University of Strathclyde
Department of Mechanical Engineering

A Dynamical Systems Approach to Space-Based Embedded Autonomy

Blair Brown

A thesis presented in fulfilment of the requirements for the degree of Doctor of Philosophy

2010

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Signed: Date:

Acknowledgements

During the time I have spend at the University of Strathclyde I have had the great pleasure of working with a number of people who have contributed, either academically or socially, towards my submission of this thesis. I would like to take this opportunity to thank them.

Primarily, Professor Colin McInnes for his continued inspiration, knowledge and understanding, without which I doubt I would have ever finished this work. Secondly, other members of academic staff in the Mechanical Engineering Department including: Dr Malcolm Macdonald, Dr James Biggs, Dr Derek Bennet and Jules Simo. Also, the industrial partners at EADS Astrium Ltd, most specifically Dr Elie Alouis for his interest and input to my research. Financial support from EADS Astrium Ltd and the Science and Technology Facilities Council is greatly acknowledged.

I would like to thank my family: mum, dad and my sister Janie for always believing in me and showing me their support whenever necessary.

Finally i would like to thank my friends, their high spirits and support have made the last three years some of the best of my life. Thanks to Mark, Anna, Drew, Nichola, Jons, Richard, Tam, Coach, Fernando, Pedro, Paula, Micheal and Glynn.

Contents

List of Figures

List of Tables

Abstract

Assessments of required technologies for space systems has highlighted the need for better autonomous control. Space-based autonomous control is a requirement of many missions, including those in Earth orbit and deep-space missions. This thesis will investigate embedded autonomous control applied to both these mission scenarios.

The purpose of this thesis is to investigate the benefits that can be made from moving towards 'bottom-up' embedded autonomy that can be provided using a dynamical systems theory approach. Dynamical systems theory permits control algorithms to be developed that are based on differential equations. Different dynamical phenomenon, such as bifurcation theory and heteroclinic cycles are investigated in this thesis, and the stability properties of each system investigated through stability analysis.

Two main themes are covered. First embedded autonomy for a deep-space asteroid lander is investigated. The descent onto, contact with, and operations on, the surface of the asteroid are modelled and a quantitative comparison between conventional and new attitude and translational controllers is presented. The second main theme is embedded autonomy for computationally limited micro-spacecraft. A range of action selection controllers are developed and implemented into a generic Earth orbiting spacecraft model. The characteristics of the controllers are assessed by stability analysis.

An overarching theme is an investigation of the ability of all the controllers to recover from failures. This is a feature that all new controllers must posses in order to provide robustness in an effective manner. A range of different failure studies are presented throughout the thesis. In conclusion, this thesis presents an alternative embedded 'bottom-up' approach to spacecraft autonomous control which can enhance mission robustness, particularly for computationally limited systems. The work has been funded by EADS Astrium Ltd and the Science and Technology Facilities Council under the CASE studentship scheme.

Chapter 1

Introduction

1.1 Preface

New science missions to locations far from Earth is one of the main goals proposed by space agencies. Also, space agencies are pushing for the continued return from investment in Earth observation (EO) missions. One common system that can be introduced (or improved) in either of these scenarios, to better achieve these goals, is on-board autonomous control. Functions that are conventionally performed by ground stations or individuals 'in-the-loop' such as: planning activities, sequencing spacecraft actions, ensuring correct functioning and recovering from failures, can be delegated to the autonomous spacecraft and real benefits can be achieved [6].

Autonomy can be defined as:

"*self-government; the capacity to make an informed decision; the capacity of a system to make a decision about its actions without the involvement of another system or operator*" [7].

Most spacecraft to date have had little autonomous control authority and any decision making capability has been restricted to preprogrammed schedules. With continued interest and investment this will change, and over time spacecraft will become more independent systems.

By investing in autonomous control, space agencies are ultimately trying to build intelligent spacecraft. These intelligent spacecraft will have the capacity to make fully autonomous decisions based on the spacecraft state and the mission goals. The introduction of these autonomous algorithms will remove the need for continuous communication with spacecraft, thereby improving performance. It is also foreseen that the introduction of autonomous ground-station operations will reduce operating costs, generally estimated to be 10% of the total mission cost [8].

The improvement of any spacecraft's ability to act autonomously is considered to be

the answer to many of the problems associated with conventional spacecraft operations. In particular autonomy has been identified as a tool that will increase robustness, reduce errors and operator stress, enhance safety and cut communication bandwidth requirements [9]. Autonomy techniques have already been used in some spacecraft operations. The first example of this was Deep Space 1 (DS1) and the Remote Agent Experiment [10]. During this mission several autonomy based concepts were validated for spacecraft including: high-level goal orientated commanding, on-board planning, robust plan execution and model-based fault protection and recovery. The results of the DS1 experiments established that aspects such as: artificial intelligence, software engineering and systems engineering are all important factors in the development of spacecraft that will be able to conduct increasingly complex tasks with decreasing interaction from other systems [11].

This thesis investigates new control concepts in spacecraft autonomy. In particular, the concept of embedded autonomy is proposed as a 'bottom-up' approach, rather than conventional 'top-down' artificial intelligence. Research into embedded autonomy will be considered over a range of applications including:

- Descent control of an asteroid lander. A comparison between a novel implementation of direct adaptive control and more conventional phase-plane and potential function controllers will be drawn. The fault detection, isolation and recovery (FDIR) attributes of the respective controllers will be compared. Also, a study into the applicability of using an adaptive controller during the sampling procedure of the lander mission will be presented.

- Autonomous action selection algorithms for micro-spacecraft. Three different types of novel low computational cost action selection algorithms are developed and implemented. The relative merits of each controller are discussed and the FDIR abilities of each are shown.

- Autonomy by embedded electronics. One of the autonomous action selection algorithms is developed further by creating an analogue electronic integrated circuit where an action is selected based upon the voltages in circuitry.

By conducting investigations into each of these points the overall objectives of the thesis are met. The objectives in question are two fold. The first objective is to demonstrate the applicability of novel implementations of adaptive control techniques to space systems as a form of embedded autonomy. Here the advantages of adaptive control laws will be demonstrated and a quantitative comparison between adaptive and conventional controllers provided. The second objective is to demonstrate the use of dynamical systems theory for implementation as an action selection control law. The use of dynamical systems theory is used to decrease the burden on CPU limited micro-spacecraft. The aim of the thesis to to show how these new control laws can be implemented successfully and efficiently into future micro-space systems.

Chapter 1 provides a brief overview of the thesis objectives and aims. Current and proposed future missions that are associated with, and will benefit from, this research are described.

Chapter 2 discusses various aspects of spacecraft autonomy. A brief introduction to conventional forms of autonomous action selection and attitude control is given. Then, the theory behind the autonomous control used throughout this thesis will be introduced and described with relevance to the work undertaken.

Chapter 3 deals with the first part of the deep-space embedded autonomy concept. A model of a micro-gravity asteroid lander is presented. The dynamics and kinematics associated with the descent onto, and landing on, the surface of the asteroid are given. The subsystems of the lander that are relevant to the work conducted here are described and an investigation into possible behaviour from encountering different asteroid surface materials is conducted.

Chapter 4 applies different autonomous controllers to the asteroid lander model. Three different controllers, including a novel implementation of adaptive control, are developed and implemented, then a quantitative comparison between the three is shown. The FDIR capabilities of the three controllers are then assessed by investigating the response of the controllers to a range of faults.

Chapter 5 deals with a range of novel autonomous action selection algorithms for Earth orbiting micro-satellites. The model into which the algorithms are implemented is defined and a comparison between the controllers is discussed. Again, an FDIR study is conducted and the responses of the controllers shown.

Chapter 6 investigates the practical application of a low computational overhead micro-satellite action selection algorithm. First, the algorithm is developed and is then applied to the model defined in chapter 5 and a study regarding the controllers effectiveness and FDIR capability is conducted. Subsequently, a practical design for implementation of the control law using analogue electronics is presented. The functionality of the controller is then embedded in the electronic circuit.

In chapter 7 the work is summarised and conclusions are drawn from the results. An overview of the outcomes from the research and areas of interest for future work are presented.

1.2 Autonomy in Space Missions

Autonomy has been highlighted as one of the main focus areas in space systems engineering. It will allow space science missions to be conducted cheaper and also permit the insertion of spacecraft into more exotic, deep-space locations. By removing the 'human-in-the-loop' aspect entirely, or to some increased degree, space systems will be expected to work as efficiently, if not more, than legacy systems.

To achieve this level of autonomy, research has been conducted in the field of artifi-

cial intelligence where autonomous systems are designed to synthesize the mechanisms of logical deduction. Space systems adopting these methods of autonomy have resulted in some success, but the limitations of the techniques have also been discovered. In particular, conventional artificial intelligence cannot be completely validated for mission critical applications.

A mathematically robust technique for autonomous control is therefore desirable. In this case, through stability analysis, the convergence of any control problem can be investigated and the control engineer can then be guaranteed of any output as a result of any sequence of inputs to the control law. In this thesis such means of mathematically robust autonomous control will be demonstrated. The space systems to which they will be applied will be discussed in the following sections.

1.2.1 Marco-Polo Asteroid Sample Return Mission

Marco-Polo [12] has been proposed as an M-class joint mission by the European Space Agency (ESA) within the Cosmic Vision programme and by the Japanese Space Exploration Agency (JAXA). The primary goal of this mission is to return a sample from a near Earth object (NEO) to the Earth, where investigations will lead to a better understanding of the evolution and origins of the solar system. Some of the investigations that must be conducted before such a mission can proceed include:

- studies into the ability of the spacecraft to operate safely, including manoeuvring, close to the asteroid
- a definition of the proposed scientific experimentation that can be conducted on the surface of the asteroid
- a parallel study to understand exactly what is being collected and how it relates to the solar system evolution

The technical requirements of the Marco-Polo mission have already been assessed. Data has already been determined for the optimal launch and transfer to the asteroid, the operations that can be performed while near or on the surface of the asteroid, and the best method by which to return the sampled asteroid material to the Earth. Alongside this, the technologies that must be developed to successfully achieve the mission have been determined and a strategy for realising these has been identified.

Mission Specifications

The general mission time-line in order to present a robust design case for this M-class mission is known. The aim is to launch the mission in 2017, with a backup in 2019. After launch the spacecraft is proposed to stay in the proximity of the asteroid for 8-24 months to conduct scientific experimentation and take relevant samples. To achieve this

time-line, new technologies necessary for successful completion of the mission should be at technology readiness level (TRL) 5 by 2009, or plans must be in place to raise the TRL to 5 by 2011.

There are a range of options regarding the spacecraft system design. Due to the complexity of landing the spacecraft on the surface of the asteroid, a dual spacecraft with a small lander and parent orbiting system was initially proposed but due to mating/demating problems a final single system option was selected. This spacecraft will use a single chemical propulsion system for both outbound and return journeys to the asteroid but with a separate propulsion system for attitude control during the landing phase. The samples collected from the asteroid will be held in a seperate capsule, which will be the only element of the spacecraft to return to the Earth's surface.

For orbit injection the spacecraft will be launched from Kourou in French-Guiana by a Soyuz-Fregat. The asteroid chosen is 1999JU3 [13] due to its close proximity to the Earth at the time of the 2017 launch.

There have been similar systems developed in the past regarding landing on and operating on the surface of other solar system bodies, examples include Rosetta/Philae [14] and elements of the Mars/Venus Express [15] [16] missions. The proposed landing spacecraft will therefore be developed using the bests aspects of these missions.

The spacecraft is designed to land at any location on the illuminated side of the asteroid using visual navigation to ensure the landing velocities are limited and a meter-level landing accuracy is achieved. The target asteroid selection (with a rotation period of 7.5 hours) ensures that there is sufficient time to carry out the descent, sampling and ascent under full illumination conditions. The impact energy is absorbed using 3 legs which are sized to operate up to 3 times for multiple sample opportunities.

A dual-floor hexagonal shape is the main body for the spacecraft. The floor towards the base of the spacecraft supports the fuel tanks, thrusters for orbital transfer and insertion, pipework, landing gear, and proximity cameras and landing sensors. The upward facing floor supports the remote sensing payload and avionics with radiators for heat transfer. The thrusters for attitude control during landing are mounted on the outer framework of the spacecraft. This baseline concept can be seen in figure 1.1.

Guidance Navigation and Control (GNC)

There has never been a mission with the intention of directly landing on and then taking samples from the surface of an asteroid before. As a consequence, the main requirements of the Marco-Polo GNC are those associated with the landing phase onto the asteroid. The mission will begin in orbit around the asteroid so that it can be characterised.

Although the GNC technology for descent onto the surface of the asteroid is a focus area of continued interest, a preliminary scenario for the descent phase is as follows (reference figure 1.2). The spacecraft will begin from a home position that is 2 km above the surface of the asteroid. From here it will await until the optimal time when a de-orbit burn is

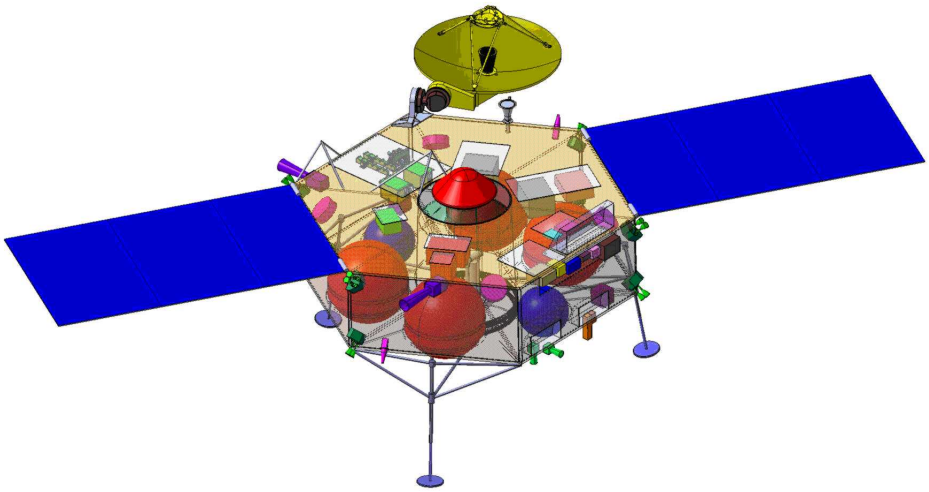

Figure 1.1: Marco-Polo lander (EADS Astrium Ltd)

performed. During this time the pointing attitude of the spacecraft will ensure electrical power through the solar arrays.

The de-orbit burn will subsequently enable the spacecraft to leave the home position en-route to the gate position. This gate position is defined at 500 m above the landing site on the surface of the asteroid. The attitude of the spacecraft during this phase is arranged so eclipse is avoided and power is maintained through the solar arrays.

Another burn is now performed to stabilise and hold the spacecraft at the gate position. The spacecraft waits at the gate position for one hour while all final communications with Earth are concluded. During this hour the spacecraft controls its velocity so it is zero relative to the asteroid surface below. This is the point where full autonomy is passed to the spacecraft for the remaining phase of the descent.

Now that the spacecraft is stable at the gate position, a controlled vertical descent can commence. An initial vertical velocity of 20 cm/s is initiated and the spacecraft moves towards the surface of the asteroid while maintaining the lateral velocity of the spacecraft with respect to the desired landing site to be zero.

When the spacecraft reaches 50 m above the surface of the asteroid the vertical velocity control is removed and the spacecraft is free to fall towards the surface of the asteroid. During this time lateral control continues. At the point where the spacecraft is 15 m above the asteroid all control is removed until contact is made.

For the purposes of mapping and then selecting the desired landing location the MAGELLAN [17] algorithm is used. MAGELLAN is an autonomous feature extraction and matching algorithm that uses visual information obtained from the on-board cameras. Being autonomous it reacts quickly to features of the asteroid encountered by the camera during descent. Determination of an appropriate descent path is provided by the navigation for planetary approach and landing system software (NPAL) [18]. This technique

alongside a doppler velocimeter gives an accurate definition of the spacecraft velocity and position.

Position and velocity control is performed by 10 N hydrazine thrusters arranged in an 8-thruster configuration. The attitude is estimated by a gyro-stellar estimator. Star trackers will also be used to detect the target in the acquisition/insertion phase.

The measurement of the relative attitude of the spacecraft with respect to the terrain is required for the last part of the descent. Vision-based relative navigation could be used for that purpose, a back-up option could be to use multiple beams such as the laser range finder that has been used successfully on the Hayabusa mission [19].

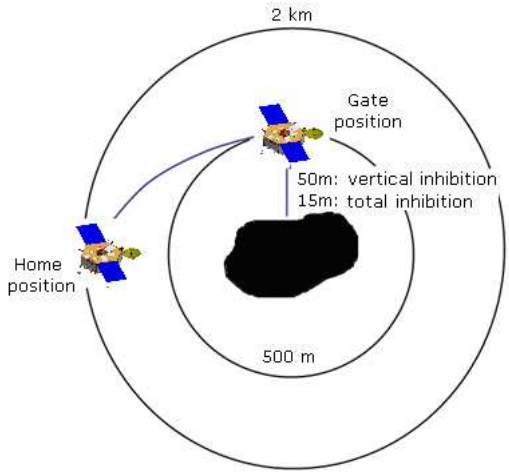

Figure 1.2: Marco-polo descent phase (EADS Astrium Ltd)

Spacecraft Leg Configuration

Landing simulations carried out have proven that two options for the leg deformation processes are feasible. Primarily, a simple crushable damper integrated into the legs shows stable landing can be achieved within all the envisaged landing scenarios. Another option considered is a spring-ratchet assembly where the kinetic energy absorbed by the spring is stored by a ratchet.

The landing gear consists of three independent legs. Each leg consists of a tube which includes either the crushable material, or spring (plus external ratchet), at its top and a piston tube that penetrates into the crush material, or spring cavity, while landing.

Technology Development

There are several technologies that need to be developed to the required TRL by 2011. The technologies outlined in an EADS Astrium Ltd assessment study all related to the new scenarios expected to be encountered in the proximity to the asteroid.

The GNC software discussed above has not been used in this situation before and so must be validated. The leg configurations for the spacecraft have not been developed and

tested. Methods for obtaining a sample of the asteroid regolith have not yet been developed. Due to the composition of the asteroid being unknown, the sampling mechanism will have to be capable of operating with a wide range of materials. Finally, the Earth re-entry capsule is an area of development that needs consideration.

For each of these technology development areas activities have been planned. This thesis aims to consider two of these areas, the GNC for proximity operations and the contact dynamics for the spacecraft legs. All of these activities are planned to be finished before 2012. Therefore they all should be available in time for the proposed Marco-Polo mission planning where the earliest launch date is November 2017.

Relevance to Thesis

The asteroid chosen for the Marco-polo mission is 1999JU3. Although the semi-major axis of the asteroid's orbit is close to the Earth's, the eccentricity and inclination of its orbit means that it does not pass close enough to the Earth to make communication delays insignificant. Therefore, a degree of autonomy is required regarding aspects of the mission where descent to the surface of the asteroid and operations on the surface of the asteroid must be conducted without continuous contact with a ground team.

A section of this thesis is dedicated to the autonomous control problems encountered in an asteroid sample return mission. The main focus will be on attitude control during the descent and contact phase on the surface of the asteroid and a subsequent translation control during the regolith sampling procedure. Alongside the attitude and translation control studies, a range of FDIR studies are conducted where the response of each of the controllers to failures in the hardware of the spacecraft system are given. The general aspects of an asteroid lander autonomous attitude (and translational) control considered and investigated in this thesis are given in table 1.1.

Table 1.1: Asteroid lander control relevance to thesis

Key Issue	**Impact of Autonomy**	**Impact from Thesis**
Uncertain contact dynamics due to unknown asteroid material	Controller must be robust and capable of dealing with changes in real time	Direct adaptive control investigated for real time gain changing properties
Uncertain regolith properties	Controller must deal with uncertain translation using minimal fuel during sampling process	Direct adaptive control algorithm capable of working in uncertain environments
Possibility of encountering unexpected faults during operation	Controller must react quickly to recover from any failure scenario	Direct adaptive control stability properties can be verified and validated

1.2.2 Action Selection in Spacecraft

The process involved when a spacecraft chooses an appropriate action is most typically defined before the mission begins. This process is known as tasking and scheduling. The long-term tasking of a spacecraft deals with accomplishing a mission objective over the life-span of the mission. Short-term tasking defines what action the spacecraft should be doing at that specific moment. It is this short-term action selection that is investigated in this thesis.

Although the main focus for autonomous action selection in current missions is in the field of rovers [20][21][22], there also exists similar requirements for orbiting spacecraft. These deep-space planetary spacecraft are used to return the data collected by rovers to Earth as the rovers themselves do not have the necessary bandwidth needed for this task. These spacecraft are subject to the same communication time delays while selecting the appropriate tasking to return data to Earth.

The following sections will discuss the evolution of the control algorithms used for conducting such tasking. The first use of algorithms are presented alongside the development of tasking algorithms that are of interest to the work in this thesis. This is mainly for micro-satellites where the CPU computational power available for the algorithm is limited.

Deep Space 1

Deep Space 1 (DS1) [10] was the first of NASA's New Millennium missions. It was used as a platform for testing a variety of new technologies and most relevant to this thesis, DS1 tested an artificial intelligence system designed to plan and execute spacecraft activities. This Remote Agent Experiment (RAX) [23] acted as a synthetic member of the operations team and scheduled tasks on-board the spacecraft. RAX defined its task schedule by combining the goals provided by the ground team with the data that instrumentation provided regarding it own internal state. These developed plans where then executed while monitored by a ground team to assess their efficiency. An additional aim of the RAX experiment was to redefine the task scheduling when artificial faults were imposed on the system.

RAX was made up of three components, each playing a significant role. The planner and scheduler (PS) produced flexible plans, specifying the basic activities that must take place in order to accomplish the mission goals. The smart executive (EXEC) carried out the planned activities while the mode identification and recovery (MIR) monitored the health of the spacecraft and attempted to correct any faults that occurred. These three elements worked together and communicated with each other to make sure that DS1 accomplished the goals of the mission.

When DS1 was launched in May 1999 a milestone in the history of development of spacecraft autonomy was achieved. In two separate experiments the remote agent was given full control of DS1 and demonstrated numerous autonomy concepts ranging from

the ability to respond to high level goals by generating and executing plans on-board, to robust plan execution and model-based fault protection.

On completion of the DS1 mission it was proposed that future work regarding autonomous control should be conducted. The RAX framework was to be improved upon in three categories: improvements in the capabilities of each of the three main components, improvements in the applicability of the framework, and an investigation into possible missions where the framework could be utilised.

The RAX framework was pursed by development of contingency planning that enabled the PS to create a plan with multiple options that could be chosen if any of a range of likely events were to occur, reducing the need to re-plan. By improving the ability of MIR to handle uncertainties permitted better tracking of multiple plans the spacecraft may have been following. A range of new user-centric tools for developing the associated algorithms for RAX were created. This implies that RAX could be implemented into similar technologies faster, while being easier to validate. The technology in DS1 and RAX was also transferred to other NASA missions [10].

PROBA - Project for Onboard Autonomy

PROBA [24] was successfully launched by ESA in 2001, initially for a one year mission. The purpose of the mission was to demonstrate new onboard technologies and the opportunities and benefits of increased spacecraft autonomy.

The mission was based around not only the spacecraft using on-board autonomy but the ground-segment of the mission being automated as well, greatly decreasing the operational costs. Some of the autonomy functions operated on-board PROBA included; operation and resource management, control of camera pointing, task scheduling and data transmission. PROBA automated ground-segment functions included: spacecraft pass operations, spacecraft performance evaluation, and user requests being autonomously transferred to the spacecraft via the Internet.

Successfully demonstrated autonomy elements in the PROBA mission were mainly attributed to the autonomous operation of the attitude and control navigation system (ACNS) [25]. A general ACNS is responsible for acquiring and maintaining on-board knowledge of the attitude and orbit for applying the necessary attitude manoeuvres. Another important aspect of the PROBA mission was the demonstration of autonomous fault detection isolation and recovery (FDIR). Any fault occurring in the ACNS system was compensated for by new control commands as a consequence of any detected failure.

When the spacecraft was operating normally all attitude information for the spacecraft was acquired autonomously. Also, knowledge of the spacecraft orbit was acquired autonomously via a GPS receiver, in another important demonstration of on-board autonomy.

Micro-spacecraft

To support future science missions a new trend is to investigate decreasing spacecraft size by orders of magnitude. This is becoming more practical with rapid electronic advances and decreasing size and in greatly increasing capability with very low power consumption. This decrease in size also directly benefits the mission in lower launch costs.

Selecting a weight of < 1 kg and the size of a 10 cm cube, a design has been established termed the CubeSat (figure 1.3). Alongside this a launch tube was developed that can hold three CubeSats. The launch tube is easily attachable in almost any location on the last stage of an expendable launch vehicle (ELV) and requires only one signal to open the launch tube door for release.

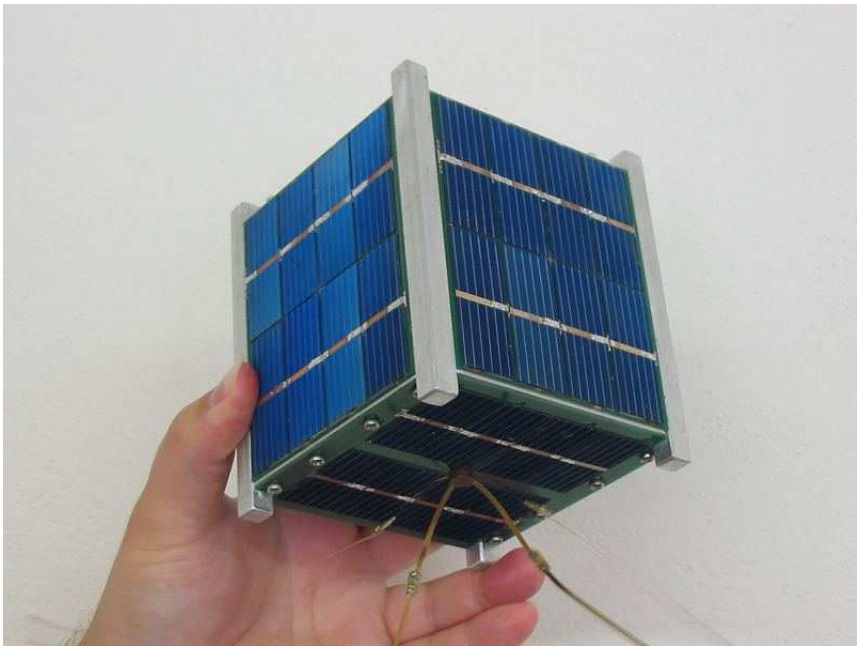

Figure 1.3: CubeSat [1]

CubeSats have become a standard in developing low cost space missions. The introduction of CubeSats into an otherwise agency dominated industry has permitted smaller institutions such as some businesses and universities to conduct their own space missions. CubeSat missions have been involved as technology demonstrators [26] and for small scale science experiments [27], but they can be utilised for complex science missions where multiple cheap spacecraft are needed in different locations for taking measurements at the same instant [28].

