps per orbit, which perturbations to include etc.).

The global variables defined in the unit include:

- the time variable;
- the time step variable;
- the orbital period; and
- the distance between satellites in a constellation.

3.3 ORBIT PROPAGATION (UNIT *SATS*)

This unit defines a number of objects, containing the core of the orbit propagation process. These objects, with their associated data structures and methods will be treated in section 4 of this chapter.

The *SATS* unit also makes use of some other units, to be discussed in the following sections.

3.3.1 Unit DENSITY

This unit contains the function to model the atmospheric density at any altitude and at any instant in time. From the previous chapter:

$$\rho = \left(\rho_{nom} + \rho_v(t)\right)\left(1 + F\cos(\phi)\right)\exp\left[-\frac{h - h_{ref}}{H_s}\right] \tag{3.7}$$

To apply the model, the term $\cos(\phi)$ in (3.7) is required. It is calculated by using the following algorithm:

1. Get the unit vector ($\hat{\mathbf{s}}$) from the origin of the co-ordinate system to the Sun.
2. Rotate $\hat{\mathbf{s}}$ eastwards by 30°, to get the unit vector to the centre of the day-time bulge ($\hat{\mathbf{b}}$).
3. Get the unit vector ($\hat{\mathbf{r}}$) from the origin of the co-ordinate system (as defined in chapter 1) to the satellite.
4. Take the dot product between $\hat{\mathbf{b}}$ and $\hat{\mathbf{r}}$ to obtain $\cos(\phi)$.

3.3.2 Unit TRIG

The Pascal package used provides only the following trigonometric functions:

- $\sin(x)$
- $\cos(x)$

- $\tan^{-1}(x)$

The *TRIG* unit realises some additional trigonometric functions from these, all necessary for the orbit propagation process:

- function *tan(x)*, computed as:

$$\tan(x) = \frac{\sin(x)}{\cos(x)} \tag{3.8}$$

- function *arcsin(x)*, computed as:

$$\sin^{-1}(x) = \tan^{-1}\left(\frac{x}{\sqrt{1-x^2}}\right) \tag{3.9}$$

- function *arccos(x)*, computed as:

$$\cos^{-1}(x) = \tan^{-1}\left(\frac{\sqrt{1-x^2}}{x}\right) \tag{3.10}$$

- function *atan2(num, den)*, giving the four-quadrant arc-tangent of the quotient of *num* and *den*.

- procedure *mod180(angle)*, giving the value between -180° and 180° for any angle.

3.3.3 Unit MATRIX

The following vector- and matrix-related functions are contained in the unit *MATRIX* and are used in the simulations:

- function *Vec3Mag*, computing the magnitude of a three-dimensional vector; and
- procedure *Vec3Diff*, computing the vector difference of two three-dimensional vectors.

3.3.4 Unit MATH

The only function used in this unit is the function *pw*, which determines the value of x to the power of y (x^y), where x and y can be arbitrary real numbers. The following formula is used:

$$x^y = e^{y\ln(x)} \tag{3.11}$$

3.4 DISPLAYING THE RESULTS (UNIT *DISPLAY*)

This unit contains routines to initialise and display the results of the simulations in a graphical format. Any combination of variables can be displayed during the simulation process on a set of up to four simultaneous graphs.

The *DISPLAY* unit makes use of the *PLOT* unit to generate the graphs.

3.4.1 Unit PLOT

The *PLOT* unit defines an object *TGraph*, with the functionality to generate a graph on the screen. Every instance of the object is a different graph. The two main methods of *TGraph* are:

- the *Init* procedure, initialising the graph position, scale of the axes etc.; and
- the *PutPoint* procedure, to plot any point (x, y) on the graph.

3.5 CONSTELLATION CONTROL ROUTINES (UNIT *CONTROL*)

The *CONTROL* unit contains all routines related to the constellation control process, including the acquisition and maintenance phases. The unit also makes use of the *FILTERS* unit.

A discussion of these units can only be relevant once the appropriate theory is covered and will therefore be postponed until chapter 4.

4. ORBIT PROPAGATION OBJECTS

Extensive use was made of Borland Pascal's object-handling capabilities. The objects were defined to represent real physical objects, thus providing easy visualisation and understanding. All the objects necessary for the accurate orbit determination are contained in the unit *SATS*. These objects are:

- *TKepSat*: object representing an unperturbed (ideal) Kepler satellite;
- *TSat*: object derived from *TKepSat*, representing a perturbed satellite;
- *TSun*: object representing the Sun.

Each object has an associated set of variables and a number of methods. The variables define the data structure associated with the object. The methods are the functions and procedures of the object, defining and realising its functionality. The essential data contained in each of the objects includes a three-dimensional position vector. The essential methods are the initialisation method (procedure *Init*) and a method that propagates the position one time-step ahead (procedure *NextPosition*). These two methods define the "interface" through which the object is accessed.

4.1 THE KEPLER SATELLITE (OBJECT *TKEPSAT*)

The *TKepSat* object represents a satellite in an unperturbed (ideal) Kepler orbit. The necessity for this object is threefold:

- the Moon is modelled as an ideal Earth satellite and is implemented as an instance of the *TKepSat* object;
- the perturbed satellite object (*TSat*) is derived from the *TKepSat* object; and
- the ability to model unperturbed satellites can be useful for applications outside the scope of this study.

4.1.1 Data structure

The variables contained in a *TKepSat* object include the orbital parameters and the position and velocity vectors. This means that every instance of this object represents

a different satellite, in a potentially different orbit and position. Note that the orbital parameters will stay constant for this unperturbed situation.

4.1.2 Initialisation (procedure Init)

The initialisation procedure sets up the orbital parameters and the initial position and velocity vectors.[3] To calculate the initial position and velocity vectors, the *semilatus rectum* (p) is first calculated:

$$p = a\left(1 - e^2\right) \tag{3.12}$$

From this, the magnitude of the position vector (r) can be calculated:

$$r = \frac{p}{1 + e\cos\left(\theta_0\right)} \tag{3.13}$$

The position vector follows:

$$\begin{aligned} r_x &= r\left[\cos\left(\omega + \theta_0\right)\cos(\Omega) - \sin\left(\omega + \theta_0\right)\sin(\Omega)\cos(i)\right] \\ r_y &= r\left[\cos\left(\omega + \theta_0\right)\sin(\Omega) + \sin\left(\omega + \theta_0\right)\cos(\Omega)\cos(i)\right] \\ r_z &= r\left[\sin\left(\omega + \theta_0\right)\sin(i)\right] \end{aligned} \tag{3.14}$$

where r_x, r_y, r_z are the components of the position vector along the x, y and z axes respectively. To calculate the initial velocity, the magnitudes of the radial and normal components of the velocity vector at epoch (v_{r0} and v_{n0}) are calculated first:

$$\begin{aligned} v_{r0} &= \sqrt{\frac{\mu}{p}}e\sin\left(\theta_0\right) \\ v_{n0} &= \sqrt{\frac{\mu}{p}}\left(1 + e\cos\left(\theta_0\right)\right) \end{aligned} \tag{3.15}$$

---

[3] The mathematics of this section is derived from the theory of the ideal Kepler orbit. See for example Chobotov (1991) or Wertz (1978).

From this follows the velocity magnitude (v) and flight angle at epoch (γ_0):

$$\begin{aligned} v &= \sqrt{{v_{r0}}^2 + {v_{n0}}^2} \\ \gamma_0 &= \text{atan2}(v_{r0}, v_{n0}) \end{aligned} \tag{3.16}$$

The function *atan2* is a four-quadrant arc-tangent. The initial velocity is finally computed as:

$$\begin{aligned} v_x &= v\left[\cos\left(\omega + \theta_0 + \frac{\pi}{2} - \gamma_0\right)\cos(\Omega) - \sin\left(\omega + \theta_0 + \frac{\pi}{2} - \gamma_0\right)\sin(\Omega)\cos(i)\right] \\ v_y &= v\left[\cos\left(\omega + \theta_0 + \frac{\pi}{2} - \gamma_0\right)\sin(\Omega) + \sin\left(\omega + \theta_0 + \frac{\pi}{2} - \gamma_0\right)\cos(\Omega)\cos(i)\right] \\ v_z &= v\left[\sin\left(\omega + \theta_0 + \frac{\pi}{2} - \gamma_0\right)\sin(i)\right] \end{aligned} \tag{3.17}$$

4.1.3 Kepler orbit propagation (procedure NextPosition)

Kepler's equation relates the mean and eccentric anomalies (M and E):

$$M = E - e\sin(E) \tag{3.18}$$

This equation can be solved by a variety of methods (see for instance Battin, 1987) to find E at any time. A simple iterative method is used for this study. Firstly, the mean anomaly at time t is calculated from the mean anomaly at epoch (M_0) and the mean motion (n):

$$M(t) = M_0 + n(t - t_0) \tag{3.19}$$

The time t_0 is the epoch time and equals zero for all simulations.

Using $M(t)$, the value of E is now determined by the following iteration:

$$E_i = E_{i-1} + \frac{M + e\sin(E_{i-1}) - E_{i-1}}{1 - e\cos(E_{i-1})} \tag{3.20}$$

The starting value is $E_0 = M$ and the iteration is continued until the correction term becomes smaller than 1×10^{-8}. This corresponds to a theoretical accuracy of less than 10 cm in a 450 km circular orbit to prevent the propagation of the iteration error.

From the value of E follows the true anomaly (θ) and the magnitude of the position vector (r):

$$\tan\left(\frac{\theta}{2}\right) = \sqrt{\frac{1+e}{1-e}}\tan\left(\frac{E}{2}\right)$$
$$r = a(1 - e\cos(E)) \tag{3.21}$$

The same procedures as in the initialisation are then used to compute the position and velocity vectors from θ and r. The position of the sub-satellite point is also mapped to longitude and latitude:

$$Lon = \text{atan2}(r_y, r_x) - \frac{2\pi(t - t_0)}{S_{day}}$$
$$Lat = \text{atan2}\left(r_z, \sqrt{r_x^2 + r_y^2}\right) \tag{3.22}$$

where t_0 is epoch time and S_{day} is the value of one sidereal day.[4]

4.1.4 Determining Sun illumination (function InSun)

The *TKepSat* object also has a method to determine whether the Sun is visible from the satellite. The Sun can either be visible or obscured by the Earth at any instant in

---

[4] A sidereal day is defined as one complete revolution of the Earth (360°), relative to the fixed stars. The value equals 86164.0918 seconds.

time. This knowledge is necessary to model the perturbation effect due to solar radiation pressure.

Figure 3.4 explains the method used to determine Sun visibility. Note the approximation that the Sun is infinitely far away so that the shade borders are parallel. Angles are measured clockwise from the sun vector. The satellite is positioned at the angle ϕ.

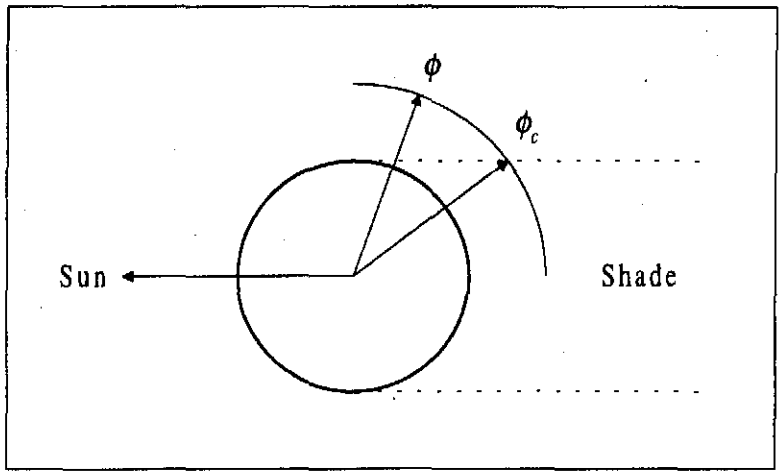


Figure 3.4: Geometry for the* InSun *routine.

The angle between the Sun- and the satellite position vectors (ϕ) is compared to a critical angle (ϕ_c) where the satellite orbit enters the shade. When $\phi < \phi_c$ then the Sun is not obscured by the Earth. The angles are given by:

$$\phi = \cos^{-1}\left(\frac{r_{Sx}r_x + r_{Sy}r_y + r_{Sz}r_z}{r_S r}\right)$$
$$\phi_c = \pi - \tan^{-1}\left(\frac{R_E}{\sqrt{r^2 - R_E{}^2}}\right) \qquad \textbf{(3.23)}$$

where R_E is the Earth's radius and (r_{Sx}, r_{Sy}, r_{Sz}) is the Sun's position vector (with magnitude r_S).

4.2 THE PERTURBED SATELLITE (OBJECT *TSAT*)

As mentioned earlier, the object for the satellite with perturbations is derived from the Kepler satellite object. This gives the two objects the same structure. Some variables and methods are added to include the various perturbations. The initialisation (*Init*) and orbit propagation (*NextPosition*) procedures are also overridden.

4.2.1 Data structure

TSat inherits the position and velocity vectors, as well as the orbital parameters from *TKepSat*. The latter will now change over time as the perturbations continually influence the satellite movement and change the instantaneous shape of the orbit.

Additional variables include the cross-sectional area perpendicular to the velocity vector, perturbation acceleration vectors and a six-dimensional integration vector. The latter is used to store the first derivatives of the three-dimensional position- and velocity vectors for use in the numerical integration scheme.

4.2.2 Orbit propagation (procedure NextPosition)

This procedure contains the fourth-order Runge-Kutta numerical integration algorithm (3.2), used to solve the differential equations of motion of the satellite. From this procedure, calls are made to the *dydt* procedure, each time calculating the six derivatives of the equations of motion (3.5).

To calculate the derivatives, the vector sum of the accelerations due to the various perturbation forces on the satellite ($\mathbf{a_p}$) is needed. This vector is calculated in the *GetVars* procedure. The following perturbations are modelled, as described in the previous chapter:

- atmospheric drag;
- Earth oblateness (J_2) perturbations;
- Sun gravitational attraction;
- Moon gravitational attraction; and
- solar radiation pressure.

Finally, the sub-satellite longitude and latitude are also calculated from the new satellite position, as was done for the Kepler satellite.

4.3 THE SUN (OBJECT *TSUN*)

The position of the Sun is necessary to compute the following perturbation-related effects:

- Sun-gravity effects on the satellite;
- solar radiation pressure; and
- the diurnal bulge in atmospheric density.

The Sun's position is modelled by the *TSun* object. The only data contained in the object is the three-dimensional position vector of the Sun. The only two methods are an initialisation (*Init*) procedure and the usual *NextPosition* procedure. The object is initialised by giving the initial Sun position vector. The Sun's trajectory is modelled as a circular orbit around the Earth, with an inclination (i_S) of 23.439°, radius (r_S) of 1 astronomical unit[5] and period of one sidereal year (S_{year}):[6]

$$\begin{aligned} r_{Sx} &= r_S \cos\left(\frac{2\pi(t-t_0)}{S_{year}}\right) \\ r_{Sy} &= r_S \sin\left(\frac{2\pi(t-t_0)}{S_{year}}\right)\cos(i_S) \\ r_{Sz} &= r_S \sin\left(\frac{2\pi(t-t_0)}{S_{year}}\right)\sin(i_S) \end{aligned} \qquad \textbf{(3.24)}$$

The terms r_{Sx}, r_{Sy}, and r_{Sz} are the components of the Sun's position vector.

---

[5] The astronomical unit (or AU) is defined as the length of the semi-major axis of the Earth's orbit around the Sun. The value is 1.495 978 70 x 10^{11} metres.

[6] The sidereal year is defined as the orbital period of the Earth relative to the fixed stars. The value is 3.155 814 945 8 x 10^7 seconds (about 365.26 days).

5. SIMULATIONS

The orbit propagation software that was described in this chapter will be used in simulations to test the various control strategies that will be introduced later. The software implementation of the various control strategies will also be discussed later.[7] The nominal orbital configuration, satellite and initial conditions will be assumed for each simulation, unless stated otherwise.[8]

The purpose of the simulations is to represent a practical satellite constellation. The solutions from the numerical integration will therefore be treated as "sampled" values and the terms "time instant" and "sampling instant" will be used interchangeably. The chosen time step for the simulation ($\approx$ 28 seconds) is thus also assumed as a sampling period for discrete measurements on the practical satellite.

---

[7] Software code listings for the control routines are given in appendix C.

[8] The orbital configuration and initial conditions are specified in chapter 1 and the nominal satellite is discussed in chapter 4.

Chapter 4

CONSTELLATION ACQUISITION

1. INTRODUCTION

With the theoretical background and simulation tools now in hand, the first constellation control objective, i.e. acquisition, can be considered. As mentioned in chapter 1, the acquisition phase normally implies large relative movements between different satellites as the constellation is set up in some operational configuration. The objective of this chapter is to design optimal control strategies for the constellation acquisition phase. The following three simplifications will initially be made:

- The constellation will only consist of two satellites.
- All temporal variations in the atmospheric density, except the diurnal variation, will be ignored. This will lead to a time-invariant model of the relative movement between the satellites that can be used to design the control strategies.
- The orbit will be circular.

The above conditions will prevail throughout sections 2 to 4. The effects of unpredictable time-variations in the atmospheric density will be considered in section

5 and the application of the proposed control strategies to elliptical orbits will be discussed in section 6. The designs will be generalised for multi-satellite constellations in section 7.

Firstly, however, it is necessary to discuss the fundamental concept used for the proposed control strategies, as well as the nominal satellite configuration on which the concepts will be illustrated.

1.1 VARIABLE DRAG CONTROL CONCEPT

The magnitude of the atmospheric drag perturbation on a LEO satellite can be controlled if the cross-sectional area of the satellite, projected on a plane perpendicular to the velocity vector, can be controlled. This variable drag force can be viewed as a control force to change the relative positions between satellites in a constellation. The fact that the atmospheric drag acts in the orbital plane limits the application of the proposed concept to satellites in the same orbital plane.

The success of the variable drag control concept is dependent on many factors, including the satellite and orbital properties. One of the most important factors is the altitude of the orbit. If the altitude is too high, the atmospheric density is so small that the drag force is virtually zero. Although the control concepts might theoretically still be successful, the resulting control times might become so large that it is impractical. Analytical expressions for the total control time will be presented in later sections so that the feasibility of the control concepts for a certain orbital configuration can be evaluated.

At very low altitudes, the density becomes so large that the effects of aerodynamic lift might become significant. Aerodynamic disturbance torques due to a flow-profile over the satellite structure and a possible centre-of-mass imbalance will also become increasingly significant as the altitude decreases. These disturbance torques will be discussed in the following sections.

A concept micro-satellite will now be proposed that will be used in the remainder of this study for the design and demonstration of the constellation control concepts.

1.2 SATELLITE CONFIGURATION

1.2.1 Physical properties

The only structural requirement for the satellite is that the cross-sectional area perpendicular to the velocity vector must be controllable. This controllable area can be one of the following:

- the body of an asymmetrical satellite;
- a dedicated control surface (drag paddle); or
- other movable surfaces, for instance controllable body-detached solar panels.

In the first case, the orientation of the satellite can be changed to control the drag force and in the last two cases, the attitude of the particular control surface must be changed. Each method has its own advantages and disadvantages. The first alternative is advantageous from a structural simplicity point of view and is chosen for the nominal satellite. The satellite has the following physical properties:

- Shape: cylindrical disc
- Diameter: 0.7 m
- Height: 0.125 m
- Mass: 10 kg

In the remainder of this study, the mechanism whereby the cross-sectional area perpendicular to the velocity vector is changed is not relevant. Only the minimum and maximum cross-sectional areas must be known. For the nominal satellite, the minimum cross-sectional area is 0.0875 m^2 and the maximum cross-sectional area is 0.3947 m^2. The calculation of these areas is described in appendix A. Requirements on the rate of change of the cross-sectional area will be given later.

The values of the solar radiation pressure reflectivity constant (K) and the drag coefficient (C_D) also depend on the physical properties of the satellite and must be known so that the particular orbit perturbation forces can be calculated.

The value of K depends on the surface material of the satellite. If K=1, the surface is perfectly absorbent and if K=2, the surface reflects all light. A value of 1.5 will be assumed for the remainder of this study.

The drag coefficient C_D is not as trivial to evaluate as it may seem. Since the atmospheric density is very low at the altitudes of satellite orbits, even low Earth orbits, the mean free path of the molecules is large. For altitudes above 200 km, the mean free path of the molecules in the Earth's upper atmosphere exceeds 200 m. Since the dimensions of the nominal satellite are at least two orders of magnitude smaller than this value, the assumption can be made that molecules reflected or re-emitted from the satellite do not interfere with incident molecules. The ordinary continuum-flow theory of conventional aerodynamics has thus ceased to apply and the appropriate regime is that of free-molecule flow (King-Hele, 1987). As a result, there should be no disturbance torque due to an aerodynamic flow profile over the satellite's body.

Various works has shown that a mean value of 2.2 can be taken for C_D under the conditions of free molecular flow with an error (standard deviation) which should not exceed 5 % (King-Hele, 1964).

1.2.2 Sensors

The relative positions of the satellites in the constellation are required for the proposed constellation control concepts. This can easily be computed if each satellite carries a GPS receiver. An accuracy of 1m is easily obtainable with commercial systems and would be more than sufficient for the constellation control requirements.

The orientation of each satellite must also be known. A magnetometer can provide the orientation information with an accuracy in the order of 5° and will be sufficient.

1.2.3 Torqueing devices and energy considerations

Torqueing devices are necessary to change the orientation of the satellite. These manoeuvres are necessary to change the cross-sectional area of the satellite perpendicular to its velocity vector. The use of magnetorquers is possible for this

purpose, although reaction wheels could provide faster re-orientation times. Even if reaction wheels are used, magnetorquers must still be present for momentum dumping. Energy requirements for orientation manoeuvres are discussed in more detail in appendix A.

Apart from the energy required during re-orientation manoeuvres of the satellite, energy might also be necessary to continuously maintain the flight angle in the presence of atmospheric drag if the satellite's centre of mass does not coincide with the geometrical centre of the satellite. If such an imbalance exists, the atmospheric drag force will cause a disturbance torque that will tend to alter the flight angle. It is shown in appendix A that magnetorquers will be able to reject such disturbance torques, even if the imbalance is at its maximum possible value for the nominal test case.

2. MODELLING THE RELATIVE MOVEMENT BETWEEN 2 SATELLITES

In the following section, the changes in the altitude and velocity of two satellites in the same circular orbit will be investigated when the atmospheric drag force on the satellites is varied through orientation adjustments. The effect of the orientation manoeuvres on the scalar distance between the satellites will also be considered. The scalar distance between the two satellites can be calculated as the magnitude of the vector pointing from the one to the other. This quantity is very easy to compute and, for the typical distances considered, the value is almost identical to the arc length (along the orbital path) between the satellites. It provides an easy way of specifying the constellation configuration.

The goal of the section is to find a suitable model to represent the relative movement between two satellites under controlled drag conditions.

2.1 INVESTIGATING THE DYNAMICS

Consider two identical Earth-orbiting satellites, S_1 and S_2. Assume that the cross-sectional areas perpendicular to the velocity vectors of the two satellites (A_1 and A_2) can be changed arbitrarily between some minimum (A_{min}) and some maximum value (A_{max}). The magnitude of the atmospheric drag on each satellite can thus be controlled. Assume further that the only perturbation force acting on the satellites is atmospheric drag, due to a constant atmospheric density.

Let S_1 and S_2 be in the same circular orbit,[1] but separated by some distance in this orbit. As long as $A_1 = A_2$, the two satellites will lose altitude at the same rate and the distance between them would remain virtually constant. If A_1 is made larger then A_2, the satellite S_1 will start to lose altitude faster than S_2 and the two satellites will start to drift apart. As long as S_1 is lower than S_2, they will have different velocities and the distance between them would continue to change.

To stabilise the distance again, S_2 must fall at a faster rate than S_1 until the altitudes (and hence velocities) of the two satellites are equal again. This can be done by setting $A_2 > A_1$ until the two satellites are at the same altitude. If the areas are made equal at this point, the new distance between the satellites will remain approximately constant.

A simulation was done to illustrate the above example. Two satellites were initialised with identical velocity and position vectors. The areas A_1 and A_2 were both kept equal to A_{min} for the first two orbits. A_1 was then switched to A_{max} and kept there for the following two orbits after which it was switched back to A_{min}. The manoeuvre was then repeated on A_2 so that it ended in the same orbit as S_1.[2] All perturbations, except atmospheric drag from a constant atmospheric density, were excluded.

---

[1] Strictly speaking, the orbit cannot be exactly circular, because the satellites slowly spiral towards the Earth due to the atmospheric drag.

[2] S_2 will lose the same amount of altitude as S_1, because the atmospheric density is assumed constant.

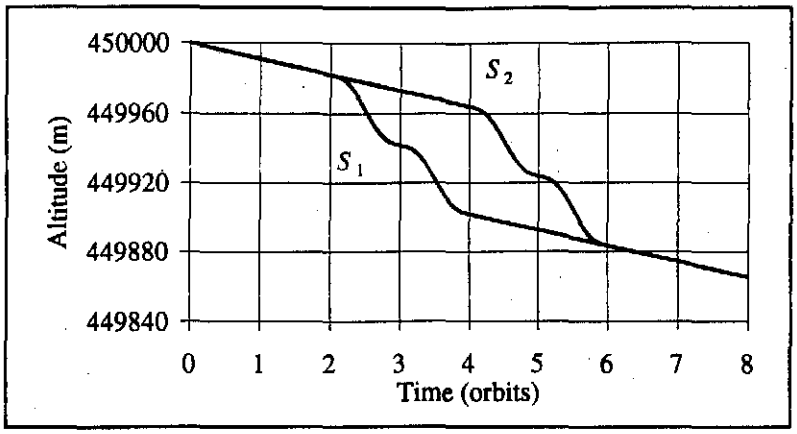


Figure 4.1: Altitude of the two satellites.

Figure 4.1 shows the instantaneous altitude of the two satellites. It is clear that S_1 starts to lose altitude faster than S_2 as soon as its area is switched from A_{min} to A_{max}. S_2 follows after two orbits. Note the corresponding increase in velocity (figure 4.2) as the altitude decreases, demonstrating the "drag paradox", as discussed in chapter 2.

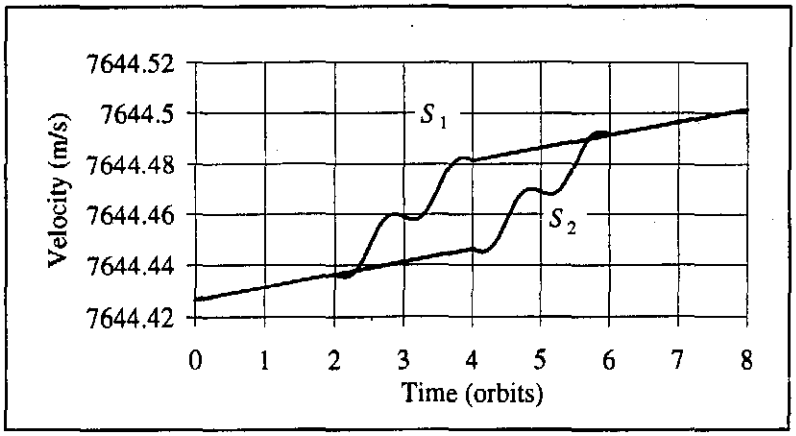


Figure 4.2: Velocity magnitude of the two satellites.

The velocity vectors of the satellites were initiated to correspond to a uniform inward spiral if the cross-sectional area equals A_{min} – hence the slow, linear decrease in altitude during these times. When A_1 or A_2 is switched from A_{min} to A_{max}, the velocity vector of the particular satellite still corresponds to the uniform inward spiral for an area of A_{min}. This causes an oscillation in the altitude and velocity during the high-drag time. The velocity magnitude actually decreases at first after the switch, but starts to increase as the satellite loses more and more altitude. Note that the oscillations in altitude and velocity have a period equal to the orbital period. This characteristic is very useful and will later be dealt with further.

In the following two figures (4.3 and 4.4), the trajectories represent the motion of S_1 relative to S_2. Each graph's origin coincides with S_2's position. The x-axes point in the direction of S_2's velocity vector and the y-axes point radially away from the Earth (to zenith), so that y-values correspond to the difference in altitude between the satellites. Figure 4.3 is a close-up of the first 2.5 orbits, showing the relative movement just after A_1 is switched to A_{max}. S_1 initially falls "behind" as it enters the high-drag stage and its velocity decreases, but then "overtakes" S_2 as it loses more altitude and the velocity starts increasing. The graph of figure 4.4 shows that the satellites are at the same altitude at the end of the complete manoeuvre, with S_1 a distance of approximately 1.2 km ahead of S_2.

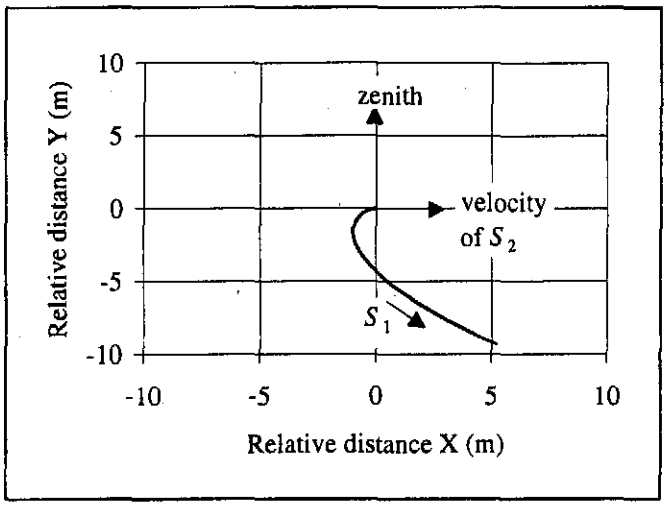


Figure 4.3:* S_1*'s motion relative to* S_2 *(close-up).

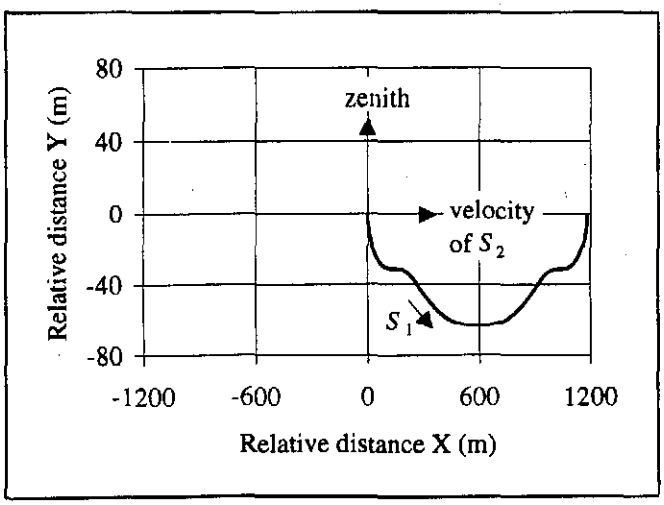


***Figure 4.4:* S_1*'s motion relative to* S_2.**

Consider now the assumptions that were made in the above example. The variation in cross-sectional area perpendicular to the velocity vector is possible through orientation adjustments of the nominal satellite described earlier. The feasibility of controlling the incidence angle of such a small micro-satellite by means of relatively cheap reaction wheels and magnetorquers is demonstrated by the development of SUNSAT[3], a 60 kg imaging micro-satellite (Steyn, 1996). An abrupt change (switch) implies a re-orientation manoeuvre of almost 80° within the duration of one time step (28 seconds). This fast slew manoeuvre is possible for the nominal satellites.[4]

The most significant short-term effect of the non-constant atmospheric density is a periodic influence due to the day-time bulge (chapter 2). Density variations due to solar activity effectively act as a gain to the day-time bulge. The characteristic period of this influence equals the orbital period. The magnitude of the resulting oscillations in altitude and velocity is of the same order as the oscillations displayed in figures 4.1 and 4.2.

The altitude dependence of the density will cause the rate of altitude loss to increase slowly as the satellites lose more and more altitude and the density increases exponentially. This will result in a slight difference between the lengths of time that S_1 and S_2 must spend in the high-drag phase to lose the same amount of altitude. As long as the total altitude loss is relatively small, however, the average density per orbit will not vary significantly and the effect should be small.

Other perturbation forces can have secular or periodic influences on the satellites, as discussed in chapter 2. Since the satellites are relatively close together in the orbit,[5] they would experience approximately the same secular effects due to perturbations and the influence on the relative velocity between them should be very small. The most significant periodic influence comes from the Earth's oblateness (J_2)

---

[3] SUNSAT is developed by the University of Stellenbosch and will be launched by NASA as a secondary payload with the ARGOS mission in August 1997.

[4] See appendix A.

[5] Even a separation distance of 500 km between the satellites is less than 1.2% of the circumference of the circular orbit at an altitude of 450 km.

perturbation, causing variations in the altitude (and velocity) at periods equal to the orbital period and half the orbital period. The magnitude of these oscillations far exceeds those caused by atmospheric drag and can be up to1% in the altitude for the 450 km orbit. The effects of lunisolar attractions and solar radiation pressure are so small that their influence on the relative distance between the satellites can be assumed insignificant for the nominal test case.