Along with the benefits that are gained from moving to smaller spacecraft systems there are inherited disadvantages. Most of these come as a result of the small physical size of the system. Everything that is necessary for the spacecraft to survive and operate must be contained within the small physical dimensions of the CubeSat. This means that all subsystems, including conventionally large subsystems such as power and propulsion,

must either be miniaturised or alternative means incorporated. Another drawback to the CubeSat system is that for orbit insertion the CubeSat is usually 'piggy-backed' with a larger more expensive system, so the CubeSat must go to the same orbit as the primary payload.

Relevance to Thesis

With regards the implications of the previous examples of spacecraft tasking and command scheduling the aim of this thesis is to develop means of autonomous action selection for use in miniature spacecraft, such as the CubeSat. One of the main problems associated with a decrease in the size of a spacecraft is the decrease in the mass, volume, and power available for computational processors.

Any micro-spacecraft action selection algorithm will not be capable of using conventional means of autonomous control, such as artificial intelligence, as these traditionally command significant computational load and memory allocation. To overcome these limitations the use of dynamical systems theory as a control method is investigated. Using this method, dynamical phenomenon, implicit in the differential equations that describe the system to be controlled, are manipulated resulting in an efficient means of controlling a computationally limited spacecraft. The general aspects of micro-satellite autonomous action selection considered and investigated in this thesis are given in table 1.2.

Table 1.2: Micro-spacecraft control relevance to thesis

Key Issue	**Impact of Autonomy**	**Impact from Thesis**
Conventional forms of autonomous action selection are computationally expensive	Micro-spacecraft need control algorithms that use low processing power and memory allocation	Dynamical systems theory investigated as alternative means of autonomous action selection
Traditional means of action selection are expensive to develop and cannot be validated and verified mathematically	Cheaply constructed micro-satellites need robust control that has good stability characteristics	Stability analysis of control algorithms provide action selection assurance
Small satellites have limited interactions with ground teams	Micro-spacecraft must operate for long time periods without assistance	Control algorithms presented can be simulated and outcomes predicted over long time frames

1.3 Thesis Goals

Throughout this thesis ideas taken from dynamical systems theory are applied to two main topics. The first of these is spacecraft autonomous attitude and translational control and the second is spacecraft autonomous action selection. Consequently the main aim of this

thesis is to provide new and novel approaches to the control problems that exist in current and proposed future space missions that result as a consequence of conventional forms of control.

Two specific space systems have been selected for investigation for during the course of this research. These were selected based on the current needs and proposed missions of space agencies, industrial companies, and academic institutes. The first type of spacecraft is an asteroid lander. This system was chosen as a result of the proposed ESA Marco-Polo mission which aims to return asteroid regolith to the surface of the Earth. The second spacecraft type is the micro-satellite. Organisations are becoming interested in micro-satellites due to the capability of launching independent experiments at much lower costs. By investigating various scenarios using the two spacecraft types, a broad range of applications for the proposed controllers can be considered.

Various problems exist with some of the conventional methods of autonomous control for spacecraft. Algorithms involving expert systems tend to be expensive in the computational cost associated with running the algorithm onboard the spacecraft. Alongside these there exists a real problem in the validation and verification of these methods. A control designer will conduct intensive testing processes on a controller covering every possible scenario that the spacecraft may encounter to ensure the controller works efficiently, but unexpected scenarios cannot of course be captured.

All of the controllers investigated in this thesis therefore aim to cover some of the conventional problems associated with autonomous control. By developing controllers based on dynamical systems theory, which results in the controllers consisting of sets of differential equations, mathematical stability analysis can be applied to determine their stability characteristics. Controllers are developed that utilise the inclusion of control parameters into the differential equations that lead to changes in the qualitative nature of the equations as these control parameters are varied. It is these changes in characteristics that constitute the action selection procedures developed. Also, as the control algorithms exist solely as a set of differential equations, the costs associated with CPU computation and memory allocation are decreased meaning that in principle they can be applied to smaller space systems.

1.4 Published Work

Dynamic Autonomous Control for Future Space Science Mission, IAC-08.E2.2.6, International Astronautical Congress, Glasgow, 29th September - 3rd October 2008 [Winner of best UK student entry to student competition].

Autonomous Control of an Asteroid Lander, European Air and Space Conference, CEAS-09.4.E4/5.1 Manchester, 26th - 29th October 2009.

Dynamic Intelligent Autonomous Control of an Asteroid Lander, 224(8), 865-879, Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, January 2010.

A Dynamical Systems Approach to Micro-Spacecraft Autonomy, AAS 10-179, AIAA AAS Specialists' Conference, San Diego, 14th - 17th February 2010.

Chapter 2

Autonomous Control

2.1 Introduction

Before a detailed description of the autonomous control methods developed in this thesis are described, a precise definition of what is implied by autonomy must be given. Autonomy is different from automation. Automation is used widely in many industrial applications and can be described as:

"*the practice of using computers or advanced electronics to conduct repetitive procedures*" [29].

Automation procedures expect a specific result as a consequence of a defined input parameter set. Autonomy, however, is an entirely different field of study. Autonomy, with regards to spacecraft engineering, may be defined as:

"*the practice of using a reasoning engine to make decisions and implement actions on the basis of a complex set of input parameters*" [29].

As discussed in chapter 1 autonomous control algorithms have begun to appear in space systems. Most notably in deep-space science missions such as past and current rovers [30][31]. Autonomy algorithms are based upon achieving a goal, therefore they tend to be adaptable to changes in the system state, or environment, during operations.

In this chapter a review of autonomy is presented. To begin the review general characteristics of autonomy with respect to application in the space industry will be defined and discussed. The importance of, and the benefits to be gained from, the implementation of autonomous systems in spacecraft is discussed. Then, the individual components of autonomy that comprise an overall framework of autonomous systems are described. Emphasis will be placed on the areas of autonomy addressed in this thesis.

Following this will be a review of artificial intelligence (AI). AI has made significant

advances and has been responsible for improvements in the ability of spacecraft to perform tasks autonomously. It is therefore prudent to have an understanding of these techniques and how they are applied to autonomy. Although modern AI methods have been proven capable of providing efficient solutions to some spacecraft autonomy problems, they are not without limitations. They are inherently hard to validate due to the nonlinear nature of the input-output mapping and they also tend to be computationally expensive.

One of the main aims of this thesis is to produce an alternative to current AI methods of autonomous control used for spacecraft. The branch of mathematics used to develop such alternative autonomous algorithms is dynamical systems and bifurcation theory, and it is with this field that the next section of this chapter is concerned. The methodology behind the utilisation of dynamical systems and bifurcation theory is presented alongside a description of the dynamical phenomenon associated with how the autonomous control algorithms are developed. Dynamical systems theory provides an alternative to AI with the benefit that the control laws all exist as one (or more) differential equations, requiring less computational effort by the spacecraft CPU. Also stability analysis techniques can be applied to the differential equations leading to methods for verifying and validating the response of controllers.

Taking the ideas of dynamical systems theory further, another of the aims of the thesis is to apply adaptive control algorithms, in a novel way, to space applications. The last section of this chapter defines the adaptive control problem, then proceeds to describe the mechanisms behind it and the stability of the control method. When a real system can be defined by a set of differential equations then there exists the possibility of a real-time control-gain varying solution that adapts to changes in the system plant and/or environment. By introducing such an adaptive controller, substantial gains can be made over conventional controllers as is demonstrated in subsequent chapters.

2.2 Autonomy in Space Systems

The promise of autonomy is that tasks which are conventionally performed by ground-station teams can be transferred to the spacecraft. By realising these autonomous algorithms costs can be saved by reducing, or completely removing, the ground-station teams and thereby reducing the responsibility these teams have for spacecrafts actions. Also, by implementing autonomous algorithms on-board the spacecraft for data analysis, the quantity of data that is transmitted back to Earth is reduced. Therefore, mission performance increases. This has been made possible by advances in fields such as miniature on-board processors and improved software development [11].

The consequence of introducing autonomous algorithms that reduce the mission operation and data analysis costs (MO and DA) and decrease the quantity of data being transmitted to Earth is that new space missions can be realised. In deep-space missions, where knowledge about the environment in which the spacecraft will operate is limited,

autonomy becomes a necessity. In such environments there is no possibility of repairing the spacecraft and increased communications delays between operator commands require that the system be capable of operating autonomously to a high level of efficiency [32].

Space-based autonomy can be characterised by six individual aspects. The development and introduction of any of these components will lead to MO and DA cost reduction and performance increases. These will be discussed next, with specific emphasis applied to the areas addressed in this thesis.

2.2.1 The Six Components of Space-Based Autonomy

The capabilities which contribute to spacecraft autonomy are categorised into six different areas that are listed below [32]:

- Guidance, navigation and control (GNC)
- Mission planning, scheduling and resource management
- Intelligent action selection
- Fault detection, isolation and recovery (FDIR)
- Onboard science data analysis
- System architecture

A brief description of each of these individual components will now be covered.

GNC

Autonomous controlled descent of an asteroid lander is one of the main topics of this thesis. This is the control aspect of GNC, and can be defined as:

"*the orientation maintenance of the spacecraft attitude and its trajectory control*" [33].

The GNC system is responsible for defining the attitude state of the spacecraft, determining the orbit the spacecraft is following, planning and executing the manoeuvres required to meet mission goals and pointing instrumentation [34].

The traditional method of GNC involves spacecraft following a predefined sequence of commands sent to the spacecraft from ground teams. The sequence of commands send to the spacecraft requires continuous assessment of the information being received and can lead to large time delays between commands. This process must be approved by each subsystem affected by the command as no command must lead to the disruption of any other task being conducted. During the implementation of any command sequence, if a fault occurs the spacecraft is placed in safe mode until a ground-based operator can help. Figure 2.1 illustrates a traditional GNC approach for spacecraft.

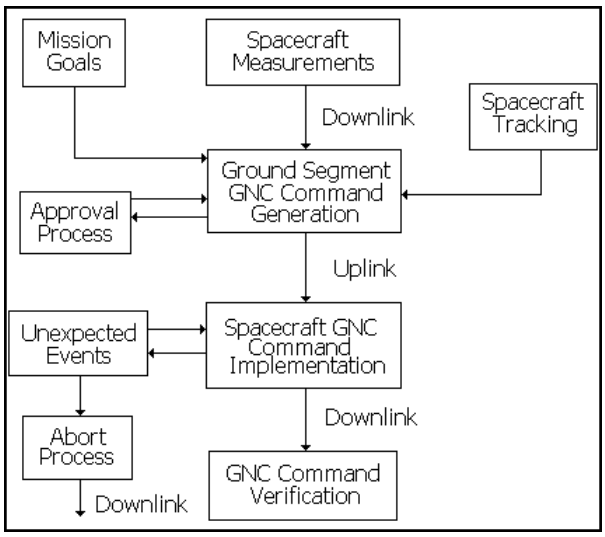

Figure 2.1: Traditional GNC (adapted from [2])

This is a lengthy and time consuming process, but with the introduction of autonomy the spacecraft can be left unattended for long periods of time. This provides a more efficient operating process and greater return on investment. Autonomous techniques succeed where traditional methods restrict the mission through ground operations and larger costs [35].

In introducing an autonomous aspect of GNC to the system the spacecraft becomes a closed-loop system. Any action that is implemented is now dependent on the internal state of the spacecraft and not parameters send from the ground station. In this approach only high-level mission commands are send to the spacecraft and all the GNC analysis is conducted on-board. The spacecraft is then in a position to handle any unexpected events by autonomously changing the applied action as the internal state of the system changes. This technique decreases the information burden placed on the data cycle between the spacecraft and the ground station and the spacecraft is left to execute its own decisions. Figure 2.2 illustrates an autonomous GNC approach for spacecraft.

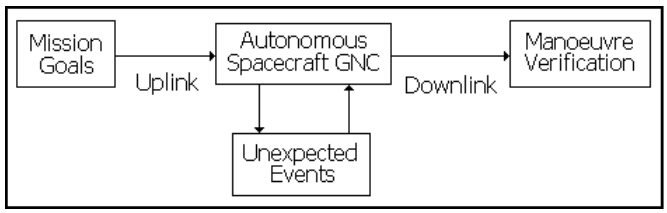

Figure 2.2: Autonomy flow chart (adapted from [2])

Mission Planning, Scheduling and Resource Management

Planning is the process which constructs the potential consequences of actions and scheduling is the assignment of resources to activities at a specific instant [36]. Similar to GNC operations, traditionally these tasks are done by ground teams and uplinked to the spacecraft. New design approaches emphasise that through autonomy, requests can be evaluated on ground and uplinked to the spacecraft, then the spacecraft later schedules the activities in an efficient manner while considering all constraints [37].

The requests that are send to the autonomous spacecraft are high-level and goal orientated. All subsequent tasking is conducted on-board. This greatly reduces mission operation costs. Also, as no task is defined by the ground team, the spacecraft is free to perform opportunity tasks based on unexpected events that would otherwise have been missed by the ground team. An extra benefit comes in a reduced communications bandwidth [38].

Resource management plays an important role in mission planning and scheduling. The resources of the spacecraft must be well managed for proper implementation of autonomy as failure of any resource can disrupt the mission planning and scheduling for a particular activity or action.

Intelligent Action Selection

The second main topic dealt with in this thesis is autonomous action selection. An autonomous action selection algorithm executes spacecraft activities while being robust against failures in the system. Each task that the spacecraft performs is either to accomplish a mission goal or maintain proper functioning.

Conventionally, action selection algorithms are based on expert heuristic rules (such as If-Then rules). Every possible outcome must be accounted for and the computational burden can be large. These algorithms can be expensive to develop and are hard to validate. This is a problem for smaller missions with limited on-board processing capabilities.

In this thesis a different approach to autonomous action selection is presented. The action selection algorithms are defined by sets of differential equations that evolve as a consequence of a change in the spacecraft internal state. This decreases the number of rules necessary for control, decreases the computational expense, and through stability analysis the algorithms can be validated.

Fault Detection, Isolation and Recovery

When a fault occurs the spacecraft system must continue to function as efficiently as possible. Spacecraft FDIR software breaks this down into separate tasks. First, there is an algorithm continually assessing the health of the spacecraft by limit checking. If any thresholds are broken in the usual operating limits of the spacecraft then that system is

isolated. The software then decides if a back-up redundancy system should be used in place. This can be expensive in doubling, or tripling, the cost of some subsystems.

There are a range of different techniques used for FDIR. Most common are fault tree analysis (FTA) [39], sneak circuit analysis (SCA) [40] and failure mode, effect, and critical analysis (FMECA) [41]. Generally a range of different control parameters are stored in the spacecraft memory in "look-up" tables that represent different operating conditions. When the operating criteria satisfies the heuristic rule a different set of control parameters are used in the spacecraft.

It is the aim of this thesis to investigate the FDIR characteristics of the autonomous GNC and action selection algorithms developed. These algorithms remove the need for separate software to conduct FDIR due to the control laws evolving with the spacecraft internal state. This introduces an embedded aspect of autonomy to the control laws, decreasing reaction times to any fault encountered.

Onboard Data Analysis

As new spacecraft are required to be smaller and lighter this poses demands in delivering sufficient bandwidth for data return. A method for reducing the need for large data bandwidths is to conduct some of the scientific analysis onboard the spacecraft. Only important and relevant information is then returned to the ground-team.

An example of this is the ESA Gaia mission [42]. At any instant there is approximately ten thousand observable objects that can be observed by space-based telescopes. The Gaia mission has onboard data analysis techniques for deciding which objects are worth investigating and only returns the information for these.

System Architecture

The system architecture brings together all the six components noted in section 2.2.1. This involves all the spacecraft subsystems using universal parameters, such as the same clock to regulate exactly when events occur in each of the individual segments of software. All individual software components are then capable of reacting to any event in any other component, improving reliability.

As many algorithms can be running simultaneously such complex software can be hard to develop. Such software can then be notoriously hard to validate. More complex missions require additional validation and this can increase total mission costs.

2.3 Artificial Intelligence

Autonomous control and artificial intelligence (AI) have been evolving simultaneous and as a result many of the techniques used in one field have been implemented in the other. Spacecraft have conventionally used AI methods for autonomous control in areas such

as automated guidance, navigation, and control, or in task management and execution. Using AI to perform these tasks leads to quicker decision making possibilities and relieves some of the burden place on the supporting team. A definition of AI that is relevant to autonomous space operations is:

"*the study and design of intelligent agents*" [43].

In this sense an intelligent agent is aware of its own environment and goals and implements procedures with the aim of achieving a goal.

2.3.1 Modelling System Structures

Modelling the structure of a system has the aim of defining the relationships between elements in the environment of the system or internal states of the system itself. These structural models do not need to be defined mathematically and are usually heuristic relationships and therefore can be learned over a period of time.

The most obvious example of a structural model is a map. Maps are based on distinctive locations and the directions and distances between them. The analogy with a mechanical system would be locations as actions (or movements) and the direction and distances would be the transitions between such actions. Robotic maps can be used for planning a sequence of tasks where the last 'location' is the desired goal of the system [44].

Rules and heuristics are another example of structural models where the model is defined as a set of conditional relationships between sensory input, from either internal state sensors or external environment sensors, and actions. Such methods form the underlying model for expert systems and fuzzy logic [44] [45].

Similar to physical modelling, the primary application for structural models is for controlling a systems. Structural models can be used to interpret sensor data, as a consequence of a change in the desired action, or to determine the appropriate control input to achieve a desired result. Planning can also be done using structural models to predict the results of actions, or to compare sensor data to expected sensor inputs. The ASPEN and COSMO [46] tools use heuristical models to determine optimal command sequences to achieve specified goal states - these systems are in use on Cassini.

2.3.2 Modelling System Unknowns

Many space-related environments are unknown and therefore the exact physics and structure of the environment is hard to model. Here a more probability oriented approach to modelling is adopted where likely outcomes as a consequence of a certain series of inputs, or subjectively known mathematical relationships, is adopted. Once the model begins to produce outputs, these results can be retained and subsequent outputs reflect the previous as the model 'learns'.

Neural Networks are the most common in the family of empirical models and are based on a perceived understanding of the workings of the human brain. The human brain has an immense number of interconnected neurons, where a number of neurons must be active to influence the actively of another neuron. Individually, each of these neurons performs the most basic of tasks but when they are combined together lead to complex behaviours. A Neural Network is a network of interconnected artificial neurons used to learn and model mathematical functions [47] [48].

Neural networks have been developed theoretically and can be trained to model practically any mathematical formula or logical expression. Limitations have been encountered where, due to the requirements for large numbers of artificial neurons to solve some simple problems, the practicality of solving complex problems is reduced. Large numbers of artificial neurons place a high burden on the controller and as a result long periods of time are needed to converge to solutions of useful meaning. One method developed, for relieving this problem, uses the idea of specialisation where a number of smaller subset neural networks are trained to individual aspects of a bigger problem and then the results are combined by a parent, higher-level, neural network.

A neural network consists of any number of interconnected layers of neurons. The input to each successive layer is a weighted sum of outputs from the neurons in the previous layer. Each neuron performs a simple mathematical calculation on this weighted sum. The first layer, called the input layer, receives inputs from the environment. There are generally as many input neurons as inputs, though inputs may outnumber input layer neurons. The output layer consists of one neuron per output and calculates the final result of the neural network calculation. The result of the network is a series of one or more numerical values, often binary. The series can represent responses to individual signals or can be combined into a single output. Figure 2.3 is an example of an architecture for a small neural network with three inputs, a two-neuron input layer, a single three-neuron hidden layer, and two final result outputs. The neuron function is represented by f, X represents inputs to the neuron and W represents the weights applied to those inputs.

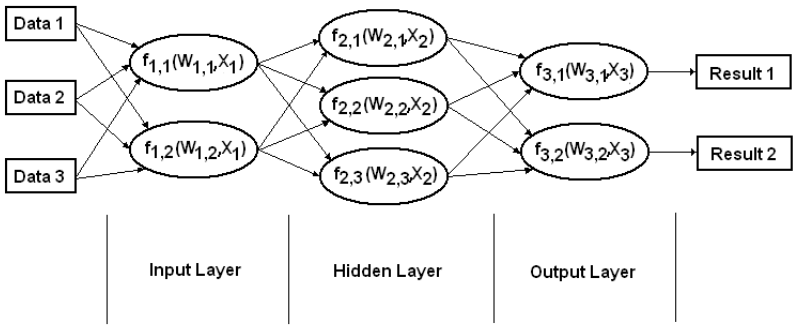

Figure 2.3: A feed-forward neural network

Each of the neurons in the network performs a simple calculation and these are defined

by that neurons internal function. These simple functions are usually based on mathematical thresholding or linear functions that contain a saturation limit. For the purposes of trained networks the shape and/or magnitude of these functions are predetermined but can be adjusted as a consequence of the eventual output of the network. Examples of variable parameters are linear gradients, magnitudes of bias terms, and magnitudes of threshold values.

A wide range of AI problems have been addressed by neural networks but only in a few specific cases have efficient solutions been found. The main application for neural networks is in highly nonlinear systems where the behaviour of the system is unpredictable and hence a degree of 'learning' is most appropriate. If the inputs and outputs of a system are being monitored and 'learned' by a neural network then a model of the system can be created without specific knowledge of the system dynamics. Such a model can then be used to control the system of unknown dynamics [49]. A space application for such a control method has been found in neutral buoyancy docking maneuvers [50].

2.3.3 Expert Systems

Expert systems are designed with the intention of imitating the response of known experts in the field. These systems use integrated mathematical formula alongside heuristic rules to gain the best from physical and structural models and determine the optimum output. Similar to neural networks, expert systems are based on human approaches to problem solving. The input to the system is based on the human operator's knowledge and hence the ability of the system to perform to a high degree of efficiency is based on the operator [43] [44] [45].

As a consequence of the operator input, an expert system is based solely on this knowledge. Any problem solving capability is therefore derived from the data interpretation through the heuristic rules. Searching methods drive the rules to determine the rules most suitable to the problem under investigation. After deciding on the specific rule, symbolic processing uses inference and the knowledge base to produce a procedure for solving the problem or obtaining the goal.

The solution to a problem is generally presented as a model: a list of rules to be followed, a network of states and goals with appropriate transitions specified, or by an explicit mathematical model. Expert systems are expected to be able to explain the reasoning behind and the method for achieving the solution.

The two classes of rules used in expert systems are procedural and heuristic. Procedural rules are specific rules that are invariant and do not alter between operators. Heuristic rules are rules built around the human operator's experience. The rules and knowledge base are generated either by human operators or by a sequence of learning. Human operators provide the data sets around how the rules solve the problems. Learning generally occurs through repetition of a task or by copying a similar task. In conventional learning some classified examples of procedures or data are shown to the system and then incorporated

appropriately. In learning by analogy, the system uses data or a rule similar, but not identical, to the circumstance and adapts it, adding the new rule or data to its program.

The most common form of expert system is fuzzy logic. Since humans generally use degrees and relative values to make decisions, fuzzy logic is often necessary to fully describe expert system rules in order to approximate them more closely. In general, fuzzy logic is applicable whenever degrees of truth are required or useful. One system using fuzzy logic in space is Earth Observing-1 (EO-1). EO-1 uses fuzzy logic to determine and maintain appropriate satellite constellation formations [51].

A system based on fuzzy logic begins with the definitions of the degrees of truth, i.e. the association of belonging to a symbolic range. The degrees of truth for various conditions, or the degree of belonging to each fuzzy set, are defined. Interpolation between these definitions determines the degree to which new data belongs to the fuzzy set, which is in turn used for mathematical logic.

While fuzzy logic allows for vague matching of symbols and rules, the rules of basic logic and sets apply. Combining rules and applying them to data allows for making implications and inferences. New sets can be generated from the old ones by set unions or set intersections.

Several types of problems have been solved with the expert system approach including data analysis, medical diagnostics, fault diagnostics, and design. Expert systems such as that on Deep Space 1 can be used to plan and execute sequences of actions in order to achieve mission goals. MYCIN [52] used the results of blood tests to diagnose blood diseases and recommend appropriate treatments. Also, Jet-X [53] is an algorithm for diagnosing military aircraft engine faults.

2.4 Dynamical Systems and Bifurcation Theory

In this section the theory associated with the concepts of dynamical systems and bifurcation theory as applied to space-based autonomy are presented. Dynamical systems theory can be used as an interesting alternative to conventional AI methods of autonomous control. Dynamical systems theory provides the control design engineer with different advantages to AI methods. Primarily, the computational cost associated with operating control algorithms based on differential equations is less than AI methods. Secondly, the differential equations that constitute the algorithm can be subject to stability analysis. This stability analysis can then be used as a means of verification and validation of the control laws to ascertain that a specific outcome will result as a consequence of a series of inputs to the control law. While these new methods provide advantages for some applications, they do not provide high-level planning and scheduling which is only possible with top down AI.

By defining the autonomous control law by a set of distinct differential equations, that vary only as a function of the system state, a different approach to autonomous control

is then realised. Conventionally, autonomous control algorithms use scenario driven tables that change the performance of the controllers based on sensor data that define the dynamic situation. The control gains that are implemented from these look-up tables are predefined from simulated control situations. A high level executive can select different look-up tables of gains in the event of failures. The aim here is to reverse this procedure by permitting the controller to define its own gains based on the current information coming from the real-time sensor information. In doing so, the controller will be able to react quickly to changes in the systems dynamics, due to the removal of the need to reference other algorithms, define certain scenarios, and implement new control gain schemes. Alongside this a reduction in memory allocation is achieved by removal of unnecessary internally stored information. This will be used as the basis of the embedded autonomy algorithms in chapter 4.

2.4.1 Dynamical Systems

The transient settling of systems that dissipate energy, by damping, is something that is seen regularly in engineering and has been thoroughly investigated. Through the transient response, the system decays towards an equilibrium position. A simple example of a system approaching an equilibrium position is to set the forcing term in equation 2.1 to zero - this is shown in figure 2.4. For this system, defining an unforced oscillator, the stable equilibrium position is at rest. Setting $k = 0.08$, if a plot of the trajectories is given on the system phase plane $(x, \dot{x})$, the solution is seen spiralling towards the origin $(x, \dot{x}) = (0, 0)$ - this is the stable equilibrium state of rest. Therefore, for any initial position on the phase plane $(x_i, \dot{x}_i)$, the equilibrium point $(0, 0)$ attracts all the trajectories. For such a system the evolution of the dynamics towards an equilibrium goal state can be verified analytically. This is unlike AI methods discussed earlier which are algorithmic and difficult to explicitly verify.

$$\ddot{x} + k\dot{x} + x^3 = Bcost \tag{2.1}$$

Similar to a system being attracted to a point, system trajectories can also be attracted to stable periodic cycles in a phase plane. Figure 2.5 describes two stable motions that can result as a consequence of different initial conditions applied to the same oscillating system defined by equation 2.1. In these cases the oscillator is now forced with $k = 0.08$ and $B = 0.2$.

The plots in figure 2.5 do not show the transient settling trajectories for the system, as there exists a range of initial conditions that would lead to different final cycles. One method for observing the trajectories settling to specific attracting cycles is the stroboscopic Poincare section [54]. In the upper example in figure 2.5 the attractor is at the fundamental frequency of the system, therefore leading to one Poincare section at point A; the transient trajectory of this system would show a continuous strobing effect converging

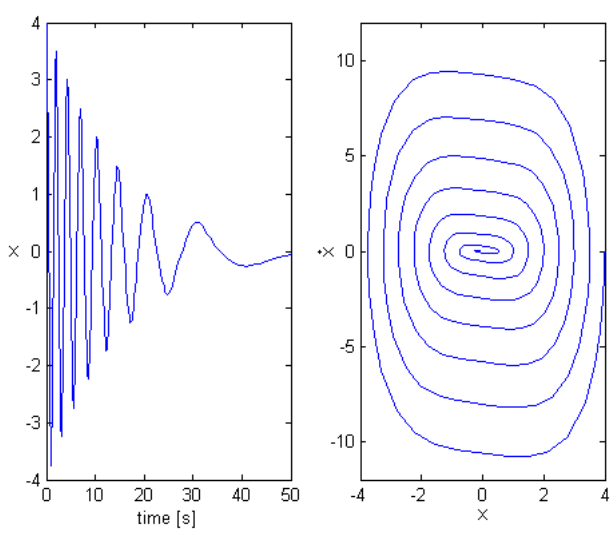

Figure 2.4: Attracting behaviour of an unforced oscillator ($\ddot{x} + k\dot{x} + x^3 = 0$)

to point A.

The four most common types of attractors are: equilibrium points, periodic cycles, quasi-periodic motion and chaotic mixing attractors. In the case for chaotic motion, a prerequisite of the attractor to develop is the existence of at least three state variables in the system. Any dynamical system can identify the final stable motions in a phase space by using these attractors and subsequent analysis.

Dynamical systems, such as the forced and unforced oscillator described above, can have their dynamic behaviour described in a qualitative manner by defining the known attractors present in its associated phase space. In more complex nonlinear dynamical systems there is the possibility of multiple attractors of different types existing at the same time, leading to complex behaviours. Conducting analysis on the systems phase space leads to an understanding of all the attractors of each type existing, and the initial conditions that lead to the specific attractors. Once this is known, the areas encompassing attracting trajectories is known as the basin of attraction. Generally, all the basins of attraction pertaining to each attractor define the entire phase-space.

By means of defining all the locations of attractors and their relative basins of attraction, the attractor-basin phase portrait of the dynamical system is defined [55]. Subsequent analysis of the phase portraits of dynamical systems using ranges of values for the system coefficients leads to a better understanding of the system behaviour. For the oscillator investigated above the system coefficients are the periodic forcing magnitude B and the damping term k. For different ranges of B and k different attractors and hence final motions exist. This can be plotted on a $k - B$ coefficient plane to gain insight into the system.

The different ranges of coefficients plotted on this plane are separated by certain boundaries. To understand what occurs at these coefficient boundaries, analysis can be

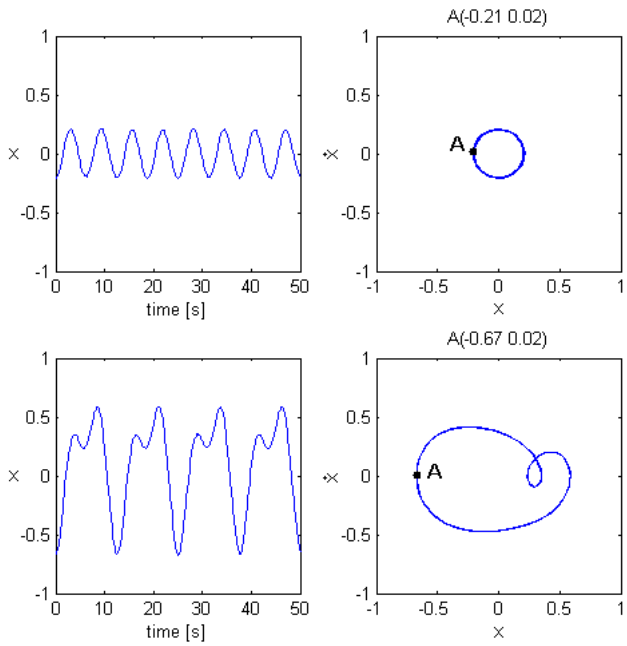

Figure 2.5: Attracting behaviour of a forced oscillator ($\ddot{x} + k\dot{x} + x^3 = Bcost$)

conducted where the coefficients k and B are assumed to be control parameters. Incrementally changing the values of the control parameters, leading to a boundary in the coefficient plane being crossed, a qualitatively different behaviour ensues. There has been a qualitative change in the long-term behaviour, associated with a change in an attractor. In theory there is a mathematically precise value of the control setting at which the qualitative change, or bifurcation, occurs. A complete understanding of such qualitative changes means knowing the topological changes in the structure of the phase portrait at the bifurcation threshold.

Throughout this thesis ideas taken from dynamical systems theory will be applied to control problems. Goal points on a phase plane will represent desired attitudes and/or translational locations, and the manner by which the trajectories on the phase planes propagate are used as a means of defining the characteristics of the controller. By manipulating sets of differential equations it is then possible to define specific attractors (or goals), and the qualitative changes in the phase plane can be developed such that they are desirable in the specific control problem. This will be used as the basis of the action selection algorithms in chapter 5.

2.4.2 Bifurcation Theory

A bifurcation is the qualitative change in the behaviour of a dynamical system as a consequence of a change in one (or more) of the control parameters present in the system. Bifurcations can occur statically through local equilibrium points, and dynamically by local periodic cycles, as well as global bifurcations in the phase portrait.

First, consider the two dimensional set of differential equations 2.2. Here a general description of the types of local attractors present can be defined by using some mathematical proof as the basis for behaviour validation.

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2, \ldots, x_n) \\ \dot{x}_2 &= f_2(x_1, x_2, \ldots, x_n) \\ &\vdots \\ \dot{x}_n &= f_n(x_1, x_2, \ldots, x_n) \end{aligned} \tag{2.2}$$

Equations 2.2 define the general form of a system of differential equations used to describe a dynamical system. The equilibrium points of this system are defined where:

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &= 0 \\ f_2(x_1, x_2, \ldots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, x_2, \ldots, x_n) &= 0 \end{aligned} \tag{2.3}$$

Stability analysis can then lead to the system being represented in vector form:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} \tag{2.4}$$

where $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_n]^T$ and:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{bmatrix} \tag{2.5}$$

with

$$\begin{aligned} a_{11} &= \tfrac{\partial f_1}{\partial x_1} \quad a_{12} = \tfrac{\partial f_1}{\partial x_2} \quad a_{1n} = \tfrac{\partial f_1}{\partial x_n} \\ a_{21} &= \tfrac{\partial f_2}{\partial x_1} \quad a_{22} = \tfrac{\partial f_2}{\partial x_2} \quad a_{2n} = \tfrac{\partial f_2}{\partial x_n} \\ a_{n1} &= \tfrac{\partial f_n}{\partial x_1} \quad a_{n2} = \tfrac{\partial f_n}{\partial x_2} \quad a_{nn} = \tfrac{\partial f_n}{\partial x_n} \end{aligned} \tag{2.6}$$

The eigenvalues of the system then result as a consequence of the determinant of the equation:

$$det\,[\mathbf{A} - \mathbf{I}\lambda] = 0 \tag{2.7}$$

where $\mathbf{I}$ is an identity matrix. This can then be used to define the characteristic equation of the system.

The stability of the system is therefore defined by the mathematical trace p and the

determinant q of the matrix $\mathbf{A}$. Another quality of the system that defines the type of stability characteristic the systems has is the discriminant ($\Delta = p^2 - 4q$) of matrix $\mathbf{A}$.

The local phase portraits of this system can be classified according to the nature of the eigenvalues of the linearised systems. All the possible cases are shown in figure 2.6. In figure 2.6 the axes of the plot define the stability properties of the dynamical system under investigation. The horizontal axis is defined by the trace of the matrix form of the differential equations and the vertical axis is the determinant of the system.

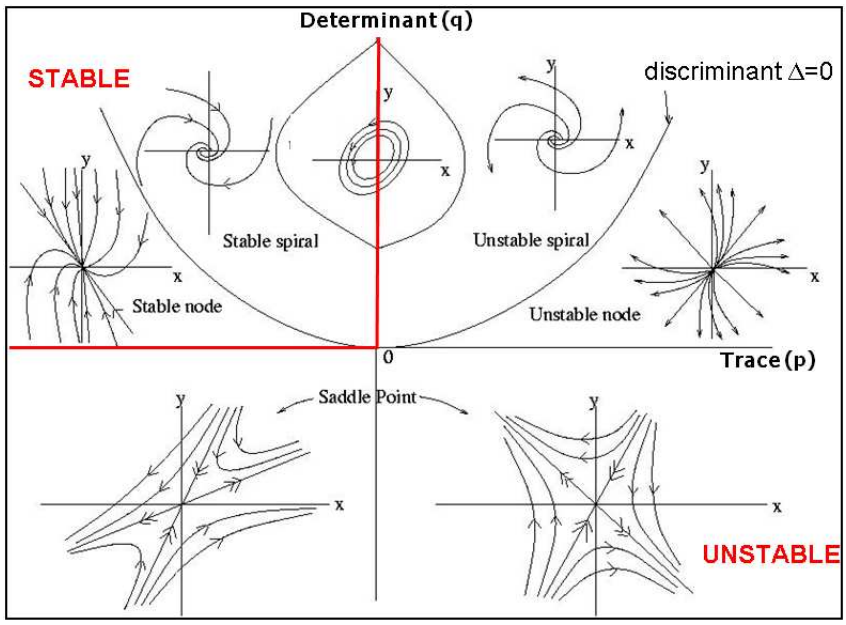

Figure 2.6: Phase portraits and root structure of a linear system [3]

Saddle-Node Bifurcation

Close to a bifurcation point the properties of a dynamical system change. Consider the one-dimensional system defined by:

$$\dot{x} = \mu - x^2 \tag{2.8}$$

where x is a scalar. The equilibrium point ($\dot{x} = 0$) gives a quadratic relationship between μ and x, and in the plane (x, μ) shows a parabolic equilibrium path. One of the branches of this parabola is unstable, and marks the separation between the basin of attraction of the stable path and the basin of attraction of the point leading to infinity.

The saddle-node bifurcation is also known as a fold bifurcation. This is because one or more of the branches of the attractor change stability 'folding' the attractor basin of the phase plane. This can be demonstrated by a second order equation analogous to equation 2.9:

$$\ddot{x} + \dot{x} + x^2 - \mu = 0 \tag{2.9}$$

In figure 2.7 three phase portraits of this systems are shown. The first is drawn at $\mu = 0.5$, so there is an attracting equilibrium point and a saddle. As the control parameter μ is decreased these two equilibria approach one another and at $\mu = 0$ they collide. The resulting critical phase portrait can been seen in the second diagram of figure 2.7. If μ is decreased past zero the equilibrium point vanishes, as illustrated in the third diagram.

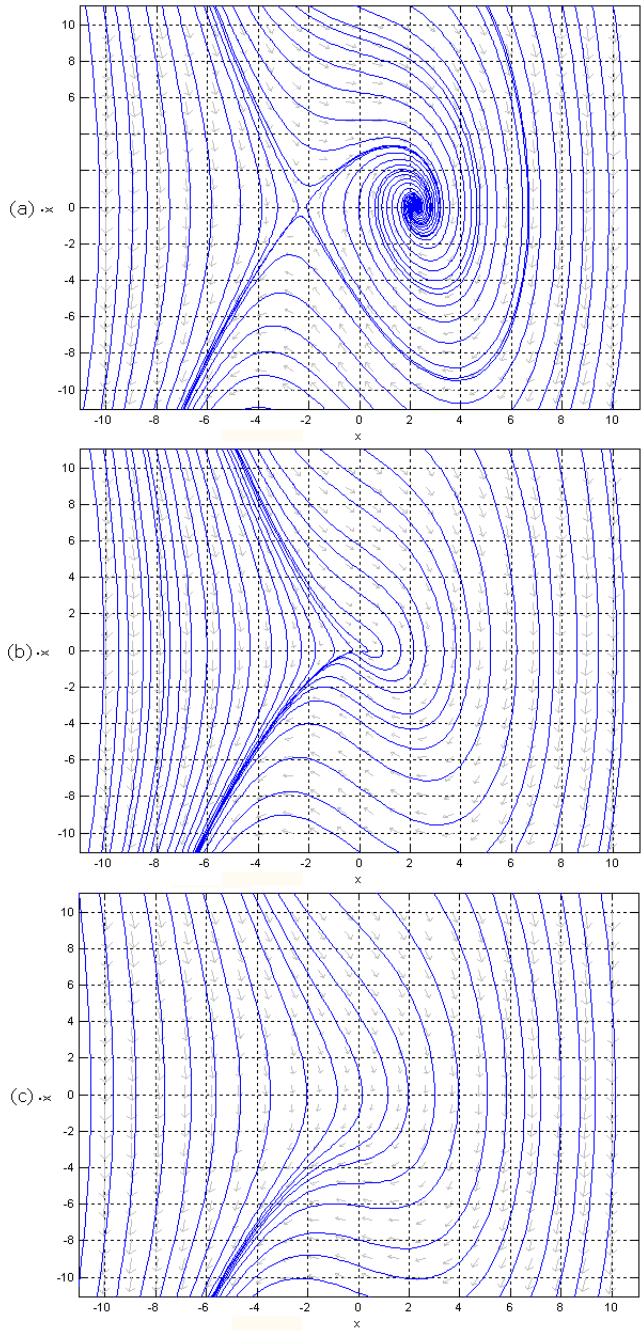

Figure 2.7: Phase portraits for saddle-node bifurcation ($\mu = 0.5$(a), $\mu = 0$(b), $\mu = -0.5$(c))

Hopf Bifurcation

Consider an unforced, damped nonlinear oscillator, whose motion can be described by:

$$\ddot{x} + b\dot{x} + \omega^2 x + \sigma = 0 \tag{2.10}$$

where σ is a non-linear function. Rewriting this equation in terms of two first-order equations gives:

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= -by - \omega^2 x - \sigma \end{aligned} \tag{2.11}$$

It can be shown that as b changes from positive to negative, the eigenvalues of the linearised system cross the imaginary axis as a pair at: $\lambda_1, \lambda_2 = \pm i\omega^2$. This bifurcation therefore addresses the change in the linear damping in the motion of an oscillator.

If $\sigma = 0$, the system reduces to being linear and the motion of the system can be qualitatively described by oscillations. The addition of a nonlinear term ($\sigma \neq 0$) leads to a different qualitative behaviour.

For analysis consider equation 2.10 where $\sigma = \dot{x}^3$ and $b = -\mu$. Here μ will act as the control parameter, so that:

$$\ddot{x} - \mu\dot{x} + x + \dot{x}^3 = 0 \tag{2.12}$$

Stability analysis of this equation shows an attracting point at the origin but exhibits a bifurcation as μ changes from a positive to a negative value. A sequence of phase portraits is shown in figure 2.8. When $\mu < -2$ the origin is an asymptotically stable node which becomes a stable spiral as the eigenvalues become complex at $\mu = -2$. When $\mu > 0$ the origin becomes an unstable spiral and the flow tends to a limit cycle. Finally at $\mu > 2$ the origin is an unstable node, and the phase paths wind away from the origin. This change in qualitative behaviour with the creation of a limit cycle is a Hopf bifurcation.

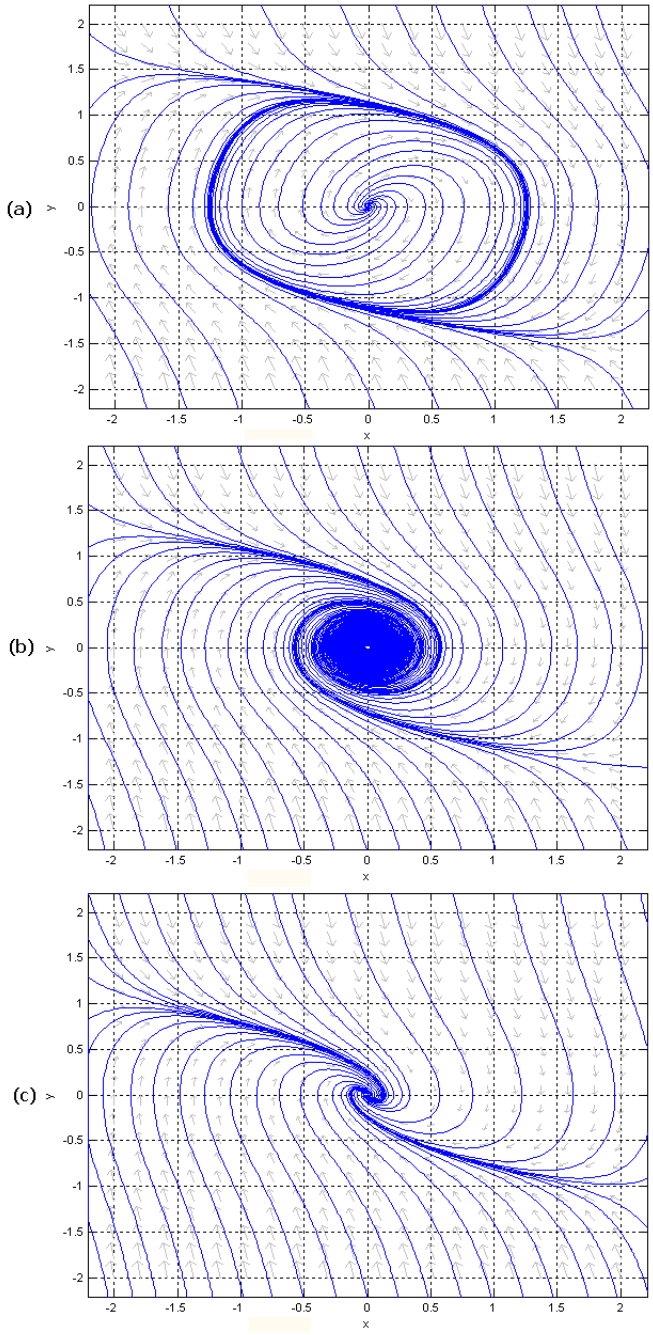

Figure 2.8: Phase portraits for Hopf bifurcation ($\mu > 2$(a), $\mu = 0$(b), $\mu < -2$(c))

During the development of the controllers in chapter 5 it will be the bifurcating qualitative changes in the behaviour of the sets of differential equations that will be used to define the response of the controllers. For example, a set of differential equations can be manipulated such that if a certain threshold is broken in a specific control parameters then a Hopf or Fold bifurcation occurs. This in turn leads to a change in the transient behaviour and hence a change in the behaviour of the control law. These bifurcations are embedded into the differential equations and hence also into the control laws, removing

the need for separate algorithm, hence reducing the computational load on the spacecraft CPU and memory allocation.

2.4.3 Heteroclinic Cycles

A heteroclinic cycle is a collection of trajectories that connect sequences of equilibria or periodic solutions via saddle-sink connections. In the phase space of a dynamical system a heteroclinic cycle is a path connecting a series of equilibrium points. The final heteroclinic connection in the series is the unstable manifold of the last equilibrium point and the stable manifold of the first equilibrium. This leads to a continuous cycling path being developed.

Heteroclinic cycles will be manipulated in this thesis in chapter 5 where sequences of saddles will be connected to produce a dynamical system that bifurcates depending on control parameters that lead to subsequent qualitative changes in the phase space of the dynamical system. The heteroclinic cycle utilised in this thesis is of the structurally stable type, therefore it has a basin of attraction that attracts all initial conditions.

A simple example of a heteroclinic cycle is shown in figure 2.9, and defined by equations 2.13 and 2.14.

$$\dot{x} = y(1 - x^2) \tag{2.13}$$

$$\dot{y} = -x(1 - y^2) \tag{2.14}$$

In figure 2.9 each of the four saddles, located at the four corners of a square, have two stable and two unstable manifolds. The connection between the stable and unstable manifolds then forms the heteroclinic cycle.

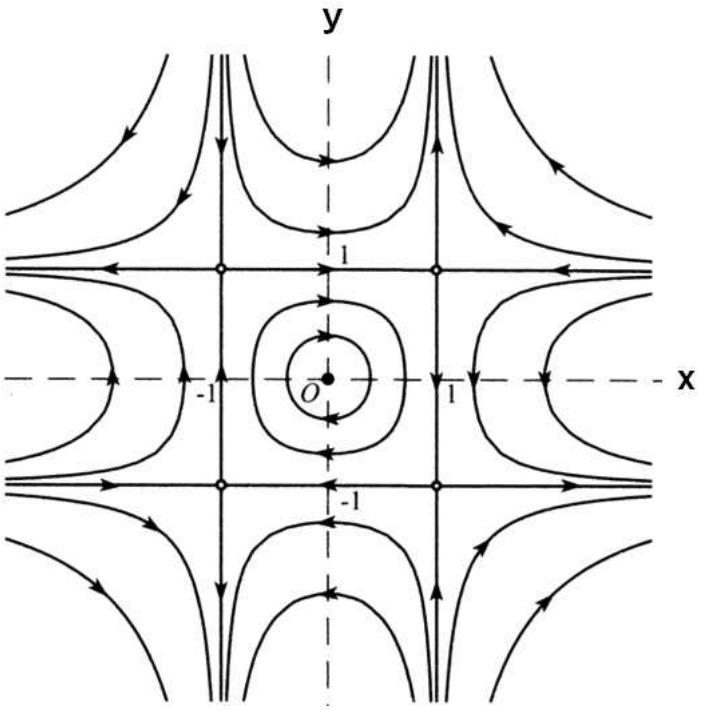

Figure 2.9: Example heteroclinic cycle

2.5 Adaptive Control

One of the main aims of this thesis is to apply adaptive control algorithms to space-based control problems as a novel means of embedded autonomy. What follows is a definition of the control problem followed by the mathematical development and reasoning behind the control algorithm that is used for two different control problems.

The first of the control problems addressed by an adaptive controller is attitude control of an asteroid lander, discussed in chapter 4. In the case of the asteroid lander the adaptive controller is used to demonstrate the advantages to be made by introducing an adaptive embedded type control in a situation where the dynamics that may be encountered are highly uncertain. The advantages gained in this case arise from the real-time gain changes that are made to deal with these uncertain situations. Following from this, the asteroid lander is subjected to FDIR studies and the advantages of having an adaptive controller are explored.

An adaptive controller is also developed for the purpose of action selection in micro-spacecraft, discussed in chapter 5. Here, the adaptive controller tries to find a solution to a resource allocation problem and in doing so actively 'searches' for the most appropriate action. Again FDIR studies are conducted and the benefits of the adaptive controller under these circumstances becomes apparent.

2.5.1 Definition of Problem

Many definitions of adaptive control exist in the literature [56] [57], but there is no standard definition. There are however an excepted range of criterion that define the commonality between all adaptive controllers, and these are used universally in the developmental process [4].

Primarily, the most common idea used in all adaptive control is a means of selecting a performance measure through which all respective controllers can be compared. Generally, this performance parameter is the same as for conventional forms of control such as settling time, rise time, or steady-state error. In a system where the goal state is changing however, other more system specific performance measures are used, for example power usage in electrical systems.

The second universal idea is the evaluation of the actual on-line process behaviour with respect to the desired or ideal behaviour. Again this is not exclusive to the adaptive controller case as the desired output of any controller can be compared to the ideal case. However, with the adaptive nature of this type of control a much wider range of useful operation can be gained and therefore this must be reflected in the evaluation process.

The third idea is the parameter adjustment procedure used to drive the process towards the desired behaviour. This is unique to adaptive control. Adaptive control can be used in two main strategies: indirect adaptive control where the algorithm implements changes in the plant dynamics and then changes the control gains accordingly, or direct adaptive control where the feedback gains change solely as a function of the internal state of the system. By defining the type of adaptive control, and subsequently the procedure used to drive the behaviour, a good understanding of the control strategy can be gained.