The idealised example, as well as the discussion above, pointed out that the relative movement between the two satellites is complex, primarily due to periodic oscillations in the altitudes and velocities. The fact that the periods of these oscillations equal the orbital period or half the orbital period, allows for a drastic simplification if *orbit-averaged* quantities are considered. In the special perturbations simulation, 200 consecutive values of any quantity represent a complete orbit, because there are 200 numerical integration steps per orbit. The *orbit-average* of the quantity can thus be computed at every time step by averaging the last 200 values. This averaging process will act as a sliding filter, removing the oscillations at the orbital rate and integer multiples of the orbital rate. In practice, this can be done by averaging the data samples in a sliding square window with a length of one complete orbit, at every sampling instant. Since the length of the window is exactly one orbit, there will be no end-effects in the average.

A simulation to illustrate the concept of orbit-averaging was carried out. All perturbation forces were now included. A_1 was switched from A_{min} to A_{max} after two orbits and kept there for the next 23 orbits, until the end of orbit 25. Here it was returned to A_{min} and A_2 was switched to A_{max} for the next 23 orbits.

Figure 4.5 (next page) shows the altitudes of the two satellites between orbits 24 and 26, where the difference is at its greatest. The instantaneous, as well as calculated orbit-averages are shown. Note that the peak-to-peak magnitude of the oscillation is approximately equal to 5 km ($\approx$ 1.1%). This oscillation is mainly caused by the J_2-perturbations.

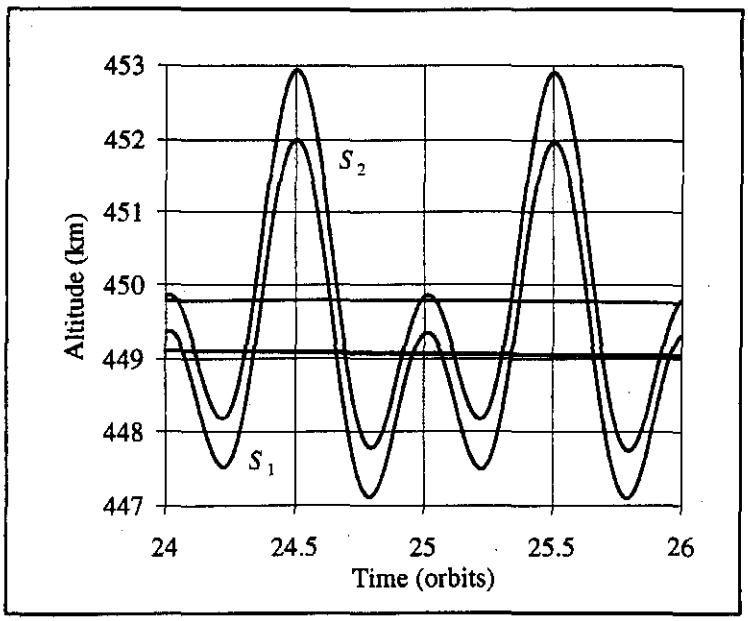


Figure 4.5: Altitudes – instantaneous values and calculated orbit-averages.

The calculated orbit-averages of the altitudes are shown in figure 4.6 over all 50 orbits. The maximum altitude difference between the satellites is less than 1 km for this particular example, which is only a fraction (less than 20%) of the peak-to-peak oscillation magnitude.

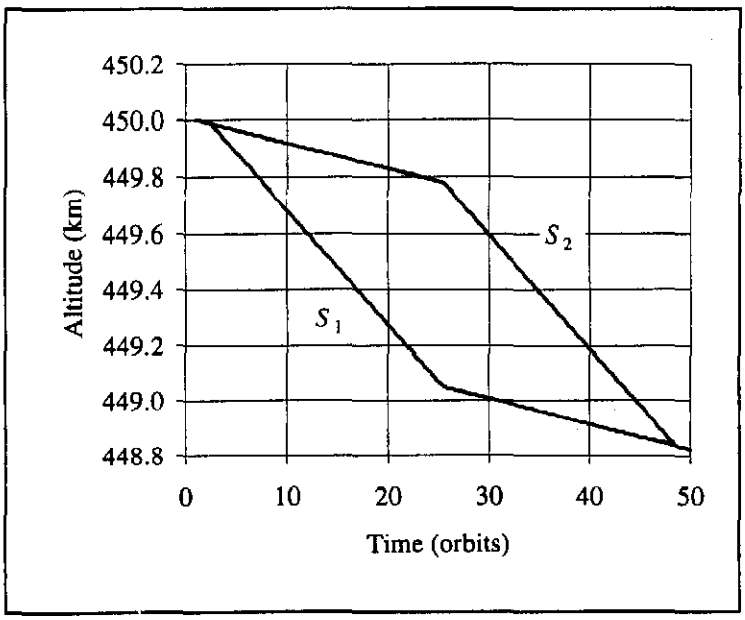


Figure 4.6: Orbit-averaged altitudes – long term tendency.

From figure 4.6 it is clear that the orbit-average altitudes of the two satellites have a piece-wise linear tendency. The rate of altitude loss for each satellite is given by:[6]

$$\Delta a_{rev} = -2\pi \frac{C_D A}{m} \rho a^2 \qquad (4.1)$$

where Δa_{rev} is the change in the semi-major axis (hence altitude) per orbit for a circular orbit. According to equation (4.1), the altitude loss per orbit will be constant as long as the atmospheric density (ρ) and cross-sectional area (A) remain constant. The orbit-average altitude loss will thus be linear under these circumstances.

Figure 4.7 shows the instantaneous and calculated orbit-average values of the straight-line distance between the two satellites, for the interval between orbits 24 and 26.

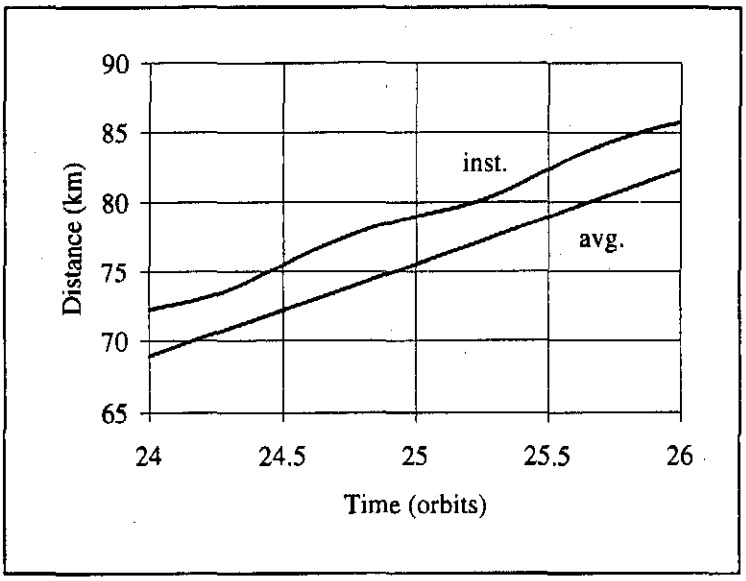


Figure 4.7: Straight-line distance – instantaneous value and calculated orbit-average.

The sliding-filter averaging process introduces a half-orbit-period delay. The reason is that a particular orbit-average value represents the average value over the full orbit up until the time of calculation. It is not the "instantaneous" orbit-average, which would be the orbit-average over the orbit which stretches from half-orbit before to half-orbit ahead of the calculation point. The delay is clearly visible in figure 4.7.

---

[6] From Larson *et al* (1992).

The orbit-average distance is shown in figure 4.8 over the entire 50 orbits. The tendency is piece-wise quadratic, mirrored around the switching point at orbit 25.

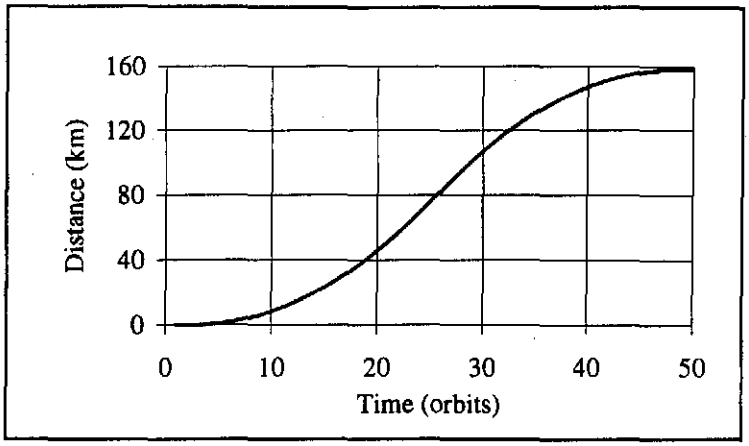


Figure 4.8: Orbit-averaged distance – long term tendency.

2.2 DOUBLE-INTEGRATOR MODEL WITH DELAY

It was shown in the previous section that the orbit-averaged behaviour of the two-satellite system is much simpler than the complex variations taking place on time scales shorter than the orbital period. If a satellite's cross-sectional area perpendicular to the velocity vector is kept constant, the orbit-average altitude loss (and hence velocity increase) is approximately linear when averaged over each orbit. Therefore, a constant difference in cross-sectional area between two satellites will lead to an approximate linear tendency in the orbit-averaged altitude difference between the two satellites. The variation in orbit-averaged distance between the two can then be approximated by a quadratic change, as shown in figure 4.8. All these approximations are valid as long as the total altitude loss is relatively small, so that the nominal density can be assumed constant.

The orbit-averaged dynamics of the two satellites are similar to the simplified case of two objects moving in a straight line under the influence of two forces, applied in the line of the velocity. If the difference between the forces is considered as the input and the distance between the objects is considered as the output, a double integrator model is appropriate to describe the input-output dynamics. In analogy to this, a double integrator model is postulated to approximate the relative dynamics between the two satellites of this simple constellation. The half-orbit-period delay that is introduced by

the calculation of the orbit-averages must be added to the double-integrator. The resulting model is shown in figure 4.9.

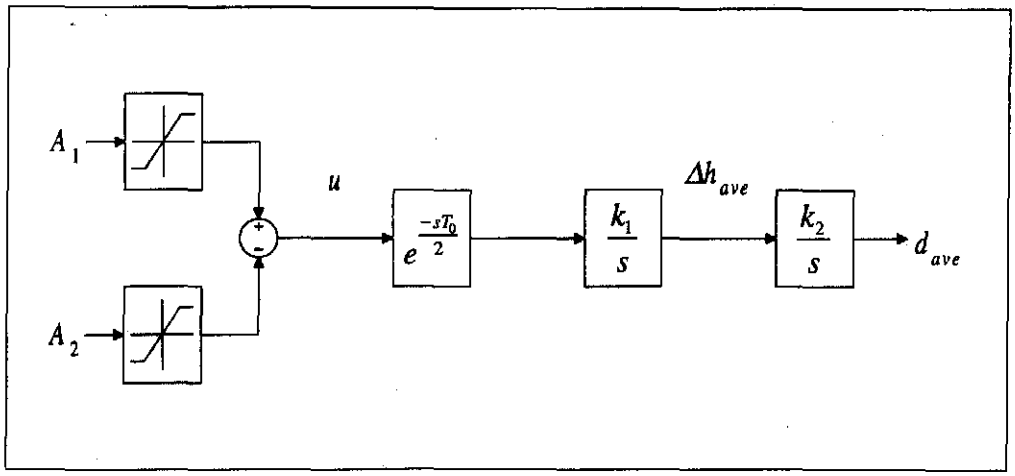


Figure 4.9: Double integrator model with delay.

In the block diagram, d_{ave} is the orbit-averaged straight-line distance between S_1 and S_2, Δh_{ave} is the orbit-averaged difference in altitude; and T_0 is the orbital period. Note that the cross-sectional areas are bounded as follows:

$$A_{min} \leq A_{1,2} \leq A_{max} \tag{4.2}$$

The values of k_1 and k_2 will be determined in section 2.4.

2.3 ELIMINATING THE DELAY

The half-orbit delay in the model of the previous section can be eliminated when the cross-sectional areas of the satellites are constant or piece-wise constant. This is accomplished by making use of the known linear tendency in the orbit-average altitude difference and the known quadratic tendency in the orbit-average distance between the satellites under these conditions. Ideally, the "instantaneous" orbit-averages over the orbit made up of the half-orbit before and half-orbit ahead of the calculation time are needed. The advantage of eliminating the delay is a completely linear model, from the control signal u to the orbit-average distance d_{ave}.

The prediction for the orbit-average altitude difference, 100 samples (half orbit) ahead, can be done with the linear equation:

$$\Delta h = b_h(n+100)+c_h \tag{4.3}$$

where Δh is the "instantaneous" orbit-average altitude difference between the two satellites. The constants b_h and c_h can be calculated at each sampling instant n by fitting the linear equation through the last two values of orbit-average altitude difference, Δh_{ave} and $\Delta h_{ave(n-1)}$:

$$\begin{aligned} b_h &= \Delta h_{ave} - \Delta h_{ave(n-1)} \\ c_h &= \Delta h_{ave} - b_h n \end{aligned} \tag{4.4}$$

The prediction for the orbit-average distance can be done with the quadratic equation:

$$d = a_d(n+100)^2 + b_d(n+100)+c_d \tag{4.5}$$

where d is the "instantaneous" orbit-average distance between the satellites. The constants a_d, b_d and c_d can be calculated at each sampling instant n by fitting the quadratic equation through the last three values of orbit-average distance; d_{ave}, $d_{ave(n-1)}$ and $d_{ave(n-2)}$:

$$\begin{aligned} a_d &= \frac{1}{2}\left(d_{ave} - 2d_{ave(n-1)} + d_{ave(n-2)}\right) \\ b_d &= d_{ave} - d_{ave(n-1)} - a_d(2n-1) \\ c_d &= d_{ave} - a_d n^2 - b_d n \end{aligned} \tag{4.6}$$

The model of figure 4.9 can now be simplified as shown in figure 4.10.

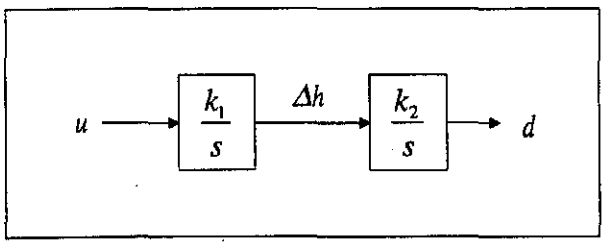


Figure 4.10: Linear model of two-satellite relative movement.

Although there are two inputs and one output in the model of figure 4.9, the system can in essence be treated as a single-input single-output (SISO) system, with the following input:

$$u = A_1 - A_2 \tag{4.7}$$

The linear model of figure 4.10 will only give an approximation of the orbit-average tendencies in the relative movement between the two satellites. It will be time-invariant as long as the density can be assumed constant. As mentioned before, this would be a valid approximation as long as the total altitude loss during the application time of the model remains relatively small so that the density can be assumed constant. The situation where this approximation is not valid anymore will be considered in section 5.

2.4 SYSTEM IDENTIFICATION

The rate of altitude difference between the satellites can be obtained from the model of figure 4.10:

$$\frac{d}{dt}[\Delta h] = k_1 u \tag{4.8}$$

If the assumption is made that the two satellites are close enough to each other so that the atmospheric properties are equal at the two locations, it follows that for each satellite the rate of altitude loss is:

$$\frac{d}{dt}[h_i] = k_1 A_i \tag{4.9}$$

where h_i is the orbit-average altitude and A_i the cross-sectional area of the particular satellite, perpendicular to its velocity vector. The rate of altitude loss is also given by equation (4.1), which can be modified to:

$$\dot{a} = -2\pi \frac{C_D A}{m T_0} \rho a^2 \tag{4.10}$$

where $\dot{a}$ is the rate of change in the semi-major axis for a circular orbit and T_0 is the orbital period. Comparing (4.9) and (4.10), the value of k_1 can be expressed analytically as:

$$k_1 = -2\pi \frac{C_D}{m T_0} \rho a^2 \tag{4.11}$$

This evaluates to -0.01818 for the nominal satellite and circular orbit.

The value of k_2 – relating the altitude difference to a rate of change in the orbit-average distance – can easily be determined, using system identification techniques. The technique used here is the *bisect* function provided in the SIMuWIN[7] block diagram simulation package. It is a gradient algorithm which minimises a cost function by adjusting a single parameter. The cost function was defined as:

$$J = \int (y - \hat{y})^2 dt \tag{4.12}$$

where y is the (orbit-averaged) output from the full special perturbations simulation and $\hat{y}$ is the output from the linear model of figure 4.10.

The value of k_1 could also be verified through system identification. Since both the altitude difference and the distance are available from the special perturbations simulation (from the GPS measurements in practice), the values of k_1 and k_2 could be determined separately using the two outputs in turn. Note that k_1 must be determined first and then it must be used to determine k_2. Figure 4.11 (next page) shows the system identification set-up for determining k_2. The input signal is the difference between the two cross-sectional areas, as given by (4.7).

---

[7]

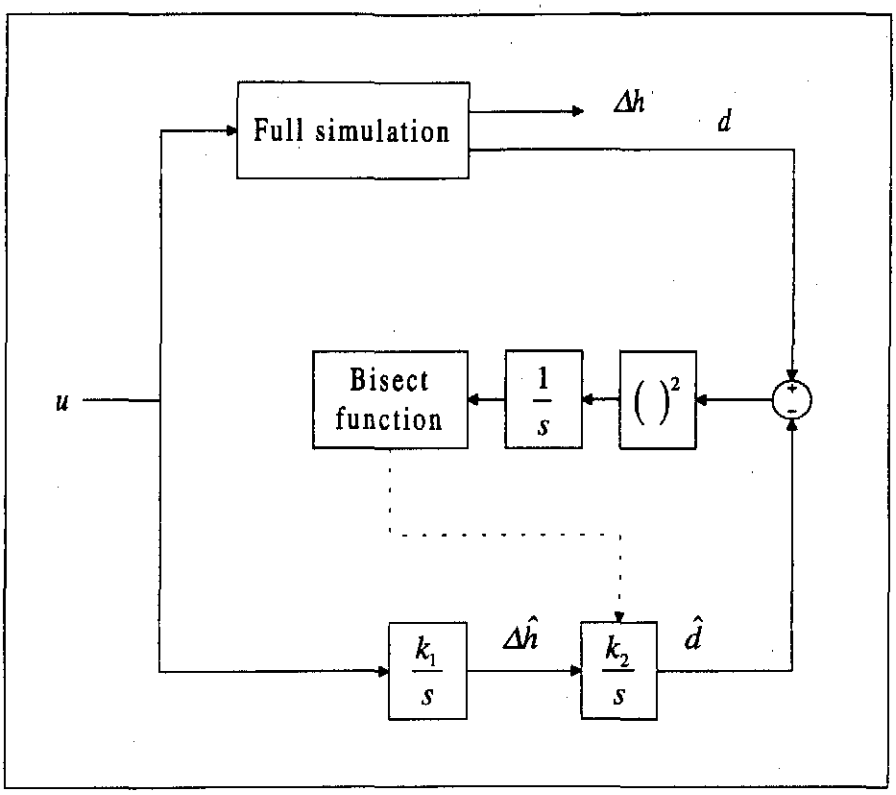


Figure 4.11: System identification set-up.

A square input signal was generated by first setting $A_1 = A_{max}$ and $A_2 = A_{min}$. After 10 orbits the values of A_1 and A_2 were swapped for the next 10 orbits. The output from simulation was written to a file and read into SIMuWIN where the system identification was done. Appendix A gives more information about the identification process. The identification over 20 orbits produced the following values:

$$k_1 = -0.01841$$
$$k_2 = -0.00167 \qquad \textbf{(4.13)}$$

The value of k_1 is close to the theoretical value of -0.01818.

Figures 4.12 to 4.15 compare the output from the special perturbations simulation with the model output, using the above values. The difference (error) between the two outputs is also shown for distance and altitude difference.

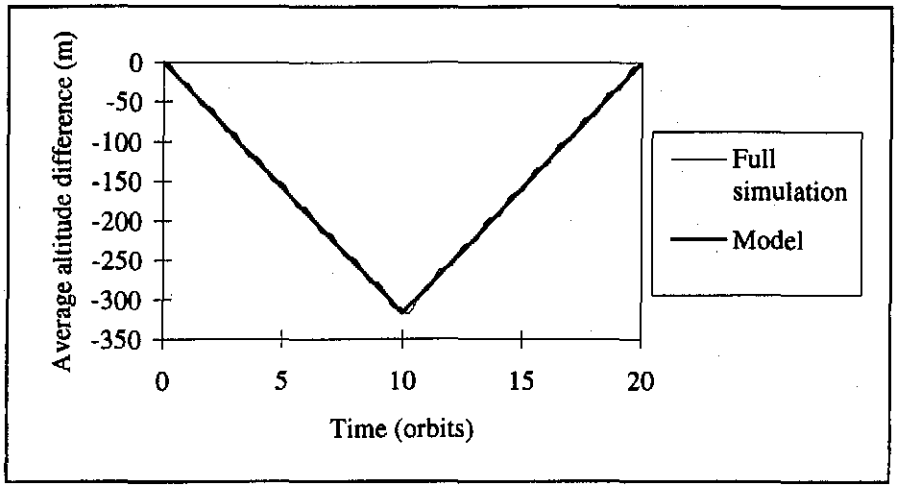


Figure 4.12: Comparing the model with the full simulation – altitude difference.

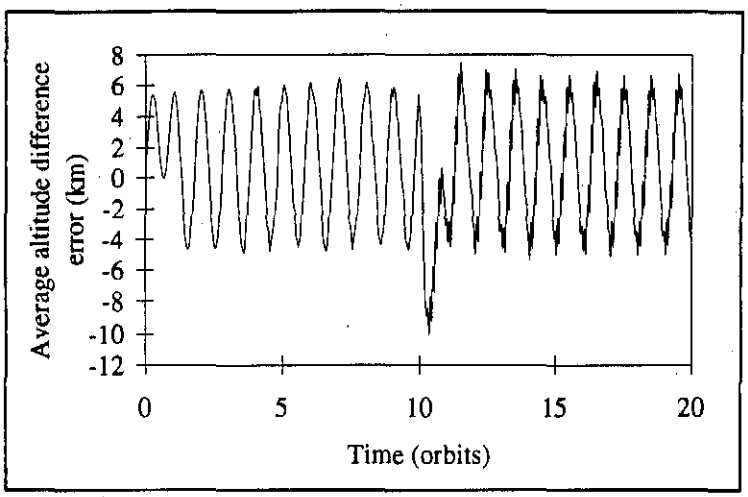


Figure 4.13: Altitude difference error.

In figure 4.12 (altitude difference), the only visual difference is as a result of some oscillation in the calculated "instantaneous" orbit-averages from the full simulation. The maximum error is approximately 10 km.

In figure 4.14 (distance), there is no visual difference. The maximum error is approximately 200 m. This close match confirms the validity of the model under the given operating conditions.

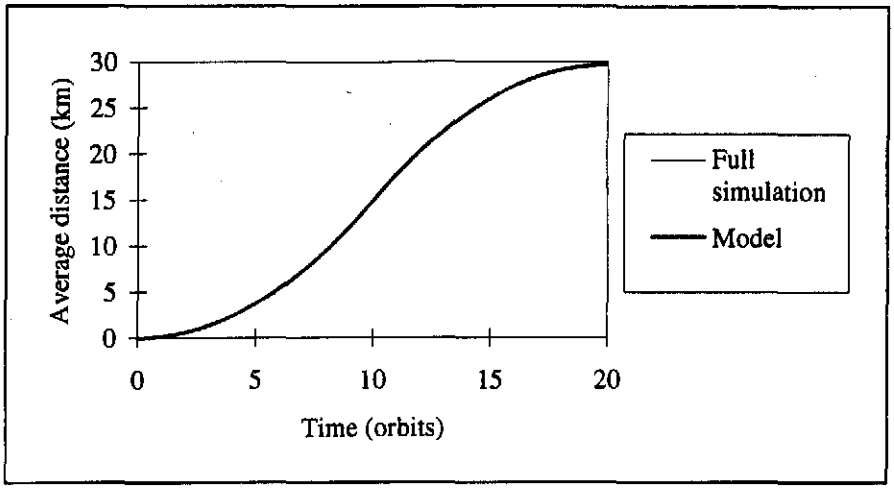


Figure 4.14: Comparing the model with the full simulation – distance.

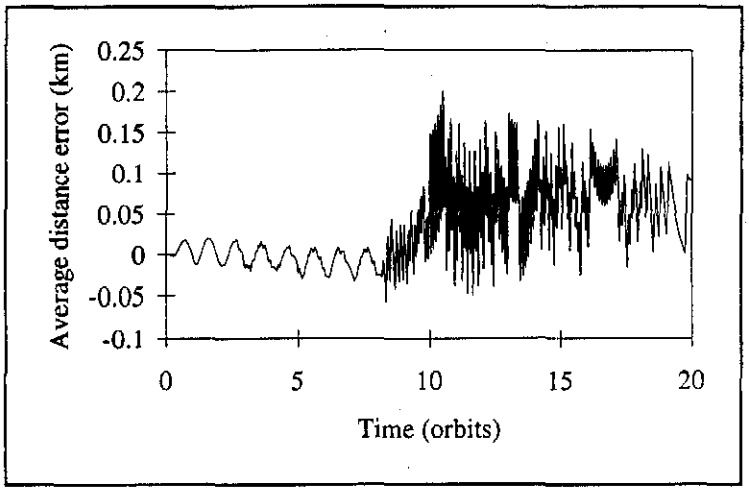


Figure 4.15: Distance error.

3. DESIGNING OPTIMAL CONTROL STRATEGIES

Using the linear, time-invariant model from the previous section, it is now possible to design strategies to control the distance between the two satellites from an initial to any specified reference value. These strategies must be optimised to conserve the limited resources in the restricted space environment. In the context of this study, the

critical resources are time and altitude. The conservation of both is important because the satellites fly in low-altitude orbits with limited lifetimes.

In the following section, a mathematical approach to the optimisation problem will be formulated. The theory will then be applied to design control strategies which are optimal in terms of the two critical resources.

3.1 THEORETICAL BACKGROUND

Consider a general system described by the following set of potentially non-linear and time-dependent differential equations:

$$\dot{\mathbf{x}} = \mathbf{a}\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \tag{4.14}$$

The explicit time dependence of the state vector (**x**) and control vector (**u**) will be dropped henceforth. Assume that the initial state $\mathbf{x}_0$ and initial time t_0 are specified, with:

$$\mathbf{x}\left(t_0\right) = \mathbf{x}_0 \tag{4.15}$$

Let the control vector (**u**, with dimension m) be restricted as follows:

$$M_{i-} \le u_i \le M_{i+} \qquad \text{for} \qquad i = 1,2,\ldots,m \qquad \text{and} \qquad t \in \left[t_0, t_f\right] \tag{4.16}$$

M_{i-} and M_{i+} are known lower and upper bounds for the i^{th} control component. For the given system , a general performance criterion (J) can be defined as follows:

$$J(\mathbf{u}) = f(\mathbf{x},t)\Big|_{t=t_f} + \int_{t_0}^{t_f} g(\mathbf{x},\mathbf{u},\tau)d\tau \tag{4.17}$$

The first term (scalar function f) is evaluated at the end of the control interval (at time t_f) and can represent a measure of the terminal control accuracy. The second term incorporates the evaluation of the scalar function g over the entire control interval. It can represent a measure of the total control effort.

The objective is to find an admissible control vector[8] (denoted as $\mathbf{u}^*$), resulting in an admissible state trajectory[9] (denoted as $\mathbf{x}^*$) that will minimise the performance criterion (4.17). If $\mathbf{u}^*$ exists it will be called an optimal control and $\mathbf{x}^*$ the optimum state trajectory. The minimum performance will be denoted by J^*.

A set of necessary conditions, from which the optimal control can be determined, can be derived from the *fundamental theorem of the calculus of variations* (see Kirk, 1970) and *Pontryagin's minimum principle* (Pontryagin *et al*, 1962). These conditions are formulated in terms of the Hamiltonian (H), defined as follows:

$$H(\mathbf{x},\mathbf{u},\mathbf{p},t) \equiv -g(\mathbf{x},\mathbf{u},t) + \mathbf{p}^T\mathbf{a}(\mathbf{x},\mathbf{u},t) \tag{4.18}$$

where $\mathbf{p}$ is a vector of Lagrange multipliers. The necessary conditions for $\mathbf{u}^*$ to be an optimal control can now be summarised. These conditions must be satisfied for all $t \in [t_0, t_f]$:

$$\dot{\mathbf{x}}^* = \left.\frac{\partial H}{\partial \mathbf{p}}\right|_{(\mathbf{x}^*,\mathbf{u}^*,\mathbf{p}^*,t)} \tag{4.19}$$

$$\dot{\mathbf{p}}^* = -\left.\frac{\partial H}{\partial \mathbf{x}}\right|_{(\mathbf{x}^*,\mathbf{u}^*,\mathbf{p}^*,t)} \tag{4.20}$$

$$H(\mathbf{x}^*,\mathbf{u}^*,\mathbf{p}^*,t) \le H(\mathbf{x}^*,\mathbf{u},\mathbf{p}^*,t) \qquad \text{for all admissible } \mathbf{u}(t) \tag{4.21}$$

The following boundary conditions must also be satisfied:

$$\left[\left[\frac{\partial h}{\partial \mathbf{x}} - \mathbf{p}^*\right]^T \delta\mathbf{x}_f + \left[H + \frac{\partial h}{\partial t}\right]\delta t_f\right]_{(\mathbf{x}^*(t_f),\mathbf{u}^*(t_f),\mathbf{p}^*(t_f),t_f)} = 0 \tag{4.22}$$

---

[8] An admissible control satisfies the control constraints (4.16) over the entire control interval $[t_0, t_f]$.

[9] An admissible state trajectory satisfies the state constraints (4.14) over the entire control interval.

The terms $\delta\mathbf{x}_f$ and δt_f in the last expression are variations on the functions $\mathbf{x}$ and t at the final time t_f. The following can be noted:

- $\mathbf{u}^*$ is a control that causes $H(\mathbf{x}^*, \mathbf{u}, \mathbf{p}^*, t)$ to assume its global, or absolute, minimum.
- Equation (4.21), stating that an optimal control minimises the Hamiltonian, is known as *Pontryagin's minimum principle*.
- Equations (4.19) to (4.22) constitute a set of necessary conditions for optimality, but are not necessarily sufficient.

Consider the boundary conditions of (4.22). The variations $\delta\mathbf{x}_f$ and δt_f are determined by restrictions on the final time and state. For the satellite constellation control problem the final state will always be specified:

$$\mathbf{x}(t_f) = \mathbf{x}_f \tag{4.23}$$

resulting in the variation $\delta\mathbf{x}_f$ being equal to zero. With a fixed final time, the variation δt_f will also be zero. Substituting $\delta\mathbf{x}_f = 0$ and $\delta t_f = 0$ into (4.22) leads to the following set of boundary conditions:

$$\begin{aligned} \mathbf{x}^*(t_0) &= \mathbf{x}_0 \\ \mathbf{x}^*(t_f) &= \mathbf{x}_f \end{aligned} \tag{4.24}$$

If the final time is free, the variation δt_f is arbitrary, resulting in the following additional boundary condition (from (4.22)):

$$\left[H + \frac{\partial h}{\partial t}\right]_{(\mathbf{x}^*(t_f), \mathbf{u}^*(t_f), \mathbf{p}^*(t_f), t_f)} = 0 \tag{4.25}$$

From the set of necessary conditions, the optimal control can now be determined for various performance criteria.

3.2 TIME-OPTIMAL CONTROL

The form of the optimal control will be determined first. If the form is known, the control function can be expressed in terms of the measurements from the GPS receiver.

3.2.1 Form of the optimal control

The objective in time-optimal control problems is to transfer the system from a given initial state to a specified target state in the shortest possible time. The cost function to be minimised is thus:

$$J = \int_{t_0}^{t_f} d\tau = t_f - t_0 \tag{4.26}$$

Assume at first that the control signal (u) will be piece-wise constant and that the delay in the model of figure 4.9 can be eliminated through linear and quadratic prediction, as shown in section 3. The resulting linear second-order model of figure 4.10 has the following state space representation:

$$\begin{aligned} \dot{\mathbf{x}} &= A\mathbf{x} + Bu \\ &= \begin{bmatrix} 0 & k_2 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ k_1 \end{bmatrix} u \end{aligned} \tag{4.27}$$

The state vector is chosen as:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} d - d_{ref} \\ \Delta h \end{bmatrix} \tag{4.28}$$

The specified (reference) distance between the satellites is d_{ref}. With this choice of x_1, the target set equals the origin of the state space. This state space representation is not unique, but is convenient for the analysis and design to follow and is adequate to demonstrate the proposed control concepts.

The control signal u is the difference of the two cross-sectional areas and is therefore bounded as follows:

$$-M \le u \le M \tag{4.29}$$

where M has been defined for convenience:

$$M \equiv A_{max} - A_{min} \tag{4.30}$$

For time-invariant linear systems, it can be shown[10] that a unique optimal control exists if all the eigenvalues of the state matrix A in (4.27) have non-positive real parts; and further that the satisfaction of the minimum principle is sufficient to determine such a control in this case. For the current model (4.27) the eigenvalues are both zero and a unique optimal control can thus be found.