Conventional and/or robust controllers are designed around the premise of using *a priori*, or known system information. Adaptive controllers use *a priori* and *posteriori* information. This *posteriori* information is used to refine the control algorithm, if the output *posteriori* information matched the *a priori* information then the desired state of the controller has been achieved, otherwise the difference between the two can be used drive the adaptive nature of the controller.

In this thesis a direct adaptive control algorithm is developed based around the concept of 'following'. That being, the desired behaviour is defined by the output of a simulated reference model *a priori* designed to account for standard specifications such as overshoot, settling and rise times, and steady-state behaviour.

This section of the thesis deals with the evaluation of the system behaviour through a reference model giving a comparison between desired and actual system states. This instantaneously computed error between the desired and actual plant state is in turn used to adjust the control gains. This adaptation rule reduces an index including a positive weighted combination of the error signals and a positive weighted combination of the differences between the instantaneous gains and their ideal values.

The adaptive control problem dealt with in this thesis is control of a continuous multivariable linear system subject to uncertainty in the parameters. The continuous time, linear model control problem is solved for the linear process equations:

$$\dot{\mathbf{x}}_p(t) = \mathbf{A}_p\mathbf{x}_p(t) + \mathbf{B}_p\mathbf{u}_p(t) \tag{2.15}$$

$$\mathbf{y}_p(t) = \mathbf{C}_p\mathbf{x}_p(t) \tag{2.16}$$

where $\mathbf{x}_p(t)$ is the $n \times 1$ plant state vector, $\mathbf{u}_p(t)$ is the $m \times 1$ control vector, $\mathbf{y}_p(t)$ is the $q \times 1$ plant output vector, and $\mathbf{A}_p$, $\mathbf{B}_p$ and $\mathbf{C}_p$ are matrices with the appropriate dimensions. The objective is to find, without knowing $\mathbf{A}_p$ and $\mathbf{B}_p$, the control $\mathbf{u}_p(t)$ such that the plant output vector $\mathbf{y}_p(t)$ approximates the output of the reference model:

$$\dot{\mathbf{x}}_m(t) = \mathbf{A}_m\mathbf{x}_m(t) + \mathbf{B}_m\mathbf{u}_m(t) \tag{2.17}$$

$$\mathbf{y}_m(t) = \mathbf{C}_m\mathbf{x}_m(t) \tag{2.18}$$

where $\mathbf{x}_m(t)$ is the $n \times 1$ model state vector, $\mathbf{u}_m(t)$ is the $m \times 1$ model output or command, $\mathbf{y}_m(t)$ is the $q \times 1$ model output vector, and $\mathbf{A}_m$, $\mathbf{B}_m$ and $\mathbf{C}_m$ are matrices with the appropriate dimensions. The dimension of the model state may be less than the dimension of the plant state. Since $\mathbf{y}_p(t)$ is to track $\mathbf{y}_m(t)$, the number of model outputs must be equal to the number of plant outputs.

Fixed gain robust controllers have numerous techniques for designing the system [58] [59], although often these controllers do not satisfy the design criterion. The main reason for this is robust controllers are optimised for a specific operational range. The controllers perform well when within this range but inherent large errors when they are applied to any regime not previously anticipated during the design phase. Not only will the operating regime play a part in the performance of the controller but the range of uncertainty in the system itself may be too great for a fixed gain to control. Another problem with a fixed gain controller is that the controlled process itself may vary during operation due to life cycle changes in the system. Fatigue and stresses not present in the system may occur over time leading an eventual degradation in the performance of the controller as a consequence.

Generally, in the case where a fixed gain controller can be applied to control a system then it should be used due to the lower degree of complexity of the system and easier implementation of the control laws. For a system that may only vary to a small degree during operation then an alternative method to full adaptive control is to use a fixed gain controller with an adaptive compensator. As a result less effort is placed on the development of the adaptive controller, leading to a less complex controller, which uses less resources as a consequence.

2.5.2 Theory for Direct Adaptive Control

The concept of 'following' means that the control law is based upon driving the response of a system to that of a known desired result. Models built to artificially represent a mechanical plant include real plant design specifications so the outputs from the model respond with a specific rise time, overshoot, and settling time. When the model controller is designed well the inputs to the plant, which drive the resultant plant output, should be an accurate representation of the desired response. A block diagram of a typical model reference control system is shown in figure 2.10.

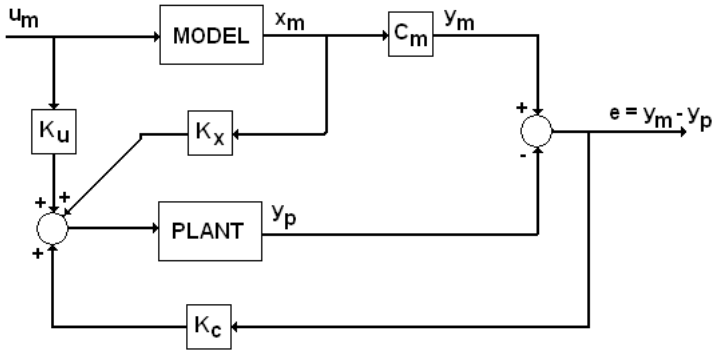

Figure 2.10: Model reference control system (adapted from [4])

Often there arises the case in space-based autonomy when the control designer has limited knowledge of the dynamics of the plant during operation, an example being deep-space missions. Therefore, benefits can be gained by adopting an adaptive control law with the premise of reducing the effect of these unknown parameters on the system. By implementing an adaptive controller a greater level of performance can be gained with little knowledge of the plant dynamics. The control engineer has then to choose between the two types of adaptive controller.

The first of these is indirect (explicit) adaptive control. Here the controller estimates the plant parameters and as a result make changes to the resultant control gains. The second type, which will be take further and used in this thesis is direct (implicit) adaptive control where the error signal in the system is driven to a minimum by adjusting the control gain as a function of this error.

In the process of developing controllers based on the idea of perfect model following (PMF), interesting properties of the controllable system arise. A PMF controller can be designed for a system of the form:

Plant:

$$\dot{\mathbf{x}}_p(t) = \mathbf{A}_p\mathbf{x}_p(t) + \mathbf{B}_p\mathbf{u}_p(t) \tag{2.19}$$

Model:

$$\dot{\mathbf{x}}_m(t) = \mathbf{A}_m\mathbf{x}_m(t) + \mathbf{B}_m\mathbf{u}_m(t) \tag{2.20}$$

where all the symbols have their usual meaning.

It can be seen that if at some time t^* the plant state vector is identical to the model state vector ($\mathbf{x}_p(t^*) = \mathbf{x}_m(t^*)$) then at all subsequent times the plant state vector will be equal to the model state vector. Also, if a controller is introduced of the form:

$$\mathbf{u}_p(t) = \mathbf{K}_c(\mathbf{x}_m - \mathbf{x}_p) + \mathbf{K}_x\mathbf{x}_m(t) + \mathbf{K}_u\mathbf{u}_m(t) \tag{2.21}$$

where $\mathbf{K}_c$, $\mathbf{K}_x$ and $\mathbf{K}_u$ are control gain matrices. Then if the plant state vector is not equal to the model state vector it should evolve to the model state asymptotically.

In the design phase for PMF controllers there are demands on the plant model structure. It has been shown that a PMF controller exists if the gains $\mathbf{K}_x$ and $\mathbf{K}_u$ can be found provided that [60]:

$$(\mathbf{A}_m - \mathbf{A}_p) = \mathbf{B}_p\mathbf{K}_x \tag{2.22}$$

$$\mathbf{B}_p\mathbf{K}_u = \mathbf{B}_m \tag{2.23}$$

Equations 2.22 and 2.23 require that either the number of controlled states is equal to the number of states, or if the case exists where the model and plant are similar. PMF controllers therefore have a number of restricting criterion to their application. In general a PMF controller is not easily applied to deep-space mission control problems because there is the likelihood that it is not possible to determine all the plant states.

To address the problem associated with knowing all the system states, a similar type of controller has been developed based on the concept of only tracking the model output vector. These types of controller work on the same principles as the controller described above only now the the system process output vector tracks the model output vector and all the associated states in the control problem are bounded. This type of controller is described next and then used subsequently as the basis for the adaptive controller in this work.

2.5.3 Output Model State Following

As discussed earlier the continuous linear model reference control problem is solved for the linear process equations 2.15 and 2.16. In addition to the criteria presented in the previous section, there are three extra assumptions made about this type of controller. Firstly, the characteristics of the system plant to be controlled ($\mathbf{A}_p, \mathbf{B}_p$) are controllable and the associated output can be stabilised. Also, the system matrices $\mathbf{A}_p$ and $\mathbf{C}_p$ are observable, i.e. all system states can be calculated. Finally, the rank of matrix $\mathbf{B}_p$ is maximum.

The requirement for this new controller is now to find the control $\mathbf{u}_p(t)$, in a manner which the plant output vector $\mathbf{y}_p(t)$ is as close as possible to the output of the reference

model, defined by equations 2.17 and 2.18. This must be done without knowledge of the plant characteristics $\mathbf{A}_p$ and $\mathbf{B}_p$. For this output model state controller the maximum dimension of the model state may be less than the dimension of the plant state, but since $\mathbf{y}_p(t)$ is to approximate $\mathbf{y}_m(t)$, the number of model outputs is always equal to the number of plant outputs.

Stability of the Output Model State Following System

For adaptive controllers it is not always possible to use the frequency domain for stability analysis techniques. This is due to the frequency domain characteristics of the modelled system changing alongside the changing plant dynamics. Other means of stability analysis do exist, such as that described in section 2.4.2, that analyse the system based on local linearisations. These techniques can provide some insight into the internal structure of the system but in general, due to the nature of the changing locations of the stability poles in the frequency domain analysis, this does not prove sufficient to guarantee stability over the operational range of the system.

There exists Lyapunov stability analysis techniques [61] that can be applied to complex dynamical systems such as nonlinear nonautonomous systems. To illustrate these concepts, consider a system:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}, t) \tag{2.24}$$

along with some definitions regarding the properties of the function [4]:

Definition*(1): "A continuous function W(*$\mathbf{x}$*) is said to be a locally positive definite function if W(0) = 0, and W(*$\mathbf{x}$*) > 0 in some neighborhood around the origin."* This implies that the function under investigation is always positive unless the goal state (W(0)) is reached.

Definition*(2): "A continuous function W(*$\mathbf{x}$*) is said to be a positive definite function if W(0) = 0, and W(*$\mathbf{x}$*) > 0 in* R^n *(W(*$\mathbf{x}$*)* $\geq$ *0 in* R^n*) and there exists some r > 0 such that:* $\inf_{\|\mathbf{x}\| > r} (W(\mathbf{x}) > 0)$" This definition states that the function is globally positive definite but now the origin is not explicitly the goal position. Note that the definition for positive definite functions require that $W(\mathbf{x})$ be strictly positive.

Definition*(3): "A continuous function* $V(\mathbf{x}, t)$ *is said to be a (locally) positive definite function if there exists a (locally) positive definite function* $W(\mathbf{x})$*, such that* $V(\mathbf{x}, t) \geq W(\mathbf{x})$ *for all* $\mathbf{x}$ *in* R^n*."* Therefore, any subset of W (here V) is positive definite if it belongs to the same neighborhood.

***Definition**(4): "A continuous function $W(\mathbf{x},t)$ is said to be a negative definite function if $-W(\mathbf{x},t)$ is positive definite."* Hence the function under investigation has the inverse properties in a similar neighborhood with negative sign.

The aim is to prove the stability properties of such a system defined by equation 2.24, given the four definitions given above. Lyapunov techniques will now be used to associate a 'Lyapunov candidate function' with equation 2.24. This Lyapunov candidate function must be a positive definite function of the system state $\mathbf{x}$.

To achieve stability it is desirable to show that the function $V(\mathbf{x},t)$ decreases along the trajectories of 2.24. As a consequence of this whenever $\mathbf{x} \neq 0$ the derivative $\dot{V}(\mathbf{x},t) < 0$ until the final goal state is reached $\mathbf{x}(t) = 0$ and $V(\mathbf{x},t) = 0$. Subsequently as $V(\mathbf{x},t)$ cannot increase, the neighborhood of operation containing $\mathbf{x}(t)$ is bounded. Also, as the end state of the system is $V(\mathbf{x},t) = 0$, $\mathbf{x}(t)$ has evolved towards the goal point $\mathbf{x} = 0$.

Validating the stability of the variation of $V(\mathbf{x},t)$ it is necessary to differentiate the function $V(\mathbf{x},t)$ along the trajectories of equation 2.24 and check whether the derivative is negative definite, hence evolving towards the goal state:

$$\dot{V}(\mathbf{x},t) = \frac{\partial V(\mathbf{x},t)}{\partial t} + \frac{\partial V(\mathbf{x},t)}{\partial \mathbf{x}} f(\mathbf{x},t) \tag{2.25}$$

The following is an example of the general stability properties of a simple system. The stability analysis of the specific control used for autonomous control of the space-based systems will be given when those problems are outlined in the subsequent chapters.

Consider the system:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) \tag{2.26}$$

To maintain the positive definite characteristic of the Lyapunov candidate function a quadratic function is chosen of the form:

$$V(\mathbf{x}) = \mathbf{x}^T(t)\mathbf{P}\mathbf{x}(t) \tag{2.27}$$

where $\mathbf{P}$ is some positive definite symmetric matrix. The derivative of $V(\mathbf{x})$ is:

$$\dot{V}(\mathbf{x}) = \mathbf{x}^T(t)\mathbf{P}\dot{\mathbf{x}}(t) + \dot{\mathbf{x}}^T(t)\mathbf{P}\mathbf{x}(t) \tag{2.28}$$

Finding the derivative of the Lyapunov function along the trajectories of equation 2.26 is achieved by substituting $\dot{\mathbf{x}}(t)$ from 2.26 into 2.28:

$$\dot{V}(\mathbf{x}) = \mathbf{x}^T(t)\mathbf{P}\mathbf{A}\mathbf{x}(t) + \mathbf{x}^T(t)\mathbf{A}^T\mathbf{P}\mathbf{x}(t) = \mathbf{x}^T(t)[\mathbf{P}\mathbf{A} + \mathbf{A}^T\mathbf{P}]\mathbf{x}(t) \tag{2.29}$$

The result is to produce the algebraic Lyapunov equation, defined as:

$$\mathbf{PA} + \mathbf{A}^T\mathbf{P} = -\mathbf{Q} \tag{2.30}$$

Since equation 2.29 must be negative definite, a theorem for the stability of this simple systems is obtained [57]:

Theorem: *"The linear time-invariant system given by equation 2.26 is asymptotically stable if and only if for any positive definitive matrix* $\boldsymbol{Q}$ *there exists a positive definite matrix* $\boldsymbol{P}$ *that satisfies the algebraic Lyapunov equation 2.30."* [62]

This theorem implies that if the system is asymptotically stable, then for any selection of $\mathbf{Q} > 0$, a positive definite solution for $\mathbf{P}$ exists.

When positive definite matrices $\mathbf{P}$ and $\mathbf{Q}$, that satisfy equation 2.30 exist, then $V(\mathbf{x})$ in equation 2.27 is positive definite and it's derivative equation 2.28 is negative definite. If the derivative is not negative definite but semidefinite, it still guarantees that $V(\mathbf{x})$ cannot increase, and thus $\mathbf{x}(t)$ remains bounded [63]. However, $\mathbf{x}(t)$ may not go to zero. Thus consider the next theorem for stability [57]:

Theorem: *"The linear time-invariant system given by equation 2.26 is stable if there exists a positive definite matrix* $\boldsymbol{P}$ *and a positive semidefinite matrix* $\boldsymbol{Q}$ *that satisfy the algebraic Lyapunov equation 2.30."* [62]

2.5.4 Controller Based on Output Model State Following

Consider the controllable system defined by:

$$\begin{aligned} \dot{\mathbf{x}}_p(t) &= \mathbf{A}_p\mathbf{x}_p(t) + \mathbf{B}_p\mathbf{u}_p(t) \\ \mathbf{y}_p(t) &= \mathbf{C}_p\mathbf{x}_p(t) \end{aligned} \tag{2.31}$$

where $\mathbf{x}_p$ is the (n $\times$ 1) state vector, and $\mathbf{u}_p$ and $\mathbf{y}_p$ are the control and output signals respectively.

The stability of the system, defined by equations 2.31, is considered for a control law based upon the Lyapunov function:

$$V(\mathbf{x}_p) = \mathbf{x}_p^T\mathbf{P}\mathbf{x}_p \tag{2.32}$$

where $\mathbf{P}$ is a positive definite symmetric matrix.

The stability of the system, with no input ($\mathbf{u}_p(t) = 0$) can be determined by the derivative of $V(\mathbf{x}_p)$:

$$\begin{aligned} \dot{V}(\mathbf{x}_p) &= \dot{\mathbf{x}}_p^T\mathbf{P}\mathbf{x}_p + \mathbf{x}_p^T\mathbf{P}\dot{\mathbf{x}}_p = \mathbf{x}_p^T\mathbf{P}\mathbf{A}\mathbf{x}_p + \mathbf{x}_p^T\mathbf{A}_p^T\mathbf{P}\mathbf{x}_p \\ &= \mathbf{x}_p^T(\mathbf{P}\mathbf{A} + \mathbf{A}_p^T\mathbf{P})\mathbf{x}_p = -\mathbf{x}_p^T\mathbf{Q}\mathbf{x}_p \end{aligned} \tag{2.33}$$

It was shown in section 2.5.3 that the linear time-invariant system is asymptotically stable if there exists positive definite matrices $\mathbf{P}$ and $\mathbf{Q}$, such that the following Lyapunov equation is satisfied:

$$\mathbf{P}\mathbf{A}_p + \mathbf{A}_p^T\mathbf{P} = -\mathbf{Q} \tag{2.34}$$

Again, consider the plant given by equation 2.31, and use the Lyapunov equation to determine if the plant can be stabilised by the constant output feedback controller:

$$\mathbf{u}_p(t) = -\mathbf{K}_c\mathbf{y}_p(t) = -\mathbf{K}_c\mathbf{C}_p\mathbf{x}_p(t) \tag{2.35}$$

where $\mathbf{K}_c$ is the constant feedback gain matrix.

The closed loop system is now:

$$\begin{aligned}\dot{\mathbf{x}}_p(t) = \mathbf{A}_p\mathbf{x}_p(t) + \mathbf{B}_p\mathbf{u}_p(t) \quad &= \mathbf{A}_p\mathbf{x}_p(t) - \mathbf{B}_p\mathbf{K}_c\mathbf{C}_p\mathbf{x}_p(t) \\ &= (\mathbf{A}_p - \mathbf{B}_p\mathbf{K}_c\mathbf{C}_p)\mathbf{x}_p(t)\end{aligned} \tag{2.36}$$

As a consequence, the closed loop system and the plant are stabilisable if positive definite matrices $\mathbf{P}$ and $\mathbf{Q}$ can be found so the algebraic Lyapunov equation is satisfied.

To develop this idea assume that there does exist some constant feedback gain $\mathbf{K}_c$ that stabilises the plant. Now a means of describing how a variable gain controller finds this associated gain is shown. Consider the controller:

$$\begin{aligned}\mathbf{u}_p(t) = \quad &-\mathbf{K}(t)\mathbf{y}_p(t); \\ \dot{\mathbf{K}}(t) = \quad &\lambda\mathbf{y}_p(t)\mathbf{y}_p^T(t); \quad \lambda > 0\end{aligned} \tag{2.37}$$

where λ is a positive control law shaping parameter. The closed loop system now contains this time varying gain:

$$\dot{\mathbf{x}}_p(t) = \mathbf{A}_p\mathbf{x}_p(t) - \mathbf{B}_p\mathbf{K}(t)\mathbf{C}_p\mathbf{x}_p(t) = [\mathbf{A}_p - \mathbf{B}_p\mathbf{K}(t)\mathbf{C}_p]\mathbf{x}_p(t) \tag{2.38}$$

To include the concept of the constant stabilising gain matrix $\mathbf{K}_c$ being present in the controllable system, add and subtract $\mathbf{B}_P\mathbf{K}_c\mathbf{C}_p\mathbf{x}_p(t)$ to equation 2.38. Note that $\mathbf{K}_c$ is known to exist but has no known value:

$$\dot{\mathbf{x}}_p(t) = [\mathbf{A}_p - \mathbf{B}_p\mathbf{K}_c\mathbf{C}_p]\mathbf{x}_p(t) - \mathbf{B}_p[\mathbf{K}(t) - \mathbf{K}_c]\mathbf{C}_p\mathbf{x}_p(t) \tag{2.39}$$

The stability of this system must now be assessed. In addition to the system state vector there is now a variable adaptive gain and this must be taken into account when defining the Lyapunov candidate equation. The final goal state will be reached when $\dot{\mathbf{x}}_p(t) = 0$ or when both $\mathbf{x}_p(t) = 0$ and $\mathbf{K}(t) = \mathbf{K}_c$.

To fulfill the requirement of a candidate Lyapunov equation that contains all variables of the controllable system consider:

$$V(\mathbf{x}) = \mathbf{x}_p^T \mathbf{P} \mathbf{x}_p + tr([\mathbf{K}(t) - \mathbf{K}_c][\mathbf{K}(t) - \mathbf{K}_c]^T)/\lambda \tag{2.40}$$

where 'tr' is the trace. The derivative of this equation is:

$$\dot{V}(\mathbf{x}) = \dot{\mathbf{x}}_p^T \mathbf{P} \mathbf{x}_p + \mathbf{x}_p^T \mathbf{P} \dot{\mathbf{x}}_p + tr([\mathbf{K}(t) - \mathbf{K}_c]\dot{\mathbf{K}}^T(t))/\lambda + tr(\dot{\mathbf{K}}(t)[\mathbf{K}(t) - \mathbf{K}_c]^T)/\lambda \tag{2.41}$$

and substituting equation 2.39 gives:

$$\begin{aligned}\dot{V}(\mathbf{x}) = \mathbf{x}_p^T \mathbf{P}[\mathbf{A}_p - \mathbf{B}_p \mathbf{K}_c \mathbf{C}_p]\mathbf{x}_p + \mathbf{x}_p^T [\mathbf{A}_p - \mathbf{B}_p \mathbf{K}_c \mathbf{C}_p]^T \mathbf{P} \mathbf{x}_p - \mathbf{x}_p^T \mathbf{P} \mathbf{B}_p [\mathbf{K}(t) - \mathbf{K}_c]\mathbf{C}_p \mathbf{x}_p \\ -\mathbf{x}_p^T \mathbf{C}_p^T [\mathbf{K}(t) - \mathbf{K}_c]^T \mathbf{B}_p^T \mathbf{P} \mathbf{x}_p + tr([\mathbf{K}(t) - \mathbf{K}_c]\dot{\mathbf{K}}^T(t))/\lambda + tr(\dot{\mathbf{K}}(t)[\mathbf{K}(t) - \mathbf{K}_c]^T)/\lambda\end{aligned} \tag{2.42}$$

By further substitution of the control law given in equations 2.37, the derivative of the Lyapunov candidate equation can now be expressed as:

$$\begin{aligned}\dot{V}(\mathbf{x}) = \mathbf{x}_p^T \mathbf{P}[\mathbf{A}_p - \mathbf{B}_p \mathbf{K}_c \mathbf{C}_p]\mathbf{x}_p + \mathbf{x}_p^T [\mathbf{A}_p - \mathbf{B}_p \mathbf{K}_c \mathbf{C}_p]^T \mathbf{P} \mathbf{x}_p - \mathbf{x}_p^T \mathbf{P} \mathbf{B}_p [\mathbf{K}(t) - \mathbf{K}_c]\mathbf{C}_p \mathbf{x}_p \\ -\mathbf{x}_p^T \mathbf{C}_p^T [\mathbf{K}(t) - \mathbf{K}_c]^T \mathbf{B}_p^T \mathbf{P} \mathbf{x}_p + tr([\mathbf{K}(t) - \mathbf{K}_c]\dot{\mathbf{K}}^T(t))/\lambda + tr(\lambda \mathbf{y}_p^T(t)[\mathbf{K}(t) - \mathbf{K}_c]^T)/\lambda\end{aligned}$$

$$= -\mathbf{x}_p^T \mathbf{Q} \mathbf{x}_p - \mathbf{x}_p^T [\mathbf{P}\mathbf{B}_p - \mathbf{C}_p^T][\mathbf{K}(t) - \mathbf{K}_c]\mathbf{C}_p \mathbf{x}_p - \mathbf{x}_p^T \mathbf{C}_p^T [\mathbf{K}(t) - \mathbf{K}_c]^T [\mathbf{B}_p^T \mathbf{P} - \mathbf{C}_p]\mathbf{x}_p \tag{2.43}$$

As $\mathbf{Q}$ is predefined to be positive definite then the first term in equation 2.43 is negative definite. However, the second and third terms in equation 2.43 do not have the same convenient property. An extra relationship is need to ensure that the full function is negative definite. This is provided by the assumption that when the matrix $\mathbf{P}$, used to satisfy equation 2.34, also satisfies the relation:

$$\mathbf{P}\mathbf{B}_p = \mathbf{C}_p^T \tag{2.44}$$

then the second and third terms in equation 2.43, relating to the varying control gain matrix, are equivalent to zero. Therefore as a consequence, equation 2.43 reduces to:

$$\dot{V}(\mathbf{x}_p) = -\mathbf{x}_p^T \mathbf{Q} \mathbf{x}_p \tag{2.45}$$

The extra condition is what is necessary to ensure that all the system states and gains are bounded. Also, as a consequence of the stability properties, all the system states will evolve to the goal state $\mathbf{x}_p = 0$. This analysis demonstrates the stability of the system with adaptive gains. Bringing together this proof using the nonlinear adaptive gains, the previous theorem states that the adaptive control algorithm is stable, allowing for the existence of two positive definite matrices, $\mathbf{P}$ and $\mathbf{Q}$, so the closed loop system, utilising the the unknown (yet real) gain $\mathbf{K}_c$, while at the same time satisfying equations 2.34 and

2.44.

Equations 2.34 and 2.44 are derived from the relationship between the input and output of the system. When both these conditions are satisfied but the matrix $\mathbf{Q}$ is positive semi-definite the system is regarded as "positive real" (PR). If $\mathbf{Q}$ is positive definite, the system is known as "strictly positive real" (SPR). As was described in section 2.4.3, when the system is PR this condition still guarantees that $V(\mathbf{x})$ cannot increase and thus $\mathbf{x}(t)$ remains bounded, hence this type of controller can be implemented.

It is a controller of the type described above that will be implemented in the cases of autonomous attitude control and autonomous action selection in chapter 4 and 5. The stability analysis described above will guarantee that convergence to the desired state will be achieved after some transient response.

Chapter 3

A Deep Space Asteroid Lander Mission

3.1 Introduction

This chapter deals with the dynamic model for the first technical challenge addressed. The problem of autonomous attitude control of an asteroid lander during descent and contact is investigated. Embedded adaptive control algorithms, developed in chapter 4, are presented and incorporated into a relevant dynamic model, developed in this chapter. The aim is to show that by the introduction of an embedded adaptive controller, issues caused by environmental uncertainties and unexpected system failures can be dealt with to a high level of robustness.

A description of the descent/landing/contact dynamics and lander model used for the lander problem is presented. First, a description of the lander model is presented. Subsequently the kinematic motion of the lander in micro-gravity conditions is described and then modelling of how the lander interaction with different types of asteroid surface material is presented and discussed.

3.2 Asteroid Lander Model

One of the proposed future flagship missions of the European Space Agency (ESA) was the asteroid sample return mission Marco-Polo, as discussed in chapter 1. Although there have been a number of past missions to asteroids, a sample has never been successfully returned. The return of asteroid material to the Earth's surface introduces new technical challenges. This thesis deals with the attitude control problem for the descent onto and contact with an asteroid in micro-gravity conditions. The concept of embedded autonomy is investigated to compensate for uncertain surface properties.

This section focuses on modelling of the Marco-Polo asteroid lander, adapted from the EADS Astrium Ltd near earth asteroid sample return technology reference study (NEA-

SR TRS) [12] shown in figure 1.1. A representation of the lander has been adopted for modelling purposes. Figure 3.1 depicts the layout of the lander as it exists in the model, where the y-axis is positive out of the page. Each thruster has a maximum thrust value of 10 N. The contractible spring constant is 1250 N/m, modelling the leg resistance to compression only. The natural length of each lander leg (before deformation) is 1 m and the mass of each leg is 7.3 kg. Laterally, the length of the lander base is 2.8 m and the mass of the main structure is 985.4 kg. These parameters shown in figure 3.1 are constant and are specified in table 3.1 and were supplied by EADS Astrium Ltd.

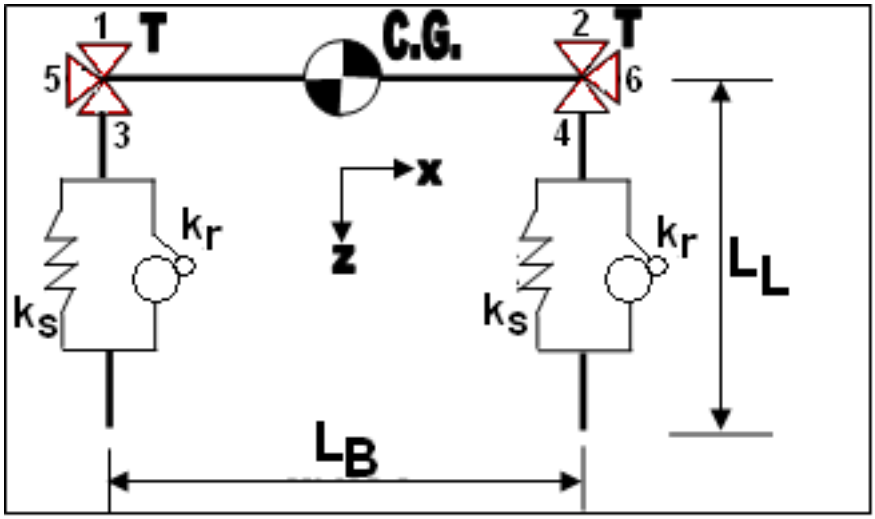

Figure 3.1: Model lander schematic

During the leg reaction to compression, the extension constant is modelled by a ratchet constant, 22500 N/m, in addition to the spring. The ratchet assembly is included as a means of absorbing the kinetic energy of the lander as it makes contact with the surface of the asteroid. The rotating ratchet allows continuous linear motion of each leg in only one direction, while preventing motion in the opposite direction by retaining the kinetic energy in the mechanical device. The general layout of the ratchet is shown in figure 3.2 - the gear can rotate around the rotational spring in only one direction, the pawl prevents rotation in the opposite direction.

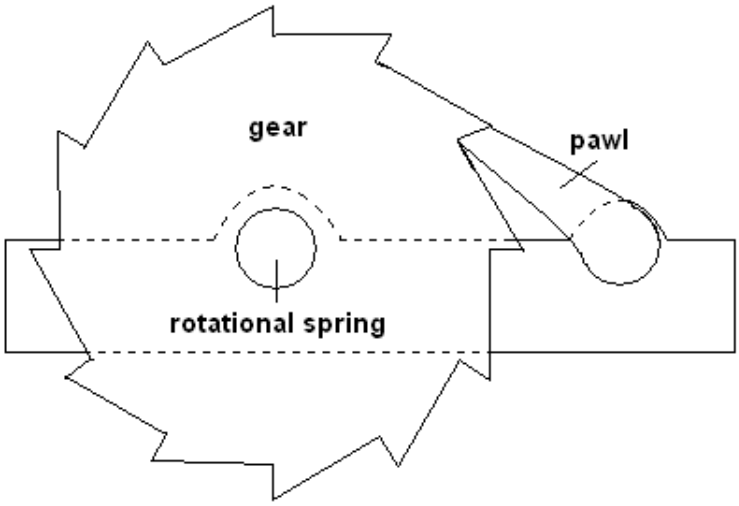

Figure 3.2: Generic ratchet

Table 3.1: Lander properties

Parameter	**Value**	**Unit**
Thrust (T)	10	N
Spring Const (k_s)	1250	N/m
Ratchet Const (k_r)	22500	N/m
Leg Natural Length (L_L)	1	m
Leg Stroke Length (L_S)	0.3	m
Base Length (L_B)	2.8	m
Base Mass (m_B)	985.4	kg
Leg Mass (m_L)	7.3	kg
Moment of Inertia (I)	677	Nm^2

Several assumptions are made about the lander:

- The mass of the lander legs (m_L) is much less than the mass of the lander base (m_B)
- The center of gravity (C.G.) of the lander lies at the mid lateral point of the lander base
- The lander base is rigid and does not deform
- The lander legs are perpendicular to the base
- The lander legs deform only in one axis, the z body-axis

The thruster layout associated with the lander is arranged in order to allocate thrusters to specific axes. The thrusters are numbered 1-6 in figure 3.1 where; 1 and 4 individually or together produce a moment in a positive sense around the y-axis, 2 and 3 produce a

moment in the negative sense, and 5 and 6 individually produce a translation positively and negatively respectively in the x-axis.

3.3 Lander Simulation Model

The lander simulation consists of a number of ordinary differential equation (ODE) solvers running at specific time intervals to integrate the relevant equations of motion in two dimensions using three degrees of freedom with respect to the translation and rotation of center of gravity of the lander. Discontinuities in the dynamics of the system, such as leg impacts, are defined heuristically and captured using Matlab algorithms.

With regards the internal dynamics of the lander model, all the equations of motion are defined in body axes, where the axis system is defined in figure 3.3. The dynamics of the model allow the system to exist in three different states, requiring three different sets of equations of motion and three separate ODE solvers. The first of these states is free-flight. This is the state the lander will assume in descent or rebound where neither leg has contact with the asteroid. This state has 14 state variables to be solved for implying a 14 degree of freedom system in the body frame of reference, including the deformation of the lander legs. These are specified in equation 3.1.

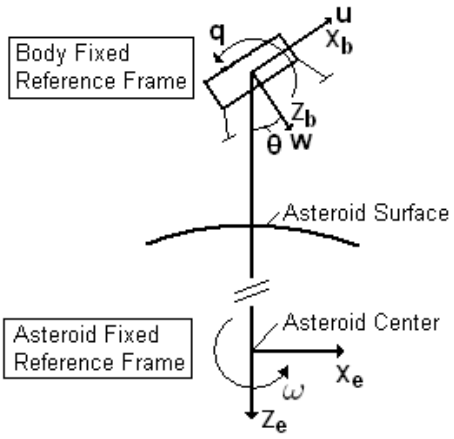

Figure 3.3: Model axes definitions

$$
\begin{array}{rcccccc}
\ddot{x}_{base} & = & \frac{Fx_{base}}{m_B} & - & qw_{base} & - & g\sin(\theta) \\
\ddot{z}_{base} & = & \frac{Fz_{base}}{m_B} & + & qu_{base} & + & g\cos(\theta) \\
\ddot{x}^L_{leg} & = & \frac{Fx^L_{leg}}{m_L} & - & qw^L_{leg} & - & g\sin(\theta) \\
\ddot{z}^L_{leg} & = & \frac{Fz^L_{leg}}{m_L} & + & qu^L_{leg} & + & g\cos(\theta) \\
\ddot{x}^R_{leg} & = & \frac{Fx^R_{leg}}{m_L} & - & qw^R_{leg} & - & g\sin(\theta) \\
\ddot{z}^R_{leg} & = & \frac{Fz^R_{leg}}{m_L} & + & qu^R_{leg} & + & g\cos(\theta) \\
\ddot{\theta} & = & \frac{M}{I} & & & &
\end{array}
\tag{3.1}
$$

where x_{base} and z_{base} are the translational components of the main lander structure in the x and z-axes respectively. x^L_{leg}, z^L_{leg}, x^R_{leg} and z^R_{leg} are the translational components in the x and z-axes of the left and right legs in the body frame of reference. q is the angular rate of the lander and θ is the lander attitude. u_{base} and w_{base} are the components of velocity in the x-axis and z-axis respectively for the main lander structure and u^L_{leg}, w^L_{leg}, u^R_{leg} and w^R_{leg} are the components of velocity in the x and z-axes for the left and right legs. M is the resultant moment acting on the lander body, I is the moment of inertia of the lander and g is the gravitational acceleration of the asteroid. Both I and g are assumed to be constant close to the asteroid surface during final descent and contact.

The leg forces are defined such that the force differs depending on the sense of the leg contraction or extension as described below:

$$Fz^L_{leg} = \begin{cases} k_s.((z_{base} - z^{rest}_{base}) - (z^L_{leg} - z^{Lrest}_{leg})), \ \dot{z}^L_{leg} > 0 \\ \\ k_s.((z_{base} - z^{rest}_{base}) - (z^L_{leg} - z^{Lrest}_{leg}))+ \\ k_r.((z_{base} - z^{rest}_{base}) - (z^L_{leg} - z^{Lrest}_{leg}) - (L_L - L_S)), \ \dot{z}^L_{leg} \leq 0 \end{cases} \tag{3.2}$$

$$Fz^R_{leg} = \begin{cases} k_s.((z_{base} - z^{rest}_{base}) - (z^R_{leg} - z^{Rrest}_{leg})), \ \dot{z}^R_{leg} > 0 \\ \\ k_s.((z_{base} - z^{rest}_{base}) - (z^R_{leg} - z^{Rrest}_{leg}))+ \\ k_r.((z_{base} - z^{rest}_{base}) - (z^R_{leg} - z^{Rrest}_{leg})) - (L_L - L_S)), \ \dot{z}^R_{leg} \leq 0 \end{cases} \tag{3.3}$$

where all the terms are in body axes. Terms defining a difference in the z locations of the base (z_{base}) and legs (z^L_{leg} and z^R_{leg}) from the base (z^{rest}_{base}) and leg (z^{Lrest}_{leg} and z^{Rrest}_{leg}) rest positions are multiplied by either the spring constant only in compression, or the spring and ratchet constant in extension to produce the leg forces that act on the system. During the extension phases of the simulation (i.e. $\dot{z}_{leg} \leq 0$) it is necessary to include a term that simulates the maximum deformation of the spring/ratchet system ($L_L - L_S$, as defined in table 3.1) so the extension forces only act when appropriate. The reason the ratchet force acts solely in one direction is to reduce the leg extension and absorb the kinetic energy from the decent of the lander when in contact with the asteroid surface.

The component of force in the z-axis, in body axes (Fz_{base}), acting on the lander base is the sum of the components of forces in the left (Fz^L_{leg}) and right (Fz^R_{leg}) legs in addition to the thruster forces (T):

$$Fz_{base} \quad = \quad Fz^L_{leg} \quad + \quad Fz^R_{leg} \quad + \quad T$$

The component of force in the x-axis, in body axes, acting on the lander base is equal to the component of force in the x-axis, in body axes, acting on the lander legs and is given by:

$$Fx_{base} \quad = \quad Fx_{leg}^{L} \quad + \quad Fx_{leg}^{R} \quad + \quad T$$

where $Fx_{leg}^{L} = Fx_{leg}^{R} = 0$ since the legs are perpendicular to the base.

One leg contact

The first dynamic state of the lander is free-flight. The second dynamic state is when there is only one leg in contact with the asteroid. This can be in a descent scenario or in the rebound ascent phase. In this system there are 10 state variables to be solved for and can be interpreted as equation 3.1 with two of the equations corresponding to one of the legs, which is in contact with the asteroid surface, removed. The remaining four variables that correspond to the displacement and velocity of the leg in contact with the asteroid surface do not change during this dynamic state. The resultant force on the base Fz_{base} is defined differently as:

$$Fz_{base} = \begin{cases} (-k_s.(z_{base} - z_{base}^{rest})) + Fz_{leg}^{free}, \ \dot{z}_{base} > 0 \\ \\ (-k_s.(z_{base} - z_{base}^{rest})) + (-k_r.(z_{base} - z_{base}^{rest})) + Fz_{leg}^{free}, \ \dot{z}_{base} \leq 0 \end{cases}$$

where Fz_{leg}^{free} is the force, in the body z-axis acting on the base from the leg which is not in contact with the asteroid surface. This force is defined as:

$$Fz_{leg}^{free} = \begin{cases} k_s.((z_{base} - z_{base}^{rest}) - (z_{leg}^{free} - z_{leg}^{FreeRest})), \ \dot{z}_{leg}^{free} > 0 \\ \\ k_s.((z_{base} - z_{base}^{rest}) - (z_{leg}^{free} - z_{leg}^{FreeRest})) \\ +(-k_r.((z_{base} - z_{base}^{rest}) - (z_{leg}^{free} - z_{leg}^{FreeRest}))) - (L_L - L_S)), \ \dot{z}_{leg}^{free} \leq 0 \end{cases}$$

where $z_{leg}^{FreeRest}$ is the body z-axis rest position of the leg not in contact with the asteroid surface. The same analysis applies as before regarding the leg compression and expansion in the case of the application of the ratchet. Here:

$$Fx_{base} \quad = \quad Fx_{leg}^{free} \quad + \quad T$$

where $Fx_{leg}^{free} = 0$ due to the legs being perpendicular to the base.

The moment acting on the center of gravity of the lander during the scenario where one leg is in contact with the surface is defined as:

$$M = \begin{cases} \frac{1}{2}(Fz_{leg}^{free} - Fz_{leg}^{contact})L_B + M_T, \ \theta > 0 \\ \\ -\frac{1}{2}(Fz_{leg}^{free} - Fz_{leg}^{contact})L_B + M_T, \ \theta < 0 \end{cases} \tag{3.4}$$

where $Fz_{leg}^{contact}$ is the force from the leg in contact with the surface ($Fz_{leg}^{contact} = Fz_{base} - Fz_{leg}^{free}$), and M_T is the moment produced by the thrusters.

Two leg contact

The third dynamic state is when both lander legs are in contact with the asteroid surface. In this system state there are 6 variables to be solved for, because the leg states now correspond to the location of the asteroid surface, defined in equation 3.5, so that:

$$\begin{array}{rcccccc} \ddot{x}_{base} & = & \frac{Fx_{base}}{m_{base}} & - & qw_{base} & - & g\sin(\theta) \\ \ddot{z}_{base} & = & \frac{Fz_{base}}{m_{base}} & + & qu_{base} & + & g\cos(\theta) \\ \ddot{\theta} & = & \frac{M}{I} & & & & \end{array} \tag{3.5}$$

where the forces obey the usual conventions for application of the ratchet force.

In all cases M, the moment applied around the center of gravity of the lander is defined by the difference in the forces in the legs and the distance between them in addition to the moment produced by the control thrusters:

$$M = \frac{1}{2}(Fz_{leg}^{L} - Fz_{leg}^{R})L_B + M_T \tag{3.6}$$

The points at which the model switches between the set of dynamic equations are defined heuristically (see figure 3.4, which demonstrates the flow of a descent). The descent or rebound scenarios terminate when either of the z-components of the legs in inertial axes is equal to the radius of the asteroid. The descent phase in which one leg has contact finishes when the z-component of the other leg in inertial axes equals the radius of the asteroid. The two leg contact phase ends when either; the rebound force in both legs is great enough to overcome the weight of the entire lander, inducing a rebound, or the lander finishes its transient response and comes to rest on the asteroid surface. The initial rebound (one leg still in contact with the asteroid) phase ends when the force acting on the leg still in contact with the asteroid is greater than the weight of one leg. In any simulation if the attitude of the lander (θ), is small ($\leq 0.5^0$) then the one leg descent and rebound phases are neglected and the model moves onto the next two leg phase.

The point where the lander may leave the surface of the asteroid and return to the free-flight state is defined by the difference between the lander weight and the resultant forces acting on the lander base from the two lander legs. This results in a rebound force acting on the lander equal to the sum of the two leg forces. If the sum of these two leg forces is less than the lander weight a bounce occurs, otherwise the lander remains in contact with the asteroid surface. Assuming momentum is conserved during contact with

the surface, the velocity component of the momentum after leg deformation is the initial velocity condition for the start of the simulation of any subsequent rebound.

The model continues in an iterative loop (see figure 3.4) until the lander is at rest on the surface of the asteroid, or alternatively falls over on its side due to instability. These processes are demonstrated in figure 3.5. In figure 3.5(a) a descent and one rebound can be seen starting from an initially vertical attitude, in figure 3.5(b) the effect of adding a 1^0 attitude error to the descent and rebound causes a subsequent tumbling of the lander. In both cases there is no attitude control. In figure 3.5 the blue line indicates the center of mass of the base and the red and green lines indicate the feet positions during the simulation. In both cases the asteroid surface is assumed to be solid.

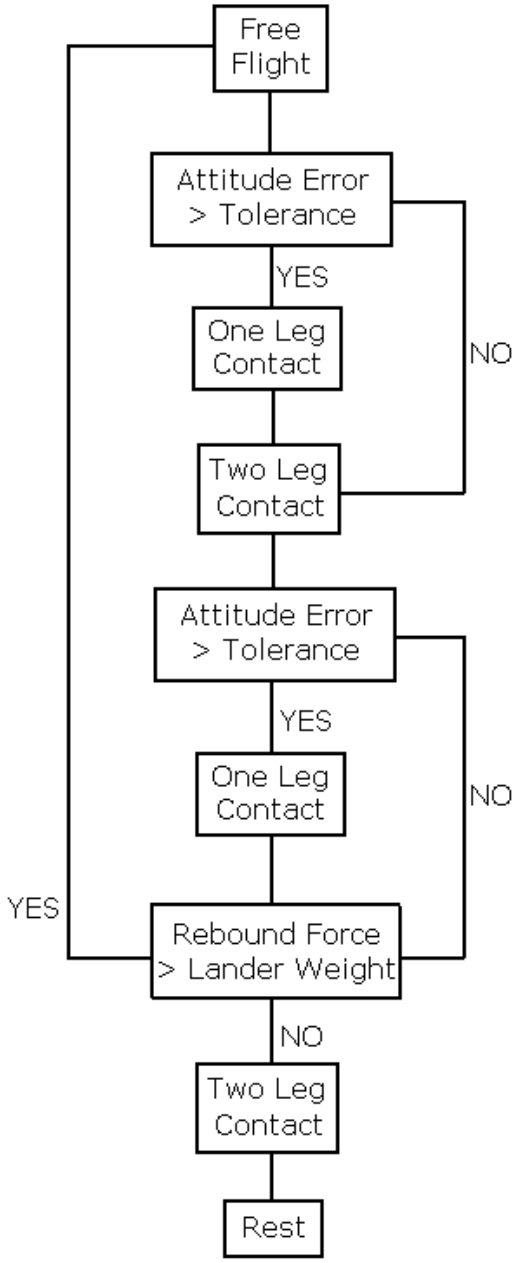

Figure 3.4: Lander descent flow

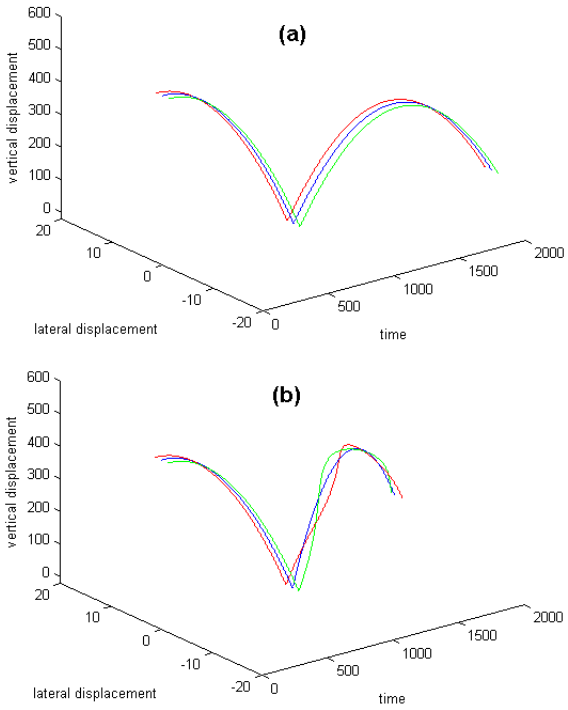

Figure 3.5: Example of descent and rebound (a) no attitude error (b) 1^0 attitude error

3.3.1 Asteroid Surface Modelling

The material that composes the surface of the asteroid is assumed to be largely unknown. Attempts at categorising material properties have been made, for example in the NASA Deep Impact mission [64]. Therefore, for the purposes of modelling the surface of the asteroid three different material types have been developed.

First, a basic solid non-deforming surface is used, see figure 3.6(a). The surface is assumed to be dense and any kinetic energy that the lander possesses after leg deformation is assumed to be conserved in the lander during rebound from the surface. Materials that have similar properties to this are rock and non-malleable metal.

The second surface type modelled is an elastic deforming material, such as viscoelastic porous materials, see figure 3.6(b). This surface is assumed to be less dense than the solid case but reforms to its original state after some transient time. In this case, in addition to leg deformation, some of the kinetic energy of the lander is absorbed by the asteroid surface. Therefore any subsequent rebound force on the lander will be less than the solid case. Modelling of the elastic surface is achieved by assuming the surface is viscoelastic and can be modelled by a Kelvin-Voigt material [65].

Finally, a plastically deforming surface is modelled, see figure 3.6(c). As a true plastic material is the most likely to be encountered on the surface of an asteroid, plastic deformation can be associated with the dynamics, and kinematics, of encountering packed

gravel or dust. For the deformation process the material obeys a linear stress-strain relationship and Hooke's law. When the legs in contact are then fully deformed any residual momentum is assumed to be conserved and produces the rebound dynamics. The plastic surface remains deformed after contact.

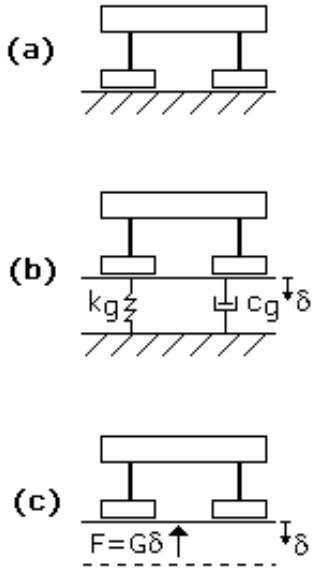

Figure 3.6: Examples of leg deformations (a) solid surface (b) elastic surface (c) plastic surface

Some material properties of the elastic and plastic surfaces must be defined and are given in table 3.2. The values for the spring and damper in the elastic material are chosen to represent a stiff porous material. Young's modulus, in the case of the plastic material, is chosen in the range where some deformation can occur [66].

Table 3.2: Asteroid surface properties for elastic and plastic materials

Parameter	**Value**	**Unit**
Elastic Spring Const (k_g)	2500	N/m
Elastic Damper Const (c_g)	2500	N/m^2
Young's Modulus (G)	8	kN/mm^2 (GPa)

A series of simulations using the different surfaces are now presented describing the changes in dynamics as a consequence of changing the asteroid surface type. Similar simulations to those defined by the figures 3.5 contain some interesting phenomena worthy of investigation when incorporating different surface types.

First, each of the trajectories of the lander making contact with the three surfaces is investigated. This is the same scenario as depicted in figure 3.5(a). As the interesting dynamics occur during the contact phase of the simulations, plots of the resultant trajec-

tories in the z-time plane during only the contact portion of simulation are given in figure 3.7.