The Hamiltonian is (from (4.18)):

$$\begin{aligned} H &= 1 + \mathbf{p}^T[A\mathbf{x} + \mathbf{B}u] \\ &= 1 + p_1 k_2 x_2 + p_2 k_1 u \end{aligned} \tag{4.31}$$

From Pontryagin's minimum principle (4.21), it is necessary that:

$$1 + p_1^* k_2 x_2 + p_2^* k_1 u^* \le 1 + p_1^* k_2 x_2 + p_2^* k_1 u \qquad \text{for all admissible } u \tag{4.32}$$

Hence, u^* is the control that causes the term $p_2^* k_1 u$ to assume a minimum value. It follows that u^* must be the smallest admissible value if the coefficient $p_2^* k_1$ is positive; and the largest admissible value if the coefficient $p_2^* k_1$ is negative.

---

[10] See Pontryagin *et al* (1962).

The requirement can be expressed mathematically as follows:

$$u^* = \begin{cases} M & \text{for} \quad p_2^* k_1 < 0 \\ -M & \text{for} \quad p_2^* k_1 > 0 \\ \text{undetermined} & \text{for} \quad p_2^* k_1 = 0 \end{cases} \tag{4.33}$$

Consider now the second necessary condition for optimal control, given by (4.20). From this condition follows:

$$\begin{aligned} \dot{p}_1^* &= 0 \\ \dot{p}_2^* &= -p_1^* k_2 \end{aligned} \tag{4.34}$$

These equations are called the *costate equations* with $\mathbf{p}(t)$ the *costate*. The solution has the form:

$$\begin{aligned} p_1^* &= c_1 \\ p_2^* &= -c_1 t + c_2 \end{aligned} \tag{4.35}$$

with c_1 and c_2 constants of integration. It is clear that p_2^* will only pass through zero once and that the optimal control will therefore change sign once. There is no finite singular interval where the optimal control is undefined. This proves that the form of the time-optimal control for the given system is a maximum effort throughout the interval of operation. This form of *extremal control* is also referred to as *bang-bang* control. Note that this also proves the validity of the initial assumption that the control signal (u) will be piece-wise constant.

3.2.2 Determining the control function

Now that the form of the optimal control is known, it is possible to determine the optimal control signal as a function of the state **x**. Firstly, the optimum trajectories in the state space will be determined. To find segments of optimum trajectories, the state equations (4.27) can be integrated analytically, since the two possible values of the control signal (u) are known. The analytical expressions for the state variables are:

$$x_2(t) = k_1 \int_0^t u d\tau + x_{20} \tag{4.36}$$

$$x_1(t) = k_2 \int_0^t x_2 d\tau + x_{10} \tag{4.37}$$

Consider firstly the case $u = M$. Then:

$$\begin{aligned} x_2 &= k_1 M t + c_3 \\ x_1 &= \frac{1}{2} k_1 k_2 M t^2 + k_2 c_3 t + c_4 \end{aligned} \tag{4.38}$$

where c_3 and c_4 are constants of integration. Time (t) can be eliminated from these two equations to yield the relationship between the state variables:

$$x_1 = \frac{k_2}{2k_1 M}(x_2)^2 + c_5 \tag{4.39}$$

with c_5 a constant. Similarly, for the case of $u = -M$, the relationship between the state variables is:

$$x_1 = \frac{-k_2}{2k_1 M}(x_2)^2 + c_6 \tag{4.40}$$

with c_6 a constant. Assume the target state is at the origin of the state space:

$$\begin{aligned} x_1(t_f) &= 0 \\ x_2(t_f) &= 0 \end{aligned} \tag{4.41}$$

Equation (4.39) then describes the terminal segment of an optimal trajectory if $c_5 = 0$ and $x_2 > 0$. Similarly, (4.40) describes the terminal segment of an optimal trajectory if $c_6 = 0$ and $x_2 < 0$. These two terminal segments can be combined to describe a continuous locus:

$$x_1 = \begin{cases} \dfrac{k_2}{2k_1 M}(x_2)^2 & \text{for} \quad x_2 \geq 0 \\ \dfrac{-k_2}{2k_1 M}(x_2)^2 & \text{for} \quad x_2 \leq 0 \end{cases} \tag{4.42}$$

which can be simplified to:

$$x_1 = \frac{k_2}{2k_1 M} x_2 |x_2| \tag{4.43}$$

If the state reaches this locus, the control must switch to M if $x_2 > 0$, or $-M$ if $x_2 < 0$ so that the origin can be reached along the optimal trajectory. For this reason, the locus given by (4.43) is referred to as the *optimal switching curve*. Define a switching function:

$$s(x_1, x_2) = x_1 - \frac{k_2}{2k_1 M} x_2 |x_2| \tag{4.44}$$

The optimal control can now be expressed in terms of the switching function:

$$u^* = \begin{cases} M & \text{for} \quad s(x_1, x_2) < 0 \\ -M & \text{for} \quad s(x_1, x_2) > 0 \\ M & \text{for} \quad s(x_1, x_2) = 0 \quad \text{and} \quad x_1 > 0 \\ -M & \text{for} \quad s(x_1, x_2) = 0 \quad \text{and} \quad x_1 < 0 \\ 0 & \text{for} \quad \mathbf{x} = 0 \end{cases} \tag{4.45}$$

The 3rd and 4th conditions refer to the situation after the switch has occurred and the state slides in towards the origin along the optimal trajectory. Note that (4.45) gives the optimal control in terms of the two known state variables for any initial conditions. The only restriction is that the final state must lie at the origin of the state space.

The implementation and simulation of the time-optimal control strategy will be shown in section 4.3.1.

3.2.3 The minimum control time

An expression for the minimum control time for a general time-optimal control manoeuvre will now be determined. Assume the initial state:

$$\begin{aligned} x_1(0) &= x_{10} \\ x_2(0) &= x_{20} \end{aligned} \tag{4.46}$$

The optimal control u^* can have one of two possible forms:

$$u^* = \begin{cases} M & \text{for} \quad 0 < t \le T_1 \\ -M & \text{for} \quad T_1 < t \le T_1 + T_2 \end{cases} \tag{4.47}$$

or:

$$u^* = \begin{cases} -M & \text{for} \quad 0 < t \le T_1 \\ M & \text{for} \quad T_1 < t \le T_1 + T_2 \end{cases} \tag{4.48}$$

T_1 and T_2 are times to be determined. Note that T_1 equals the time of the first *bang* and T_2 the time of the second *bang* and therefore $t_f = T_1 + T_2$. The first possible control (4.47) corresponds to the initial condition $s(x_{10}, x_{20}) < 0$ and the second possible control (4.48) to the case $s(x_{10}, x_{20}) > 0$. In the equations to follow, the upper signs correspond to the first case and the lower signs to the second case. Evaluating (4.36) at $t = t_f$ yields:

$$x_2(t_f) = 0 = x_{20} \pm k_1 M T_1 \mp k_1 M T_2 \tag{4.49}$$

from which:

$$T_2 = T_1 \pm \frac{x_{20}}{k_1 M} \tag{4.50}$$

Evaluating (4.37) $t = t_f$ and substituting the above value of T_2 yields:

$$x_1(t_f) = 0 = \pm k_1 k_2 M T_1^2 + 2k_2 x_{20} T_1 + \frac{2k_1 M x_{10} \pm k_2 x_{20}^2}{2k_1 M} \tag{4.51}$$

from which T_1 can be solved:

$$T_1 = \mp \frac{x_{20}}{k_1 M} + \sqrt{\frac{k_2 x_{20}^2 \mp 2k_1 M x_{10}}{2k_1^2 k_2 M^2}} \tag{4.52}$$

The time T_2 is found by back-substitution:

$$T_2 = \sqrt{\frac{k_2 x_{20}^2 \mp 2k_1 M x_{10}}{2k_1^2 k_2 M^2}} \tag{4.53}$$

and the minimum control time is thus:

$$t_f = \mp \frac{x_{20}}{k_1 M} + 2\sqrt{\frac{k_2 x_{20}^2 \mp 2k_1 M x_{10}}{2k_1^2 k_2 M^2}} \tag{4.54}$$

The above expression gives the minimum control time for any initial state (x_{10}, x_{20}). For the typical case where the satellites start in the same orbit and end in the same orbit, the initial state is:

$$\begin{aligned} x_{10} &= d_0 - d_{ref} \\ x_{20} &= 0 \end{aligned} \tag{4.55}$$

The initial orbit-average distance between the satellites is d_0 and the final (desired) orbit-average distance between the satellites is d_{ref}. If the initial conditions of (4.55) are substituted in (4.54), the control time is found:

$$t_f = 2\sqrt{\frac{\mp x_{10}}{k_1 k_2 M}} \tag{4.56}$$

For a 500 km change in distance between the satellites, evaluation of (4.56) yields a theoretical minimum control time of 82 orbits. With a nominal orbital period (T_0) of 5606 seconds, this control time is just over 5 days, 7 hours and 40 minutes.

3.2.4 The altitude loss

The theoretical altitude loss for each satellite during the typical time-optimal control effort, with initial conditions given by (4.55), will now be determined. Substitution of (4.55) in (4.53) and (4.52) gives:

$$\begin{aligned} T_2 &= T_1 \\ T_1 &= \sqrt{\frac{\mp x_{10}}{k_1 k_2 M}} \end{aligned} \tag{4.57}$$

The first equality makes intuitive sense. Both satellites must spend the same length of time in the high-drag orientation so that they will lose the same amount of altitude. The total altitude loss (Δa) for each satellite can now be determined. The (equal) times that each satellite spends in the high-drag and low-drag phases are multiplied with the rate of altitude loss, given by (4.1):

$$\Delta a = \frac{-2\pi C_D \rho a^2}{m T_0} \left[A_{\max} + A_{\min} \right] \sqrt{\frac{\mp x_{10}}{k_1 k_2 M}} \tag{4.58}$$

For the 500 km change in distance, the theoretical altitude loss is 2016 m.

3.2.5 Optimal state trajectory

The theoretical optimal trajectory in the state space[11] can now be described. It starts along the trajectory:

---

[11] The state space will be represented as a two-dimensional Cartesian graph with x_1 on the x-axis and x_2 on the y-axis.

$$x_1 = \frac{\pm k_2}{2k_1 M}(x_2)^2 + x_{10} \tag{4.59}$$

until the optimal switching curve is reached. The switching point (x_{1s} , x_{2s}) can be determined by evaluation of (4.36) and (4.37) at time $t = T_1$ to determine the state variables:

$$x_{2s} = \mp\sqrt{\frac{\mp x_{10} k_1 M}{k_2}}$$
$$x_{1s} = \tfrac{1}{2} x_{10} \tag{4.60}$$

After the switch, the state slides to the origin along the optimal trajectory given by (4.43). Figure 4.16 shows a typical time-optimal state trajectory for both cases $x_{10} < 0$ and $x_{10} > 0$. The optimal switching curve is also shown.

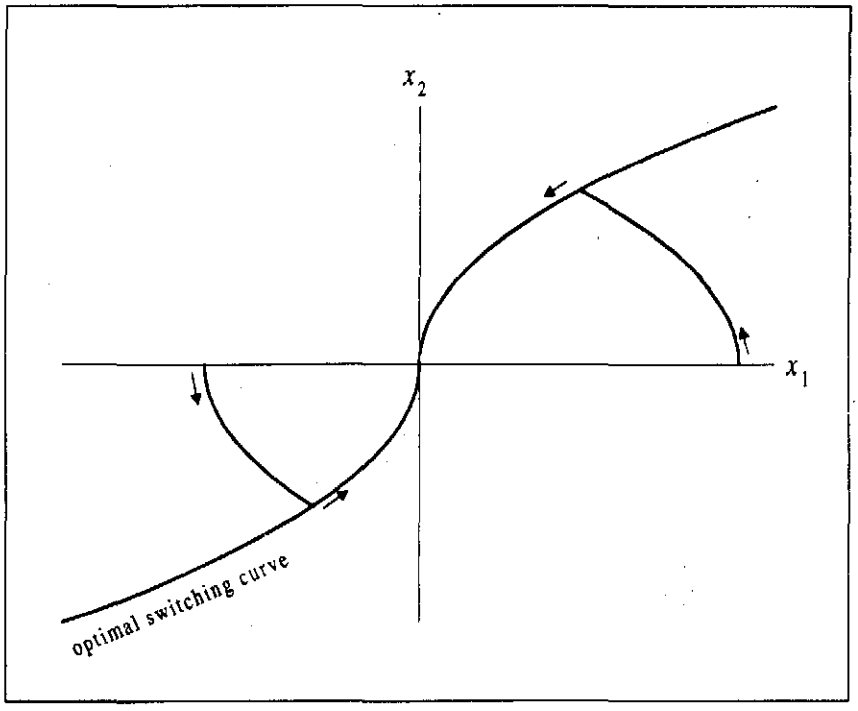


Figure 4.16: Typical time-optimal state space trajectories.

Note that units are not given on the axes since the relative variations of distance and altitude difference are dependent on the variable atmospheric density. The graph intends only to display the general form of the trajectories.

The trajectories of figure 4.16 are theoretical approximations, based on the linear time-invariant model of the relative movement between the two satellites. In reality, the trajectories might deviate from these idealised ones, due to a variety of physical factors. A strategy to keep the real trajectories close to the theoretical trajectories will be discussed in section 5.

3.3 ALTITUDE-LOSS-OPTIMAL CONTROL

The form of the optimal control will again be determined first, before the control function will be expressed in terms of the state variables.

3.3.1 Form of the optimal control

The rate of altitude loss for a satellite is directly proportional to the cross-sectional area perpendicular to its velocity vector (4.1). To minimise the total altitude loss during the control effort, the following cost function must be minimised:

$$J = \int_{t_0}^{t_f} \left(A_1 + A_2 \right) d\tau \tag{4.61}$$

To continue, it is necessary to express the integrand $(A_1 + A_2)$ in terms of the control signal u. Equation (4.7) gives a relationship between u and the areas A_1 and A_2, but is not sufficient to determine $(A_1 + A_2)$ uniquely. An additional relationship between the areas and u is necessary. Given a certain u, it must be decided how to choose A_1 and A_2 so that (4.7), as well as the constraints on the areas (4.2) are satisfied. The following choice meets the criteria:

$$\begin{aligned} A_1 &= A_{\min} + u \qquad \text{for} \quad u \geq 0 \\ A_2 &= A_{\min} - u \qquad \text{for} \quad u \leq 0 \end{aligned} \tag{4.62}$$

This choice also ensures that at least one of the areas equals the minimum area ($A_{\min}$) at any given time, hence representing the best possible choice for the minimisation of the total altitude loss.

The cost function (4.61) can now be expressed as:

$$J = \int_0^{t_f} \left(2A_{\min} + |u|\right) d\tau \tag{4.63}$$

Assuming the same linear, time-invariant model of the previous section, the Hamiltonian (4.18) is:

$$H = 2A_{\min} + |u| + p_1 k_2 x_2 + p_2 k_1 u \tag{4.64}$$

From Pontryagin's minimum principle (4.21), it is necessary that:

$$2A_{\min} + |u^*| + p_1^* k_2 x_2 + p_2^* k_1 u^* \le 2A_{\min} + |u| + p_1^* k_2 x_2 + p_2^* k_1 u \tag{4.65}$$

for all admissible u. The term $|u| + p_2^* k_1 u$ must thus be minimised by the appropriate choice of u. The following choice results in the minimum:

$$u^* = \begin{cases} M & \text{for} \quad p_2^* k_1 < -1 \\ 0 & \text{for} \quad -1 < p_2^* k_1 < 1 \\ -M & \text{for} \quad p_2^* k_1 > 1 \\ \text{undetermined nonnegative} & \text{for} \quad p_2^* k_1 = -1 \\ \text{undetermined nonpositive} & \text{for} \quad p_2^* k_1 = 1 \end{cases} \tag{4.66}$$

The costate equations are again given by (4.34) with the form of the solution the same as for the time-optimal case (4.35). This again proves that there are no finite singular intervals for which the optimal control is undefined. The form of the optimal control is now *bang-off-bang* and the initial assumption that control signal (u) will be piece-wise constant is again valid.

3.3.2 Determining the control function

Segments of optimum state space trajectories can again be obtained by integrating the state equations for the possible values of u^*. For the cases $u = \pm M$, the optimal trajectory segments are identical to the parabolas of the time-optimal case in the

previous section. If the target set is at the origin of the state space, the optimal switching curve (4.43) is applicable here too and describes the optimal state trajectory in the last stage of the control effort. The switching function (4.44) can thus be used to determine where the second switch (from *off* to the second *bang*) must occur. The state will approach the optimal switching curve while $u = 0$. The optimal trajectory segments during this *off*-phase are:

$$\begin{aligned} x_2 &= c_7 \\ x_1 &= k_2 c_7 t + c_8 \end{aligned} \tag{4.67}$$

with c_7 and c_8 constants of integration. In physical terms this means the two satellites have a constant orbit-average altitude difference and the orbit-average distance between them changes linearly. If $x_2 > 0$, the state slides to the left of the state space (distance decreases) along $x_2 = c_7$ until the optimal switching curve is reached. If $x_2 < 0$, the state slides to the right of the state space (distance increases) along $x_2 = c_7$ until the optimal switching curve is reached. The only unknown left to be determined is when the first switch (from *bang* to *off*) must occur. Consider a general *bang-off-bang* control manoeuvre. Once again there are two possible control sequences:

$$u = \begin{cases} M & \text{for} \quad 0 < t \le T_1 \\ 0 & \text{for} \quad T_1 < t \le T_1 + T_2 \\ -M & \text{for} \quad T_1 + T_2 < t \le T_1 + T_2 + T_3 \end{cases} \tag{4.68}$$

or:

$$u = \begin{cases} -M & \text{for} \quad 0 < t \le T_1 \\ 0 & \text{for} \quad T_1 < t \le T_1 + T_2 \\ M & \text{for} \quad T_1 + T_2 < t \le T_1 + T_2 + T_3 \end{cases} \tag{4.69}$$

where T_1, T_2 and T_3 are arbitrary times to be determined. Note that T_1, T_2 and T_3 represent the time duration of the first *bang*, *off* and second *bang* respectively and therefore $T_1 + T_2 + T_3 = t_f$. The first possible control (4.68) again corresponds to the initial condition $s(x_{10}, x_{20}) < 0$ and the second possible control (4.69) to the initial

condition $s(x_{10}, x_{20}) > 0$. In the equations to follow, the sign convention will be the same as before, with the upper sign corresponding to the first case.

The times T_1, T_2 and T_3 can be determined uniquely so that the altitude loss during the control effort is minimised for each satellite. For the case $s(x_{10}, x_{20}) < 0$, the satellite S_1 spends the time T_1 in the high-drag orientation and the time $(T_2 + T_3)$ in the low-drag orientation. The total altitude loss for S_1 can be derived from (4.1):

$$\Delta a = \frac{-2\pi C_D \rho a^2}{mT_0}\left[A_{\max} T_1 + A_{\min}\left(T_2 + T_3\right)\right] \tag{4.70}$$

To determine the T_1 that minimises Δa, T_2 and T_3 must be expressed in terms of T_1. Evaluation of (4.36) at $t = t_f$ gives the relationship between T_3 and T_1:

$$T_3 = T_1 \pm \frac{x_{20}}{k_1 M} \tag{4.71}$$

If (4.37) is next evaluated at $t = t_f$ and the above relationship is substituted, the relationship between T_2 and T_1 is found:

$$T_2 = \frac{-x_{10}}{k_2\left(x_{20} \pm k_1 M T_1\right)} \pm \frac{k_1 M T_1^2}{2\left(x_{20} \pm k_1 M T_1\right)} \mp \frac{x_{20}}{2k_1 M} - \frac{3T_1}{2} \tag{4.72}$$

Now the expressions for T_2 and T_3 in terms of T_1 can be substituted into (4.70) to give Δa as a function of T_1 only:

$$\Delta a = \frac{-2\pi C_D \rho a^2}{mT_0}\left[\frac{A_{\min}\left(\pm k_1 k_2 M T_1^2 - 2x_{10}\right)}{2k_2\left(x_{20} \pm k_1 M T_1\right)} + T_1\left(A_{\max} - \frac{A_{\min}}{2}\right) \pm \frac{A_{\min} x_{20}}{2k_1 M}\right] \tag{4.73}$$

Figure 4.17 on the next page shows a graph of Δa as a function of T_1, for x_{10} = -500 km and $x_{20} = 0$. It is clear that a single minimum point exists.

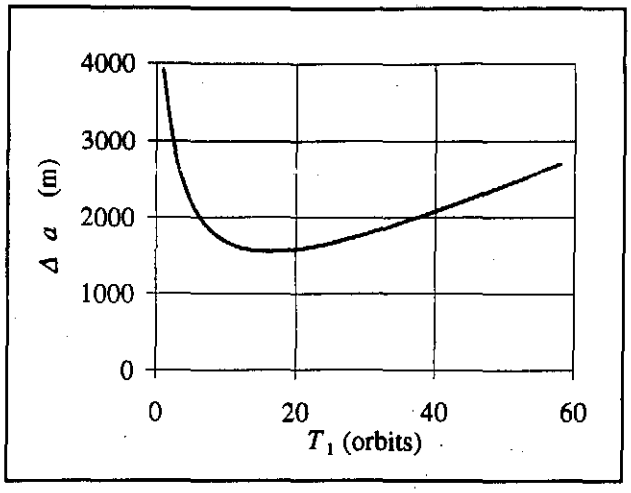


Figure 4.17: Altitude loss of S_1 as a function of T_1.

The minimum point can be found by evaluating the gradient of the function. Differentiating Δa with respect to T_1 and setting the result equal to zero gives the minimum point, which corresponds to the optimal value of T_1:

$$T_1^* = \sqrt{\frac{A_{\min}\left(x_{20}{}^2 k_2 \mp 2x_{10}k_1 M\right)}{2k_1{}^2 k_2 A_{\max} M^2}} \mp \frac{x_{20}}{k_1 M} \qquad \textbf{(4.74)}$$

The same results are obtained if S_1's altitude loss is minimised for the second possible control sequence, with $s(x_{10}, x_{20}) > 0$, or if S_2's altitude loss is minimised for either case.

The optimal control is thus completely determined. The same mathematical formulation as in (4.45) can be used, with the addition that the control equals zero after a time T_1^* has elapsed and stays zero until the optimal switching curve is reached.

The specification of the final time (t_f) is normally required in minimum-energy (*bang-off-bang*) control problems to find a non-trivial, unique optimal control function. It is interesting to note that it was not necessary in this case. The reason being that the satellites continue to loose altitude when the control signal is zero. If this altitude loss during the low-drag phase is not taken into account, the final time would have to be specified.

3.3.3 The control time

Consider again the typical case where the satellites start with the same initial altitude (4.55). Substitution of these conditions in (4.74), (4.72) and (4.71) gives the following optimal times:

$$\begin{aligned} T_1^* &= \sqrt{\frac{\mp x_{10} A_{\min}}{k_1 k_2 A_{\max} M}} \\ T_2^* &= \sqrt{\frac{\mp x_{10} M}{k_1 k_2 A_{\min} A_{\max}}} \\ T_3^* &= T_1^* \end{aligned} \tag{4.75}$$

The control time for the altitude-loss-optimal strategy is the sum of these times:

$$t_f = 2\sqrt{\frac{\mp x_{10} A_{\min}}{k_1 k_2 A_{\max} M}} + \sqrt{\frac{\mp x_{10} M}{k_1 k_2 A_{\min} A_{\max}}} \tag{4.76}$$

For the 500 km example, the theoretical times T_1^* and T_3^* equal 19.32 orbits and the time T_2^* equals 67.81 orbits. The total control time is 106.45 orbits (about 7 days).

3.3.4 The minimum altitude loss

The minimum altitude loss is given by (4.70), with T_1, T_2 and T_3 substituted by the optimal values of (4.75):

$$\Delta a = \frac{-2\pi C_D \rho a^2}{m T_0} \left[(A_{\max} + A_{\min}) T_1^* + A_{\min} T_2^* \right] \tag{4.77}$$

For the 500 km change in distance, the minimum altitude loss is 1554 m. The corresponding increase in the rate of altitude loss (4.1) is less than 2.5%, so that the assumption of a piece-wise linear altitude decrease is validated.

3.3.5 *Optimal state trajectory*

The theoretical optimal trajectory in the state space starts along the curve:

$$x_1 = \frac{\pm k_2}{2k_1 M}(x_2)^2 + x_{10} \tag{4.78}$$

until the first switching point is reached. This point is obtained by evaluating the state variables at $t = T_1^*$, using (4.36) and (4.37):

$$\begin{aligned} x_{1s1} &= x_{10}\left[1 - \frac{A_{\min}}{2A_{\max}}\right] \\ x_{2s1} &= \mp\sqrt{\frac{\mp x_{10} A_{\min} k_1 M}{k_2 A_{\max}}} \end{aligned} \tag{4.79}$$

At this point, the control switches to zero and the state slides along the trajectory:

$$\begin{aligned} x_1 &= x_{10}\left[1 - \frac{A_{\min}}{2A_{\max}}\right] \pm \left[k_1 k_2 M \sqrt{\frac{\mp x_{10} A_{\min}}{k_1 k_2 A_{\max} M}}\right] t \\ x_2 &= x_{2s1} \end{aligned} \tag{4.80}$$

during the *off* phase, until the second switching point is reached. The latter is obtained by evaluating the state variables at $t = T_1^* + T_2^*$:

$$\begin{aligned} x_{1s2} &= x_{10}\left[1 - \frac{A_{\min} \pm 2M}{2A_{\max}}\right] \\ x_{2s2} &= x_{2s1} \end{aligned} \tag{4.81}$$

Finally, after the second switch, the state slides to the origin along the optimal trajectory given by (4.43). Figure 4.18 (next page) shows the form of typical state trajectories.

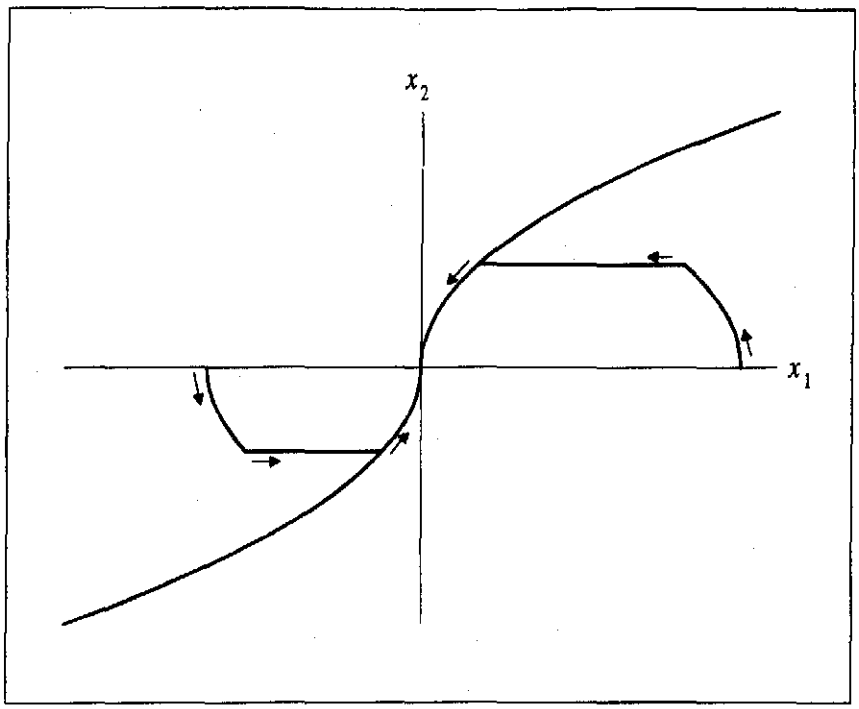


Figure 4.18: Typical state trajectories for altitude-loss-optimal control.

3.4 COMPARISON OF STRATEGIES

It is interesting to compare the total control time and altitude loss of the two optimal strategies. For the case where the satellites start in the same orbit ($x_{20} = 0$), the ratio of the total control time of the altitude-loss-optimal strategy ($t_{h\text{-}opt}$) to that of the time-optimal strategy ($t_{t\text{-}opt}$) is obtained by dividing (4.76) by (4.56) and using the definition of M (4.30):

$$\frac{t_{h-opt}}{t_{t-opt}} = \frac{A_{max} + A_{min}}{2\sqrt{A_{max} A_{min}}} \tag{4.82}$$

Note that this ratio is only dependent on the maximum and minimum cross-sectional areas. For the nominal satellite, the factor is approximately 1.3. This means that the duration of the constellation acquisition strategy that minimises the altitude loss is only 1.3 times the duration of the time-optimal strategy. This time difference is relatively small and the altitude-loss optimal strategy can thus be considered as the preferred strategy.

The ratio of altitude loss for the two control strategies is obtained by dividing (4.77) by (4.58):

$$\frac{h_{h-opt}}{h_{t-opt}} = \frac{2\sqrt{A_{max} A_{min}}}{A_{max} + A_{min}} \tag{4.83}$$

It is interesting to note that this ratio is the inverse of (4.82). The time-optimal control strategy would thus lose 1.3 times the altitude of the altitude-loss-optimal control strategy.

4. IMPLEMENTATION AND SIMULATION OF CONTROL STRATEGIES

This section will show the results of simulations to test the control strategies that have been designed in previous sections. The additional software to calculate the optimal control signals will also be discussed briefly. Firstly, however, it is necessary to consider some practical aspects pertaining to the simulation and physical implementation of the control strategies.

4.1 PRACTICAL CONSIDERATIONS

4.1.1 Control signal chatter

The calculation of the control signal (u) during the final phase of the control effort presents an important practical consideration. For both extremal control strategies, the theoretical optimal control is given by equation (4.45) during the final control phase. During this phase, the state should ideally slide along the optimal switching curve (4.43) and the switching function (4.44) should remain zero. In practice, however, it can be expected that the state will always tend to deviate from the ideal, theoretical trajectory. This will cause a "chatter" in the control signal as it continuously switches between $+M$ and $-M$ to try and keep the state on the theoretical trajectory. Such a chattering control signal is undesired, because it represents a large

number of re-orientation manoeuvres, every time slewing both satellites through almost 80°. To conserve on-board energy, the number of re-orientation manoeuvres must be minimised. The approach adopted in this case is to use equation (4.45) to determine the control signal until the optimal switching curve is reached. At this point, the control signal changes for the last time and is then not allowed to switch again until the end of the control effort, where it is returned to zero. This guarantees a minimum number of switches – and hence re-orientation manoeuvres – for both the optimal strategies. In both cases only two re-orientation manoeuvres are required from each satellite.

The result of the above approach is that the state is allowed to deviate from the optimal trajectory and will not, in general, reach the origin of the state space exactly. The control is terminated ($u = 0$) when the satellites have the same orbit-average altitude again ($x_2 = 0$). An error in the final distance between the satellites should thus be present. The magnitude of this error depends on the deviation from the optimal state trajectory and it should be small if the model of the system is a good representation of the real system. A strategy to keep the state close to the optimal trajectory, even in cases where the model is not accurate anymore, will be discussed in section 5. A strategy to eliminate the terminal error is presented in chapter 5, where constellation maintenance is the objective.

4.1.2 Supervisory vs. autonomous control

The nature of the constellation acquisition control strategies requires a centralised computation of the control signal, based on the relative positions of both satellites. It is not possible to have an autonomous controller on board each satellite without inter-satellite communications. The supervisory control strategy should thus be co-ordinated from the ground station, especially if the constellation consists of more than two satellites.

4.2 CONTROL SOFTWARE

The relevant routines to implement the optimal control strategies are contained in the two Pascal units that are discussed in the following sections. Source code listings are given in appendix C.

4.2.1 Unit CONTROL

The core of the control process is contained in this unit. An object *TController* is defined to provide the data structure and methods to calculate the optimal control signal (u) from the state variables x_1 and x_2 at every sampling instant.

4.2.2 Unit FILTERS

This unit provides the routines to calculate orbit-averages and predicted values necessary for the control process. Three objects are defined:

- *TFilter*: This is the sliding average filter, calculating orbit-averages in the square window over 200 samples of a particular quantity.
- *TLpred*: Linear prediction, using the formula of (4.3) to calculate Δh at every sampling instant from the previous two values of the orbit-averaged altitude difference.
- *TQpred*: quadratic predictor, using the formula of (4.5) to calculate d at every sampling instant from the previous three values of the orbit-averaged straight-line distance.

4.3 SIMULATION RESULTS

Simulations to demonstrate the optimal control strategies were carried out for a specified distance of 500 km between the satellites. The satellites were initiated with identical velocity and position vectors. All perturbations were included in the simulations, as well as the exponential altitude dependence of the atmospheric density. The cross-sectional areas were chosen from u using (4.62).

4.3.1 Time-optimal control

The results of the simulation are shown in the state space of figure 4.19. Keep in mind that x_1 is the difference between the predicted orbit-average distance (d) and the reference distance (d_{ref} = 500 km); and x_2 is the predicted orbit-average altitude difference (Δh) between the satellites. Note also that the origin of the graph is in the upper right corner. The black dots give an indication of time. A black dot is displayed every ten orbits. This convention will be used for all the phase-plane graphs in this chapter.

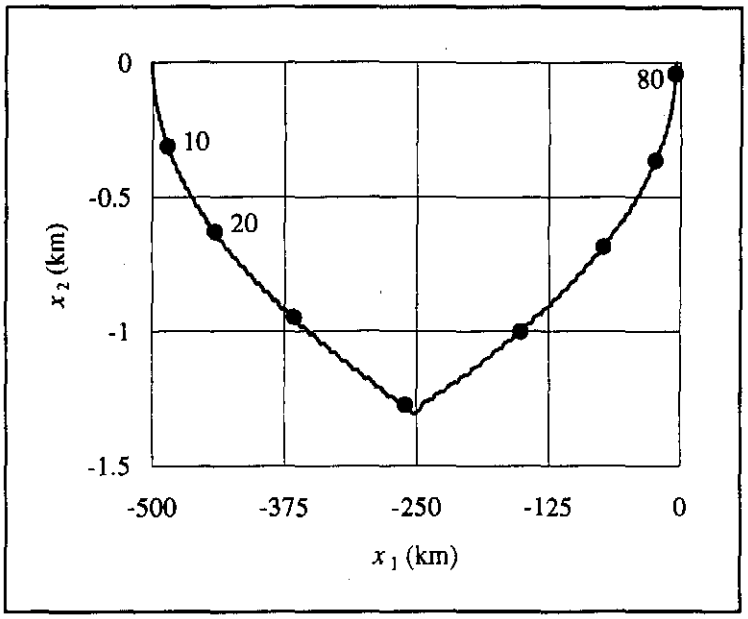


Figure 4.19: Time-optimal constellation acquisition – state space.

The terminal error in the orbit-average distance is 4.4 km (0.88%), mainly due to the variation of the atmospheric density with altitude which is ignored in the linear, time-invariant model of (4.27).

The total control time is 81.24 orbits, made up by a 40.64 orbit first phase (T_1) and a 40.60 orbit second phase (T_2). This compares very well to the theoretical prediction of two equal 41 orbit phases for a control time of 82 orbits, given by equation (4.56). The fact that the times are slightly shorter than the theoretical predictions is consistent with the fact that the atmospheric density increases slightly as the satellites lose altitude during the control effort. The same effect causes the total altitude loss of 2029 m for both satellites to be slightly more than the theoretical value of 2016 m from equation (4.58).

4.3.2 Altitude-loss-optimal control

Figure 4.20 gives the phase-plane results for an altitude-loss-optimal constellation acquisition simulation. The same scale as in figure 4.19 is used for easy comparison.

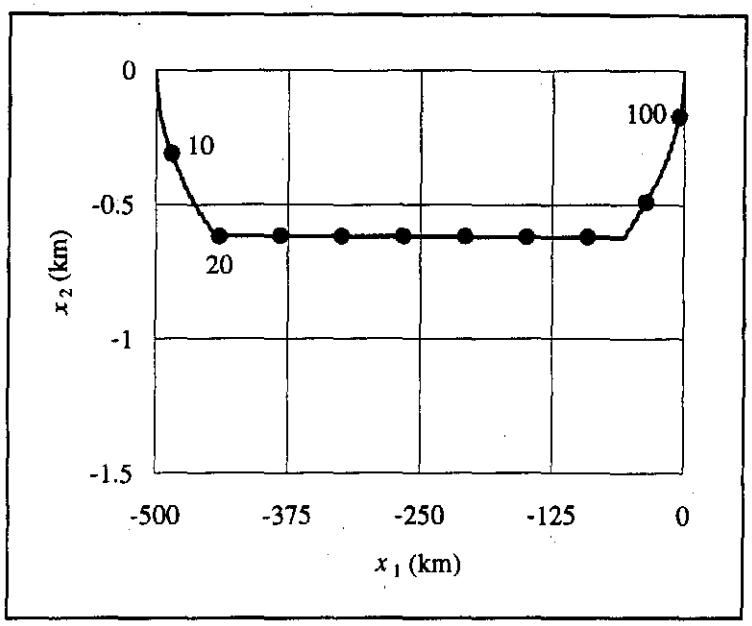


Figure 4.20: Altitude-loss-optimal constellation acquisition – state space.

The final error in the distance is approximately 413 m (0.08%).

The first phase of the control effort was timed, using the calculated duration of 19.32 orbits from equation (4.75). The values of T_2 and T_3 were 66.53 and 19.42 orbits respectively, for a total control time of 105.27 orbits. The times are again shorter than the theoretical values of 67.81 and 19.32 orbits for T_2 and T_3 respectively. The altitude loss during the control effort equals 1566 m. The theoretical value, using equation (4.77) is 1554 m.

5. ADAPTIVE CONTROL

If the nominal density varies with time, the second-order model of (4.27) will no longer be time-invariant and the control strategies based on this model will no longer be accurate. This section will present a strategy for real-time estimation of the changing parameters of the model. The two optimal control strategies, based on this

model, can then be adapted continuously to maintain accuracy and robustness during the constellation acquisition effort.

5.1 CHARACTERISING THE VARIATIONS IN ATMOSPHERIC DENSITY

From chapter 2, the atmospheric density is modelled as follows in the special perturbations simulation:

$$\rho = \left(\rho_{nom} + \rho_v(t)\right)\left(1 + F\cos(\phi)\right)\exp\left[-\frac{h - h_{ref}}{H_s}\right] \tag{2.84}$$

Up until now, the term $\rho_v(t)$ was assumed zero, thus neglecting all temporal variations in the atmospheric density, besides the diurnal variation. Additional temporal variations will now be included through $\rho_v(t)$.

The variations in atmospheric density due to short-term solar activity are of particular interest, as discussed in chapter 2. The duration of these variations is typically a few days which could drastically influence the performance of the constellation acquisition effort. Since the occurrence of solar flares are irregular and difficult to describe, the best that can be done is to characterise the corresponding variations in density according to some typical magnitude, form and time-scale.

It has been observed that the variation in atmospheric density due to short-term solar activity correlates with the geomagnetic disturbance of the Earth's atmosphere, also due to the solar activity.[12] The form of a typical *magnetic storm* is described in Wertz (1978). From this, the following rough approximation of the variation in atmospheric density, starting at time t_0, is derived:

$$\rho_v(t) = c_1\left(t - t_0\right)^2 e^{-\frac{t - t_0}{c_2}} \tag{4.85}$$

---

[12] See Wertz (1978) or King-Hele (1969, 1987).

The constants c_1 and c_2 determine the magnitude and time-scale respectively. The form of the function resembles the form of the magnetic disturbance, with a sudden initial increase (main phase) and a more gradual decrease (initial recovery and final recovery phases). Values of $c_2 = 8$ and $c_1 = 0.0285$ result in the graph of figure 4.21, with time expressed in orbit units and $t_0 = 0$.

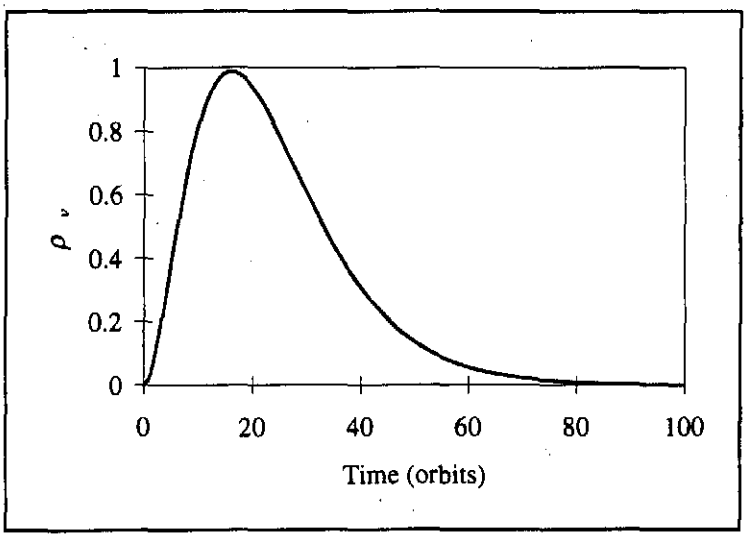


Figure 4.21: Approximated form of a typical variation in atmospheric density due to short-term solar activity.

The time from start to peak is approximately 17 orbits (≈ 26 hours) and the variation has practically died out (< 1%) after 80 orbits (≈ 5.2 days).

5.2 REAL-TIME PARAMETER ESTIMATION

From (4.11) it is clear that k_1 is directly proportional to the atmospheric density. The parameter k_2 is not influenced by density variations since it simply relates the altitude difference to the rate of change of the distance between the satellites.

For a small enough time step (δ), the orbit-average altitude at time step n can be estimated from the orbit-average altitude at time step n-1 (h_{n-1}), using the following difference equation, derived from the continuous equation (4.9) for a particular satellite:

$$h_{est(n)} = h_{n-1} + \delta k_1 A_n \qquad \textbf{(4.86)}$$

where $h_{est(n)}$ is the estimated altitude and A_n is the cross-sectional area at sampling point n. This regression model is linear in the parameter k_1 and the recursive least squares (RLS) method can be used to estimate this parameter. Since k_1 will be time-varying, an exponential forgetting factor (λ) is introduced in the standard RLS algorithm, from Åström *et al* (1989):

$$\begin{aligned} K_n &= P_{n-1}\Psi_n\left[\lambda \mathbf{I} + \Psi_n^T P_{n-1}\Psi_n\right]^{-1} \\ \theta_n &= \theta_{n-1} + K_n\left(y - \Psi_n^T\theta_{n-1}\right) \\ P_n &= \tfrac{1}{\lambda}\left(\mathbf{I} - K_n\Psi_n^T\right)P_{n-1} \end{aligned} \qquad \textbf{(4.87)}$$

This is a matrix formulation for the general case where more than one parameter must to be determined. The vector Ψ is the *regression vector* and θ is the *parameter vector*, so that the model output is given by $\Psi^T\theta$. The measured output of the system is y. K is a vector and P a matrix of variable gains, updated every time-step. Since k_1 is the only parameter to be estimated, the RLS algorithm reduces to:

$$\begin{aligned} K_n &= \frac{P_{n-1}A_n}{\lambda + A_n^2 P_{n-1}} \\ k_{est(n)} &= k_{est(n-1)} + K_n\left(h_n - h_{est(n)}\right) \\ P_n &= \tfrac{1}{\lambda}\left(1 - K_n A_n\right)P_{n-1} \end{aligned} \qquad \textbf{(4.88)}$$

where $k_{est(n)}$ is the estimate of k_1 at the time step n and $h_{est(n)}$ is given by (4.86). The measured output at the time step n is h_n (from the full special perturbations simulation). The estimation of k_1 can be based on the movement of either one of the satellites.

Since there is some oscillation present in the orbit-average altitude h, there will also be some oscillation in the estimate of k_1. This can be eliminated by calculating the orbit-average of the estimate. The net effect is that the half-orbit delay due to averaging the altitude and the half-orbit delay due to averaging the estimate will add together, so that a full-orbit delay can be expected in the estimate of k_1. The typical short-term variation in density, however, has a duration of a few days, so that the error due to the delay should be small.

Figure 4.22 shows the results of the RLS estimation[13] if there is a sinusoidal variation in the atmospheric density. The amplitude of variation is half the nominal density and the period is 20 orbits. This hypothetical variation serves to demonstrate the ability of the RLS algorithm to track a relatively rapid-varying density. The forgetting factor (λ) was set to 0.98 and the gain P was initiated as 0.1. The normalised atmospheric density $(\rho_{nom} + \rho_v)/\rho_{nom}$ is plotted with the lagging normalised estimate (k_{est}/k_1).

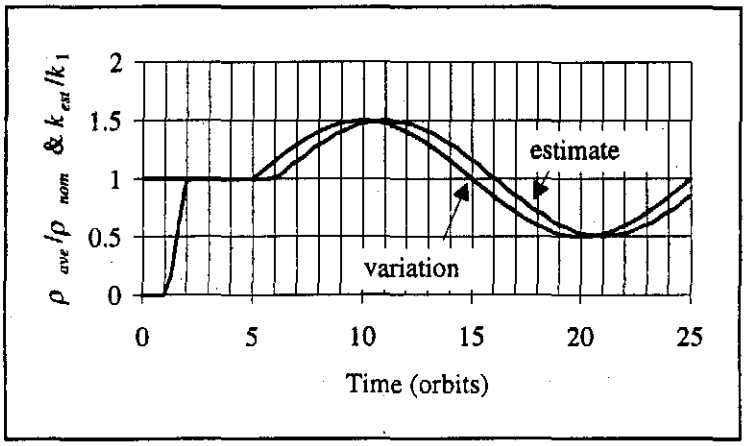


Figure 4.22: RLS – tracking a sinusoid.

The estimation starts after one orbit and the transient ($\approx$ 1 orbit) is mainly due to the orbit-averaging process. The tracking is accurate, with the full-orbit delay clearly present.

All temporal variations in atmospheric density, including the typical variation due to short-term solar activity, take place over longer times than the period of the sinusoidal variation in the above example. The RLS algorithm can thus be expected to be perform satisfactorily.

5.3 FOLLOWING THE OPTIMAL TRAJECTORY

Now that the time-varying model can be estimated in real-time, it is relatively easy to adapt the optimal control strategies to maintain accuracy in the presence of temporal variations in atmospheric density. The parameter k_1 is influenced by such variations in atmospheric density, but it can be estimated in real time. Since k_1 is a gain in the

---

[13] The Pascal unit *RLS* contains the algorithm for real-time estimation. See appendix C.

feed-forward path (see figure 4.9), an inverse variation in the control signal u could counter the effect of its variation. The control signal can thus be adapted as follows:

$$u_a = u\frac{k_{10}}{k_{est}} \tag{4.89}$$

where u_a is the adapted control signal, u is the calculated optimal control signal, k_{10} is the original value of k_1, used to calculate u and k_{est} is the estimated value of k_1. Since u is an extremal control, it is clear that (4.89) will result in an admissible control only if $k_{est} > k_{10}$ during the times that u is not zero. This corresponds with an increase in the atmospheric density and a required decrease in the cross-sectional area of the particular satellite in the high-drag orientation. Fortunately, the short-term solar activity normally causes an increase in atmospheric density.

If the density does however decrease rapidly during the control effort and k_{est} becomes smaller than k_{10}, the cross-sectional area cannot increase beyond A_{max}. This problem can be overcome if M is scaled to some fraction of its maximum possible value, hence replacing equation (4.30) by:

$$M = \gamma\left(A_{max} - A_{min}\right) \tag{4.90}$$

with γ some constant between 0 and 1. The calculated control u will now be based on this non-maximum value of M, resulting in a non-optimal strategy. If γ is 0.5, the calculation of the switching curve will be based on satellites falling at half the maximum rate during their high-drag phases. The drag can thus decrease to half the initial value before saturation in the adapted control signal occurs. Further decreases in the density will result in a deviation from the switching curve and a corresponding error in the final distance. If this error is too large for the constellation maintenance phase (chapter 5) to take over, another acquisition phase can be done to reduce the error.

The proposed adjustment in u is only necessary during the final phase of both optimal constellation acquisition efforts. The terminal error in distance will be small, regardless of the variations in density until the point where the switching curve is

reached, as long as the system stays close to this curve. Variations prior to the final phase, however, will generally result in a deviation from the optimal state trajectory and a corresponding loss of optimality.

For every new constellation acquisition effort, a new value of k_{10} can be used. This value can be obtained from the RLS estimate just prior to the start of the effort. This ensures that longer-term variations in atmospheric density (solar cycle, semi-annual, sun-rotation etc.) are also tracked and will be included through the updated value of k_{10}. The proposed adaptation scheme of (4.89) is only necessary to counter the effects of rapid density variations during the control effort.

The proposed adaptation scheme will generally not keep the state trajectory exactly on the optimal switching curve, mainly because the estimate k_{est} suffers a time delay of one orbit. The resulting error in the final distance between the satellites is however drastically smaller than the error if the short-term variations in density were not catered for. Simulations to verify this will be described in the following section.

5.4 IMPLEMENTATION AND SIMULATION

The adaptive scheme of the previous section has been implemented for both the optimal acquisition strategies and the results of simulations are shown in the sections to follow. The satellites were again initialised with identical position and velocity vectors and the specified distance was 500 km.

As mentioned earlier, only short-term variations in atmospheric density during the final phase of the control effort will have an influence on the terminal accuracy of the effort. The short-term variation in atmospheric density of figure 4.21 was thus included during this final phase for all simulations.

5.4.1 Bang-bang control

The density variation is shown in figure 4.23, starting at orbit number 46 orbits, just after the switch has occurred.

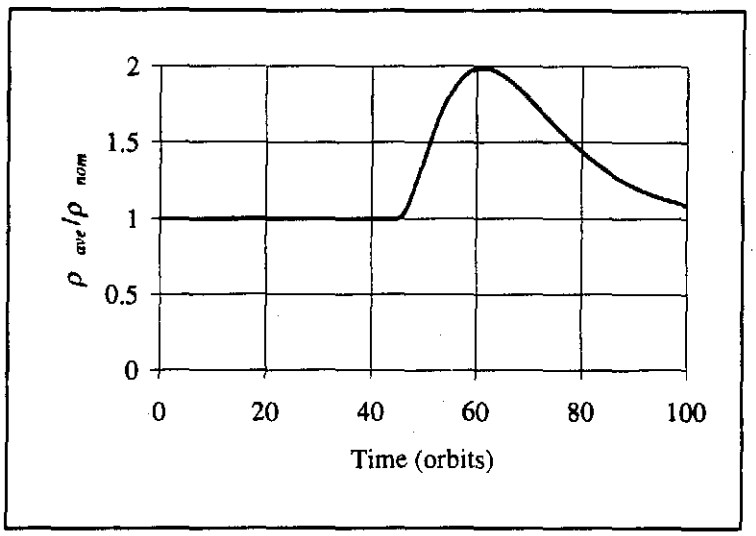


Figure 4.23: Variation in atmospheric density.

The results in the state space (figure 4.24) show the improvement as a result of the adaptive control scheme.

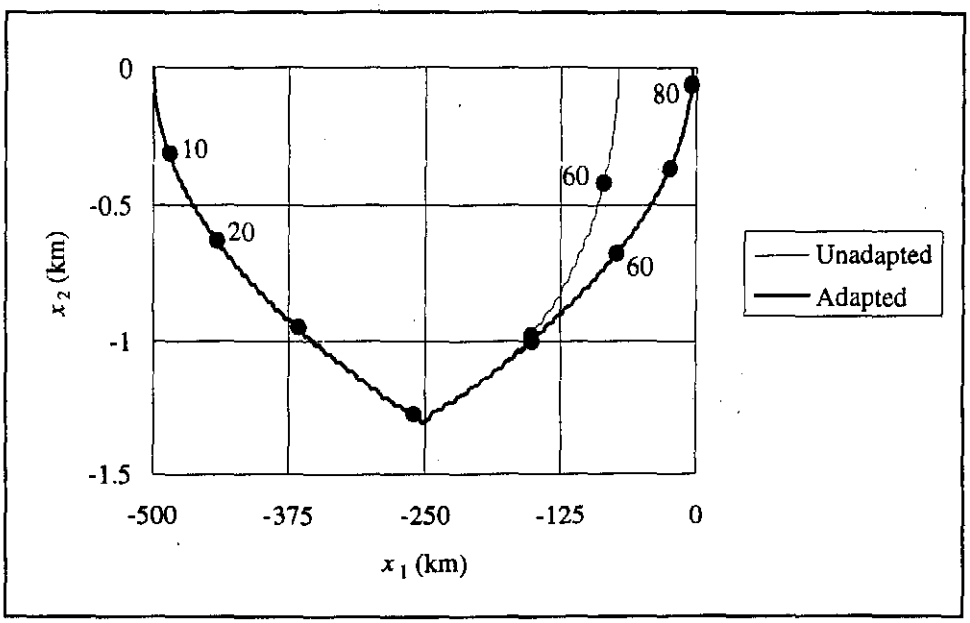


Figure 4.24: Adaptive bang-bang acquisition – state space.

The deviation from the optimal switching curve is obvious when no adaptive control is done. The terminal error in this case is approximately 73 km (14.6%). The adaptive scheme reduces the error to 2.9 km (0.64%).

The estimate of k_1 is shown in figure 4.25. The glitch just before the variation is a result of the discontinuity in the cross-sectional area (input to the RLS algorithm) when the satellite switches from the high-drag to the low-drag orientation.

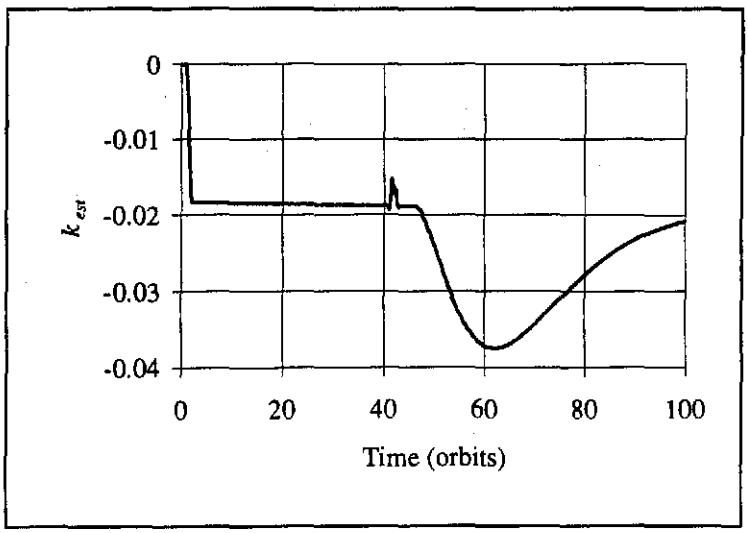


Figure 4.25: Estimate of k_1.

The adapted control signal (u_a) is shown in figure 4.26.

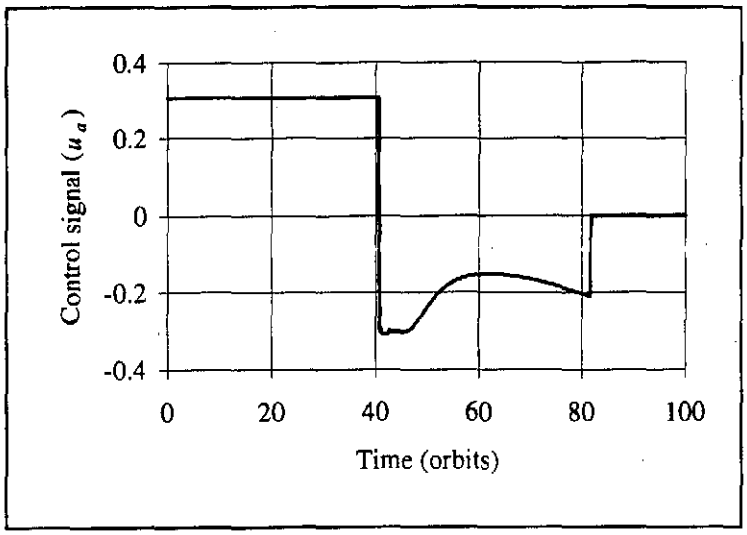


Figure 4.26: Adapted control signal.

5.4.2 Bang-off-bang control

The state space results are shown in figure 4.27. The density variation started after 85 orbits (figure 4.28) to coincide with the final phase of the control effort.

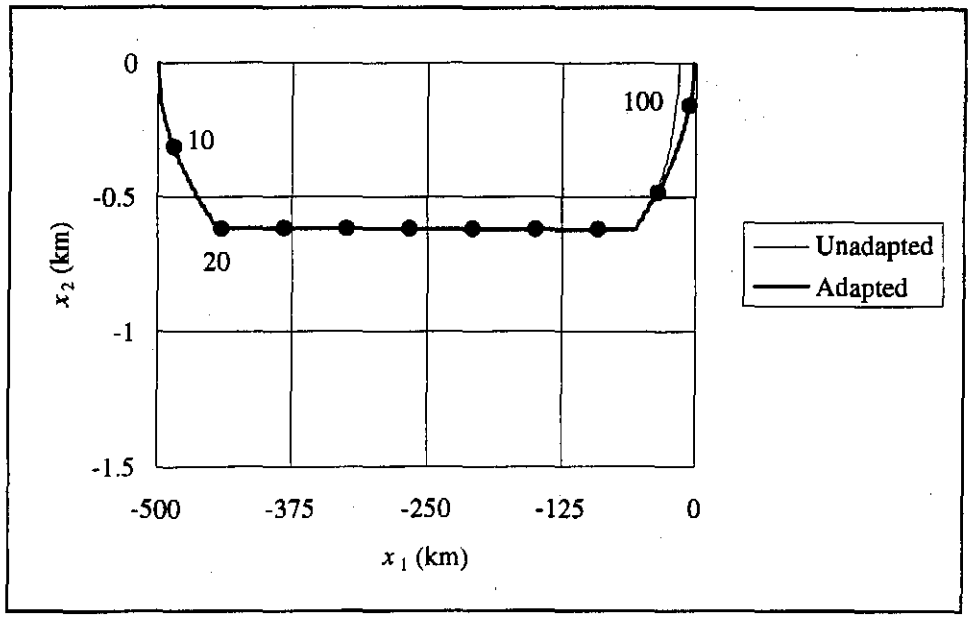


Figure 4.27: Adaptive bang-off-bang acquisition – state space.

The terminal error is in this case reduced from 16.3 km (3.6%) to 2.3 km (0.5%).

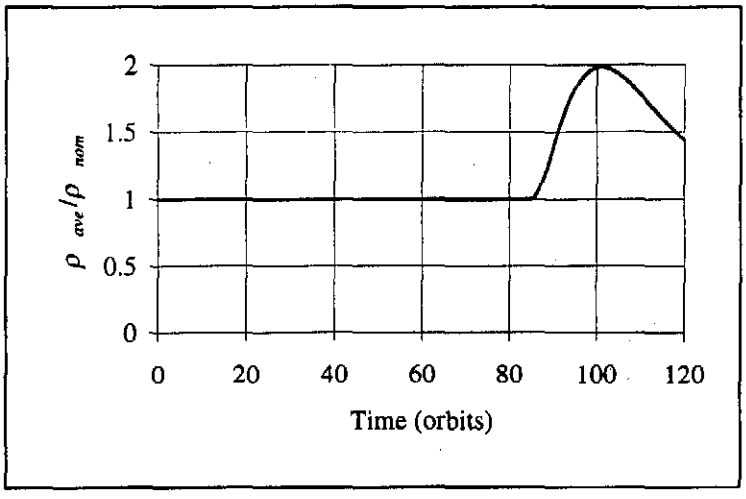


Figure 4.28: Variation in atmospheric density.

Figure 4.29 shows the parameter estimate and figure 4.30 the adapted control signal.

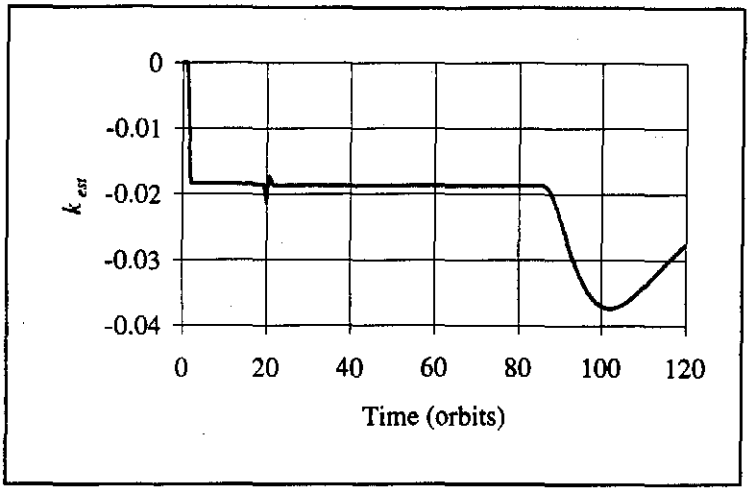


Figure 4.29: Estimate of k_1.

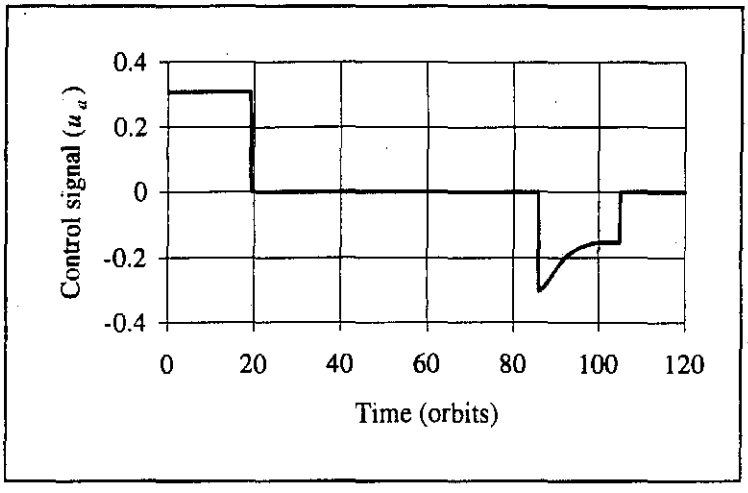


Figure 4.30: Adapted control signal.

6. CONSTELLATION ACQUISITION IN ELLIPTICAL ORBITS

Up until now, only circular orbits were considered. In this section, the proposed concepts and designs will be extended to elliptical orbits. Although an elliptical orbit is not desirable for a gravity mapping mission like CHIPSAT, the consideration of this case is still relevant because the available launch opportunity might not leave the satellites in a perfectly circular orbit. Some missions might also require a slightly elliptical orbit so that the atmosphere can be sampled over a range of altitudes. It is thus necessary to verify the application of the proposed concepts for these orbits.

6.1 THEORY

The earlier analyses and designs, based on the model of figure 4.9, hinge on the approximation that the variations in altitude and relative distance between satellites can be linearised if averaged over the full orbital period. This approximation remains valid for elliptical orbits. The instantaneous variations in velocity and altitude due to the orbit's ellipticity have a characteristic period equal to the orbital period. When averaged over the orbit, the altitude of the satellite in the elliptical orbit still decreases approximately linearly if observed over the relatively short time duration of the typical constellation acquisition phase. This implies that the model used before can be used in exactly the same way. The model parameters k_1 and k_2 will have different values than before.

The instantaneous variations in altitude and velocity are however much larger in an elliptical orbit than in a circular orbit. If two satellites are in the same elliptical orbit, but separated by some distance, this instantaneous distance will also vary much more than before. The large variation in the instantaneous altitude also causes a large variation in the drag acceleration magnitude, due to the exponential relationship between atmospheric density and altitude. Figure 4.31 shows the variation in the magnitude of the perturbation acceleration due to atmospheric drag on the nominal satellite in the nominal elliptical orbit (450 km perigee and e = 0.02), with the drag area equal to A_{max}.

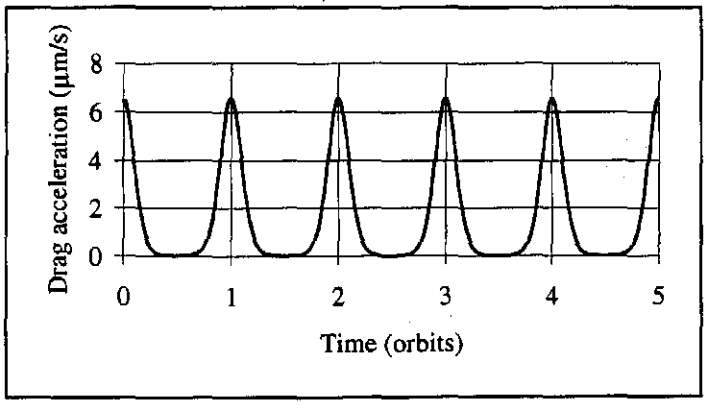


Figure 4.31: Drag acceleration magnitude in elliptical orbit ($A=A_{max}$).

It is clear that the exponential altitude dependence of the density causes the atmospheric drag perturbation to vanish almost completely near the apogee. The influence of atmospheric drag is thus only significant near the perigee. For a perigee

height equal to the altitude of the nominal circular orbit used until now (450 km), it can thus be expected that the orbit-average effect of drag on the satellite in the elliptical orbit will be smaller than in the circular orbit. Intuitively, this would result in a smaller parameter k_1 and longer control times.

Following the observation that the effect of atmospheric drag is very small near the apogee, the question can be asked whether it is worthwhile applying control to the satellites in this region. Clearly it could be a waste of on-board energy if the cross-sectional areas perpendicular to the velocities – and hence orientations – of the satellites were controlled where there is a minimal atmospheric drag effect. Although perfectly valid, this particular concern has little or no bearing on the constellation acquisition phase. The reasons are as follows:

1. In the case of an unadapted control signal (when the atmosphere can be assumed stationary), both the optimal control strategies guarantee a minimum number of orientation "switches" for the constellation to acquire the specified new configuration. Each satellite is only required to slew once from low-drag orientation to high-drag orientation and once from high-drag orientation to low-drag orientation. Whether these orientation changes take place near the apogee or not has no implication on the usage of on-board energy.