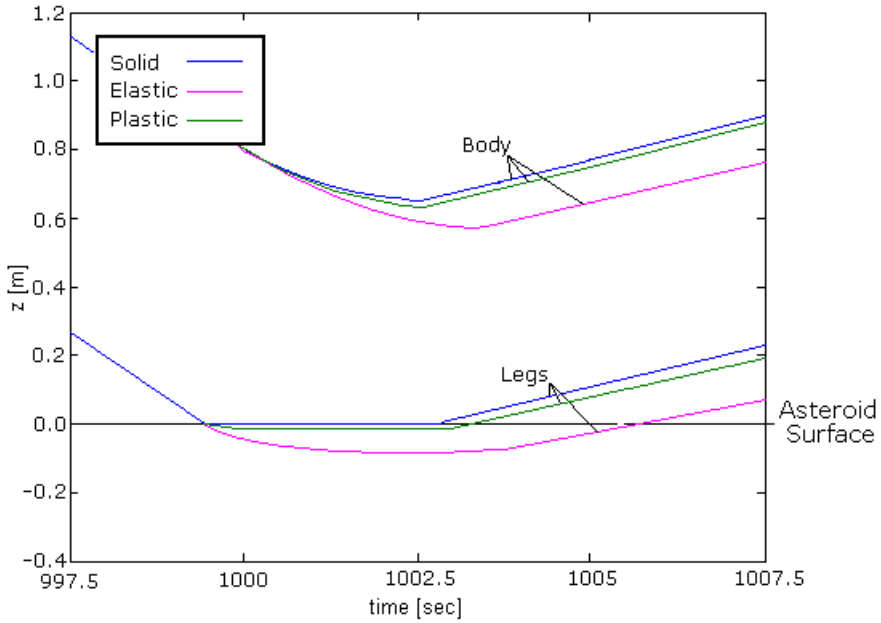

Figure 3.7: Lander body and leg trajectories during contact with different surface types

The differences in how the lander interacts with the three surfaces is evident in figure 3.7. For the solid surface, trajectories for the body and legs, figure 3.7, shows how the leg deformation starts when both legs make contact with the asteroid surface. Then, when leg deformation is complete, the lander rebounds due to all the momentum still being conserved in the system. For the plastic surface the dynamics are similar, although slightly more kinetic energy has been absorbed due to the slight surface deformation. The elastic case differs most as the surface has the greatest ability to deform. More kinetic energy is absorbed by the surface and the subsequent rebound is smaller.

The second of these phenomena is an investigation into the attitude change as a consequence of the legs contacting when in contact with the surface of the asteroid. A plot of the attitude change during the contact phases of the simulation is shown in figure 3.8. In the plot the initial attitude of 1^0 is evident and then the subsequent changes, for each of the surfaces, to uncontrolled spins can be seen. The lander, in each case, begins to rotate when contact with the surface with one leg is made. Rotation slows slightly when the second leg makes contact, but a constant rate of spin is maintained when the lander leaves the surface of the asteroid due to residual kinetic energy.

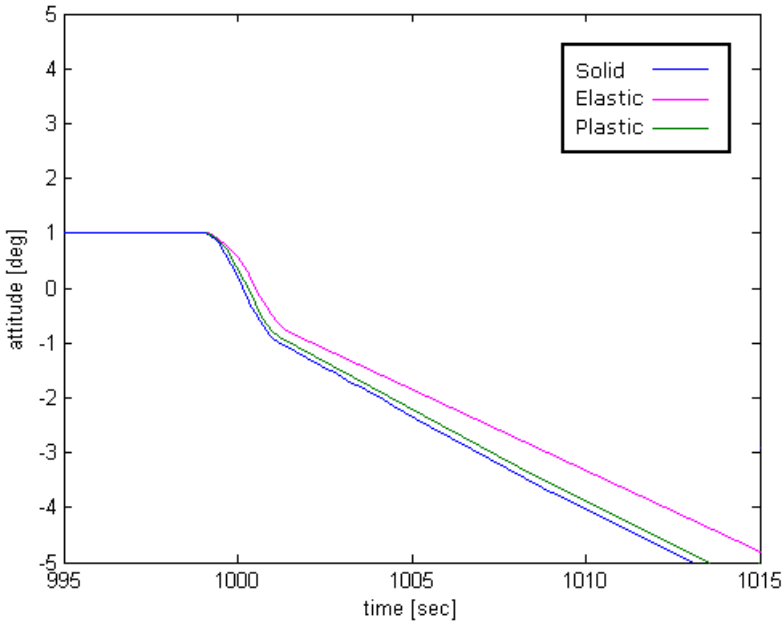

Figure 3.8: Attitude changes for three surfaces with 1^0 initial incidence

The contact dynamics with the solid surface shows the most extreme case of rotation. As the plastic surface material properties are close to a solid case, the change in attitude is close to that of the solid case. The rotation in the plastic case is slower however due to the deforming surface, but the final rate of spin is almost identical. In the elastic surface case more kinetic energy is absorbed over a longer period of time leading to a slower spin rate for the lander.

Another phenomenon is that of the lander leg deformations as a consequence of the spring and ratchet which compose the mechanics of the legs. Results for these simulations are shown in figure 3.9. Due to the initial attitude of the lander the left leg makes contact with the surface of the asteroid first and begins deforming. At the point where the right leg makes contact with the surface and begins deforming, the rate at which the left leg deforms begins to decrease until it reaches its fully deformed value of 0.7 m. In these particular examples the reaction forces in the legs of the lander become greater than the weight of the lander before the right legs fully deform, explaining why the right legs reach only semi-deformed states.

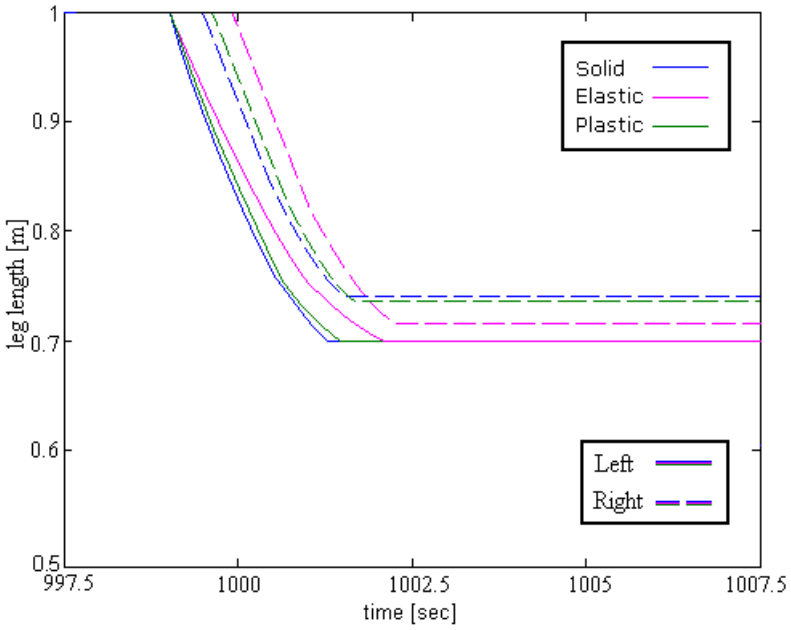

Figure 3.9: Examples of leg deformations

The different surfaces cause slight variations in the contact dynamics relative to the solid case. The plastic case is similar, only due to the plastic surface the leg deformations starts 0.01 s behind the solid case, and the right (partially deformed) leg deforms by somewhat more. The elastic case is more complex. As the elastic surface contact time is greatest, this means more time for leg deformation. This can be seen in figure 3.9, where the right leg deforms over a longer period of time and to a greater extent.

3.3.2 Reaction Control System (RCS)

In order to implement the control laws that will be defined, there are physical characteristics of the propulsion system that must be included in the lander model. The first of these results as a consequence of the nature of the desired rotational torque (or translational force) that is demanded from the control laws.

These demanded controller torques (and forces) take the form of continuously changing thrust magnitudes during the course of any simulation. In reality the torques (and forces) that can be produced are discrete in nature due to the mechanical processes involved in expelling reaction mass to either rotate or translate the lander. The process that derives a discrete thrust from the continuous input signal is pulse width modulation (PWM) [67].

The simplest way to generate a PWM signal is the intersective method [68], which requires only a sawtooth or a triangle waveform. When the value of the reference signal (the green sine wave in figure 3.10) is more than the modulation waveform (blue), the PWM signal (red) is in the high state, otherwise it is in the low state. An example of the applied PWM process is shown in figure 3.11. Figure 3.11 shows one of the simulations

that is conducted later and the PWM process can be seen. The triangular waveform in the simulations has an amplitude of 1 Nm, a frequency of 200 Hz, and a threshold of 0.5 Nm to avoid unwanted chattering. In figure 3.11 the triangular waveform is not shown for clarity but the reference (demanded) torque is depicted in red and the resultant modulated waveform in black.

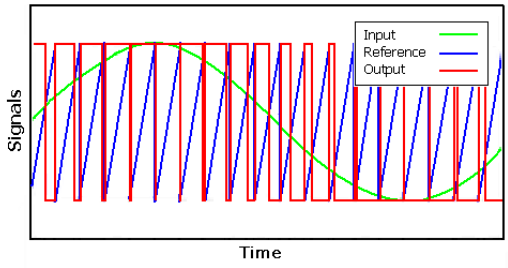

Figure 3.10: Pulse width modulation

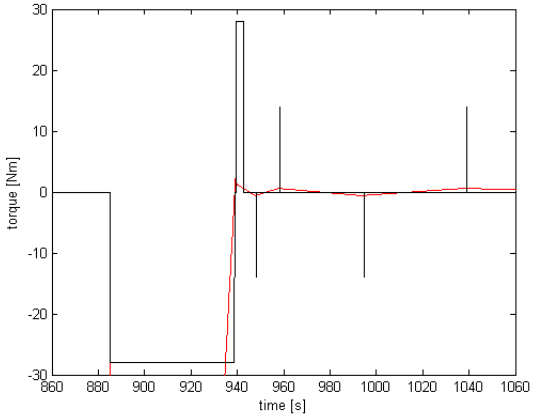

Figure 3.11: Simulation example of PWM

The second physical characterisitic of the propulsion system that must be included to represent a realistic system is the modelling of the thruster minimum impulse bit (MIB). The minimum impulse bit of a propulsion system is the minimum quantity of momentum a thruster can deliver. This is characterized by mechanical properties such as time to open and close the solenoid valves. Here the MIB is taken at a value of 0.05 Ns (corresponding to 10 N for 0.005 s) [33].

The last point to note is the relation between the MIB and the simulation integration time step. The simulation time step must be kept smaller than the MIB of the propulsion subsystem.

Thruster Mappings

There exists one final process before these discrete torques or forces from the RCS are implemented. This is the method through which the discrete moments and forces demanded

are distributed (or mapped) onto the thrusters that are available to produce these demands. The technique applied is the cost function approach [69]. The same type of cost function is applied to both controllers (developed later in chapter 4) in both control cases (attitude and lateral translation) individually.

As a primary goal of the work is to define qualitative and quantitative differences between the controller types a simple cost function is used in all simulations. The cost functions are defined as:

$$C_M = |M_{des} - M_{prod}|^2 \tag{3.7}$$

$$C_F = |F_{des} - F_{prod}|^2 \tag{3.8}$$

where C_M and C_F are the cost functions for the attitude and translational control respectively, M_{des} and F_{des} are the moment and forces demanded by the RCS respectively, and M_{prod} and F_{prod} are the moment and force produced respectively as a consequence of any combination of thrusters.

With reference to figure 3.1, this leads to a range of different thruster options, but the selection of which thrusters is simply the option that minimises the cost functions defined by equations 3.7 and 3.8. To make the process more efficient the translational and rotational cost functions are separated. This is due to the fact that there are only two translational thrusters (one for each sense of translation) and the required thruster selection is obvious. In the case of rotation the problem resolves to whether to apply one or two thrusters in the case of rotation in each sense, and the applied moment is that which minimises equation 3.7.

3.4 Descent in a Rotational Frame

In section 1.2.1 the mission concept for the Marco Polo mission was described. There is a need for the lander to descend onto the surface of the asteroid with a linear descent path so the final contact site can be defined accurately. In order to achieve this, due to the high rotation rate of the asteroid, a rotating frame of reference has been chosen for the descent phase. What follows is a description of the differences in inertial and rotational frames and the implications of implementing a rotational reference frame into the descent model.

Figure 3.12 compares an inertial descent with a rotational descent with the rotating frame of reference located on the asteroid. The asteroid rotates at a constant angular velocity represented by a vector $\mathbf{\Omega}$ with magnitude ω with the y-axis pointing outward according to the right-hand rule ($\mathbf{\Omega} = (0, \omega, 0)$). If the lander descends radially towards the surface of the asteroid at a constant speed, in a straight line path (initial descent position ϕ) in figure 3.12, then it is in a rotational frame of reference. In the inertial frame however the descent path a spiral.

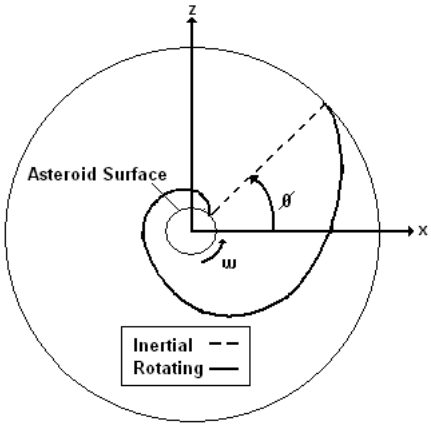

Figure 3.12: Frame of reference comparison

3.4.1 Defining an Inertial Frame

An inertial frame of reference describes the path followed by the lander as a spiral. Adopting the coordinate system defined in figure 3.12, the descent path is described by $\mathbf{r}(t)$:

$$\mathbf{r}(t) = R(t)\mathbf{u}_R = (x(t), z(t)) = (R(t)cos(\omega t + \phi), R(t)sin(\omega t + \phi)) \tag{3.9}$$

where $\mathbf{u}_R$ is a unit vector in the radial direction pointing from the center of the asteroid to the lander at time t. The radial distance $R(t)$ decreases uniformly with time according to:

$$R(t) = R_0 + ct \tag{3.10}$$

where c is the radial descent velocity ($c < 0$) and R_0 is the initial radial displacement. The velocity in the inertial frame is then the first derivative of the position:

$$\mathbf{v}(t) \quad = \tfrac{d}{dt}R(t)\mathbf{u}_R + \omega R(t)\mathbf{u}_\perp \tag{3.11}$$

where $\mathbf{u}_\perp$ is a unit vector that is always perpendicular to $\mathbf{u}_R$. Taking the derivative of the the velocity provides the acceleration term:

$$\mathbf{a}(t) = \quad 2c\omega\mathbf{u}_\perp - \omega^2 R(t)\mathbf{u}_R \tag{3.12}$$

The last term in equation 3.12 is radially inward with magnitude $\omega^2 R$, which is defined as the centripetal acceleration due to the circular motion. The first term in equation 3.12 is perpendicular to the radial direction and is defined as the Coriolis acceleration.

3.4.2 Defining the Rotational Frame

In a rotating frame of reference the lander travels in a straight line from the initial orbit to the asteroid surface, as shown by the dashed line in figure 3.12. The rotating observer sees the lander move at a constant speed so there is zero resultant force on the lander. The rotating observer has to introduce *fictitious* forces that appear to exist in the rotating frame.

The force acting on the lander for the rotating frame is given by:

$$\mathbf{F} = -2m\boldsymbol{\Omega} \times \mathbf{v}_B - m\boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{r}_B) \tag{3.13}$$

where the velocity observed in the rotating frame is:

$$\mathbf{v}_B = c\mathbf{u}_R, \tag{3.14}$$

where $\mathbf{u}_R$ again has the same definition.

The position of the lander seen during the descent in the rotating frame is:

$$\mathbf{r}_B = R(t)\mathbf{u}_R, \tag{3.15}$$

where it can be seen that:

$$\boldsymbol{\Omega} \times \mathbf{u}_R = \omega \mathbf{u}_\perp \tag{3.16}$$

and

$$\boldsymbol{\Omega} \times \mathbf{u}_\perp = -\omega \mathbf{u}_R \tag{3.17}$$

therefore:

$$\mathbf{F} = -2m\omega c\mathbf{u}_\perp + m\omega^2 R(t)\mathbf{u}_R \tag{3.18}$$

To obtain straight line motion in the rotating frame, a force exactly opposite in sign to the fictitious force must be applied to reduce the net force on the lander to zero. The fictitious forces that must be combated are the Coriolis force (first term) and the centrifugal force (second term). By applying thrust to counter these fictitious forces, the lander follows a linear descent path in the rotating frame.

3.4.3 Rotating Asteroid Model

As discussed above, moving to a rotating asteroid frame of reference introduces two new *fictitious* forces that must be implemented into the descent model. These new forces are the Coriolis force (equation 3.19) and the Centrifugal force (equation 3.20):

$$\mathbf{F}_{COR} = -2m\boldsymbol{\Omega} \times \mathbf{v} \tag{3.19}$$

$$\mathbf{F}_{CEN} = -m\boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{r}) \tag{3.20}$$

where $\boldsymbol{\Omega} = (0, \omega, 0)$, $\mathbf{v} = (u, 0, w)$ and $\mathbf{r} = (x, 0, z)$.

The process of introducing these new forces will result in the lander appearing to descend in a straight line towards the surface of the asteroid although in reality it will curve around the surface of the asteroid with angular velocity ω, as shown schematically in figure 3.13.

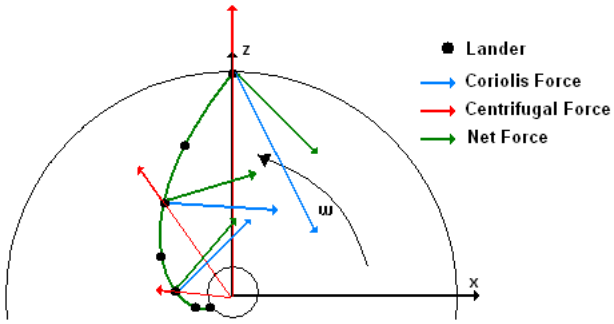

Figure 3.13: Rotational frame force breakdown

The *fictitious* forces are present in the model, but if these forces are not counteracted the lander will drift in a lateral sense. To demonstrate this a non-controlled descent simulation from 1000 m altitude is shown in figure 3.14 where figure (a) depicts the inertial frame descent (the circular arc is the surface of the asteroid) and (b) the rotational frame descent.

If the *fictitious* forces are not included in the model then translation drift will occur during descent onto the asteroid surface. A consequence of this is for linear descent to the landing site lateral control must be introduced to the model, as will be seen chapter 4.

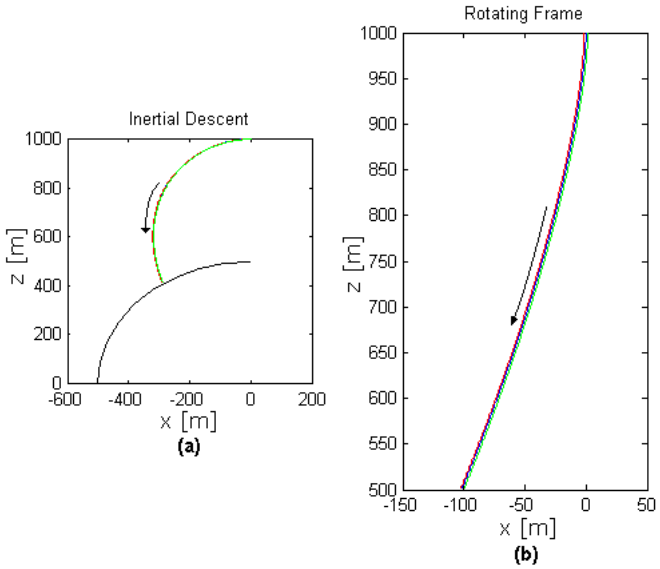

Figure 3.14: Reference frame comparison (a) inertial (b) rotational

Chapter 4

Embedded Autonomy for a Deep Space Mission

4.1 Asteroid Lander Controllers

When the asteroid lander makes contact with the asteroid surface it is required to remain there for a period of time while surface sampling operations are conducted. Reaching the surface of the asteroid with an attitude error leads to a subsequent unstable tumble due to the contact dynamics between the lander and the asteroid surface. It is therefore necessary to introduce autonomous control of the lander attitude to maintain this error to within an acceptable tolerance. Also, as described in section 3.4, autonomous lateral control is required to compensate for any lateral drift.

In this section the controllers used to autonomously control the asteroid lander are developed and presented. Conventional phase-plane and potential function controllers are defined and the direct adaptive control developed for embedded autonomy. The ability of each controller to recover from attitude errors is investigated and a quantitative comparison is presented. Subsequently a range of failures are induced into the lander system and the ability of each controller to recover from these failures is quantified and discussed. Also, a study detailing the ability of the adaptive control algorithm to control the translational effects of the sampling procedure is presented.

4.1.1 Phase-Plane Control

The conventional phase-plane (PP) controller that is implemented is an adaptation of a similar control algorithm used in applications such as the Apollo Lunar Module (LM) [70].

Figure 4.1 shows the logic for firing the control thrusters. The parabolas in the figure are the 'switch curves' that determine when the thrusters are turned on and off. To clarify this logic, a typical phase path is shown. The rate and the attitude are measured, and based on this measurement a set of thrusters are turned on. For the measurements shown at the initial position, the decision must be made to fire thrusters that provide a

negative acceleration. The logic for doing this is based on where the state is relative to the parabolas shown in the figure. As the thrusters fire, the trajectory in the phase plane is shown. When the phase path crosses the 'off-switch curve' (the lower parabola), the thrusters are turned off. Since the rate is negative at this point, the attitude drifts to the left at a constant rate, until the trajectory crosses the switch curve for turning the thrusters on with a positive acceleration.

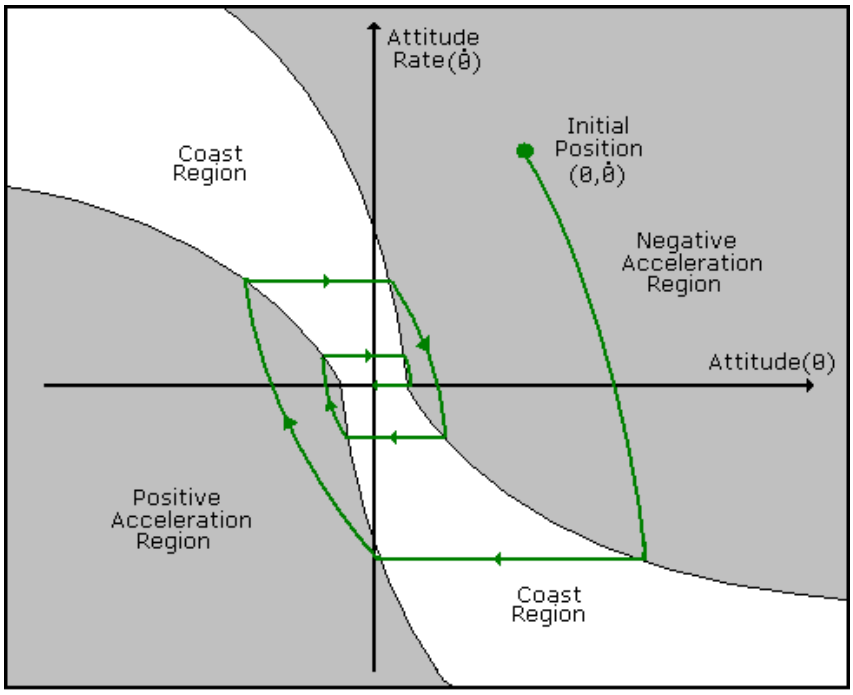

Figure 4.1: Phase plane switching logic

A relation is necessary to define the parabolas for the switching curves. Four such curves are needed, two for defining the points that the thrusters switch on and two that define the points that the thrusters switch off. The curve definitions are given below (equation 4.1 [70]) and they are denoted in figure 4.2.

$$\begin{aligned} f_n(\theta,\dot{\theta}) &= \theta + \frac{\dot{\theta}^2}{2.N_\theta.Accel_\theta} - DB_\theta = 0 \quad - Negative\ Acceleration \\ f_p(\theta,\dot{\theta}) &= \theta - \frac{\dot{\theta}^2}{2.N_\theta.Accel_\theta} + DB_\theta = 0 \quad - Positive\ Acceleration \\ c_n(\theta,\dot{\theta}) &= \theta + \frac{\dot{\theta}^2}{2.N_\theta.SwitchAccel_\theta} + DB_\theta = 0 \quad - Negative\ Coast \\ c_p(\theta,\dot{\theta}) &= \theta - \frac{\dot{\theta}^2}{2.N_\theta.SwitchAccel_\theta} - DB_\theta = 0 \quad - Positive\ Coast \end{aligned} \tag{4.1}$$

where θ and $\dot{\theta}$ are the attitude and attitude rates respectively, N_θ is the number of thrusters firing, $Accel_\theta$ is the acceleration produced from one thruster and $SwitchAccel_\theta$ is the acceleration at which the thrusters are turned on/off. DB_θ is the magnitude of the dead-band. These numerical values are shown in table 4.1.

Table 4.1: Phase plane for rotation switching curve parameters

Parameter	Value	Unit
Number of thrusters (N_θ)	2	
$Accel_\theta$	0.02	rad/s^2
$SwitchAccel_\theta$	0.035	rad/s^2
DB_θ	0.0025	rad

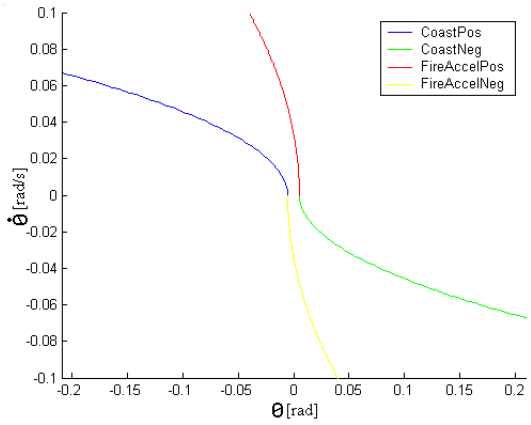

Figure 4.2: Phase plane for asteroid lander switching logic

The numerical values specified in equation 4.1 will determine the shape of the parabolas depicted in figure 4.2. The performance of the controller is therefore dominated by these parameters. Some can be selected by the user, such as the switching acceleration, but others will be predefined, such as the single thruster acceleration, which is a function of the thrust magnitude.