2. If the control signal is adapted, the cross-sectional areas of the satellites can change continuously, depending on the estimated value of the atmospheric density. This estimate will not vary significantly on time scales shorter than the orbital period, for the following two reasons: firstly because the value is averaged over the full orbit and it is also calculated from orbit-averaged values; secondly because the atmospheric density characteristics cannot vary rapidly. Even the fastest variations in the atmospheric density characteristics take place over several hours, corresponding to a number of orbits. The cross-sectional area will therefore not be changed rapidly by the control algorithm on time scales shorter than the orbital period and on-board energy is not wasted. The typical slow change in the cross-sectional area of a particular satellite during apogee passage is necessary to create the correct *orbit-average* drag effect.

6.2 IMPLEMENTATION AND SIMULATION

It has been found that the large oscillations in the instantaneous values of the altitude difference and the straight-line distance between the satellites in the elliptical orbit cause some significant oscillation to be left even after the orbit-averaging process. This leads to similar oscillations in the "instantaneous orbit-averages", calculated by the linear and quadratic predictions of equations (4.3) and (4.5). To overcome this problem, the output values from the sliding orbit-average filter are filtered for a second time by the same filter. This introduces another half-orbit delay for a total delay of one orbital period. To calculate the "instantaneous" orbit-averages, the linear and quadratic predictions of equations (4.3) and (4.5) are modified to predict a full orbit ahead:

$$\Delta h = b_h(n+200)+c_h \tag{4.91}$$

$$d = a_d(n+200)^2 + b_d(n+200)+c_d \tag{4.92}$$

The values of the constants are calculated as before.

To calculate the optimal control, the model parameters k_1 and k_2 must be determined. The identification procedure is the same as in section 2.4 and will thus not be described here. Additional information is given in appendix A. The identification yields:

$$\begin{aligned} k_1 &= -0.00841 \\ k_2 &= -0.00163 \end{aligned} \tag{4.93}$$

Notice that k_1 is now much smaller than before. This implies that the average altitude reduction per orbit due to atmospheric drag is lower than in the circular orbit. As mentioned before, this could be expected, since the drag perturbation is smaller in this particular elliptical orbit. The value of k_2, relating the altitude difference to the rate of distance variation, remains more or less the same as before.

Figure 4.32 shows the phase-plane results of a time-optimal constellation acquisition effort for the nominal elliptical orbit. The specified distance is again 500 km.

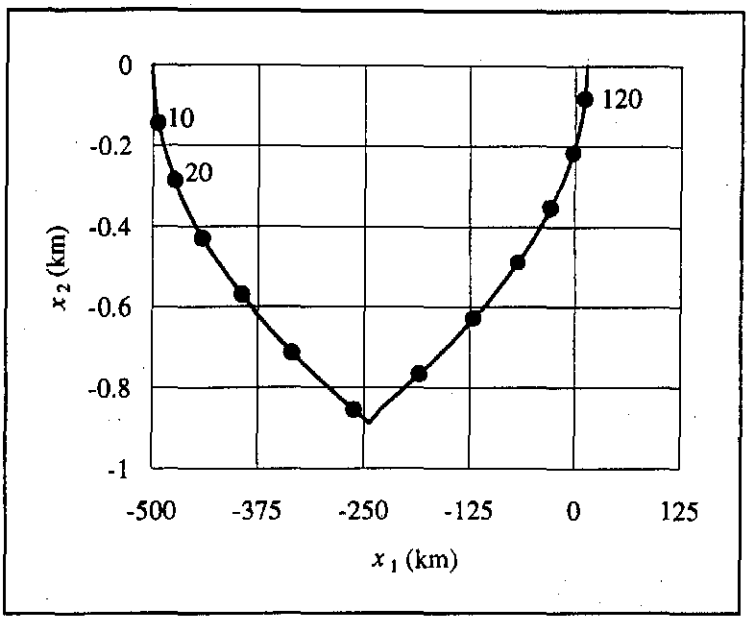


Figure 4.32: Time-optimal constellation acquisition in an elliptical orbit – state space.

The final orbit-average distance exceeds the specified distance by 13.8 km (2.76%). The oscillation in the instantaneous distance has a peak to peak magnitude of approximately 20 km (4%) at the end of the control effort.

The time of the first control phase (T_1) equals 61.8 orbits and T_2 equals 64.4 orbits.[14] This gives a total control time of 126.2 orbits (≈ 8 days, 10 hours). The theoretical value, using equation (4.56), is 119.11 orbits. The control time is roughly about 50% longer than before.

It must be kept in mind that the phase plane plot of figure 4.32 shows the orbit-averaged values and thus hides the instantaneous variations in the movement of the satellites. Figure 4.33 (next page) shows the instantaneous and orbit-averaged altitude difference (S_1's altitude minus S_2's altitude) for the first few orbits of the control effort. It demonstrates the underlying variations on time scales shorter than the orbital period. Note that these variations are not a result of the characteristic dynamics of the system,

---

[14] When considering times, it must be kept in mind that the orbital period is not the same as for the circular orbit. It is now 5778.9 sec, which is about 3% longer than before.

but can be considered as the dynamics due to the "forcing function" from the elliptical orbit. The two satellites are initiated with the same position and velocity, but start to move apart because S_1 is switched to the high-drag orientation while S_2 remains in the low-drag orientation. The two satellites start at the perigee. Notice that the altitude difference undergoes the most significant change at the apogee. This is consistent with the theory that atmospheric drag reduces the apogee height much faster than the perigee height in elliptical orbits. Near the perigee, the altitudes of the two satellites remain close to each other.

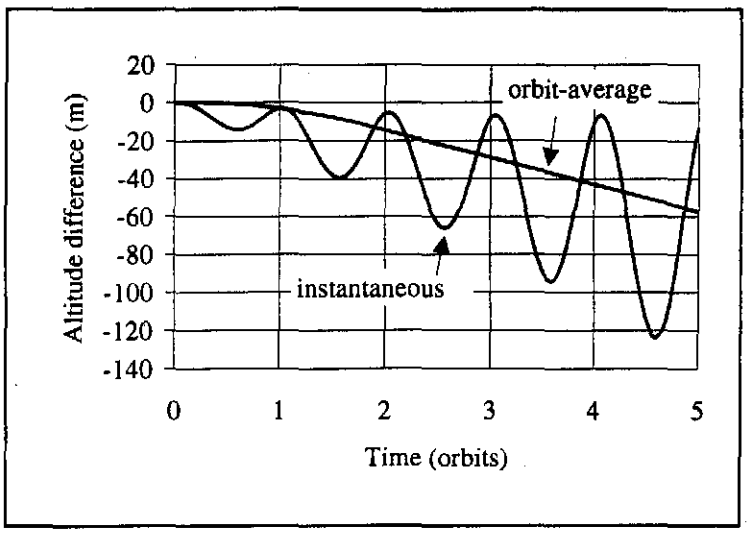


Figure 4.33: Altitude difference between the two satellites.

7. CONSTELLATION ACQUISITION FOR MULTI-SATELLITE CONSTELLATIONS

This section will address the problem of finding the optimum strategy to reach a specified constellation from some initial configuration for constellations with more than two satellites. Optimality will once again be measured in terms of control time and altitude loss.

If a number of satellites have the same initial altitude,[15] with constant separation distances between them and a new set of separation distances is specified, all the satellites must lose the same amount of altitude during the constellation acquisition process, so that they end up in the same orbit and the new set of separation distances can be constant.

The key to reaching to correct final configuration is to time the start of each high-drag manoeuvre correctly. A general method to determine the time at which each satellite must switch to the high-drag orientation will be developed in the following sections.

The linear, time-invariant model of (4.27) to describe the relative movement between two satellites will again be used. The adaptive scheme of section 5 can be applied to each pair of controlled satellites to handle temporal variations in the atmospheric density.

7.1 TIME-OPTIMAL CONTROL

Consider an arbitrary pair of satellites (say S_i and S_j) in a multi-satellite tandem constellation.[16] Assume that the initial orbit-average distance between them is d_{ij}. This distance will be defined as positive if S_i is in front of S_j and negative if S_i is behind S_j. If a new distance (d_{ij_ref}) is specified, the required distance change (or increment) is:

$$\Delta d_{ij} = d_{ij\_ref} - d_{ij} \tag{4.94}$$

Note that this increment can be positive or negative. The following can easily be shown: if $\Delta d_{ij} > 0$, S_i must switch to the high-drag orientation first and S_j second, regardless of which one is initially in front. If $\Delta d_{ij} < 0$, the order is always reversed.

The required distance increment can be determined for every combination of two satellites in the constellation. The largest relative movement between any two

---

[15] The initial conditions of (4.55) will be assumed throughout this section.

satellites will correspond to the increment with the largest magnitude. This largest relative movement is important since it determines the shortest possible control time for the specified constellation change. Any change in separation distance with a smaller magnitude can theoretically be achieved in a shorter control time.

Let S_A and S_B be the two satellites associated with the largest required increment in orbit-average distance (Δd_{AB}). Define S_A as the satellite to enter the high-drag orientation first. The increment Δd_{AB} will thus always be positive. If all the other satellites in the constellation are ignored for the moment, it is easy to calculate the time-optimal control signal between S_A and S_B, as it has been done in section 3.2. In this case the initial state is:

$$\begin{aligned} x_{10} &= -\Delta d_{AB} \\ x_{20} &= 0 \end{aligned} \tag{4.95}$$

Both S_A and S_B will (ideally) spend the same time (T_1) in the high-drag orientation. This time can be obtained from (4.57):

$$T_1 = \sqrt{\frac{\Delta d_{AB}}{k_1 k_2 M}} \tag{4.96}$$

Thus, S_A will be in the high-drag orientation for the time interval $[0, T_1]$ and S_B will be in the high-drag orientation for the time interval $[T_1, 2T_1]$. The total control time (t_f) equals $2T_1$.

Consider now the other satellites in the constellation. The required distance increment between S_A and every other satellite can be calculated from the given initial and specified final constellation configurations. Say the required increment between S_A and some satellite S_i is Δd_{Ai}. It can also be shown that since $\Delta d_{AB} > 0$, Δd_{Ai} will be positive for any satellite S_i. Further, since $\Delta d_{Ai} < \Delta d_{AB}$, the satellite S_i must switch to the high-drag orientation before S_B does. Thus S_A switches first, followed

---

[16] Note that there can be other satellites between S_i and S_j.

chronologically by all the other satellites of the constellation. The order of switching will correspond with increasing magnitudes of required distance increments. S_B will switch last since $\Delta d_{AB} > \Delta d_{Ai}$ for all satellites S_i. All the satellites will stay in the high-drag orientation for a time equal to T_1.

Say satellite S_i switches to the high-drag orientation after a time T_i, with $T_i < T_1$. The control signal between S_A and S_i can thus be described by:

$$u_{Ai} = \begin{cases} M & \text{for} \quad 0 \le t < T_i \\ 0 & \text{for} \quad T_i \le t < T_1 \\ -M & \text{for} \quad T_1 \le t < T_1 + T_i \end{cases} \tag{4.97}$$

During the time that $u_{Ai} = 0$, both S_A and S_i are in the high-drag orientations. If S_A and S_i are viewed in isolation, the control is clearly not optimal anymore.[17] For the constellation as a whole, however, there exists no strategy to acquire the target configuration from the initial configuration in a shorter time.

The time T_i can be determined through analytical evaluation of the state x_2, using equation (4.36), at the time $T_1 + T_i$:

$$\begin{aligned} T_i &= \frac{\Delta d_{Ai}}{k_1 k_2 M T_1} \\ &= \frac{\Delta d_{Ai}}{\sqrt{\Delta d_{AB} k_1 k_2 M}} \end{aligned} \tag{4.98}$$

The last expression comes from the substitution of T_1 given by (4.96). The time-optimal control for the constellation is thus completely determined. To summarise: the boundary pair, S_A and S_B, can be controlled by the controller of (4.45), with S_A entering the high-drag orientation first and the switch occurring when the optimal switching curve (4.43) is reached. All the other satellites must switch to the high-drag

---

[17] It is interesting to note that although the control now has a *bang-off-bang* form, it is not altitude optimal, since the *off* ($u = 0$) is a result of both the satellites being in the high-drag orientation and not both being in the low-drag orientation.

orientation at the times T_i, given by (4.98) for the respective increments Δd_{Ai}. These other satellites are switched back to the low-drag orientation when they reach the altitude of S_A. This should theoretically correspond to all high-drag times being equal to T_1.

7.2 ALTITUDE-LOSS-OPTIMAL CONTROL

The arguments leading to the altitude-loss-optimal control for the constellation are almost identical to those in the previous section. The distance increment with the largest magnitude determines the minimum amount of altitude loss for the constellation acquisition effort. All satellites will lose the same amount of altitude. The control has a *bang-off-bang* form and the times T_i can theoretically be larger than T_1, depending on the values of the required increments Δd_{Ai}. The times T_i are now given by:

$$T_i = \frac{\Delta d_{Ai}}{\sqrt{\frac{\Delta d_{AB} k_1 k_2 M A_{\min}}{A_{\max}}}} \qquad \textbf{(4.99)}$$

This is obtained by substituting the value of T_1, given by (4.75), in the first expression of (4.98). Once again, isolated pairs of satellites will, in general, not be controlled altitude-loss-optimally, but the strategy for the constellation as a whole is altitude-loss-optimal.

7.3 IMPLEMENTATION AND SIMULATION

The following example will demonstrate both optimal control strategies for a constellation of three satellites (S_1, S_2 and S_3). The three satellites were initiated with the same position and velocity vectors and two successive constellation acquisitions were done – the first time-optimally and the second altitude-loss-optimally.

The first constellation specification required S_1 to be in front of the constellation, followed by S_2, 100 km behind S_1 and then S_3, another 300 km behind S_2. The

acquisition phase starts immediately and is done time-optimally. The largest increment is clearly between S_1 and S_3, so that S_1 will enter the high-drag orientation first and S_3 last, with S_2 in between.

The second acquisition phase starts after 80 orbits and is done in an altitude-loss-optimal way. The requirement is for S_1 to stay in front, followed by S_2, now 300 km behind S_1 and then S_3, 200 km behind S_2. The largest increment is now in the distance between S_1 and S_2, thus S_1 enters the high-drag orientation first, followed by S_3 and S_2, in that order.

Figure 4.34 shows the orbit-average distance between S_1 and S_2 (d_{12}) and between S_1 and S_3 (d_{13}). The changes in the cross-sectional areas of the three satellites are shown in figure 4.35.

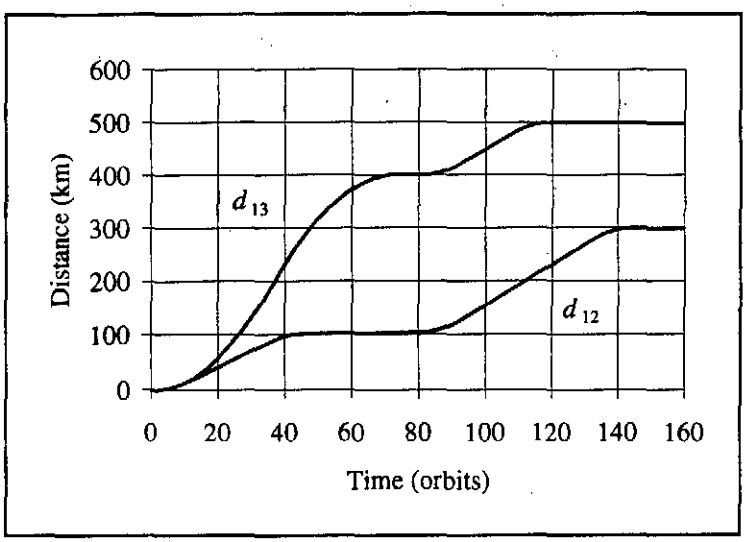


Figure 4.34: Multi-satellite optimal constellation acquisition – distances.

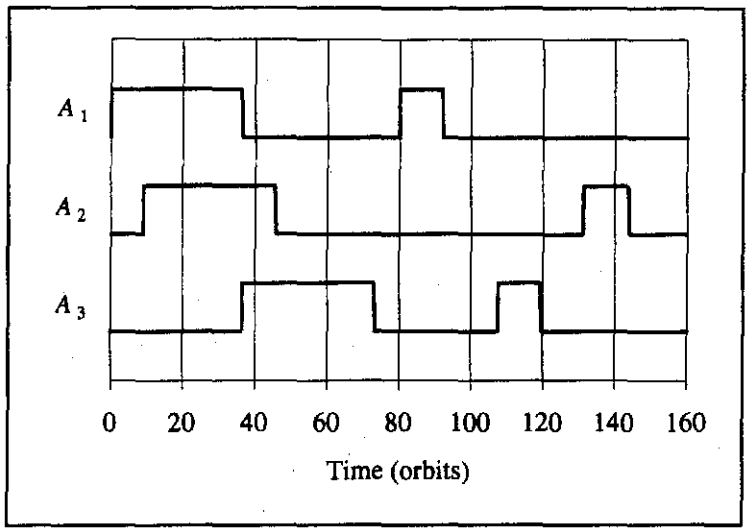


Figure 4.35: Multi-satellite optimal constellation acquisition – cross-sectional areas.

From figure 4.35 it is clear that the first acquisition between S_1 and S_3 (the largest increment) is done time-optimally. S_3 switches to high-drag as S_1 switches to low-drag (bang-bang). Note that S_2 switches before S_3, so that the acquisition phase between S_1 and S_2 is finished in a shorter time than the control time between S_1 and S_3.

For the second acquisition phase, the relative movement between S_1 and S_2 is the largest and determines the minimum altitude loss. The switching order S_1, S_3, S_2 is clear from figure 4.35, now with a delay between S_1 and S_3 for the altitude-loss-optimal bang-off-bang controller.

For both acquisition phases, the high-drag times of the three satellites are equal, implying equal altitude losses.

Chapter 5

CONSTELLATION MAINTENANCE

1. INTRODUCTION

The optimal strategies that were designed in the previous chapter will control the constellation of satellites from a certain initial configuration to a specified final configuration. At the end of the constellation acquisition effort, the various orbit-average distances between the satellites might not be exactly as specified due to a variety of factors, including model inaccuracies and external disturbances. The same factors might cause the orbit-average distances between the satellites to drift slowly during the normal operational phase of the constellation. It is thus necessary to design a constellation maintenance strategy, that can be activated at any time after or between constellation acquisition phases, to keep the constellation in the reference configuration.

The design process will again be simplified by initially considering a single pair of satellites in a circular orbit and a stationary atmosphere. The theory of linear quadratic control[1] will be used to design a near-optimal regulator to keep the

---

[1] The theory of linear quadratic control is well documented in many handbooks. See for example Dorato *et al* (1995).

constellation in the specified configuration (section 2). The regulator will also be made adaptive to accommodate a non-stationary atmosphere with unpredictable time-variations in density (section 3). As before, the philosophy will be to estimate the changing model parameters and to update the design of the regulator continuously. The application of the proposed control strategy will also be extended to the case of an elliptical orbit (section 4). Finally, in section 5, multi-satellite constellations will be considered. Certain steps must be taken to decouple the regulator so that the orbit-average distance between neighbouring pairs of satellites can be maintained without cross-coupling effects.

The type of constellation maintenance proposed in this chapter can be classified as relative constellation maintenance, because only the relative positions of the satellites are controlled. The constellation as a whole will continue to lose altitude due to atmospheric drag. If it is required that the absolute constellation configuration must be maintained, additional energy must be supplied to the orbit (typically by thrusters) to increase the altitude from time to time.[2]

2. LQR REGULATION

2.1 LINEAR MODEL

In the previous chapter it was shown that the half-orbit delay in the calculated orbit-average altitude difference and orbit-average distance between two satellites can be eliminated if the cross-sectional areas of the satellites perpendicular to their velocity vectors are piece-wise constant. This led to the following linear model to approximate the relative movement between the two identical satellites in the same orbit:

---

[2] See Collins *et al* (1996) for absolute station-keeping in a tandem circular orbit.

$$\dot{\mathbf{x}} = A\mathbf{x} + Bu$$
$$= \begin{bmatrix} 0 & k_2 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ k_1 \end{bmatrix} u \tag{5.1}$$

with the state vector chosen as:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} d - d_{ref} \\ \Delta h \end{bmatrix} \tag{5.2}$$

During the constellation maintenance phase, the cross-sectional areas perpendicular to the velocity vectors will no longer be piece-wise constant. The above model will however still be used, thus simply ignoring the half-orbit delay in the calculated orbit-averages. The resulting error should be small as long as the dynamic response of the closed-loop control system is relatively slow – i.e. much longer than half the orbital period.

2.2 REGULATOR DESIGN

The proposed regulator is a full state-feedback control law:

$$u = -K\mathbf{x} \tag{5.3}$$

The gain matrix K must be determined to minimise the following performance index, or cost function:

$$J = \int_0^{t_f} \left[\mathbf{x}^T Q \mathbf{x} + \mathbf{u}^T R \mathbf{u}\right] d\tau \tag{5.4}$$

The matrix Q is typically real, symmetric and positive semi-definite and determines the relative weighting of the state during the optimisation process. The matrix R is real, symmetric and positive definite and determines the relative weighting of the control vector, which is a scalar in this case. Each choice of Q and R will result in a different optimal controller. A physical interpretation of the cost function is that the state must be kept close to the origin without an excessive control effort. The latter

restriction is important, not only to prevent the control signal from saturation, but also to restrict the satellites from large orientation changes during the maintenance phase. This conserves on-board energy as well as altitude. A small control signal also implies a slow dynamic response, which is necessary to justify the approximation that the half-orbit delay in the orbit-average values can be neglected.

The optimisation can be done with the LQR2 function of MATLAB. This function uses the *Schur* algorithm to find the steady-state value of K that minimises J, subjected to the constraints (5.1) and (5.3). The following choice of weighting matrices produced an acceptable controller:

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$
$$R = 1.5 \times 10^{9} \qquad \textbf{(5.5)}$$

The fact that the diagonal elements of Q are the same size, implies that equal weighting is given to the two state variables. The distance and altitude difference are thus considered equally important in minimising the terminal accuracy. The reason for the large choice of R is to ensure a slow dynamic response with a control signal that does not saturate.

The resulting gain matrix is:[3]

$$K = 10^{-5} \times \begin{bmatrix} 2.582 & -216.56 \end{bmatrix} \qquad \textbf{(5.6)}$$

The values of the model parameters k_1 and k_2 were taken as identified in the previous chapter for the circular orbit (k_1 = -0.01841 and k_2 = -0.00167).

2.3 IMPLEMENTATION AND SIMULATION

The linear quadratic regulator of the previous section was implemented and simulated with the simple linear model of equation (5.1), as well as with the full special

perturbations simulation of the two-satellite constellation. The initial offset in the orbit-average distance between the satellites was 10 km.

The outputs from the linear model are shown as the thin lines in the graphs of figures 5.1 to 5.4.

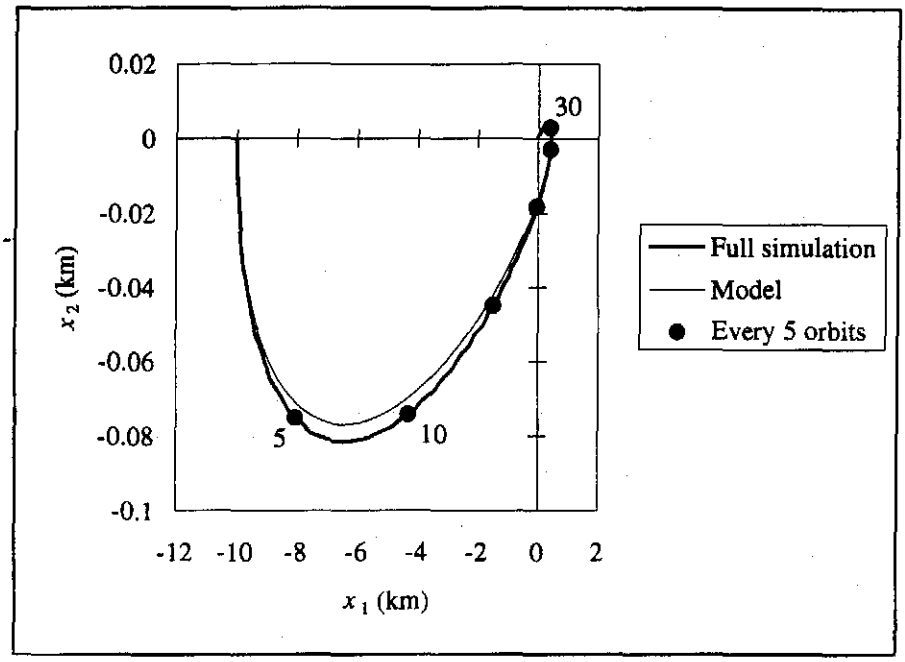


Figure 5.1: LQR constellation maintenance – state space.

Figure 5.1 shows the state trajectories. The black dots are now spaced at intervals of 5 orbits. There is a slight difference between the ideal (thin) and the "real" (heavy) trajectory, mainly because the half-orbit delay was ignored. The difference is small enough that the two dynamic responses are almost identical. This can also be seen from the time-waveforms of the two state variables x_1 (orbit-average distance offset) and x_2 (altitude difference) in figures 5.2 and 5.3 (next page).

---

[3] See appendix D for the MATLAB script file.

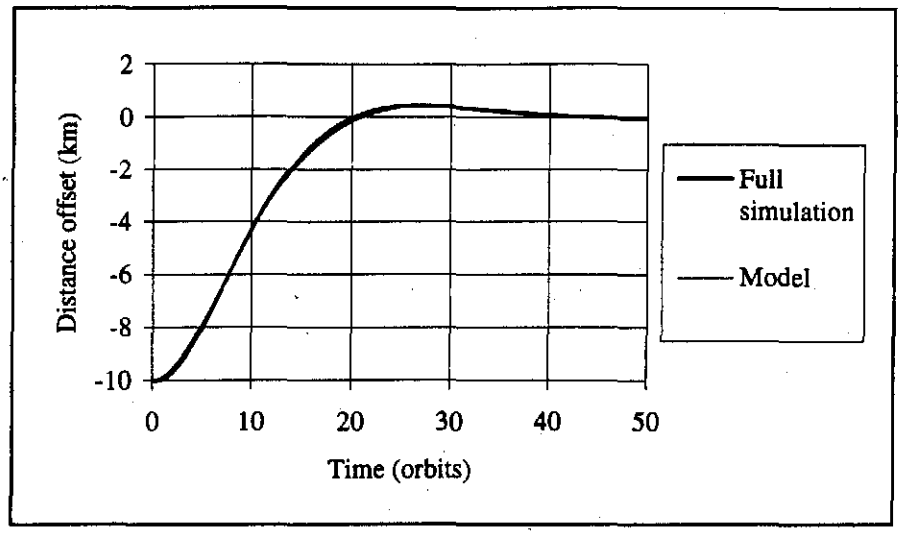


Figure 5.2: LQR constellation maintenance – orbit-average distance offset (x_1).

The 1% settling time in the orbit-average distance is 39.75 orbits for the full simulation and 41.75 orbits for the ideal model. The overshoot is 4.48% and 4.32% respectively.

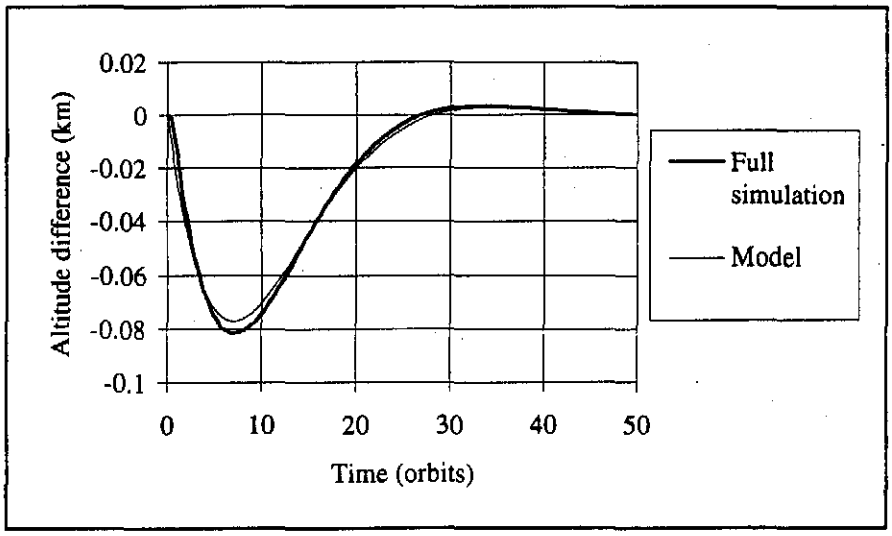


Figure 5.3: LQR constellation maintenance – altitude difference (x_2).

The control signal (u) is shown on the next page in figure 5.4. The maximum value (0.258) is 84% of the saturation value (M = 0.307). For the current regulator, the control signal starts to saturate when the distance offset (error) exceeds 12 km. This range can be increased by designing the regulator with more weight on the control signal, thus increasing the value of the weighing factor R in equation (5.4). The penalty will be a slower response (longer settling time).

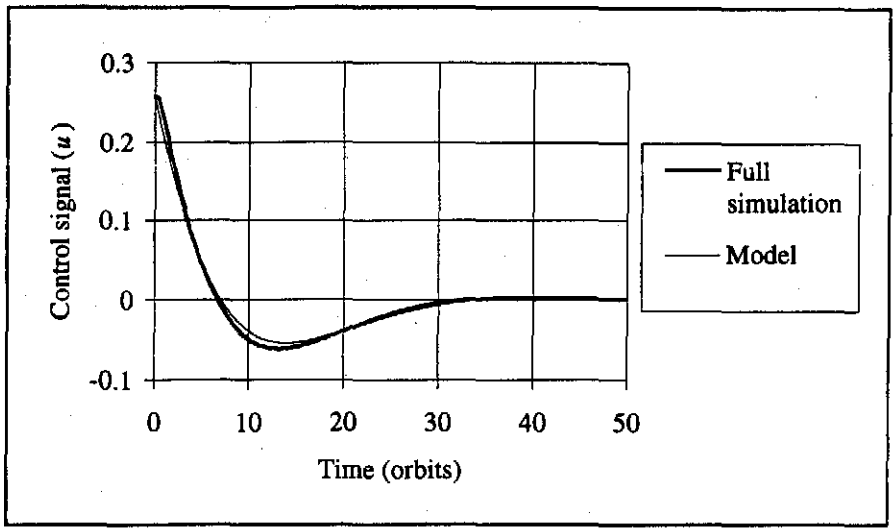


Figure 5.4: LQR constellation maintenance – control signal (u).

The dynamic response of the regulator, implemented with the linear model of (5.1), is optimal in the sense that it minimises the cost function J, given by (5.4). If this same regulator is implemented with the full non-linear simulation, the dynamic response is very close to the ideal linear case, as shown in the above example. The regulator will thus be considered as near-optimal and sufficient for the purposes of constellation maintenance.

The typical application of the feedback regulator can be to eliminate the terminal error in the orbit-average distance at the end of a constellation acquisition phase. It can then be considered as the final phase in a dual-mode control scheme. It can be switched off at any time, for instance if the orbit-average distance error is less then a certain margin. If this error margin is exceeded again, the regulator can again be activated.

3. ADAPTIVE CONSTELLATION MAINTENANCE

It was shown in chapter 4 that a temporal variation in the atmospheric density causes a variation in the parameter k_1 of the linear model of equation (5.1). This will result in a change in the dynamic response of the constellation maintenance regulator. A decreasing density will lead to more overshoot and longer settling times. If the density increases, it might lead to undesired limit cycles if the control signal starts to

saturate. Note that the changing atmospheric density does not appear as an external disturbance and it would therefore still be possible to drive the distance error to zero.

The effects of a time-varying density can be countered in the same way as in chapter 4, i.e. by scaling the control signal with an estimated value of k_1. The state feedback control law (5.3) is modified to:

$$u = -\left(\frac{k_{10}}{k_{est}}\right)K\mathbf{x} \tag{5.7}$$

where k_{10} is the original value of k_1 (used to design the gain matrix K) and k_{est} is the estimated value of k_1. This estimate can be obtained with the same RLS algorithm as in chapter 4. The value of k_{10} can also be fixed just before each maintenance phase, so that the long-term variations in the atmospheric density are tracked.

A simulation was done to demonstrate the adaptive constellation maintenance scheme. Figure 5.5 shows the (normalised) variation in the atmospheric density. The same typical short-term variation has been used as in chapter 4, starting after 2 orbits.

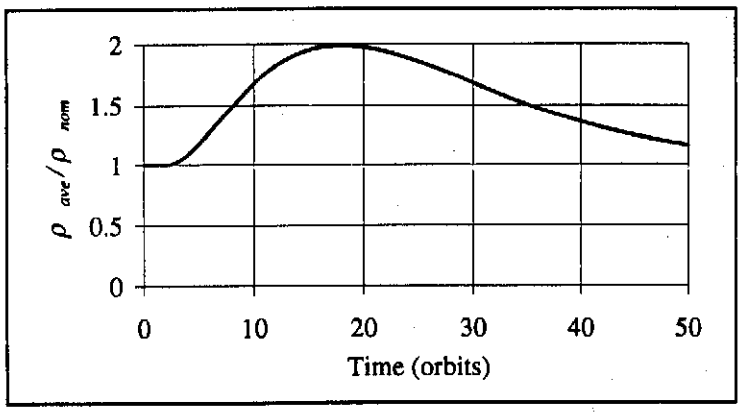


Figure 5.5: Adaptive constellation maintenance – density variation.

The estimated value of k_1 is plotted in figure 5.6.

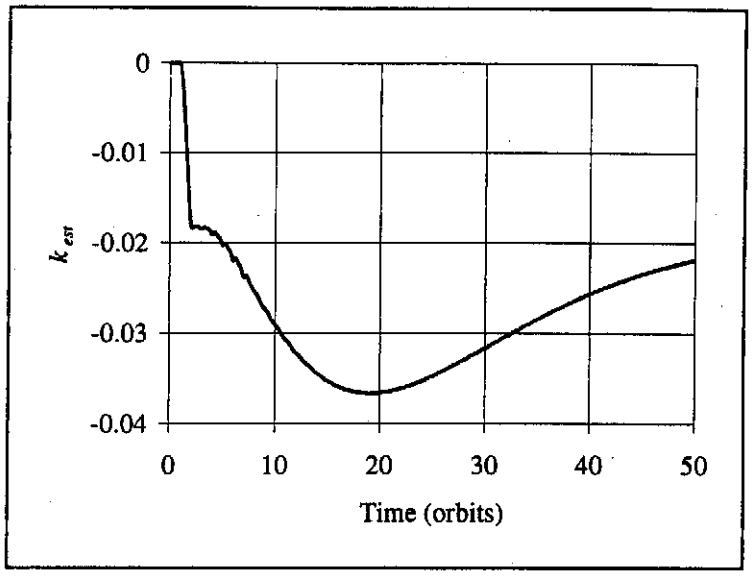


Figure 5.6: Adaptive constellation maintenance – estimated parameter k_1.

The simulation was repeated twice, once with no adaptation of the control signal and once with the adapted control signal of equation (5.7).

The two control signals are shown in figure 5.7. The thin line is the unadapted control signal and the heavy line is the adapted control signal.

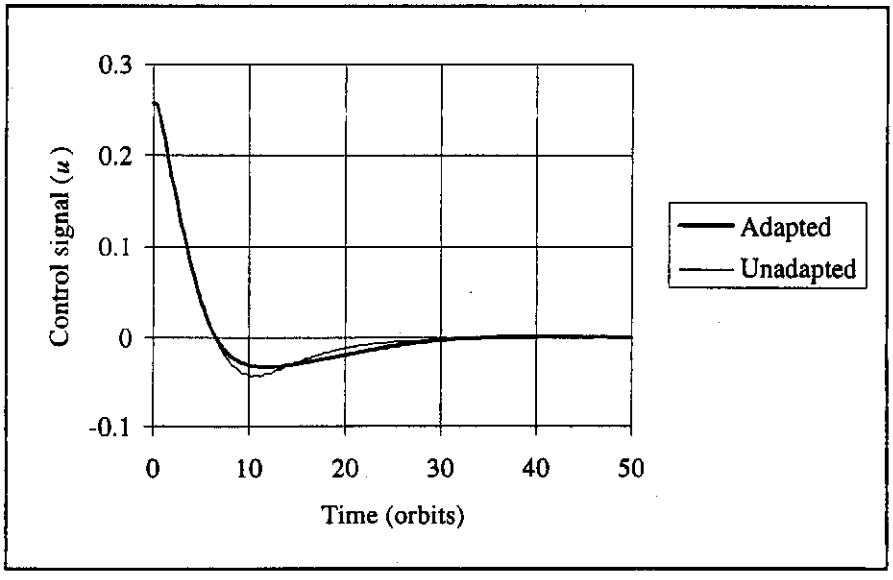


Figure 5.7: Adaptive constellation maintenance – control signal (u).

The state-space trajectories are shown in figure 5.8. The thin lines are for the unadapted case. The difference in the dynamic response is clear.

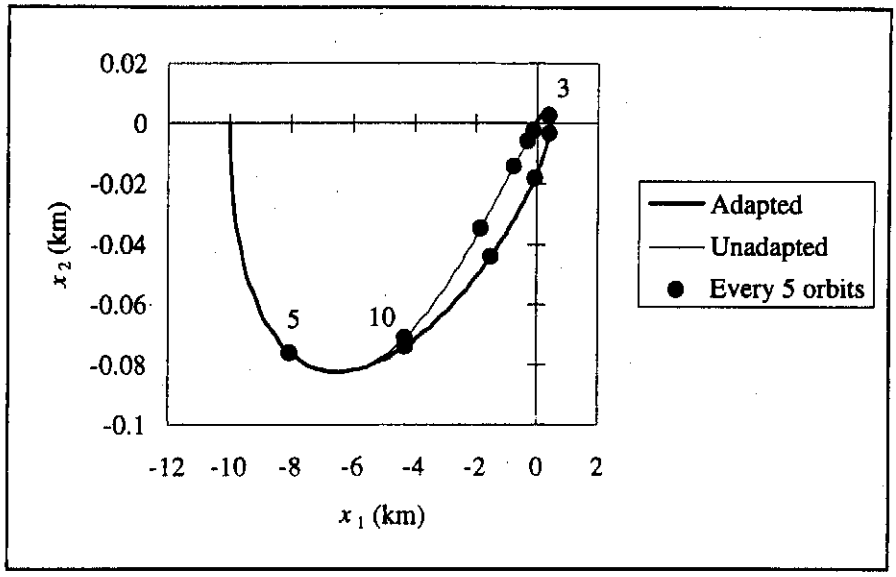


Figure 5.8: Adaptive constellation maintenance – state space.

The time-waveforms of the two state variables are shown in figures 5.9 and 5.10 (on the next page).

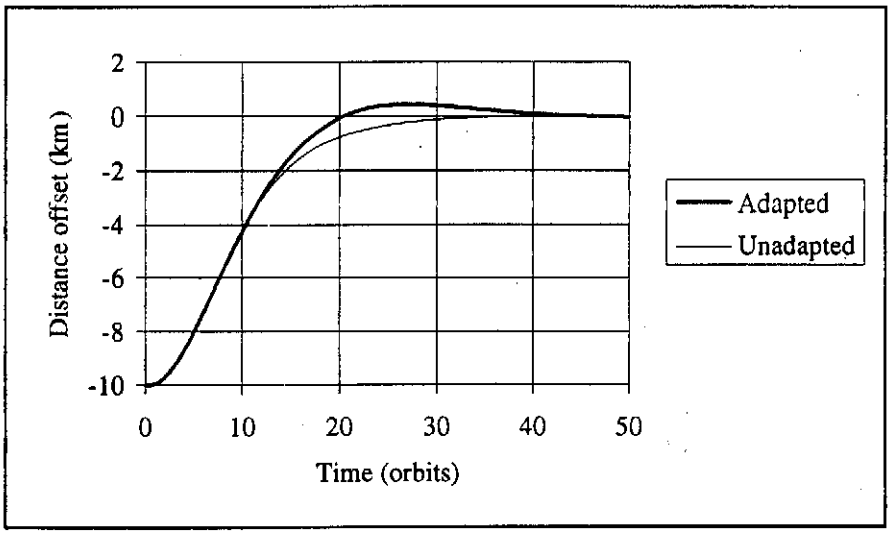


Figure 5.9: Adaptive constellation maintenance – orbit-average distance offset (x_1).

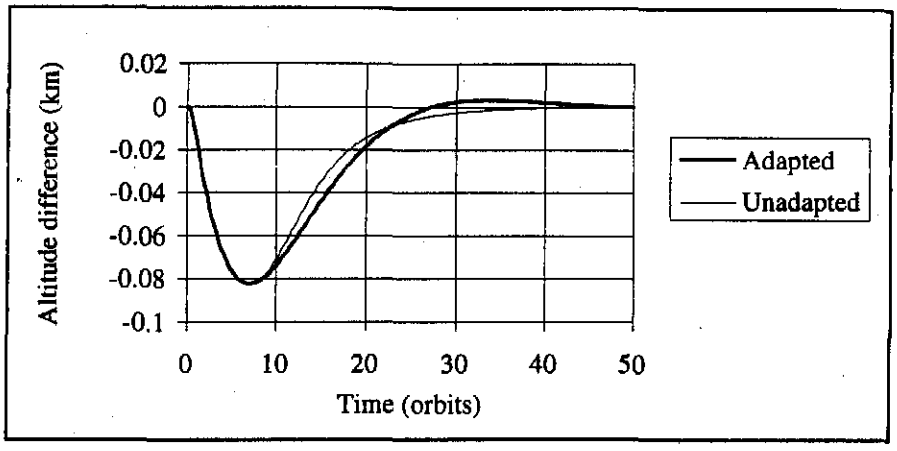


Figure 5.10: Adaptive constellation maintenance – altitude difference (x_2).

The overshoot in x_1 (orbit-average distance offset) for the adapted case is 4.29% and the settling time (1%) is 39.875 orbits. This compares very well with the 4.48% overshoot and 39.75 orbits settling time obtained earlier in section 2.3, when the possible time-variation in the density was ignored.

The above example pointed out that variations in the operating conditions are not necessarily critical to the success of the maintenance strategy. The unadapted strategy also succeeded in driving the error to zero (figures 5.8 to 5.10). The application of the adaptive strategy during the maintenance phase can therefore be considered as optional, especially during relative small changes in the operating conditions. It does however provide the means to maintain a desired dynamic response if required.

4. CONSTELLATION MAINTENANCE IN ELLIPTICAL ORBITS

As far as the design of the constellation maintenance strategy is concerned, there is no difference between circular and elliptical orbits. Using the values of k_1 and k_2 obtained in chapter 4 for the 450 km to 728 km (e = 0.02) elliptical orbit, with the values of the weighting matrices Q and R as before, the following feedback gain matrix is obtained:

$$K = 10^{-5} \times \begin{bmatrix} 2.582 & -316.6 \end{bmatrix} \qquad \textbf{(5.8)}$$

The above maintenance regulator was implemented in a dual-mode simulation.

The first phase was the constellation acquisition phase, demonstrated by the simulation in chapter 4. At the end of the acquisition phase, which lasted 126.2 orbits, the error in the orbit-average distance was 13.8 km. The two satellites were then controlled with the above regulator to eliminate the steady-state error. The graph of figure 5.11 shows the state-space trajectory during the maintenance phase. The system performs satisfactory, reducing the steady-state error in the orbit-average distance to less than 1% within the time of 50 orbits.

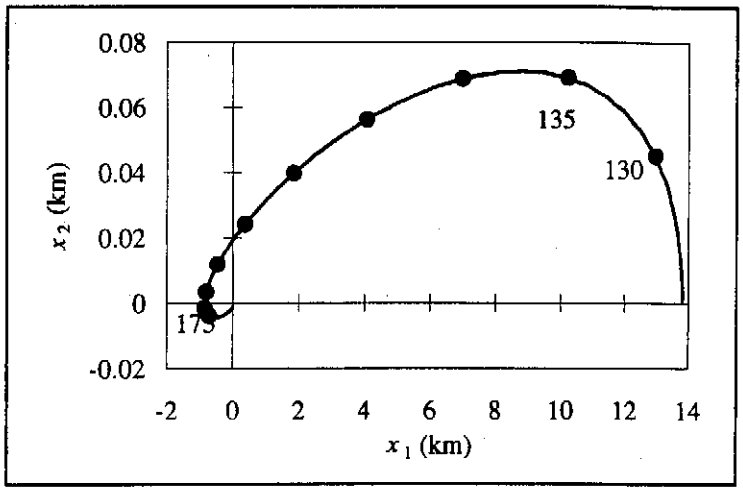


Figure 5.11: Constellation maintenance in an elliptical orbit – state space.

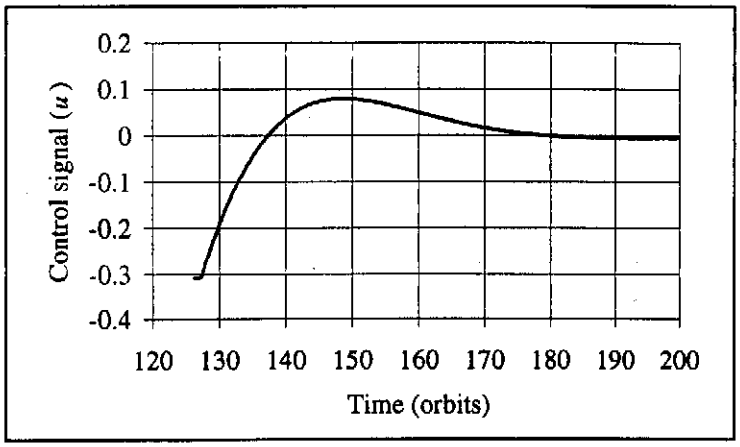


Figure 5.12: Constellation maintenance in an elliptical orbit – control signal.

Figure 5.12 shows the control signal during the feedback regulation phase. Note that the control signal is not influenced by the large variations in drag due to the orbit's ellipticity, since it is calculated from orbit-averaged values. There is thus no unnecessary short-term orientation changes at the expense of on-board energy. The

relatively slow orientation change, even through apogee passage where the drag force is very small, is necessary to create the correct orbit-average drag effect.

5. CONSTELLATION MAINTENANCE FOR MULTI-SATELLITE CONSTELLATIONS

During the constellation maintenance phase, the orbit-average distances between adjacent pairs of satellites must be regulated. Since every adjacent pair of satellites will have a common satellite in the middle, care must be taken with the selection of cross-sectional areas to avoid possible cross-coupling effects. The next section will model three satellites in a tandem constellation as a multiple-input multiple output system. From this model a decoupled regulator will be designed. The decoupling procedure will be generalised for constellations with more satellites in section 5.4.

5.1 MODELLING A THREE-SATELLITE CONSTELLATION

Consider a constellation of three identical satellites – S_1, S_2 and S_3, orbiting the earth in the same circular orbit. The linear model of (5.1) can be used to model the relative movement between S_1 and S_2, as well as between S_2 and S_3.

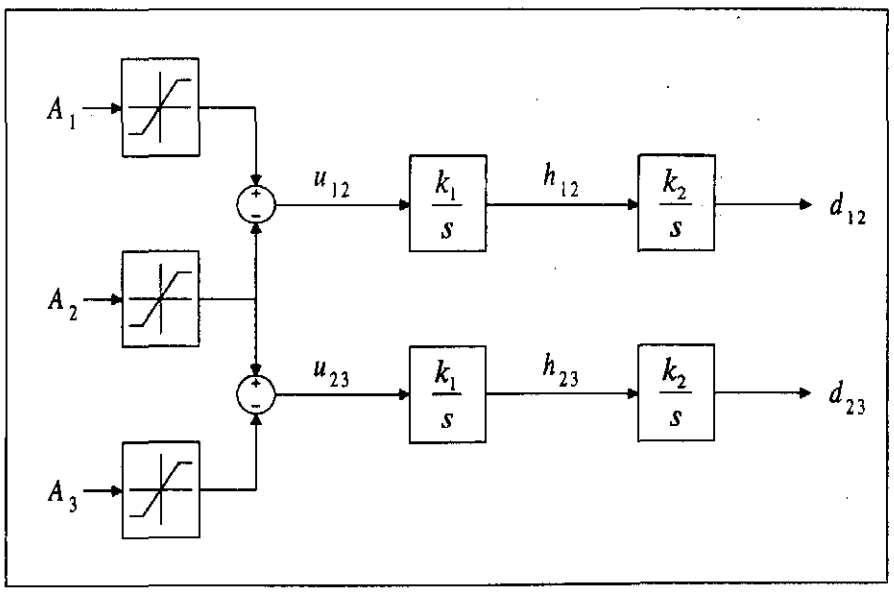


Figure 5.13: Block diagram of MIMO system.

The block diagram of figure 5.13 represents the three-satellite system. In the diagram, d_{12} is the orbit-average distance between S_1 and S_2 and h_{12} is the orbit-average altitude difference between S_1 and S_2. Similarly, d_{23} and h_{23} represent the orbit-average distance and altitude difference between S_2 and S_3. Setting the gains k_1 and k_2 equal for both pairs of satellites implicitly assumes that the three satellites have the same physical properties and that the state of the atmosphere can be considered equal in the three satellite positions. The model of figure 5.13 has three inputs (A_1, A_2 and A_3) and two outputs (d_{12} and d_{23}). It is clear that A_1 will only affect d_{12} and A_3 will only affect d_{23}, but A_2 will affect both d_{12} and d_{23}.

The system can be modelled in the state-space form:

$$\begin{aligned} \dot{\mathbf{x}} &= A\mathbf{x} + B\mathbf{a} \\ \mathbf{y} &= C\mathbf{x} + D\mathbf{a} \end{aligned} \tag{5.9}$$

The matrices A and B and the state vector $\mathbf{x}$ are not the same as those in equation (5.1). Note that the input vector ($\mathbf{a}$) is the three areas and not the signals u_{12} and u_{23}. The state vector is chosen as follows:

$$\mathbf{x} = \begin{bmatrix} d_{12} - d_{ref\,12} \\ h_{12} \\ d_{23} - d_{ref\,23} \\ h_{23} \end{bmatrix} \tag{5.10}$$

and the output vector $\mathbf{y} = \mathbf{x}$. The system matrices in (5.9) are then:

$$A = \begin{bmatrix} 0 & k_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & k_2 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 & 0 \\ k_1 & -k_1 & 0 \\ 0 & 0 & 0 \\ 0 & k_1 & -k_1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.11}$$

The given system is controllable since each state variable can be influenced through controlling the atmospheric drag on all satellites and no cancellation terms are present. Observability is ensured by the GPS receivers on the satellites. To design a decoupled regulator, the transfer function matrix of the system will be necessary. This can be obtained by taking Laplace transforms of (5.9) and setting the initial conditions to zero:

$$G_p(s) = C[s\mathbf{I} - A]^{-1}B + D$$
$$= \frac{k_1}{s}\begin{bmatrix} \frac{k_2}{s} & \frac{-k_2}{s} & 0 \\ 1 & -1 & 0 \\ 0 & \frac{k_2}{s} & \frac{-k_2}{s} \\ 0 & 1 & -1 \end{bmatrix} \qquad \textbf{(5.12)}$$

A decoupled regulator can now be designed.

5.2 DECOUPLED REGULATOR DESIGN

Consider the regulator architecture of figure 5.14 on the next page. Note that all signal paths now designate vectors.

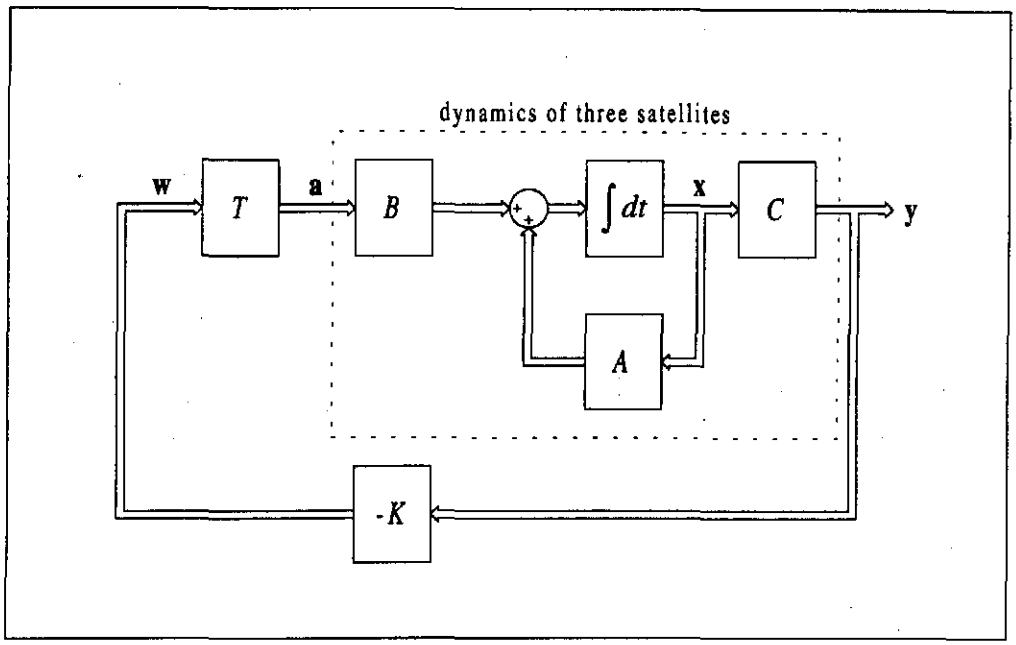


Figure 5.14: Block diagram of MIMO feedback regulator.

The vector **w** has a dimension of 2 for three satellites and would in general be of dimension n-1 for n satellites. This is a direct result of the fact that n-1 distances must be controlled to control the constellation topography of n satellites.

The regulator matrices T (3×2 transformation matrix) and K (2×4 gain matrix) must be selected to meet the following criteria:

- no cross-coupling must occur between the two pairs (S_1 , S_2) and (S_2 , S_3); and
- the dynamic response of the two satellite pairs must be the same.

These criteria will be treated as design specifications. If Laplace transforms are taken, the system can be transformed to the s-domain and transfer function matrices can be used for the design process. The transfer function matrix of the 3-satellite system – G_p in equation (5.12) – and the transformation matrix (T) can be multiplied to form the combined transfer function matrix G_0 as follows:

$$G_0(s) = G_p(s)T = \frac{k_1}{s}\begin{bmatrix} \frac{k_2}{s}(T_{11} - T_{21}) & \frac{k_2}{s}(T_{12} - T_{22}) \\ (T_{11} - T_{21}) & (T_{12} - T_{22}) \\ \frac{k_2}{s}(T_{21} - T_{31}) & \frac{k_2}{s}(T_{22} - T_{32}) \\ (T_{21} - T_{31}) & (T_{22} - T_{32}) \end{bmatrix} \tag{5.13}$$

A simplified block diagram of the MIMO system can now be constructed as in figure 5.15. All signals are in the *s*-domain.

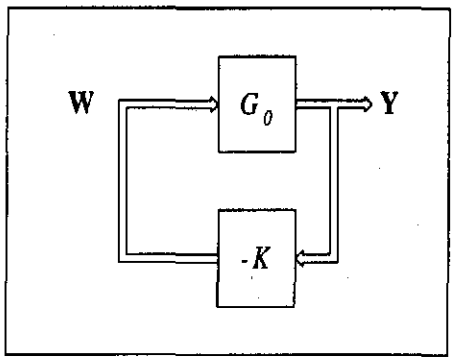


Figure 5.15: Simplified block diagram of MIMO control system.

From the block diagram in figure 5.15, it follows that:

$$\mathbf{W} = -K\mathbf{Y} \tag{5.14}$$

This can be expanded as follows:

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = -\begin{bmatrix} k_{11}(d_{12} - d_{ref\,12}) + k_{12}h_{12} + k_{13}(d_{23} - d_{ref\,23}) + k_{14}h_{23} \\ k_{21}(d_{12} - d_{ref\,12}) + k_{22}h_{12} + k_{23}(d_{23} - d_{ref\,23}) + k_{24}h_{23} \end{bmatrix} \tag{5.15}$$

The first step in decoupling the two pairs of satellites is to ensure that w_1 is not influenced by d_{23} or h_{23} and w_2 is not influenced by d_{12} or h_{12}. This is easily done by setting:

$$k_{13} = k_{14} = k_{21} = k_{22} = 0 \tag{5.16}$$

To ensure similar dynamics in the two pairs, choose:

$$\begin{aligned} k_{11} &= k_{23} \\ k_{12} &= k_{24} \end{aligned} \tag{5.17}$$

The resulting gain matrix K has the form:

$$K = \begin{bmatrix} \alpha & \beta & 0 & 0 \\ 0 & 0 & \alpha & \beta \end{bmatrix} \tag{5.18}$$

The gains α and β can be designed as in the previous section for a single satellite pair. From figure 5.15 also follows that:

$$\mathbf{Y} = \mathbf{G}_0 \mathbf{W} \tag{5.19}$$

This can be simplified to:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} d_{12} - d_{ref\,12} \\ h_{12} \\ d_{23} - d_{ref\,23} \\ h_{23} \end{bmatrix} = \begin{bmatrix} g_{011} w_1 + g_{012} w_2 \\ g_{021} w_1 + g_{022} w_2 \\ g_{031} w_1 + g_{032} w_2 \\ g_{041} w_1 + g_{042} w_2 \end{bmatrix} \tag{5.20}$$

To let (x_1 , x_2) be independent of w_2 and (x_3 , x_4) be independent of w_1, set:

$$g_{012} = g_{022} = g_{031} = g_{041} = 0 \tag{5.21}$$

If this is compared with the expression for G_0 in (5.13), it follows that:

$$\begin{aligned} T_{12} &= T_{22} \\ T_{21} &= T_{31} \end{aligned} \tag{5.22}$$

For similar dynamics in the two satellite pairs, set:

$$\begin{aligned} g_{011} &= g_{032} \\ g_{021} &= g_{042} \end{aligned} \tag{5.23}$$

From (5.13), this implies that:

$$T_{11} - T_{21} = T_{22} - T_{32} \tag{5.24}$$

The following choice of the transformation matrix T meets the criteria of (5.22) and (5.24):

$$T = \begin{bmatrix} \psi_1 & \psi_2 \\ \psi_1 - 1 & \psi_2 \\ \psi_1 - 1 & \psi_2 - 1 \end{bmatrix} \tag{5.25}$$

where ψ_1 and ψ_2 are arbitrary scalars. The matrix G_0 of (5.13) can now be simplified:

$$G_0(s) = \frac{k_1}{s} \begin{bmatrix} \frac{k_2}{s} & 0 \\ 1 & 0 \\ 0 & \frac{k_2}{s} \\ 0 & 1 \end{bmatrix} \tag{5.26}$$

It is clear that the two pairs of satellites are uncoupled and that they have the same dynamic response.

Consider now the selection of the scalars ψ_1 and ψ_2. The matrix T transforms the two signals (w_1 and w_2) to the three areas:

$$\begin{aligned} A_1 &= \psi_1 w_1 + \psi_2 w_2 \\ A_2 &= \psi_1 w_1 + \psi_2 w_2 - w_1 \\ A_3 &= \psi_1 w_1 + \psi_2 w_2 - w_1 - w_2 \end{aligned} \tag{5.27}$$

Using these relationships, the signals u_{12} and u_{23} in figure 5.13 can be expressed as:

$$\begin{aligned} u_{12} &= A_1 - A_2 = w_1 = -(\alpha d_{12} + \beta h_{12}) \\ u_{23} &= A_2 - A_3 = w_2 = -(\alpha d_{23} + \beta h_{23}) \end{aligned} \tag{5.28}$$

This verifies that the two pairs of satellites are uncoupled. It must also be noted that the inputs u_{12} and u_{23} of the two pairs are independent of the scalars ψ_1 and ψ_2. These scalars can therefore be changed arbitrarily at any time during the control process.

From (5.27) it is clear that:

$$\begin{aligned} A_1 &= \psi_1 w_1 + \psi_2 w_2 \\ A_2 &= A_1 - w_1 \\ A_3 &= A_2 - w_2 \end{aligned} \qquad \textbf{(5.29)}$$

The choice of ψ_1 and ψ_2 thus determines the value of A_1 and the other two areas are derived from this area. Since ψ_1 and ψ_2 are arbitrary, A_1 can be chosen arbitrarily at any instant and the other areas will be based on this choice. The following algorithm can now be formulated to determine the areas at every time step so that all the design specifications are met:

1. $A_1 = 0$ (arbitrary choice of A_1)
2. $A_2 = A_1 + \alpha d_{12} + \beta h_{12}$ (from equations (5.28) and (5.29))
3. $A_3 = A_2 + \alpha d_{23} + \beta h_{23}$ (from equations (5.28) and (5.29))
4. $m = \min(A_1, A_2, A_3)$ (get minimum value m)
5. $A_1 = A_{\min} - m$ (shift A_1 so that minimum will equal $A_{\min}$)
6. Repeat steps 2 and 3. (recalculate A_2 and A_3)
7. Limit all areas to $A_{\max}$. (upper limit on areas)

This algorithm ensures that the areas are chosen properly, to prevent cross-coupling effects between the two satellite pairs. It also causes the minimum value of all three areas to be $A_{\min}$ at any time, thus conserving altitude as best as possible.

5.3 IMPLEMENTATION AND SIMULATION

The feedback regulator is designed to keep the orbit-average distance between a pair of satellites at a certain reference value. If this reference value is changed, the regulator will register it as an error and the satellite pair will be controlled to the new specified orbit-average distance.

To demonstrate the decoupled regulator, three satellites were initiated with identical position and velocity vectors in a circular orbit. The reference orbit-average distance was initialised as 5 km between S_1 and S_2 as well as between S_2 and S_3. After 50 orbits, the reference orbit-average distance between S_1 and S_2 was changed to 2.5 km and after 100 orbits, the reference orbit-average distance between S_2 and S_3 was changed to 7.5 km. The gains α and β were obtained from the optimal regulator gain matrix K designed for the two-satellite case.

Figure 5.16 (next page) displays the orbit-average distance between S_1 and S_2 (d_{21}, heavy line) and between S_2 and S_3 (d_{23}, thin line). The identical dynamic response between the two satellite pairs is clear during the first control effort (the trajectories of d_{21} and d_{23} coincide in the figure). The second two control efforts show the absence of cross-coupling effects between the two satellite pairs. When d_{12} is decreased after 50 orbits it doesn't influence d_{23} and when d_{23} is increased after 100 orbits, it doesn't influence d_{12}.

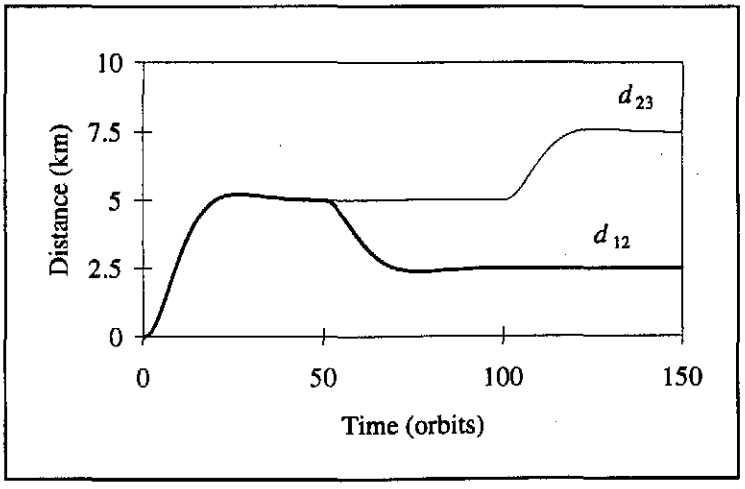


Figure 5.16: MIMO constellation maintenance –orbit-average distances.

The cross-sectional areas perpendicular to the velocity vectors of the three satellites are shown in figure 5.17.

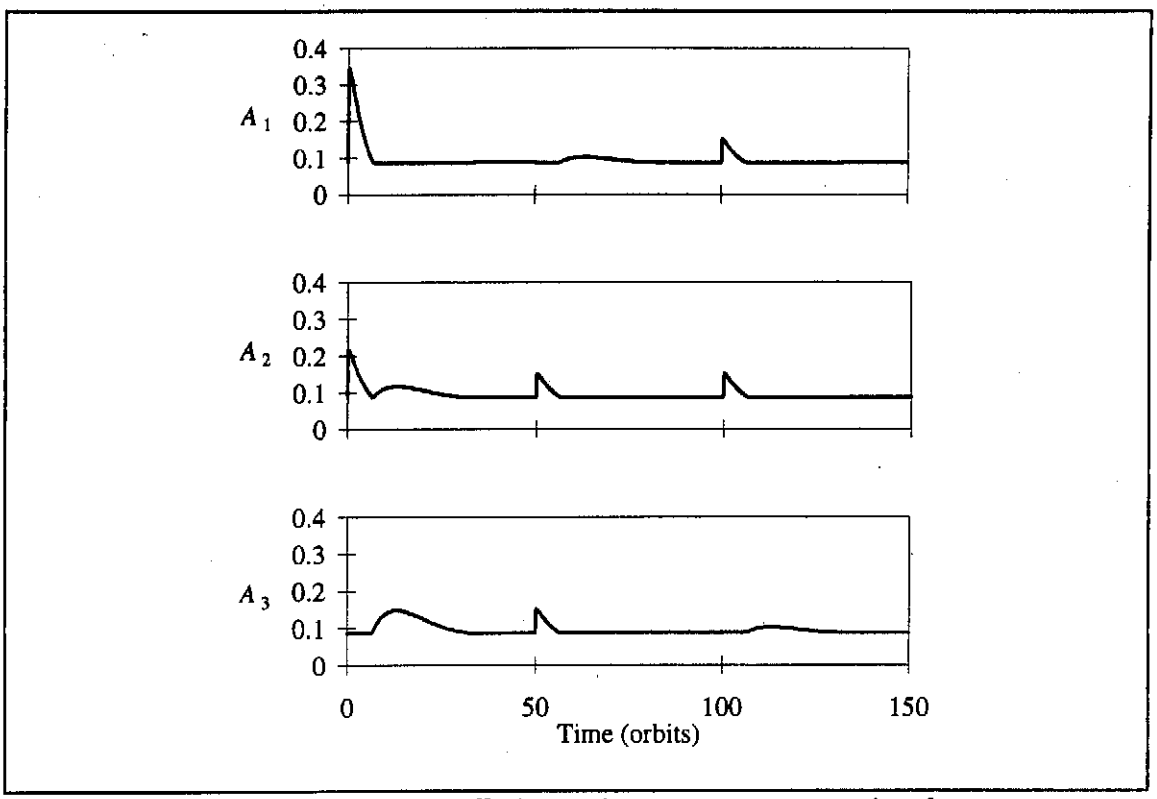


Figure 5.17: MIMO constellation maintenance – cross-sectional areas.