A phase plane controller for the lateral translational control of the lander is developed in the same fashion. The switching curves in this case are defined by equations 4.2. Here X_e and $\dot{X}_e$ are the lateral displacement error and error rate respectively. In the rotational frame of reference X_e is the lateral displacement from the vector that projects up perpendicular from the desired landing location below. N_{X_e} is the number of thrusters acting to counteract the lateral displacement error. $Accel_{X_e}$ is the acceleration produced from one thruster and $SwitchAccel_{X_e}$ is the acceleration at which the thrusters are turned on/off. DB_{X_e} is the magnitude of the lateral displacement dead-band. These numerical values are shown in table 4.2.

$$\begin{aligned}
f_n(\theta,\dot{\theta}) &= X_e + \frac{\dot{X_e}^2}{2.N_{X_e}.Accel_{X_e}} - DB_{X_e} = 0 \quad - Negative\ Acceleration \\
f_p(\theta,\dot{\theta}) &= X_e - \frac{\dot{X_e}^2}{2.N_{X_e}.Accel_{X_e}} + DB_{X_e} = 0 \quad - Positive\ Acceleration \\
c_n(\theta,\dot{\theta}) &= X_e + \frac{\dot{X_e}^2}{2.N_{X_e}.SwitchAccel_{X_e}} + DB_{X_e} = 0 \quad - Negative\ Coast \\
c_p(\theta,\dot{\theta}) &= X_e - \frac{\dot{X_e}^2}{2.N_{X_e}.SwitchAccel_{X_e}} - DB_{X_e} = 0 \quad - Positive\ Coast
\end{aligned} \tag{4.2}$$

Table 4.2: Phase plane for translational switching curve parameters

Parameter	Value	Unit
Number of thrusters (N_{X_e})	1	
$Accel_{X_e}$	0.01	m/s^2
$SwitchAccel_{X_e}$	0.035	m/s^2
DB_{X_e}	0.01	m

4.1.2 Potential Function Control

The potential function controller uses the attitude and attitude rate of the lander to define an artificial potential function for the lander. The aim is then to minimise this potential. This is achieved using Lyapunov's second theorem.

The aim of Lyapunov's second method is to guarantee the stability of a set of differential equations which describe a dynamical system. In physical terms, this has been described by Kalman and Bertram [71] [72]:

"If the rate of dE(x)/dt of the energy E(x) of an isolated physical system is negative for every possible state x, except for a single equilibrium state x_e, then the energy will continue to decrease until it finally assumes its minimum value E(x_e)."

This intuitively corresponds to the definition of all stability problems. If a stable system is perturbed from its equilibrium state, it will always return to it. In mathematical terms this can be expressed as:

"A dynamical system is stable (in the sense that it returns to equilibrium after any perturbation) if and only if there exists a Lyapunov function, a scalar function V(x) of the state with the following properties":

$$\begin{aligned} V(\mathbf{x}) > 0 \quad and \quad \dot{V}(\mathbf{x}) < 0 \qquad for \quad \mathbf{x} \neq \mathbf{x}_e \\ V(\mathbf{x}) = 0 \quad and \quad \dot{V}(\mathbf{x}) = 0 \qquad for \quad \mathbf{x} = \mathbf{x}_e \end{aligned} \tag{4.3}$$

where $\mathbf{x}_e$ is the equilibrium state of the system. If these conditions are satisfied, it is possible to guarantee that the origin of the state-space is a point of asymptotic global attraction and that all the trajectories inside the space, regardless of initial conditions, converge to this point.

An extended form of the Lyapunov function is the potential function. This function can be defined analytically and will be used to force the state vector of the dynamical system to converge to the desired goal. The mechanism that drives the convergence is based upon the rate of change of the potential function. If the rate of change of the potential $\dot{V}(\mathbf{x})$ is negative definite the state vector will converge to the goal point, which is the global minimum of the potential function. If $\dot{V}(\mathbf{x})$ is positive however, the state

vector will diverge from the goal point. In this case, to render $\dot{V}(\mathbf{x})$ once again negative, control is required. It is therefore possible to derive a control methodology which forces the convergence to a desired goal of the dynamical system. Defining a potential function based on some state vector $\mathbf{x}$ so that:

$$V = f(\mathbf{x}) \tag{4.4}$$

and differentiating with respect to time gives:

$$\dot{V} = \nabla f.\dot{\mathbf{x}} \tag{4.5}$$

where

$$\nabla f = \frac{\partial f}{\partial \mathbf{x}} \tag{4.6}$$

Therefore, if $\dot{\mathbf{x}}$ is a function of the control variable, by analytically determining the potential derivative $\dot{V}(\mathbf{x})$, it is possible to calculate the control inputs, which are required to render $\dot{V}(\mathbf{x})$ negative, and so ensure the convergence of the dynamical system to the desired goal point.

The problem at hand is to bring the lander to rest at some desired attitude. The terms which must be controlled are therefore the Euler angle θ, and the body rate $\dot{\theta}$. The potential function, as defined in Radice [5], will be:

$$V = V_{EULER} + V_{BODYRATE} \tag{4.7}$$

The component of the potential function due to the Euler angles will take the form of a power law function:

$$V_{EULER} = \frac{1}{2}(\theta - \theta_{Desired})^n, n = 2, 4, 6, \ldots \tag{4.8}$$

where $\theta_{Desired}$ is the goal attitude. The potential function component due to the body rates will have a simpler form with the goal corresponding to null body rates:

$$V_{BODYRATE} = \frac{1}{2}\omega^m, m = 2, 4, 6, \ldots \tag{4.9}$$

where ω ($= [0\ q\ 0]^T$, in this case) is the angular rate of the lander.

The total potential is the sum of the Euler and body rate components and will therefore take the form:

$$V = \frac{1}{2}(\theta - \theta_{Desired})^n + \frac{1}{2}\omega^m \tag{4.10}$$

To satisfy Lyapunov's theorem, the rate of change of the potential $\dot{V}$ must be rendered negative definite. Therefore, differentiating the potential leads to:

$$\begin{aligned} \dot{V} &= \tfrac{n}{2}(\theta - \theta_{Desired})^{n-1}\dot{\theta} + \tfrac{m}{2}\dot{\theta}^{m-1}\ddot{\theta} \\ &= \tfrac{n}{2}(\theta - \theta_{Desired})^{n-1}\dot{\theta} + \tfrac{mM}{2I}\dot{\theta}^{m-1} \end{aligned} \tag{4.11}$$

which will be used to generate the control law. Here, $\ddot{\theta} = M/I$ is selected to maintain the linearity of the system to be controlled. The assumption is that the cross moments of inertia are negligible with respect to the principle moment of inertia. A possible control torque M which will render $\dot{V}$ negative definite is expressed by:

$$M = -\frac{\dot{\theta}.I}{m\dot{\theta}^{m-1}}[2\lambda_1\dot{\theta} + n.(\theta - \theta_{Desired})^{n-1}] \tag{4.12}$$

where λ_1 is a positive definite shaping parameter for arbitrary m and n. When the control torque is substituted into equation 4.11 the potential derivative then takes the form:

$$\dot{V} = -\lambda_1\dot{\theta}^2 \tag{4.13}$$

so that the control law that rotates the lander to the desired goal attitude is available in analytical form. This solution to the control problem equates to a proportional derivative (PD) controller for m=n=2.

The above example for the creation of a potential function control law for rotational stability of the lander is repeated again for the case of the lateral displacement error of the lander during descent. Following the same procedure and replacing the rotational terms for the translational error terms lead to the following control law:

$$T_{lat} = -\frac{\dot{X}_e.(m_B + 2m_L)}{m\dot{X}_e^{\,m-1}}.[2\lambda_2\dot{X}_e + n.(X_e)^{n-1}] \tag{4.14}$$

where T_{lat} is the lateral thruster force required, m_B and m_L are the masses of the lander main body and leg respectively. All the other parameters have their usual meanings and λ_2 is a second positive shaping parameter for the case of lateral control.

The continuous demanded translational forces and rotational torques are then delivered by the reaction control system (RCS) using pulse width modulation as discussed in section 3.2.1 to produce a discrete thrust.

4.1.3 Direct Adaptive Control

Robust controllers try to desensitise a control system to the uncertainties of the plant. An adaptive controller aims to allow controller gains to vary continuously allowing larger levels of uncertainty and therefore increasing system performance during control implementation, as discussed in section 2.6.

Here, a direct adaptive control law is presented that provides robust attitude control for the asteroid lander. It is shown that the control law is a function of known system parameters and does not require any model dependant parameters. The control problem under

investigation has the potential for significant levels of uncertainty to occur, particularly in the unknown characteristics of the asteroid surface upon which the lander contacts, as discussed in section 3.3.1. With the embedded, bottom-up approach to autonomy that the direct adaptive controller introduces, the response of the controller to any fault or failure is fast as no other FDIR algorithm must be implemented. This is particularly important for the lander problem where the delay due to a traditional top down FDIR approach may be catastrophic due to the fast time scales associated with the contact dynamics.

Adaptive Controller Development

The direct adaptive control algorithm used in the lander problem is now developed. A definition of the general control problem and the associated general solution are shown, along with the proof for the stability of the controller. The linear system to be controlled is:

$$\dot{\mathbf{x}}(t) = \mathbf{A}.\mathbf{x}(t) + \mathbf{B}.\mathbf{u}(t) + \mathbf{d} \tag{4.15}$$

where $\mathbf{x}(t)$ is the system state vector, $\mathbf{u}(t)$ is the control vector, $\mathbf{d}$ is the disturbance, and $\mathbf{A}$ and $\mathbf{B}$ are matrices with the appropriate dimensions.

A controller is required to force the state error to zero. Also, the controller is required to deal with any uncertainty it might encounter. Although the system is linear, the nature of the disturbance can be nonlinear and this must be controlled in an acceptable manner.

The controller presented here uses full-state feedback stabilisation subject to unknown disturbances.

Theorem[73]: "*Assume there exists* $K_s \in \Re^{n_x}$ *such that* $\mathbf{A}_s = \mathbf{A}+\mathbf{B}.\mathbf{K}_s$ *is asymptotically stable. Let* $\mathbf{Q} \in \Re^{n_x \times n_x}$ *be positive semidefinite and assume* $(\mathbf{A}_s, \mathbf{Q})$ *is controllable. Let* $\mathbf{P} \in \Re^{n_x \times n_x}$ *be the positive-definite solution to the Lyapunov equation* $0 = \mathbf{A}_s^T\mathbf{P}+\mathbf{P}\mathbf{A}_s+\mathbf{Q}$. *Finally, let* $\mathbf{\Gamma} \in \Re^{n_u \times n_u}$ *and* $\mathbf{\Lambda} \in \Re^{n_x \times n_x}$ *be positive definite, and let* $\lambda > 0$. *Then the control law*

$$\mathbf{u}(t) = \mathbf{K}(t)\mathbf{x}(t) + \mathbf{L}(t), \tag{4.16}$$

where

$$\dot{\mathbf{K}}(t) = -\mathbf{\Gamma}\mathbf{B}^T\mathbf{P}\mathbf{x}(t)\mathbf{x}^T\Lambda, \tag{4.17}$$

$$\dot{\mathbf{L}}(t) = -\lambda\mathbf{B}^T\mathbf{P}\mathbf{x}(t) \tag{4.18}$$

yields $\mathbf{Q}\mathbf{x}(t) \to 0$ *as* $t \to \infty$ *where* $\mathbf{K} \in \Re^{n_x}$".

The stability of the above controller is proven by considering the closed loop system defined by applying the above theorem:

$$\dot{\mathbf{x}}(t) = (\mathbf{A}_s + \mathbf{B}(\mathbf{K}(t) - \mathbf{K}_s))\mathbf{x}(t) + \mathbf{B}\mathbf{L}(t) + \mathbf{d} \tag{4.19}$$

where all the vectors and matrices have the same definitions as above and it can be shown that due to the structures of $\mathbf{B}$ and $\mathbf{d}$ that $\mathbf{B}\mathbf{L}_s = \mathbf{d}$ and this represents the unknown disturbance in the system [73]. To analyse the controller define:

$$\hat{\mathbf{K}}(t) = \mathbf{K}(t) - \mathbf{K}_s \tag{4.20}$$

and

$$\hat{\mathbf{L}}(t) = \mathbf{L}(t) + \mathbf{L}_s \tag{4.21}$$

giving:

$$\dot{\mathbf{x}}(t) = (\mathbf{A}_s + \mathbf{B}\hat{\mathbf{K}}(t))\mathbf{x}(t) + \mathbf{B}\hat{\mathbf{L}}(t) \tag{4.22}$$

As discussed in section 2.4.4, a positive definite Lyapunov candidate equation must now be selected that includes all of the dynamic states of the system, i.e. $\mathbf{x}$, $\mathbf{K}$ and $\mathbf{L}$. This can be provided by:

$$V(\mathbf{x}, \hat{\mathbf{K}}, \hat{\mathbf{L}}) = \mathbf{x}^T\mathbf{P}\mathbf{x} + tr(\mathbf{\Gamma}^{-1}\hat{\mathbf{K}}\mathbf{\Lambda}^{-1}\hat{\mathbf{K}}^T) + (1/\lambda)tr(\hat{\mathbf{L}}\hat{\mathbf{L}}^T) \tag{4.23}$$

where 'tr' is the trace of a matrix. The function is required to decrease so the derivative must be negative.

$$\dot{V}(\mathbf{x}, \hat{\mathbf{K}}, \hat{\mathbf{L}}) = \dot{\mathbf{x}}^T\mathbf{P}\mathbf{x} + \mathbf{x}^T\mathbf{P}\dot{\mathbf{x}} + 2tr(\mathbf{\Gamma}^{-1}\hat{\mathbf{K}}\mathbf{\Lambda}^{-1}\dot{\hat{\mathbf{K}}}^T) + (2/\lambda)tr(\hat{\mathbf{L}}\dot{\hat{\mathbf{L}}}^T) \tag{4.24}$$

Substituting the closed loop system equation 4.19 gives:

$$\dot{V}(\mathbf{x}, \hat{\mathbf{K}}, \hat{\mathbf{L}}) = \dot{\mathbf{x}}^T(\mathbf{A}_s^T\mathbf{P}+\mathbf{P}\mathbf{A}_s)\mathbf{x}+2\mathbf{x}^T\mathbf{P}\mathbf{B}\hat{\mathbf{K}}\mathbf{x}+2\mathbf{x}^T\mathbf{P}\mathbf{B}\hat{\mathbf{L}}+2tr(\mathbf{\Gamma}^{-1}\hat{\mathbf{K}}\mathbf{\Lambda}^{-1}\dot{\hat{\mathbf{K}}}^T)+(2/\lambda)tr(\hat{\mathbf{L}}\dot{\hat{\mathbf{L}}}^T) \tag{4.25}$$

Now as $\mathbf{A}_s^T\mathbf{P} + \mathbf{P}\mathbf{A}_s + \mathbf{Q} = 0$, the equation can be rearranged to:

$$\dot{V}(\mathbf{x}, \hat{\mathbf{K}}, \hat{\mathbf{L}}) = -\dot{\mathbf{x}}^T\mathbf{Q}\mathbf{x} + 2tr\hat{\mathbf{K}}(\mathbf{\Lambda^{-1}}\dot{\hat{\mathbf{K}}}^{\mathbf{T}}\mathbf{\Gamma}^{-1} + \mathbf{x}\mathbf{x}^T\mathbf{P}\mathbf{B}) + (2/\lambda)tr\hat{\mathbf{L}}(\dot{\hat{\mathbf{L}}}^T + \mathbf{x}^T\mathbf{P}\mathbf{B}) \tag{4.26}$$

It follows that by inclusion of the two control law equations 4.17 and 4.18 for the variable control law gains that the last two terms of equation 4.26 are equivalent to zero. Therefore, as a consequence of the closed loop system being bounded, as discussed in section 2.4.4, and $\mathbf{Q}$ being positive semi-definite, for every initial condition $(\mathbf{x}(0), \hat{\mathbf{K}}(0), \hat{\mathbf{L}}(0))$, $\mathbf{Q}\mathbf{x}(t) \to 0$ as $t \to \infty$ [63].

For implementation of the control law governed by equations 4.17 and 4.18 a solution

for $\mathbf{P}$ in the Lyapunov equation $\mathbf{A}_s\mathbf{P} + \mathbf{P}\mathbf{A}_s + \mathbf{Q} = 0$ is necessary. Since $\mathbf{A}_s = \mathbf{A} + \mathbf{B}\mathbf{K}_s$ or $\mathbf{K}_s = \mathbf{B}^{-1}(\mathbf{A}_s - \mathbf{A})$ and $\mathbf{B} \neq 0$, $\mathbf{A}_s$ can be defined arbitrarily without knowledge of $\mathbf{A}$ or $\mathbf{B}$.

To develop the solution for a system with two state variables ($n_x = 2$), then $\mathbf{A} \in \Re^{2\times 2}$ and all other matrices and vectors in the system given, by equation 4.15, have the corresponding dimensions. Define $p > 0$, $a_{s1} < 0$, $a_{s2} < -p$ and [73]:

$$\mathbf{Q} = \begin{bmatrix} -2pa_{s1} & 0 \\ 0 & -2p - 2a_{s2} \end{bmatrix}$$

Then,

$$\mathbf{P} = \begin{bmatrix} -pa_{s2} - a_{s1} & p \\ p & 1 \end{bmatrix}$$

satisfies $\mathbf{A}_s\mathbf{P} + \mathbf{P}\mathbf{A}_s + \mathbf{Q} = 0$ and equation 4.17 and equation 4.18 can be given in a general form by:

$$\dot{K}_1(t) = -[\lambda_1 p x_1^2(t) + (\lambda_1 + \lambda_{12}p)x_1(t)x_2(t) + \lambda_{12}x_2^2(t)] \tag{4.27}$$

$$\dot{K}_2(t) = -[\lambda_{12} p x_1^2(t) + (\lambda_{12} + \lambda_2 p)x_1(t)x_2(t) + \lambda_2 x_2^2(t)] \tag{4.28}$$

$$\dot{L}(t) = -\lambda[px_1(t) + x_2(t)] \tag{4.29}$$

where [73]

$$\begin{bmatrix} \lambda_1 & \lambda_{12} \\ \lambda_{21} & \lambda_2 \end{bmatrix} = \mathbf{\Gamma}^{-1}\mathbf{\Lambda}$$

is positive definite because $\mathbf{\Gamma}$ and $\mathbf{\Lambda}$ are both positive definite, and $\lambda > 0$ (as defined in the above theorem). Using the control system defined by equations 4.27 - 4.29, then for any value of $p, \lambda_1, \lambda_2, \lambda_{12}, \lambda_{21} > 0$ and for any value of λ_{12} and λ_{21} such that $\lambda_{12}\lambda_{21} < \lambda_1\lambda_2$ leads to the solution $\mathbf{x}(t) \to 0$ as $t \to \infty$.

A simplified scenario where $\lambda_{12} = \lambda_{21} = 0$ leads to the solution of the control problem as:

$$\dot{K}_1(t) = -\lambda_1[px_1^2(t) + x_1(t)x_2(t)] \tag{4.30}$$

$$\dot{K}_2(t) = -\lambda_2[x_2^2(t) + px_1(t)x_2(t)] \tag{4.31}$$

$$\dot{L}(t) = -\lambda[px_1(t) + x_2(t)] \tag{4.32}$$

Direct Adaptive Control for the Lander Attitude

The methodology defined above can now be applied to the lander attitude control problem. Consider the lander attitude dynamics modelled by:

$$\ddot{\theta}(t) = M/I \tag{4.33}$$

where M is the sum of all the torques acting on the body, including the control torque $u(t)$ and the disturbing torque d, which is defined by the moment resultant on the lander as a result of a difference in the force acting on the base from the individual legs, as defined in section 3.2. This disturbance is unknown due to the highly uncertain physical properties of the asteroid surface. I is the moment of inertia of the lander about the axis of rotation, and so:

$$M = u + d \tag{4.34}$$

The control objective is to require θ to approach θ_{des} without knowledge of d, where normally $\theta_{des} = 0$. Defining the error signal $x_1(t) = \theta(t) - \theta_{des}$ and the state $\mathbf{x} = \begin{bmatrix} x_1 & \dot{x}_1 \end{bmatrix}^T$, equation 4.33 becomes:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 & 0 \\ I^{-1} & 0 \end{bmatrix} \mathbf{u}(t) + \begin{bmatrix} 0 \\ d \end{bmatrix} \tag{4.35}$$

where $\mathbf{u}(t) = [0\ u(t)]^T$.

Now solving Lyapunov's equation using the general solutions to $\mathbf{Q}$ and $\mathbf{P}$ maintaining $\mathbf{A}_s$ asymmetrically stable gives the variable control gains to the problem as (with comparison to equations 4.30 - 4.32):

$$\dot{K_1}(t) = -(1.5\theta^2 + 2\theta\dot{\theta}) \tag{4.36}$$

$$\dot{K_2}(t) = -(1.5\theta\dot{\theta} + 2\dot{\theta}^2) \tag{4.37}$$

$$\dot{L}(t) = -(7.5\theta + 10\dot{\theta}) \tag{4.38}$$

where the control parameters are: $p = 0.75$, $\lambda = 10$, and $\lambda_1 = \lambda_2 = 2$. These values were selected based on an iterative trial and error approach that lead to a satisfactory controller response, while maintaining the stability properties outlined previously.

By introducing different control parameters, consistent with the stability properties outlined in the previous section, to equation 4.38 a more efficient controller can be developed:

$$\dot{L}(t) = -(0.1\theta + 10\dot{\theta}) \tag{4.39}$$

This in turn leads to the required control moment as:

$$u(t) = (K_1\theta + K_2\dot{\theta} + L)I \tag{4.40}$$

The advantage of using this control algorithm is due to the fact it can react in real time to any disturbing moment, in this case d. Due to the unknown composition of the asteroid, the contact dynamics are unknown and the adaptive controller is well placed to deal with these uncertainties. Also included is an embedded form of autonomy which is quite different from traditional top-down FDIR.

Direct Adaptive Control for Lander Lateral Movement

The lander will be required to contact at a specified point on the surface of the asteroid, as discussed in section 1.2.1. A rotational reference frame has been chosen for kinematic modelling of the descent to ensure the lander remains in the correct lateral position during descent, as discussed in chapter 3. This introduces an extra requirement on the lander for lateral control during descent.

Introducing a DAC control law for lateral displacement using the same principles as for the rotational control ensures that the lander will be in the correct position as it approaches the asteroid surface. Drawing comparison with equation 4.33 and equation 4.35 produces:

$$\ddot{x}_1(t) = T_{lat}/(m_B + 2.m_L) \tag{4.41}$$

and

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 & 0 \\ \frac{1}{m_B+2.m_L} & 0 \end{bmatrix} \mathbf{u}(t) + \begin{bmatrix} 0 \\ d \end{bmatrix} \tag{4.42}$$

where, $\mathbf{x} = [x_1\ \dot{x}_1]$, T_{lat} is the lateral force produced by the thrusters, therefore $\mathbf{u}(t) = [0\ T_{lat}]^T$, $m_B + 2.m_L$ is the total spacecraft mass, $x_1(t) = x - x_{des}$ is the error signal and is the lateral displacement x from the desired landing location x_{des}. Following the earlier analysis a similar control algorithm is produced:

$$\dot{K_1}(t) = -(1.5x_1^2 + 2x_1\dot{x_1}) \tag{4.43}$$

$$\dot{K_2}(t) = -(1.5x_1\dot{x_1} + 2\dot{x_1}^2) \tag{4.44}$$

$$\dot{L}(t) = -(0.1x_1 + 10\dot{x_1}) \tag{4.45}$$

subsequently leading to the required lateral control thrust:

$$T_{lat} = (K_1 x_1 + K_2 \dot{x_1} + L).(m_B + 2m_L) \tag{4.46}$$

With regards the lateral control the disturbance d in equation 4.42 is now comprised of the Coriolis and centrifugal forces defined in equation 3.19 and equation 3.20 in addition to any translational forces that may be induced by the control thrusters minimising the attitude error.

4.2 Controlled Lander Descent

Using the lander model and the three controllers a comparative simulation can be undertaken to asses the performance of the controllers. The first step is to define a performance metric that can be used as a basis for comparison. The performance metric selected is the total impulse used by the thrusters to bring the lander to rest on the surface of the asteroid. Total impulse is defined as:

$$I_T = \sum_{i=1}^{n} \Delta t.T \tag{4.47}$$

where Δt is the integration step, T is the total thrust magnitude at the integration interval from all thrusters and n is the number of integration steps. It is now possible to develop a comparison between the three controllers.

The descent takes place in a rotational frame of reference as depicted in figure 3.3 and the equations of motion will be those defined by equation 3.1. A simple terminal bounce alleviation algorithm is defined: if the attitude of the lander is within a small error limit ($-0.5^0 \leq \theta_{des} \leq 0.5^0$) and the vertical velocity of the lander is negative (i.e. ascending) then two thrusters are fired in a positive vertical sense (in body axes) to reduce any residual bounces.

Before a direct comparison can be made, nominal values must be selected for the controller constants in each controller. For the potential function controller this is a single shaping parameter, λ, and the thruster threshold (0.1 N). The thruster threshold defines the lower limit of demanded thrust before the thruster is switched on. This eliminates any unwanted chattering by the thrusters. For the phase plane controller there are two parameters: $SwitchAccel_\theta$ and the dead band described earlier.

For the purpose of this comparison, for each lander descent scenario, the control parameters have been selected using a heuristic trial and error procedure. Therefore, an appropriate value for the total impulse in each case has been established.

There also exists a number of other parameters. These are specified below in table 4.3 and are common to all the lander descent simulations. This data was supplied by EADS Astrium Ltd for the Marco-Polo mission [12].

The scenarios selected for the comparison each have a different starting attitude. In

Table 4.3: Scenario parameters

Mass of Base (kg)	985.4
Mass of Leg (kg)	7.3
Inertia of Lander (kg.m^2)	677
Gravitational Accel (m/s^2)	0.000145
Asteroid Rotational Rate (^{0}/s)	0.013
Initial Inertial Position ([m,m])	[0,1000]
θ_{des} (^{0})	0
Minimum Impulse Bit (Ns)	0.05
Leg Stroke Length (m)	0.3
Asteroid Radius (m)	500

each case the lander starts at [0,1000] m as specified in table 4.3, corresponding to an initial altitude of 500 m. The lander initial attitude is then specified as 10^0, 20^0 or 30^0 and is permitted to descend towards the asteroid. In each case the lander must correct the initial attitude error and the translational disturbance forces due to the rotation of the asteroid.