Note how A_3 follows A_2 to maintain the orbit-average distance d_{23} while d_{12} changes after 50 orbits. The same can be seen after 100 orbits, with A_1 following A_2 to maintain d_{12} when d_{23} changes.

5.4 GENERALISATION FOR MORE SATELLITES

The design of the decoupled regulator can be generalised for tandem constellations with any number of satellites. The gain matrix (K) for a 5-satellite constellation is:

$$K = \begin{bmatrix} \alpha & \beta & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha & \beta & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha & \beta & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha & \beta \end{bmatrix} \tag{5.30}$$

and the transformation matrix (T) is:

$$T=\begin{bmatrix} \psi_1 & \psi_2 & \psi_3 & \psi_4 \\ \psi_1-1 & \psi_2 & \psi_3 & \psi_4 \\ \psi_1-1 & \psi_2-1 & \psi_3 & \psi_4 \\ \psi_1-1 & \psi_2-1 & \psi_3-1 & \psi_4 \\ \psi_1-1 & \psi_2-1 & \psi_3-1 & \psi_4-1 \end{bmatrix} \tag{5.31}$$

The form of these matrices for any number of satellites should be clear. The algorithm to determine the areas at every time step can also be generalised very easily. The additional areas must be calculated after step 3. When the minimum is determined, all the areas must be included.

Chapter 6

SUMMARY, CONCLUSIONS AND FUTURE RESEARCH

In this final chapter, an overall perspective on the work will be established so that conclusions can be drawn. The most significant assumptions and simplifications that were made in the previous chapters will be summarised and the results will be evaluated against the initial objectives of this study. The chapter will conclude with some comments and suggestions regarding possible future research in this field.

The objectives were formulated in chapter 1 and are repeated here for reference:

a) Obtaining a thorough understanding of orbital theory, orbit perturbations and perturbation methods for LEO satellites.

b) The development of a high-precision orbit propagator which could provide accurate short-term solutions for the velocity and position of LEO satellites.

c) From the results of (a) and (b), to design and test optimal control strategies, using the concept of variable atmospheric drag, for
   i. constellation acquisition and
   ii. constellation maintenance
   of LEO tandem constellations.

The following section will summarise the results of the study with reference to the above objectives. The emphasis will be placed on the development of the optimal control strategies, *i.e.* objective (c).

1. SUMMARY

The first two objectives were the themes of chapter 2 and 3. A sufficient theoretical and mathematical foundation was presented and the special perturbations orbit propagation routines provided an efficient simulation tool for the rest of the study. The simulations could be used to "substitute" the measurements from a real satellite constellation.

The first important aspect in the process of designing constellation acquisition strategies was to find a suitable model. The relative movement between the two satellites was investigated and it was shown that the dynamics could be simplified greatly if the average tendencies per orbit were considered. The sliding orbit-averages of the altitude difference and distance between the satellites were calculated by averaging the instantaneous values of these two quantities in a square window with the length of one orbital period. A consideration of these orbit-average tendencies lead to the double-integrator model to describe the orbit-averaged dynamics of the two-satellite constellation. The important assumption for this model to be valid was that the total altitude loss would be small enough so that the average atmospheric density per orbit could be assumed constant during the application time of the model. The validity of this assumption was confirmed through later simulations of the typical constellation acquisition manoeuvres in both circular and elliptical orbits. The decision to model the satellite pair as a single-input, single-output system simplified the design process that followed.

Two separate parameters were presented as the main criteria to measure the feasibility of the proposed control concept: the total control time and total altitude loss during the constellation acquisition effort. The two optimal control forms, minimising the respective cost functions, were determined with Pontryagin's minimum principle. The use of this principle provided a sound mathematical foundation for the development of the optimal control strategies for constellation acquisition. Once the forms of the optimal control strategies were determined, the actual control functions could be expressed in terms of known quantities.

The adaptive control scheme that was introduced provided the key to maintaining control accuracy and robustness in a changing operating environment. The effects of

unpredictable short-term variations in the atmospheric density can be neutralised by the continuous adaptation of the control signal. The RLS identification algorithm to determine the model parameter k_1 is easy to implement and has a quick convergence rate. An additional advantage of the identification scheme is that the long-term variations in the density can also be tracked. It must be kept in mind that some of the long-term variations in the atmospheric density can have a significant influence on the performance of the proposed control concepts. For example, the control strategy will remain successful throughout the typical density variations of the sunspot cycle (even for orders of magnitude variations), but the control times might become excessively long during the times when the density is very low. These considerations must be kept in mind when assessing the feasibility of the proposed constellation control scheme for a specific mission.

The extension of the control strategies to elliptical orbits was considered next. The key to the success of the strategies was again the fact that the dynamics can be averaged over an orbital period, using the same sliding square window as before, to obtain a simple linear model.

Finally, it was shown that the time-optimal and altitude-loss-optimal acquisition strategies can easily be implemented in the case of a multiple-satellite constellation. An important aspect that became clear is that individual pairs of satellites, when viewed in isolation, are not controlled optimally anymore, but that the constellation as a whole is still controlled optimally.

The constellation maintenance phase was considered in chapter 5. The goal of this phase is to maintain the relative constellation configuration. The same linear model as before was used to design a full state-feedback regulator to keep the satellites at the same orbit-average altitude and to keep the orbit-average distance between them at the specified reference value.

The linear quadratic optimisation process produces an optimal controller for each choice of the weighting matrices. This flexibility in the choice of the weighting matrices allows the dynamic response of the system to be tailored. In the given examples, the control signal was kept small and the response relatively slow.

Changes in operating conditions during the maintenance phase are in a certain sense less critical to the system's performance than during the acquisition phase. Slight variations in the density normally cause a change in the dynamic response, but the regulator will still keep the state at the origin of the state space. The adaptive strategy that was introduced does however provide an effective way of cancelling the effects of large short-term changes in the atmospheric density and maintaining the desired dynamic response.

The application of the maintenance strategy to the elliptical orbit was also demonstrated successfully.

To apply the maintenance strategy in multiple-satellite constellations, the cross-sectional areas must be selected in a way that avoids cross-coupling effects between neighbouring pairs of satellites. This decoupling process was explained mathematically. Each individual pair of satellites – and hence the constellation as a whole – is still controlled optimally.

2. CONCLUSIONS

In this dissertation, the feasibility of controlling the relative positions of micro-satellites in LEO tandem constellations by utilising atmospheric drag was established. The proposed constellation control concept provides a simple, accurate and cheap alternative for the acquisition and maintenance of low Earth orbit tandem constellations.

It has the advantage of not adding any significant complexity or cost to the satellite mission. The only structural requirement on the satellites is that the cross-sectional area, projected perpendicular to the velocity vector, must be controllable. The simplest way to meet this requirement is by changing the orientation of an asymmetrical satellite. This minimal structural requirement presents a great advantage over conventional constellation control techniques where thrusters are required on the satellite, especially for LEO missions where complexity and cost must be minimised.

Another advantage of the proposed control strategies is that they are very energy-efficient, requiring onboard energy only for orientation manoeuvres on each satellite. The strategies are in essence passive, continuously dumping energy from the orbits. This implies that absolute constellation maintenance is not possible, but only relative constellation maintenance. The proposed concepts can however be used in conjunction with thrusters if absolute constellation maintenance is required. It must be kept in mind, though, that such a scheme would imply a non-optimum use of the thrusters. Less thruster energy is required to continuously maintain the altitude of the constellation over a period of time if it is not allowed to first fall to a lower altitude. This is a direct result of the exponential increase of density as altitude decreases.

The optimal control strategies for constellation acquisition and constellation maintenance are suitable for application in circular as well as slightly elliptical orbits. As mentioned before, the strategy would be unable to maintain constellations where the satellites are in slightly different orbital planes.

The control strategies are also robust to the typical variations in the atmospheric density. These changes cause a variation in the parameters of the equivalent model of the constellation and do not do act as external disturbances. The control strategies can therefore remain accurate under these mostly unpredictable circumstances.

The effectiveness and feasibility of the proposed control strategies depend on many factors, including the satellite's physical properties and the orbital configuration. All analyses and results were generalised so that the feasibility of the control strategies could easily be determined for any satellite and orbital configuration.

One of the most critical parameters is the altitude of the constellation. If the altitudes are too high, the control times become excessively long due to the low atmospheric density. For very low altitudes, the altitude loss during the control effort will become large, resulting in a significant reduction of the constellation's lifetime. Secondary effects like aerodynamic lift might also become significant at low altitudes. The useful range of altitudes was extended as far as possible by designing the control strategies to be optimal in terms of control time and altitude loss.

The nominal test case serves as a generic example for the successful application of the control strategies. The emphasis of this example is not on the specific designs that were carried out, but on a demonstration of the proposed control concepts.

3. FUTURE RESEARCH

Future research on the concept of using atmospheric drag for constellation control would probably revolve around the application of the concept to specific practical missions. From this point of view, the interaction of the primary payload and mission objectives with the constellation control strategy and its accompanying requirements on the satellite's structure and on-board resources will have to be considered in detail.

Other areas that may be investigated include:

- *The control surface.* The feasibility of the concept, specifically in terms of minimum control time, can be enhanced by increasing the ratio of the maximum to minimum drag area. This could be done by using mechanically deployable, or even inflatable surfaces.

- *Aerodynamic lift.* At very low altitudes, the atmospheric density can become large enough for the effects of aerodynamic lift to become significant. Although this force acts perpendicular to the velocity vector, it doesn't add energy to the orbit and can thus not increase the orbit-average altitude of the satellite. It could however be utilised for out-of-plane control manoeuvres if the satellite is turned on its "side". The effect is very small, but it could theoretically counter-act certain perturbation effects that tend to change the orbit's orientation, especially secular changes in the right ascencion of the ascending node and the inclination. The possibilities and feasibility are worthwile investigating.

The investigation of similar concepts to optimise the available resources and utilise the physical phenomena in the restricted space environment, will become more critical in the future, especially as the challenges grow and the cost-restrictions become more stringent for the micro-satellite community.

REFERENCES

Anderson, B.D.O. and Moore, J.B. (1971), *Linear Optimal Control*, Prentice-Hall Inc., Englewood Cliffs, New Jersey.

Åström, K.J. and Wittenmark, B. (1989), *Adaptive Control*, Addison-Wesley Publishing Company, Reading MA.

Baker, R.M.L. (jr.) and Makemson, M.W. (1960), *An Introduction to Astrodynamics*, Academic Press, New York.

Bate, R.R., Mueller, D.D. and White, J.E. (1971), *Fundamentals of Astrodynamics*, Dover, New York.

Battin, R.H. (editor) (1987), *An Introduction to the Mathematics and Methods of Astrodynamics*, AIAA Educational Series, New York.

Bryson, A.E. (1994), *Control of Spacecraft and Aircraft*, Princeton University Press, Princeton, New Jersey.

Bryson, A.E. and Ho, Y (1969), *Applied Optimal Control*, Ginn and Company, Waltham, Massachusetts.

Chao, C.C. and Bernstein, H. (1994), "Onboard Stationkeeping of Geosynchronous Satellites Using a Global Positioning System Receiver," *Journal of Guidance, Control and Dynamics*, Vol. 17, No. 4, pp. 778-786.

Chobotov, V.A. (editor) (1991), *Orbital Mechanics*, AIAA Educational Series, Washington DC.

Collins, J.T., Dawson, S. and Wertz, J.R. (1996), "Autonomous Constellation Maintenance System," *Proceedings of the 10th Annual AIAA/USU Conference on Small Satellites*, Utah State University, Logan, Utah.

Danby, J.M.A. (1962), *Fundamentals of Celestial Mechanics*, Macmillan and Co. Ltd., New York.

De Villiers, R. (1994), *The Block Diagram Simulation of Satellite Systems*, M. Eng. Thesis, University of Stellenbosch.

Dorato, P., Abdallah, C. and Cerone, V. (1995), *Linear-Quadratic Control: An Introduction*, Prentice-Hall Inc., Englewood Cliffs, New Jersey.

Du Toit, D.N.J., Du Plessis, J.J. and Steyn, W.H. (1996), "Using Atmospheric Drag for Constellation Control of Low Earth Orbit Micro-satellites," *Proceedings of the 10th Annual AIAA/USU Conference on Small Satellites*, Utah State University, Logan, Utah.

Ei Compendex Abstracts (1992 to 1996), OnDisc compact disc database, Knight-Ridder Information Inc.

El'yasberg, P.Ye., Mersov, G.A., Bakhshiyan, B.Ts. and Kugaenko, B.V. (editors) (1967), *Determining the Motion of Space Vehicles*, NASA, Washington DC.

Escobal, P.R. (1965), *Methods of Orbit Determination*, Robert E. Krieger Publishing Co., Malabar, Florida.

Fortescue, P.W. and Stark, J.P.W. (editors) (1991), *Spacecraft Systems Engineering*, October 1992 reprint, John Wiley and Sons, Chichester, England.

Fox, K (1984), "Numerical Integration of the Equations of Motion of Celestial Mechanics," *Celestial Mechanics*, Vol. 33, pp. 127-142.

Franklin, G.F., Powell, J.D. and Workman, M.L. (1990), *Digital Control of Dynamic Systems*, Addison-Wesley, Reading MA.

Griffin, M.D. and French, J.R. (editors) (1991), *Space Vehicle Design*, AIAA Educational Series, Washington DC.

Hildebrand, F.B. (1965), *Advanced Calculus for Applications*, Prentice-Hall Inc., Englewood Cliffs, New Jersey.

Inspec Electronics and Computing Abstracts (1993 to 1996), ProQuest compact disc database, UMI Company and Fulcrum Technologies.

Kaplan, M.H. (1976), *Modern Spacecraft Dynamics & Control*, John Wiley & Sons, New York.

King-Hele, D. (1964), *Theory of Satellite Orbits in an Atmosphere*, Butterworths, London.

King-Hele, D. (1969), *The Upper Atmosphere and its Influence on Satellite Orbits*, Proceedings of COSPA-IAU-IAG/IUGG-IUTAM Symposium, Prague, May 20-24.

King-Hele, D. (1987), *Satellite Orbits in an Atmosphere: Theory and Applications*, Blackie and Son Ltd., London.

Kirk, D.E. (1970), *Optimal Control Theory: An Introduction*, Prentice-Hall Inc., Englewood Cliffs, New Jersey.

Langton, N.H. (editor) (1969), *Space Research and Technology: Volume 1, The Space Environment*, University of London Press Ltd, London.

Larson, W.J. and Wertz, J.R. (editors) (1992), *Space Mission Analysis and Design (Second Edition)*, published jointly by Microcosm Inc., California USA and Kluwer Academic Publishers, Dordrecht, The Netherlands.

Morando, B. (editor) (1970), *Dynamics of Satellites*, Proceedings of COSPA-IAU-IAG/IUGG-IUTAM Symposium held in Prague on May 20-24 1969, Springer-Verlag, New York.

Ogata, K.O. (1990), *Modern Control Engineering, Second Edition*, Prentice-Hall International, Englewood Cliffs, New York.

O'Neil, P.V. (1987), *Advanced Engineering Mathematics*, Second Edition, Wadsworth Publishing Co., Belmont, California.

Pontryagin, L.S., Boltyanskii, V.G., Gamkrelidze, R.V. and Mishchenko, E.F. (1962), *The Mathematical Theory of Optimal Processes*, Interscience Publishers Inc., New York.

Ross, I.M. and Alfriend, K.T. (1995), "Low-Earth-Orbit Maintenance: Reboost vs Thrust-Drag Cancellation," *Journal of Guidance, Control and Dynamics*, Vol. 18, No. 4, pp. 930-932.

Roy, A.E. (1988), *Orbital Motion*, 3rd ed., Adam Hilger, Bristol, UK.

Sehnal, L. (1969), *Radiation Pressure Effects in the Motion of Artificial Satellites*, Proceedings of COSPA-IAU-IAG/IUGG-IUTAM Symposium, Prague, May 20-24.

Smith, D.E. (1969), "Earth-Reflected Radiation Pressure," Proceedings of COSPA-IAU-IAG/IUGG-IUTAM Symposium, Prague, May 20-24.

Steyn, W. H. (1996). "A Multi-mode Attitude Determination and Control System for SUNSAT," *Proceedings of the 3rd International Symposium on Small Satellite Systems and Services*, June 24-28, Annecy, France.

Wertz, J.R. (editor) (1978), *Spacecraft Attitude Determination and Control, Astrophysics and Space Science Library*, Vol. 73, D. Reidel Publishing Co., Boston.

Westphal, L.C. (1995), *Sourcebook of Control Systems Engineering*, Chapman and Hall, London.

Appendix A

THE NOMINAL TEST CASE

This appendix contains some additional information on the nominal satellite and orbital configuration used in the study. The contents of the appendix is as follows:

1. SATELLITE LIFETIME

The satellite lifetime can be estimated from figure 8-13 in Larson *et al* (1992). From the graph, the lifetime can be obtained as a function of the ballistic coefficient. The ballistic coefficient is given by:

$$\frac{m}{C_D A} \tag{A.1}$$

and equals 51.9 kg.m^{-2} for the nominal satellite with $A = A_{min}$. Interpolating from the curves on the graph, the lifetime of a 450 km circular orbit would be between 1 and 3 years, depending on whether the orbit starts during a solar minimum or solar maximum.

2. CROSS-SECTIONAL AREA

For the nominal satellite, the cross-sectional area perpendicular to the velocity vector is a function of the incidence angle (ψ). Figure A.1 shows the definition ψ, which, as a result of the cylindrical symmetry, is sufficient to specify the orientation of the satellite.

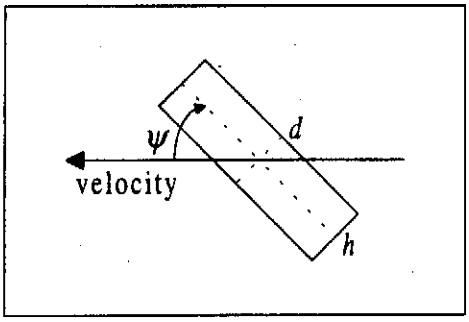


Figure A.1: Definition of incidence angle (ψ).

The projected cross-sectional area, perpendicular to the velocity vector is given by:

$$A(\psi) = \frac{\pi d^2}{4}\sin(\psi) + dh\left|\cos(\psi)\right| \qquad \textbf{(A.2)}$$

where d is the diameter and h is the thickness of the satellite. The function of equation (A.2) is plotted in figure A.2.

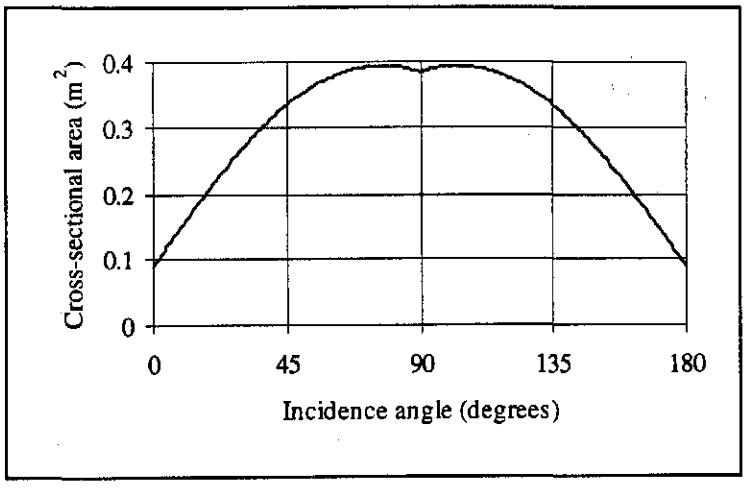


Figure A.2: Cross-sectional area as a function of the incidence angle.

The minimum cross-sectional area (A_{min}) is found at $\psi = 0°$ and equals 0.0875 m^2. It is clear that, due to the finite thickness of the satellite, the maximum cross-sectional area (A_{max}) is not at the orientation $\psi = 90°$.

To find the incidence angle that results in the maximum cross-sectional area, the expression (A.2) can be differentiated and the result set to zero. Solving for the angle yields:

$$\psi_{max} = \tan^{-1}\left[\frac{\pi d}{4h}\right] \tag{A.3}$$

The value is 77.2° for the nominal satellite. The value of the maximum cross-sectional area is found by back-substitution, yielding $A_{max} = 0.3947$ m^2.

3. ATMOSPHERIC DRAG DISTURBANCE TORQUE

Atmospheric drag can be modelled as a force acting at the geometrical centre of the projected surface perpendicular to the velocity vector of the satellite. The magnitude of the drag force is given by:

$$F_d = \frac{\rho C_D A v^2}{2} \tag{A.4}$$

If the centre of mass does no coincide with the geometrical centre of the satellite, the drag force will cause a disturbance torque that will change the orientation of the satellite, unless an equal and opposite torque is applied. The largest possible disturbance torque will be present if the incidence angle of the satellite is 90° and the centre of mass is right on the edge of the satellite's body. This extreme imbalance serves as a worst-case scenario to verify the ability of the onboard torqueing devices to counter-act the disturbance torque. The maximum disturbance torque is found by multiplying the drag force (A.4) with the satellite's radius ($0.5d$):

$$T_d = \frac{\rho C_D \pi d^3 v^2}{16} \tag{A.5}$$

The area A has been substituted by $0.25\pi d^2$. For the nominal satellite in the elliptical obit, the value of T_d is a maximum at the perigee and equals 14 μN.m if the average density at perigee is 1.585×10^{-12} kg.m^{-3} (value from Wertz, 1978). The value of the maximum disturbance torque in the circular orbit is 13.7 μN.m.

It has been shown (Steyn, 1996) that a continuous torque of at least 100 μN.m can be obtained from three orthogonal magnetorquers, powered by solar panels on a micro-satellite with similar dimensions and in a similar orbit as the nominal satellite. The worst-case disturbance torque due to atmospheric drag can thus easily be rejected using such magnetorquers on the nominal satellite.

4. SYSTEM IDENTIFICATION

This section provides some additional information on the system identification process, used for the determination of the parameters k_1 and k_2 in the second-order model of the relative movement between two satellites. The *bisect* function of SIMuWIN was used for the identification. The identification set-up is described in chapter 4.

4.1 CIRCULAR ORBIT

The parameter k_1 was initialised with a value of zero and the first step was specified to be -0.001. Figure A.3 shows the convergence of the parameter.

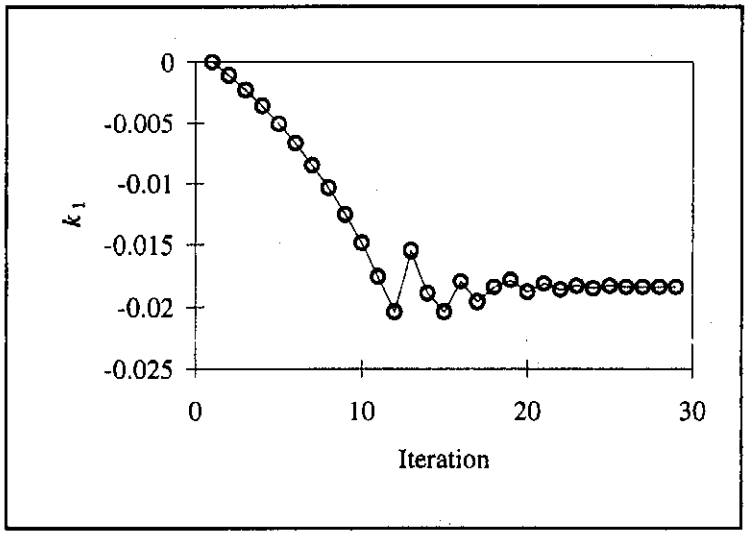


Figure A.3: Identification of k_1 in circular orbit.

The parameter reaches a stable value within 30 iterations. Note that each iteration involves a simulation of the model over 20 orbits, each time comparing the output from the model with the output of the full simulation. The full simulation was only done once and the results were stored. The final value of k_1 is -0.01841.

The parameter k_2 was initialised with a value of zero and the first step was specified to be -0.0001. A stable value, minimising the chosen error measure, is again reached

within 30 iterations. The convergence of the parameter k_2 is shown in figure A.4. The final value in this case equals -0.00167.

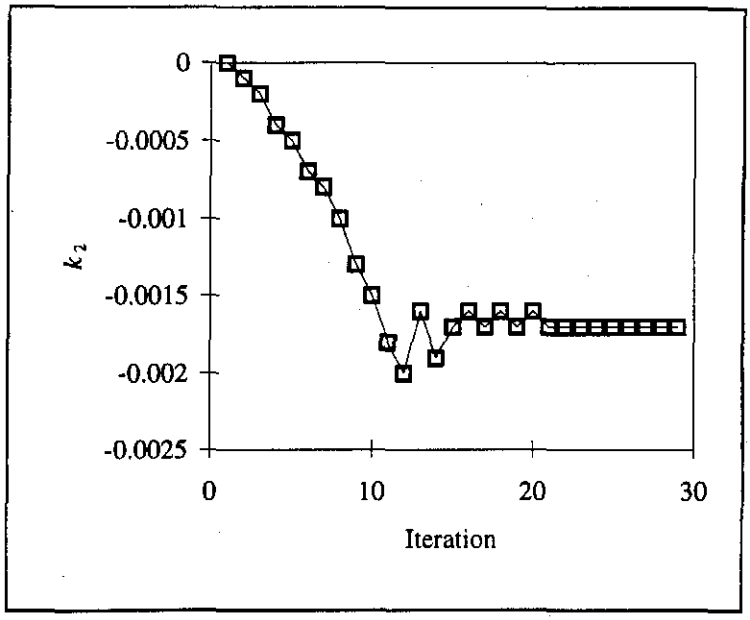


Figure A.4: Identification of k_2 in circular orbit.

4.2 ELLIPTICAL ORBIT

For the elliptical orbit, the parameter values and step sizes were initialised the same as before. Figures A.5 shows the convergence of the parameter k_1 to the final value of -0.00841.

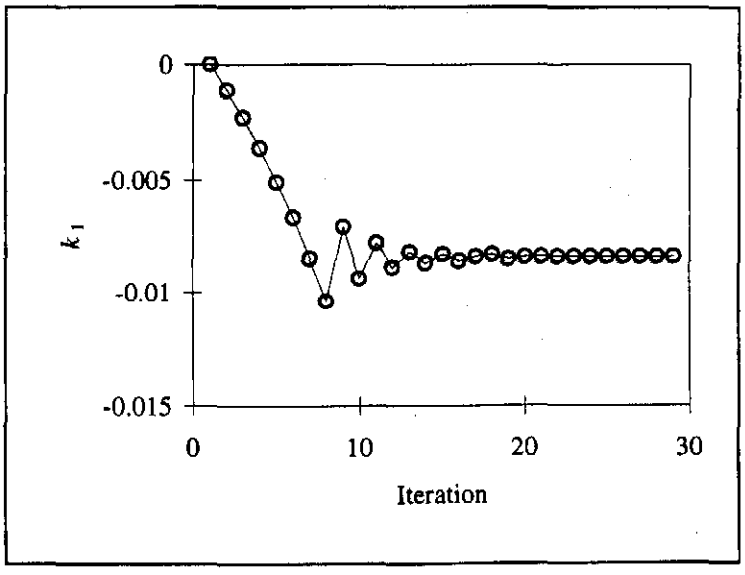


Figure A.5: Identification of k_1 in elliptical orbit.

The convergence of k_2 is shown in figure A.6. The final value is -0.00163.

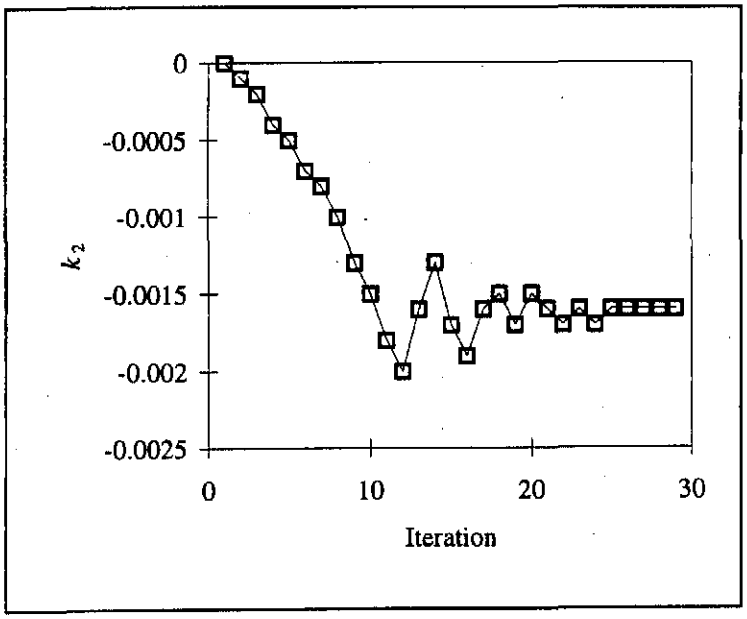


Figure A.6: Identification of k_2 in elliptical orbit.

From all the convergence graphs (figures A.3 to A.6) it is clear that the step size is initially increased. The step is then gradually decreased as the parameter converges to the value that minimises the cost function.

5. ORIENTATION MANOEUVRES

This simple analysis will show that it is possible to re-orientate the nominal satellite within one time step (≈ 28 seconds) through the required 77.2° to go from the lowest to the highest drag orientation, or *vice versa*, using cheap reaction wheels. This type of re-orientation manoeuvre ("switch" from one orientation to another) is necessary during the constellation acquisition phase, as described in chapter 4.

Assume that the centre of mass of the satellite is at the geometrical centre and that the mass is evenly distributed. The moment of inertia around the rotation axis is given by:

$$I = \frac{md^2}{16} \tag{A.6}$$

and equals 0.30625 kg.m^2 for the nominal satellite. Assume that an extremal bang-bang control manoeuvre is done to re-orientate the satellite. Since the applied torque is piece-wise constant, the orientation angle can be calculated as follows during the first control phase (before the switch) if the initial angle is zero:

$$\psi = \frac{T}{I} t^2 \tag{A.7}$$

If the control effort is specified to last exactly 28 seconds and the total change in orientation angle is 77.2°, the required torque can be calculated by evaluating (A.7) after 14 seconds. This yields a desired torque of 1.05 mN.m, which can easily be provided by relatively small and cheap reaction wheels (Steyn, 1996).

Appendix B

ORBIT PROPAGATION SOFTWARE

This appendix contains listings of the Pascal programs and units necessary for orbit propagation. The contents is as follows:

1. PROGRAM: ORBIT.PAS

```
{ Main orbit propagation program }

program Orbit;

{ include units }
uses Crt,
     Plot,                                { display routines }
     Globals, Density, Display, Sats;     { other orb. prop. units }

{ initialisation }
procedure Initialise;

begin

  { calculate orbital period }
  OrbPer := 2*pi/(sqrt(Earth_mu/(a0*a0*a0)));

  h := OrbPer/NRev;              { fixed time step for num. integr. }
  Revs := 3;                     { number of revolutions to simulate }
  Init_Sats;                     { initialise satellites }
  Init_Display;                  { initialise graphical output }
end;

{ main program ------------------------------------------------------}

var
  ch    : Char;

begin

  Initialise;
  n := 0;

  { main loop }
  Repeat
    time := n*h;                 { update time }
    EveryStep_Display;           { display results }
    EveryStep_Sats;              { propagate positions & velocities }
    inc(n);                      { increment counter }
```

```
    until ( (n = Revs*NRev + 1) or KeyPressed);

    { wait for keystroke }
    if KeyPressed then ch := ReadKey;
    ch := ReadKey;

    DonePlot;

end.
```

2. UNIT: GLOBALS.PAS

```
{ Global constants and variables }

unit Globals;

INTERFACE

{ global constants ------------------------------------------------}

const

  { satellite identification constants }
  id_Kep      = 0;                { unperturbed Kepler satellite }
  id_Moon     = 1;                { the Moon }
  id_Drag     = 2;                { only drag included }
  id_All      = 3;                { all perturbations included }
  id_DragJ2   = 4;                { drag and J2 included }

  { simulation constants }
  NRev        = 200;              { number of steps per orbit }
  rho_vary    = True;             { constant density ? }
  SatID       = id_All;           { identify satellite }

  { initial Kepler orbit parameters }
  a0          = 6371e3 + 450e3;   { semi-major axis }
  e0          = 0.0;              { eccentricity }
  i0          = 90*pi/180;        { inclination }
  Om0         = 0*pi/180;         { right ascension }
  w0          = 0*pi/180;         { argument of perigee }
  th0         = 0*pi/180;         { true anomaly }

  { satellite properties }
  Mass0       = 10;               { mass }
  Cd          = 2.2;              { drag coefficient }
  K           = 1.5;              { reflectivity }
  AMax        = 0.3946669;        { max cross-sect area }
  AMin        = 0.0875;           { min cross-sect area }

  { general constants }
  AU          = 1.49597870e11;    { 1 astronomical unit }
```

```
  { Earth constants }
  Earth_mu    = 3.986005e14;       { G*M }
  Re          = 6371e3;            { mean Earth radius }
  Day         = 24*60*60;          { one day in sec }
  Sday        = 86164.0918;        { sidereal day in sec }
  Syear       = 365.25636051*day;  { sidereal year in sec }
  J2          = 1082.7e-6;         { oblateness term }

  { sun constants }
  Sun_dist       = AU;             { distance }
  Sun_i          = 23.439*pi/180;  { inclination }
  Sun_period     = Syear;          { Period in seconds }
  Sun_mu         = 1.32712438e20;  { Sun grav. constant }
  Sun_MFlux      = 4.4e-6;         { momentum flux }

  { moon constants }
  Moon_mu          = 4.902786e12;  { gravitational const }
  Moon_a           = 384e6;        { semimajor axis }
  Moon_e           = 0.055;        { eccentricity }
  Moon_i           = 23*pi/180;    { inclination }
  Moon_Om          = 0*pi/180;     { right ascension }
  Moon_w           = 0*pi/180;     { argument of perigee }
  Moon_th          = 0*pi/180;     { true anomaly }

{ global variables ------------------------------------------------}

var
  n           : LongInt;           { time step counter }
  h           : Real;              { time step }
  time        : Real;              { global time }
  Revs        : Real;              { no. of revolutions }
  OrbPer      : Real;              { orbital period }

IMPLEMENTATION

begin
end.
```

3. UNIT: SATS.PAS

```
{ main orbit propagation objects and routines }

unit Sats;

INTERFACE

uses Matrix;

{ types ------------------------------------------------------------}
type
  IntegrateVector = Array[1..6] of Real;

{ TKepSat object ---------------------------------------------------}
type
  TKepSat = Object

    ID            : Word;     { satellite ID no }
    n0            : Real;     { mean motion at epoch}
    p0            : Real;     { semilatus rectum at epoch}

    { other orbital parameters }
    a             : Real;     { semi-major axis }
    e             : Real;     { eccentricity }
    i             : Real;     { inclination }
    Om            : Real;     { right ascension of the asc. node }
    w             : Real;     { argument of the perigee }
    th            : Real;     { true anomaly }

    { position and velocity }
    r             : Vector3;    { position vector }
    rM            : Real;       { dist. from Earth centre }
    v             : Vector3;    { velocity vector }
    vM            : Real;       { velocity magnitude }
    Lon           : Real;       { subsatellite longitude }
    Lat           : Real;       { subsatellite latitude }

    procedure Init(An_ID: Word);
    procedure NextPosition(tt: Real);
    procedure Get_r(var rr: Vector3; rrM, ii, OOm, ww, tth: Real);
```

```
Type
  TSun = object
    r: Vector3;                { position vector }
    procedure Init(Arx, Ary, Arz: Real);
    procedure NextPosition(tt: Real);
  end;

{ variables ---------------------------------------------------------}
var
  Sun              : TSun;         { Sun }
  Moon             : TKepSat;      { Moon }
  S1               : TSat;         { perturbed satellite }
  S2               : TKepSat;      { Kepler satellite }

{ procedures --------------------------------------------------------}
procedure Init_Sats;
procedure EveryStep_Sats;

IMPLEMENTATION

uses
  Trig, Math,
  Density, Globals;

{ TKepSat methods ---------------------------------------------------}

procedure TKepSat.Init(An_ID: Word);

var
  vr0              : Real;      { radial velocity }
  vn0              : Real;      { normal velocity }
  gamma0           : Real;      { flight path angle }

begin

  ID := An_ID;

  { initialise orbital parameters }
  if ID = id_Moon then
  begin
    a := Moon_a;
    e := Moon_e;
    i := Moon_i;
    Om := Moon_Om;
```

```
    w := Moon_w;
    th := Moon_th;
  end else begin
    a := a0;
    e := e0;
    i := i0;
    Om := Om0;
    w := w0;
    th := th0;
  end;

  { mean motion and semilatus rectum }
  n0 := sqrt(Earth_mu/(a*a*a));
  p0 := a*(1 - e*e);

  { initial position and velocity }
  rM := p0/(1 + e*cos(th));

  Get_r(r, rM, i, Om, w, th);

  vr0 := sqrt(Earth_mu/p0)*e*sin(th);
  vn0 := sqrt(Earth_mu/p0)*(1 + e*cos(th));
  vM := sqrt(vn0*vn0 + vr0*vr0);
  gamma0 := atan2(vr0, vn0);        { flight path angle }
  Get_v(v, vM, gamma0, i, Om, w, th);

end;

procedure TKepSat.Get_r(var rr: Vector3;
                        rrM, ii, OOm, ww, tth: Real);
begin
  rr[1] := rrM*(cos(ww+tth)*cos(OOm)-sin(ww+tth)*sin(OOm)*cos(ii));
  rr[2] := rrM*(cos(ww+tth)*sin(OOm)+sin(ww+tth)*cos(OOm)*cos(ii));
  rr[3] := rrM*(sin(ww+tth)*sin(ii));
end;

procedure TKepSat.Get_v(var vv: Vector3;
                        vvM, gg, ii, OOm, ww, tth: Real);
begin
  vv[1] := vvM*(cos(ww+tth+(pi/2)-gg)*cos(OOm)
                -sin(ww+tth+(pi/2)-gg)*sin(OOm)*cos(ii));
  vv[2] := vvM*(cos(ww+tth+(pi/2)-gg)*sin(OOm)
                +sin(ww+tth+(pi/2)-gg)*cos(OOm)*cos(ii));
  vv[3] := vvM*(sin(ww+tth+(pi/2)-gg)*sin(ii));
```

```
end;

procedure TKepSat.NextPosition(tt: Real);

var
  EE              : Real;
  dum             : Real;
  ta, vr, vn      : Real;
  gamma           : Real;
  M               : Real;

begin

  { mean anomaly }
  M := 0 + n0*(tt + h);

  { solve Kepler's equation by iteration }
  if M <> 0 then
  begin
    EE := M;
    dum := 1;
    while abs(dum/(1-e*cos(EE)))>1e-8 do
    begin
      dum := EE-e*sin(EE)-M;
      EE := EE-dum/(1-e*cos(EE));
    end;
  end
  else EE := 0;

  { position }
  ta := 2*atan2(sqrt((1+e)/(1-e))*tan(EE/2),1);
  rM := a*(1-e*cos(EE));
  Get_r(r, rM, i, Om, w, ta);

  { velocity }
  vr := sqrt(Earth_mu/p0)*e*sin(ta);
  vn := sqrt(Earth_mu/p0)*(1 + e*cos(ta));
  vM := sqrt(vn*vn + vr*vr);
  gamma := atan2(vr, vn);
  Get_v(v, vM, gamma, i, Om, w, ta);

  { longitude and latitude }
  Lon := (atan2(r[2],r[1])-tt*2*pi/Sday)*rad_deg;
  mod180(Lon);
```

```
  Lat := (atan2(r[3],sqrt(r[1]*r[1]+r[2]*r[2])))*rad_deg;

end;

function TKepSat.InSun1(r_Sun: Vector3): Boolean;

var
  x4,y4,z4: Real;
  sa,ca,sb,cb: Real;
  rr: Real;
  cc: Real;

begin

  cc := Re/AU;

  rr := sqrt(sqr(r_Sun[1]) + sqr(r_Sun[3]));
  sa := r_Sun[3]/rr;
  ca := r_Sun[1]/rr;
  sb := r_Sun[2]/AU;
  cb := rr/AU;

  x4 :=  cb*(sa*r[3] + ca*r[1]) + sb*r[2] - AU;
  y4 := -sb*(sa*r[3] + ca*r[1]) + cb*r[2];
  z4 :=  ca*r[3] - sa*r[1];

  if (((sqr(y4/cc) + sqr(z4/cc)) > sqr(x4))
    or ((r_Sun[1]*r[1] + r_Sun[2]*r[2] + r_Sun[3]*r[3])
       > 0))
    then InSun1 := True else InSun1 := False;

end;

function TKepSat.InSun2(r_Sun: Vector3): Boolean;

var
  phi     : Real;
  phic    : Real;

begin
  { angle between satellite- and sun-vectors }
  phi := arccos((r_Sun[1]*r[1]
                +r_Sun[2]*r[2]
                +r_Sun[3]*r[3])/(rM*Vec3Mag(r_Sun)));
```

```
  if phi < 0 then phi := pi + phi;

  { critical angle }
  phic := pi - arctan(Re/sqrt(sqr(rM) - sqr(Re)));

  { compare }
  if phi < phic then InSun2 := True else InSun2 := False;
end;

{ TSat methods ------------------------------------------------------}

procedure TSat.Init(An_ID: Word; r_Sun, r_Moon: Vector3);

var
  vr0       : Real;   { radial velocity at epoch }
  vn0       : Real;   { normal velocity at epoch }
  gamma0    : Real;   { flight path angle at epoch }

begin

  ID := An_ID;

  { initialise orbital parameters }
  a := a0;
  e := e0;
  i := i0;
  Om := Om0;
  w := w0;
  th := th0;

  { mean motion and semilatus rectum at epoch }
  n0 := sqrt(Earth_mu/(a*a*a));
  p0 := a*(1 - e*e);

  { initial position and velocity }
  rM := p0/(1 + e*cos(th));

  Get_r(r, rM, i, Om, w, th);

  vr0 := sqrt(Earth_mu/p0)*e*sin(th);
  vn0 := sqrt(Earth_mu/p0)*(1 + e*cos(th));
  if ID = id_Drag
    then vM := 1.0000000078*sqrt(vn0*vn0 + vr0*vr0)
```

```
    else vM := 1.0003545*sqrt(vn0*vn0 + vr0*vr0);
  gamma0 := atan2(vr0, vn0) - 0.000000208;
  Get_v(v, vM, gamma0, i, Om, w, th);

  { satellite properties }
  Mass := Mass0;
  Area := AMin;
  SolarArea := AMax;

  { initialise integration vector }
  Y[1] := r[1];  Y[2] := r[2];  Y[3] := r[3];
  Y[4] := v[1];  Y[5] := v[2];  Y[6] := v[3];

  { initialise all other variables }
  Adrag[1] := 0;  Adrag[2] := 0;  Adrag[3] := 0;
  AdragM := 0;

  Aj2[1] := 0;    Aj2[2] := 0;    Aj2[3] := 0;
  Aj2M := 0;

  Asun[1] := 0;   Asun[2] := 0;   Asun[3] := 0;
  AsunM := 0;

  Amoon[1] := 0;  Amoon[2] := 0;  Amoon[3] := 0;
  AmoonM := 0;

  Asolar[1] := 0; Asolar[2] := 0; Asolar[3] := 0;
  AsolarM := 0;

  GetVars(0.0, Y, r_Sun, r_Moon);
end;

procedure TSat.GetVars(tt: Real; yy: IntegrateVector;
                       r_Sun, r_Moon: Vector3);

var
  rds, z     : Vector3;
  rdsM       : Real;
  rds3, q, f : Real;
  Angle      : Real;

begin

  { get variables from integration vector }
```

```
r[1] := yy[1];  r[2] := yy[2];  r[3] := yy[3];
v[1] := yy[4];  v[2] := yy[5];  v[3] := yy[6];

{ calculate s.c. distance from Earth centre }
rM := Vec3Mag(r);

{ calculate s.c. velocity magnitude }
vM := Vec3Mag(v);

{ aerodynamic drag }
if rho_vary
  then dens := Rho(tt, r, r_Sun)
  else dens := rho_ave;
AdragM := 0.5*dens*Cd*Area*sqr(vM)/Mass;
Adrag[1] := -AdragM*v[1]/vM;
Adrag[2] := -AdragM*v[2]/vM;
Adrag[3] := -AdragM*v[3]/vM;

{ Earth oblateness}
if ((ID = id_DragJ2) or (ID = id_All)) then
begin
  Aj2[1] := -1.5*r[1]*Earth_mu*Re*Re*J2*(r[1]*r[1]
            + r[2]*r[2] - 4*r[3]*r[3])/pw(rM,7);
  Aj2[2] := -1.5*r[2]*Earth_mu*Re*Re*J2*(r[1]*r[1]
            + r[2]*r[2] - 4*r[3]*r[3])/pw(rM,7);
  Aj2[3] := -1.5*r[3]*Earth_mu*Re*Re*J2*(3*r[1]*r[1]
            + 3*r[2]*r[2] - 2*r[3]*r[3])/pw(rM,7);
  Aj2M := Vec3Mag(Aj2);
end;

if ID = id_All then
begin

  { sun gravity }
  Vec3Diff(rds, r_Sun, r);
  rds3 := pw(Vec3Mag(rds),3);
  z[1] := r[1] - 2*r_Sun[1];
  z[2] := r[2] - 2*r_Sun[2];
  z[3] := r[3] - 2*r_Sun[3];
  q := (r[1]*z[1] + r[2]*z[2] + r[3]*z[3])/(AU*AU);
  f := q*(3 + 3*q +q*q)/(1 + pw(1+q,(3/2)));
  Asun[1] := -(Sun_mu/rds3)*(r[1] + f*r_Sun[1]);
  Asun[2] := -(Sun_mu/rds3)*(r[2] + f*r_Sun[2]);
  Asun[3] := -(Sun_mu/rds3)*(r[3] + f*r_Sun[3]);
```

```
    AsunM := Vec3Mag(Asun);

    { moon gravity }
    Vec3Diff(rds, r_Moon, r);
    rds3 := pw(Vec3Mag(rds),3);
    z[1] := r[1] - 2*r_Moon[1];
    z[2] := r[2] - 2*r_Moon[2];
    z[3] := r[3] - 2*r_Moon[3];
    q := (r[1]*z[1] + r[2]*z[2] + r[3]*z[3])
        /(r_Moon[1]*r_Moon[1] + r_Moon[2]*r_Moon[2]
          + r_Moon[3]*r_Moon[3]);
    f := q*(3 + 3*q +q*q)/(1 + pw(1+q,(3/2)));
    Amoon[1] := -(Moon_mu/rds3)*(r[1] + f*r_Moon[1]);
    Amoon[2] := -(Moon_mu/rds3)*(r[2] + f*r_Moon[2]);
    Amoon[3] := -(Moon_mu/rds3)*(r[3] + f*r_Moon[3]);
    AmoonM := Vec3Mag(Amoon);

    { solar radiation pressure }
    Vec3Diff(rds, r_Sun, r);
    rdsM := Vec3Mag(rds);
    if Insun2(Sun.r)
      then AsolarM := K*SolarArea*Sun_MFlux/Mass
      else AsolarM := 0;
    Asolar[1] := AsolarM*rds[1]/rdsM;
    Asolar[2] := AsolarM*rds[2]/rdsM;
    Asolar[3] := AsolarM*rds[3]/rdsM;

  end; { ID = id_All }

  { total perturbation  acceleration vector }
  PertAcc[1] := Adrag[1] + Aj2[1] + Asun[1] + Amoon[1] + Asolar[1];
  PertAcc[2] := Adrag[2] + Aj2[2] + Asun[2] + Amoon[2] + Asolar[2];
  PertAcc[3] := Adrag[3] + Aj2[3] + Asun[3] + Amoon[3] + Asolar[3];
  PertAccM := Vec3Mag(PertAcc);

end;

procedure TSat.dydt(var yprime: IntegrateVector;
                    tt: Real; yy: IntegrateVector;
                    r_Sun, r_Moon: Vector3);
begin

  GetVars(tt, yy, r_Sun, r_Moon);
```

```
  { derivatives }
  yprime[1] := v[1];
  yprime[2] := v[2];
  yprime[3] := v[3];
  yprime[4] := -(Earth_mu/(pw(rM,3)))*r[1] + PertAcc[1];
  yprime[5] := -(Earth_mu/(pw(rM,3)))*r[2] + PertAcc[2];
  yprime[6] := -(Earth_mu/(pw(rM,3)))*r[3] + PertAcc[3];

end;

procedure TSat.NextPosition(tt: Real; r_Sun, r_Moon: Vector3);

var
  AA,BB,CC,DD    : IntegrateVector;
  yB,yC,yD       : IntegrateVector;
  ii             : Integer;

begin

  { Runge-Kutta 4 algorithm }

  dydt(AA, tt, Y, r_Sun, r_Moon);
  for ii := 1 to 6 do yB[ii] := Y[ii] + 0.5*h*AA[ii];
  dydt(BB, tt + h/2, yB, r_Sun, r_Moon);
  for ii := 1 to 6 do yC[ii] := Y[ii] + 0.5*h*BB[ii];
  dydt(CC, tt + h/2, yC, r_Sun, r_Moon);
  for ii := 1 to 6 do yD[ii] := Y[ii] + h*CC[ii];
  dydt(DD, tt + h, yD, r_Sun, r_Moon);
  for ii := 1 to 6 do Y[ii] := Y[ii] +
    h*(AA[ii] + 2*BB[ii] + 2*CC[ii] + DD[ii])/6;

  GetVars(tt + h, Y, r_Sun, r_Moon);

  { longitude and latitude }
  Lon := (atan2(r[2],r[1])-tt*2*pi/Sday)*rad_deg;
  mod180(Lon);
  Lat := (atan2(r[3],sqrt(r[1]*r[1]
         + r[2]*r[2])))*rad_deg;

end;

{ TSun methods ------------------------------------------------------}

procedure TSun.Init(Arx, Ary, Arz: Real);
```

```
begin

  { initialise position vector }
  r[1] := Arx;
  r[2] := Ary;
  r[3] := Arz;

end;

procedure TSun.NextPosition(tt: Real);

var
  ww: Real;

begin
  ww := 2*pi*(tt + h)/Sun_period;

  r[1] := Sun_dist*cos(ww);
  r[2] := Sun_dist*sin(ww)*cos(Sun_i);
  r[3] := Sun_dist*sin(ww)*sin(Sun_i);
end;

{ Init ------------------------------------------------------------}
procedure Init_Sats;
begin
  Sun.Init(Sun_dist, 0, 0);
  Moon.Init(id_Moon);
  S1.Init(SatID, Sun.r, Moon.r);
  S2.Init(id_Kep);
end;

{ EveryStep -------------------------------------------------------}
procedure EveryStep_Sats;

begin
  if SatID = id_All then Moon.NextPosition(time);
  Sun.NextPosition(time);
  S1.NextPosition(time, Sun.r, Moon.r);
  S2.NextPosition(time);
end;

begin
end.
```

4. UNIT: DENSITY.PAS

```
{ atmospheric density function }

unit Density;

INTERFACE

uses
  Globals, Matrix;

const
  rho_ave    = 1.585e-12;  { nominal density }
  rp0        = Re + 450e3; { initial perigee point }
  H0         = 62.2e3;     { constant scale height }
  Bulge_F    = 0.7;        { day-time bulge factor }
  Bulge_lag  = 30*pi/180;  { day-time bulge lag behind sun }

var
  rhop0      : Real;

function Rho(tt: Real; r_Sat, r_Sun: Vector3): Real;

IMPLEMENTATION

uses Trig;

function Rho(tt: Real; r_Sat, r_Sun: Vector3): Real;

var
  alpha    : Real;
  rr       : Real;
  r_Bulge  : Vector3;
  r_BulgeM : Real;
  r_SatM   : Real;
  cos_phi  : Real;

begin

  { nominal }
  rhop0 := rho_ave;
```

```
    cos_phi := 0;

    { day-time bulge }
    alpha := atan2(r_Sun[2], r_Sun[1]);
    rr := sqrt(sqr(R_Sun[1]) + sqr(R_Sun[2]));
    r_Bulge[1] := rr*cos(alpha + Bulge_lag);
    r_Bulge[2] := rr*sin(alpha + Bulge_lag);
    r_Bulge[3] := r_Sun[3];
    r_BulgeM := Vec3Mag(r_Bulge);
    r_SatM := Vec3Mag(r_Sat);
    cos_phi := (r_Bulge[1]*r_Sat[1] + r_Bulge[2]*r_Sat[2]
                + r_Bulge[3]*r_Sat[3])/(r_SatM*r_BulgeM);

    { resulting density }
    Rho := (rhop0 + rho_ave*Bulge_F*cos_phi)
           *exp((rp0-Vec3Mag(r_Sat))/H0);
end;

begin
end.
```

5. UNIT: DISPLAY.PAS

```
{ display routines }

unit Display;

INTERFACE

procedure Init_Display;
procedure EveryStep_Display;

IMPLEMENTATION

uses
  Graph,                { this units contains the graph objects }
  Plot,
  Globals, Density, Sats;

var
  G1, G2 : TGraph;   { 2 graph object instances }

procedure Init_Display;
begin
  InitPlot;

  { initialise graphs }
  G1.Init(gp_4TopLeft,0,Revs,0,1,
      round(Revs),4,0,0,5,3,'Pert Accel');
  G2.Init(gp_4BottomLeft,0,Revs,445,455,
      round(Revs),4,0,0,6,2,'Altitude');

end;

procedure EveryStep_Display;

begin

  { perturbation accelerations }
  G1.PutPoint(n/NRev, 1e5*S1.AdragM, LightRed);
  G1.PutPoint(n/NRev, 3e1*S1.Aj2M, LightGreen);
  G1.PutPoint(n/NRev, 1e6*S1.ASunM, LightBlue);
```

```
    G1.PutPoint(n/NRev, 3e5*S1.AMoonM, LightCyan);
    G1.PutPoint(n/NRev, 1e6*S1.ASolarM, LightMagenta);

    { altitude }
    G2.PutPoint(n/NRev, (S1.rM - Re)/1000, LightRed);
    G2.PutPoint(n/NRev, (S2.rM - Re)/1000, LightGreen);

end;

begin
end.
```

6. UNIT: TRIG.PAS

```
unit Trig;

INTERFACE

const
  deg_rad       = pi/180;    { degrees to radians conversion factor }
  rad_deg       = 180/pi;    { radians to degrees conversion factor }

function tan(x: Real): Real;
function atan2(num, den: Real): Real;
function arcsin(x: Real): Real;
function arccos(x: Real): Real;
procedure mod180(var angle: Real);

IMPLEMENTATION

{ trig functions ------------------------------------------------}

function tan(x: Real): Real;
begin
  if cos(x) <> 0 then tan := sin(x)/cos(x) else tan := 1e20;
end;

function atan2(num, den: Real): Real;

var
  re: Real;

begin
  if den <> 0 then re := Arctan(num/den) else
  begin
    if num > 0 then re := pi/2;
    if num < 0 then re := -pi/2;
  end;
  if ((num >= 0) and (den < 0.0)) then re := re + pi;
  if ((num < 0) and (den < 0)) then re := re - pi;
  Atan2 := re;
end;
```

```
function arcsin(x: Real): Real;
begin
  if x <> 1
    then ArcSin := ArcTan (x/sqrt (1-sqr (x)))
    else ArcSin := pi/2;
end;

function arccos(x: Real): Real;
begin
  if x <> 0
    then ArcCos := ArcTan (sqrt (1-sqr (x)) /x)
    else ArcCos := pi/2;
end;

procedure mod180(var angle: Real);
begin
  while angle < -180.0 do angle := angle + 360.0;
  while angle > 180.0 do angle := angle - 360.0;
end;

begin
end.
```

Appendix C

CONTROL SOFTWARE

This appendix contains the Pascal code listings for the constellation control-related units. The contents is:

1. UNIT: CONTROL.PAS

```
{ constellation control routines }

unit Control;

INTERFACE

type

{ TController object ------------------------------------------------}
  TController = object
    u              : Real;      { control signal }
    M              : Real;      { Amax - Amin }
    x1,x10         : Real;      { state variable x1, x1 initial value }
    x2             : Real;      { state variable x2 }
    s              : Real;      { switching function }
    p1, p2, p3, p4: Boolean;    { control phases }
    nn             : LongInt;   { counter }
    ot             : Real;      { timer }
    dref           : Real;      { reference distance }
    alpha          : Real;      { feedback regulation gain 1 }
    beta           : Real;      { feedback regulation gain 2 }
    procedure Init(dr,d,h,t1: Real);
    procedure EveryStep(d,h: Real);
  end;

var
  C: TController;

IMPLEMENTATION

uses Globals;

{ second order model parameters -----------------------------------}
const
  k1 = -0.0184079;
  k2 = -0.00167156;

{ TController methods ---------------------------------------------}
procedure TController.Init(dr,d,h,t1: Real);
```

```
begin

  dref := dr;
  x10 := d - dref;
  x1 := x10;
  x2 := h;
  M := (AMax - AMin);
  p1 := True;
  p2 := False;
  p3 := False;
  p4 := False;

  if t1 = 0.0 then ot := 1e10;  { min time case }
  if t1 < 0           { min alt }
    then ot := sqrt(abs(x10)*Amin/(k1*k2*Amax*(Amax-Amin)))/OrbPer;
  if t1 > 0 then ot := t1;      { t1 specified }

  nn := 0;
  s := x1 - (k2/(2*k1*M))*x2*abs(x2);
  if s > 0 then u := -M;
  if s < 0 then u := M;
  alpha := 2.582e-5;
  beta :=  -0.0021656;
end;

procedure TController.EveryStep(d,h: Real);

begin
  { update state variables }
  x1 := d - dref;
  x2 := h;

  s := x1 - (k2/(2*k1*M))*x2*abs(x2); { switching function }

  { phase 1, first "bang" }
  if p1 then
  begin
    if (( ((x1 < 0) and (s > 0)) ) or ( ((x1 > 0) and (s < 0)) ))
    then begin
      p1 := False;
      p3 := True;
    end;

    if nn >= ot*NRev then
```

```
  begin
    p1 := False;
    p2 := True;
  end;

  if s > 0 then u := -M;
  if s < 0 then u := M;
end;

{ phase 2, "off" }
if p2 then
begin

  if (( ((x1 < 0) and (s > 0)) ) or ( ((x1 > 0) and (s < 0)) ))
  then begin
    p2 := False;
    p3 := True;
  end;

  u := 0;
end;

{ phase 3, second "bang" }
if p3 then
begin

  if (( ((u < 0) and (x2 > 0)) ) or ( ((u > 0) and (x2 < 0)) ))
  then begin
    p3 := False;
    p4 := True;
  end;

  if x2 < 0 then u := -M;
  if x2 > 0 then u := M;
end;

{ phase 4, feedback regulation (maintenance) }
if p4 then u := -alpha*x1 - beta*x2;;

Inc(nn);     { increment counter }

{ check for saturation }
if u > M then u := M;
if u < -M then u := -M;
```

```
end;

begin
end.
```

2. UNIT: FILTERS.PAS

```
{ orbit-averaging sliding filter and prection filters }

unit Filters;

INTERFACE

uses Globals;

{ TFilter object ----------------------------------------------------}
type
  TFilter = Object
    Input          : Array[1..NRev] of Real;   { array to hold samples
}
    Output         : Real;      { filter output }
    procedure Init;
    procedure Filter(Value: Real);
  end;

{ TLpred object -----------------------------------------------------}
type
  TLpred = Object
    b, c      : Real;
    Output    : Real;
    Out_1     : Real;
    procedure Init;
    procedure Pred(Value: Real);
  end;

{ TQpred object -----------------------------------------------------}
type
  TQpred = Object
    a, b, c   : Real;
    Output    : Real;
    Out_1     : Real;
    Out_2     : Real;
    procedure Init;
    procedure Pred(Value: Real);
  end;
```

```
{ variables ----------------------------------------------------}
var
  Dist_f    : TFilter;        { orbit-average distance }
  dh_f      : TFilter;        { orbit-average altitude difference }
  hlp       : TLpred;         { linear predicted altitude diff. }
  dqp       : TQpred;         { quadratic predicted distance }

procedure Init_Filters;
procedure EveryStep_Filters;

IMPLEMENTATION

uses Sats, Density;

{ TFilter methods ----------------------------------------------------}

procedure TFilter.Init;

var
  ii: Integer;

begin
  for ii := 1 to NRev do Input[ii] := 0;
  OutPut := 0;
  OutPut101 := 0;
end;

procedure TFilter.Filter(Value: Real);

var
  ii: Integer;

begin
  OutPut := OutPut + (Value - Input[1])/NRev;
  for ii := 1 to NRev - 1  do Input[ii] := Input[ii+1];
  Input[NRev] := Value;
end;

{ TLpred methods ----------------------------------------------------}
procedure TLpred.Init;
begin
  b := 0;  c := 0;
  Out_1 := 0;
  Output := 0;
```

```
end;

procedure TLpred.Pred(Value: Real);
begin
  b := Value - Out_1;
  c := Value - b*n;
  Out_1 := Value;
  OutPut := b*(n + (Nrev/2)) + c;
end;

{ TQpred methods ------------------------------------------------}
procedure TQpred.Init;
begin
  a := 0;  b := 0;  c := 0;
  Out_1 := 0;  Out_2 := 0;
  Output := 0;
end;

procedure TQpred.Pred(Value: Real);
begin
  a := 0.5*(Value - 2*Out_1 + Out_2);
  b := Value - Out_1 - a*(2*n - 1);
  c := Value - a*n*n - b*n;
  Out_2 := Out_1;
  Out_1 := Value;
  Output := a*sqr(n+(Nrev/2)) + b*(n+(Nrev/2)) + c;
end;

{ Init ----------------------------------------------------------}
procedure Init_Filters;
begin
  Dist_f.Init;
  dh_f.Init;
  hlp.Init;
  dqp.Init;
end;

{ EveryStep -----------------------------------------------------}
procedure EveryStep_Filters;
begin
  Dist_f.Filter(Dist);
  dh_f.Filter(S1.rM - S2.rM);
  hlp.Pred(dh_f.Output);
  dqp.Pred(Dist_f.Output);
```

```
end;

begin
end.
```

3. UNIT: RLS.PAS

```
{ RLS estimation of parameter k1 }

unit RLS;

INTERFACE

var
  out_mod: Real;    { model output }
  TH     : Real;    { estimated parameter }
  ERR    : Real;    { error }
  PP     : real;
  PH     : real;

procedure Init_RLS(hh, tth: Real);
procedure EveryStep_RLS(inp, outp: Real);

IMPLEMENTATION

uses Globals, Filters;

const
  alpha = 0.1;
  lamda = 0.98;

var
  PPH          : real;

procedure Init_RLS(hh, tth: Real);
begin
  PP := alpha;
  TH := tth;
  PH := Amin;
  PPH := hh;
  out_mod := hh;
end;

procedure EveryStep_RLS(inp, outp: Real);

var
```

```
  KK: Real;

begin
  PH := inp;

  { compute new KK }
  KK := PP*PH/(lamda + PH*PP*PH);

  { estimate parameters }
  out_mod := PPH + h*PH*TH;
  ERR := outp - out_mod;
  TH := TH + KK*ERR;

  { calculate new P matrix }
  PP := (1 - KK*PH)*PP/lamda;

  PPH := outp;

end;

begin
end.
```

Appendix D

MATLAB SCRIPT FILE

This appendix contains the Matlab script file used to design the constellation maintenance regulator.

1. FILE: OPTIMAL.M

```
% LQR design

% initialise
  clear;
  clc;

  OrbPer = 5606;              % orbital period
  NRev = 200;                 % number of steps per orbit
  h = OrbPer/NRev;            % time step
  Revs = 50;                  % number of revolutions
  AMax = 0.3946669;           % max cross-sectional area
  AMin = 0.0875;              % min cross-sectional area
  M = AMax - AMin;

% second order model
  k1 = -0.0184079;
  k2 = -0.00167156;

% state space representation
  A = [0 k2; 0 0];
  B = [0; k1];

% choose weighing matrices
  Q = [1 0; 0 1]
  R = 1.5e9

% calculate optimal gain
  K = LQR2(A,B,Q,R);

% closed-loop state space model
  Ac = A - B*K;
  Bc = [0; 0];
  Cc = [1 0];
  Dc = 0;

% initialise simulation
  x0 = [-10e3;0];
  time = 0:h:Revs*OrbPer;
```

```
% simulate
  [y,x] = lsim(Ac,Bc,Cc,Dc,zeros(size(time)),time,x0);
  u = [-K*x']';

% display results
  figure(1);
  plot(time/OrbPer,x(:,1));
  title('x1');

  figure(2);
  plot(time/OrbPer,x(:,2));
  title('x2');

  figure(3);
  plot(time/OrbPer,u);
  axis([0 Revs -M M]);
  title('u');

  format long;
  alpha = K(1)
  beta = K(2)
  format;
```