Selection of the control parameters for each controller in each case gives the values specified in table 4.4. In the case of the PF and DAC controllers a thruster threshold is included to reduce chattering and is assigned to be 0.1 N. In the case of the PP controller the dead band is selected to be 0.0025 radians. An optimal solution for the three initial states of the DAC controller were found to be: $\kappa_1(0) = 0.0, \kappa_2(0) = -2.75$ and $L(0) = 0.0$, for all three initial attitude states demonstrating the robustness of DAC.

Table 4.4: Controller optimised parameters

Initial Attitude (^{0})	**PF** λ	**PP** $SwitchAccel_\theta$
10	1.8	0.035
20	2.35	0.035
30	2.8	0.035

The values for total impulse calculated in the scenarios described above are given in table 4.5. The interesting result that appears is when the potential function controller corrects the 30 deg initial attitude error more efficiently that the DAC algorithm. This result is explained by analysis of the transient correction time necessary in the manoeuvre, given in table 4.6. It follows that the DAC controller, in the 30 deg case, takes slightly longer than the actual descent time of the lander to correct the error. Control is hence carried through the contact phase and into the rebound resulting in a considerable increase in impulse necessary.

A different result to that of the rotational control is found in the joint rotational and translational control. Here, the thrusters are used to compensate for the Coriolis force

Table 4.5: Total impulse for descent scenarios

Scenario		**PP Impulse** (Ns)	**PF Impulse** (Ns)	**DAC Impulse** (Ns)
$\theta_i = 10^0$	Rot	14.08	5.98	1.79
	Trans	272	647	183
	Σ	286.08	652.98	184.79
$\theta_i = 20^0$	Rot	18.05	8.26	4.86
	Trans	290	666	195
	Σ	308.05	674.26	199.86
$\theta_i = 30^0$	Rot	21.56	19.07	21.67
	Trans	311	685	241
	Σ	332.56	704.07	262.67

Table 4.6: Settling times for descent scenarios

Initial Attitude (^{0})	**Settling Time** (s) PP	**Settling Time** (s) PF	**Settling Time** (s) DAC
10	6.9	19.2	820
20	9.9	16.2	840
30	12.4	20.1	1010

and ensure a vertical descent. The magnitude of the impulse required to control the lander in translational descent is largest for the potential function controller and smallest for the direct adaptive controller with the phase plane magnitude somewhere between the two results. As with the rotational control each controller is capable of maintaining translational control but the DAC algorithm is the most efficient.

The difference between the results for the two studies (rotational and translational) can be explained by the dynamics required to compensate for the Coriolis force. In the attitude control case the aim is to correct a step error input, but in the translational case the coriolis force is continually changing due to the continuous rotation of the asteroid. The DAC algorithm is capable of reacting to both cases due to a real-time change in its gains. The phase-plane and potential function controllers contain constants and these are selected for a specific scenario, leading to a greater impulse as the time for the descent increases.

An example of a descent and rebound scenario can be seen in figures 4.3 - 4.8. Here the potential function controller corrects an initial attitude error of 10^0. The first of these plots (figure 4.3) depicts the full kinematic process of the descent and bounce. The bounce alleviation algorithm can also be seen working efficiently. Starting from 500 m above the asteroid surface the micro-gravity conditions result in the lander making contact with the

surface at approximately 1000 s. A subsequent bounce ensues but the bounce alleviation algorithm quickly suppresses the kinematics.

A simulation of the initial attitude correction can be seen in figure 4.4, again for the same potential function controller - this is a detailed plot of the first 20 s of descent, as in most descent scenarios the attitude of the lander is quickly corrected. The black frames in figure 4.4 are a representation of the lander, giving an appreciation for the manoeuvre that has been conducted.

For the same simulation a more detailed depiction of the bounce kinematics is seen in figure 4.5. Here the brief leg contact phase can be seen along with the resultant rebound. Due to the kinetic energy absorbed by the lander legs and the bounce alleviation algorithm the gradient of the ascending rebound is considerably less that of the descending approach. Figure 4.6 shows how the legs deform during the contact phase of the simulation. The plots for the left and right legs are almost identical as the attitude error has already been corrected.

The associated attitude changes for the same simulated scenario are shown in figure 4.7. Again the fast response of the potential function controller can be seen with slight corrections needed to maintain the attitude within the required error margins. The associated thruster commands for the torque (for rotation) and force (translation) are shown in figure 4.8. In figure 4.8 the thruster mappings can be seen as the two different magnitudes of thrust that can be selected for the rotation of the lander.

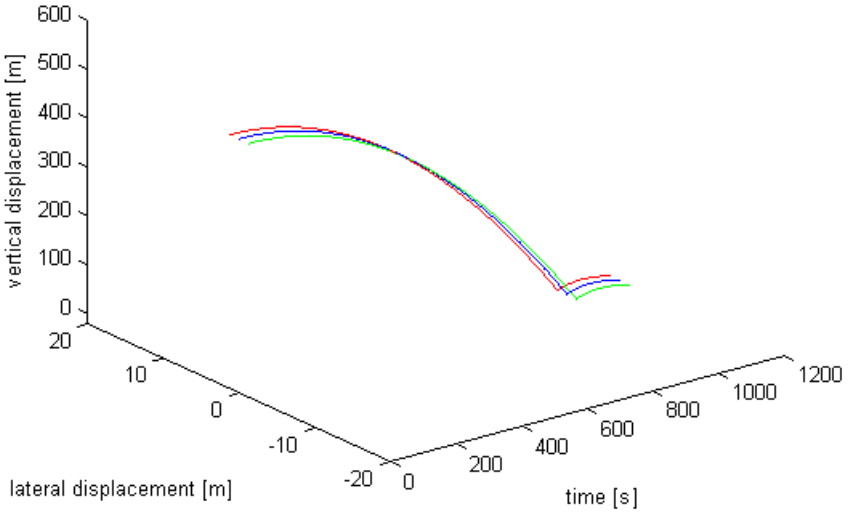

Figure 4.3: Example of descent and rebound

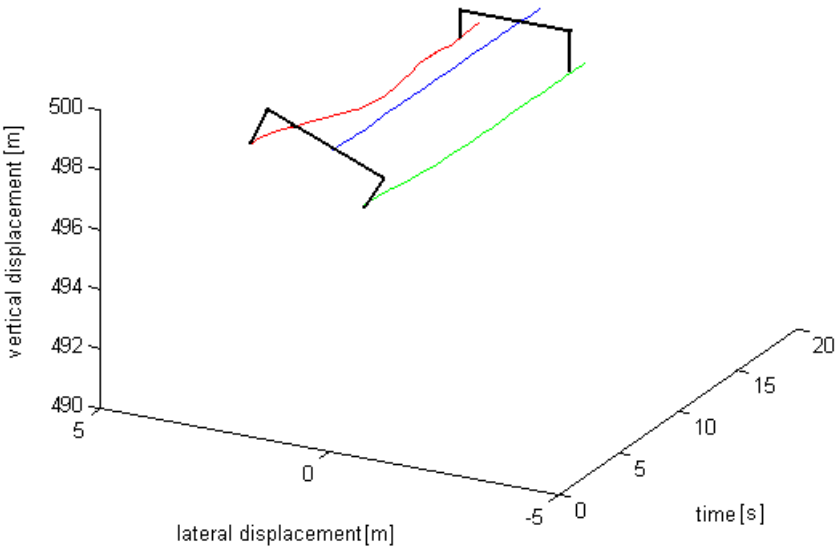

Figure 4.4: Initial attitude correction schematic

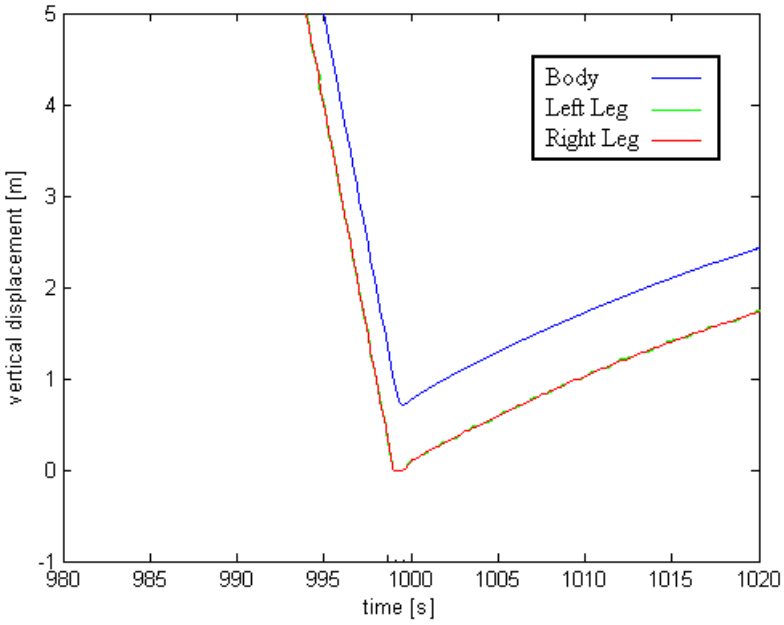

Figure 4.5: Bounce kinematics for descent example

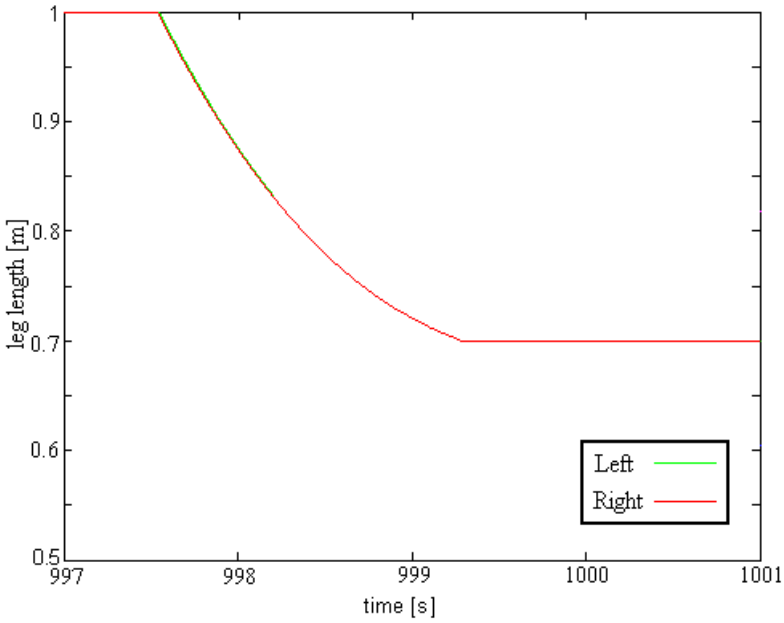

Figure 4.6: Leg deformations for descent example

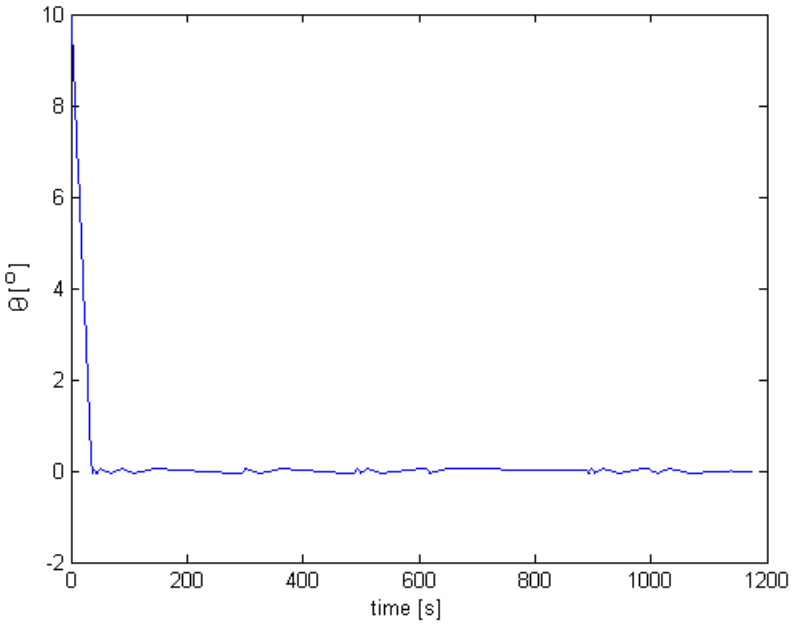

Figure 4.7: Attitude change for descent example

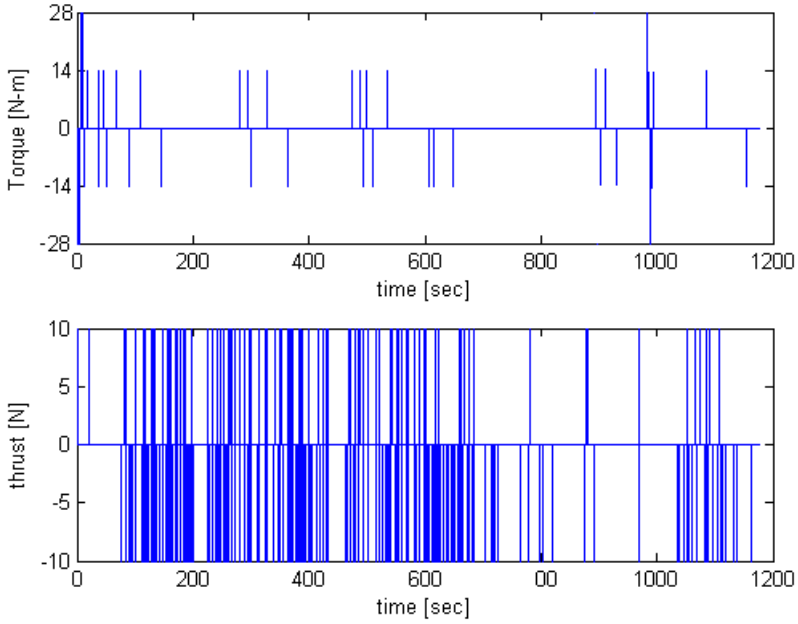

Figure 4.8: Moments and forces for descent example

4.3 FDIR Studies

If an unexpected event were to arise during descent or during science operations on the asteroid it is necessary for the system to have the ability to react. To this end it is desirable to have fault detection, isolation and recovery (FDIR) embedded into the system to minimise the chance of mission failure [74]. This section will explore a range of failure scenarios and demonstrate the utility of DAC as an effective means of embedded autonomy.

4.3.1 Lander Leg Failures

The first analysis of the FDIR properties of the three controllers is a comparison of the ability of the controllers to recover from three different leg failure scenarios. Firstly, one of the two legs is induced to fail by becoming overly 'stiff' with regards to the other leg. The failing leg is assumed to have a doubling of the spring constant in that particular leg leading to an imbalance between the two legs. Similar to the first case, the second failure scenario is when one of the lander leg spring stiffness constants is half the other. Finally, the third failure scenario is when one leg is artificially 'stuck', where no movement is permitted in this leg.

Ideally the lander is required to settle on the surface of the asteroid using as little propellant as possible while maintaining a small attitude error. An example of the initial descent, contact and first rebound phase of the scenario is given in figure 4.9. The insert shows a close up of the contact phase. Also, the black frames show a representation of the

attitude of the lander as the simulation progresses. During contact the insert shows that the lander attitude increases and the potential function control law is capable of returning the lander to zero attitude error before the next contact with the asteroid occurs.

The initial conditions for the simulations are the same as for the non-failure scenarios in section 4.2. The lander simulation is again started at a height of 500 m above the surface of a 500 m radius asteroid. The lander has an initial lateral velocity to match the rotational rate of the asteroid at 1000 m radial distance from the center. The aim of the simulation is to land at rest with zero attitude and lateral displacement error.

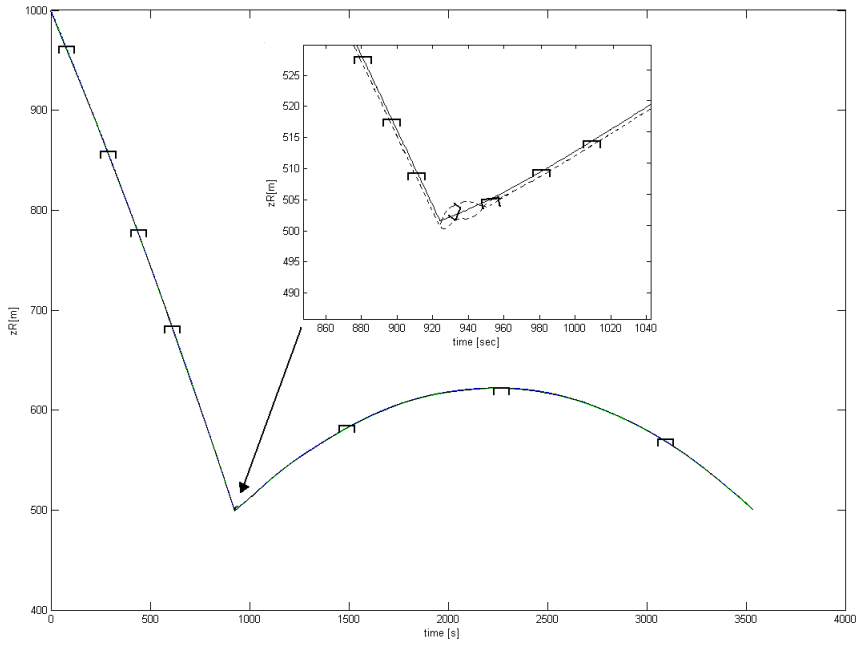

Figure 4.9: Scenario example using a conventional controller

For analysis there are two properties of the simulation that will be used for comparison. As before the total impulse of the scenario will be used to compare the propellant necessary to recover from each of the leg failures. Another property of interest here is the maximum attitude that is induced by the leg failures. This property is representative of the ability of the lander to maintain the necessary track on the asteroid surface for such instrumentation as radar/LASER altimetry [75] as this instrumentation will be positioned on the lander such that it points in the positive vertical body axis direction.

Table 4.7: Impulse for leg failure scenarios

Scenario		PP Impulse (Ns)	PF Impulse (Ns)	DAC Impulse (Ns)
$k_s \times 0.5$	Rot	852	871	653
	Trans	995	1942	810
	Σ	1847	2813	1463
$k_s \times 2$	Rot	850	878	654
	Trans	1237	2022	900
	Σ	2115	2872	1554
$k_s \approx \infty$	Rot	2062	1767	1880
	Trans	2623	6017	930
	Σ	4686	7784	2810

The results for the analysis are shown in table 4.7 and figures 4.10 - 4.16, where the contact dynamics are evident at approximately 880 s. The relative attributes of the different controllers now begin to become more apparent. As was evident with the previous studies, without disturbances, qualitatively the algorithms have inherent benefits/weaknesses. The DAC algorithm is in general the most effective.

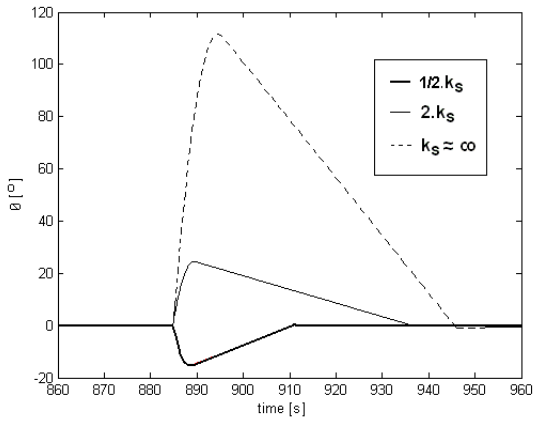

Figure 4.10: θ for potential function control

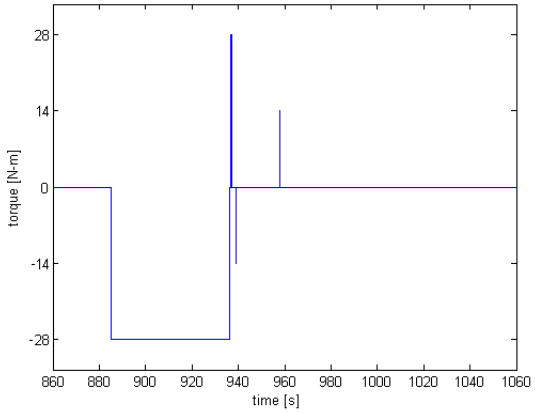

Figure 4.11: Thruster commands for potential function control with $2k_s$

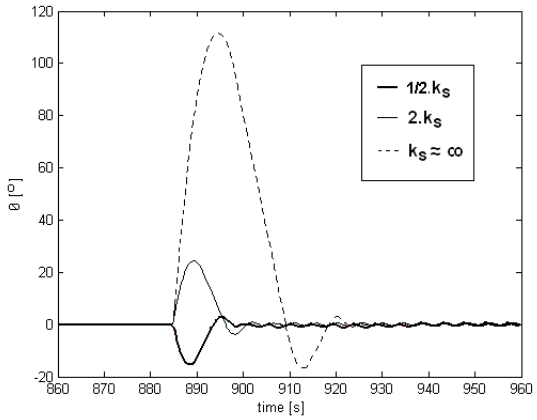

Figure 4.12: θ for phase plane control

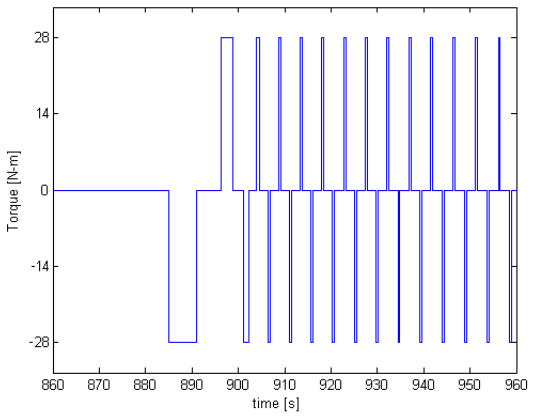

Figure 4.13: Thruster commands for phase plane control with $2k_s$

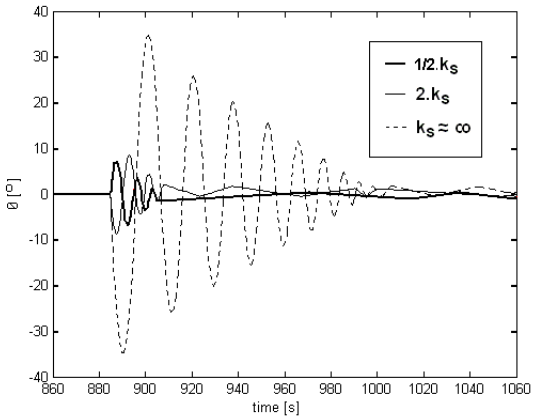

Figure 4.14: θ for direct adaptive control

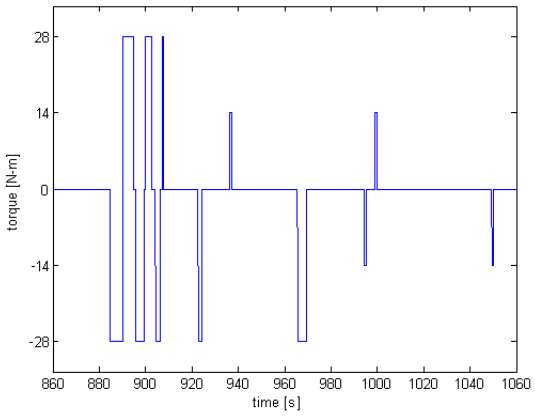

Figure 4.15: Thruster commands for direct adaptive control with $2k_s$

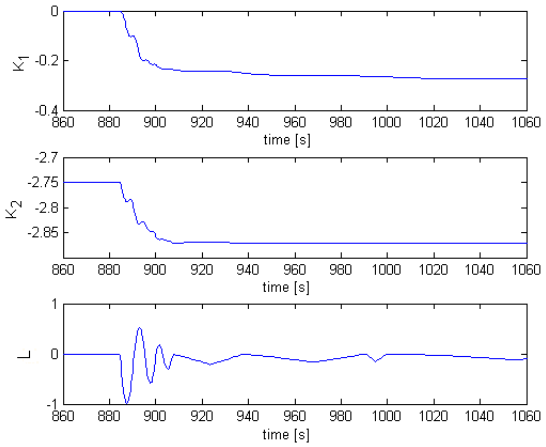

Figure 4.16: Controller gains for direct adaptive control with $2k_s$

Evaluating the evolution of the attitude during the controlled descent demonstrates the weaknesses of the phase plane and potential function controllers with the maximum overshoots being larger that the DAC results. In all cases the controllers are capable of maintaining control of the lander. However, an overshoot of over 90^0 would be too extreme for the altimetry to continue to function and hence the potential function and phase plane controllers may be deemed to fail in some cases.

These large overshoots will also contribute to extra necessary translational control. As the attitude of the lander increases any corrective control will contribute to the height of the rebound. The higher this rebound the more time the lander will spend correcting for translational errors leading to greater impulse requirements.

Alongside the plots of the attitude of the lander during each of the failure cases are plots of the associated torque commands for each of the controllers in the doubled spring constant scenarios. The first of these (for the potential function controller) seen in figure 4.11 shows how the controller leads to minimal thrust requirements after the failure has been stabilised. The phase plane plot in figure 4.13 shows how, after dealing with the failure, the controller periodically moves from one side of the dead-band to the other, leading to a continuous need for thrusting.

For the case of the DAC controller two plots are provided, one for the thrust (figure 4.15) and one for the associated controller gains (figure 4.16). In the figure denoting the torques the periodic nature of the controller can be seen, but this is not as extreme as the phase-plane case. The gains plot shows how the controller deals with the fault. At the point where the failure is encountered (approximately $t = 885$ s) the gains can be seen to be rapidly changing. It is these gains that drive the demanded torque and hence the thrusters. By using direct adaptive control, a large unknown disturbance is compensated for without using FDIR and gain look-up tables.

4.3.2 Lander Thruster Failure

The second FDIR scenario to be investigated is when one of the thrusters used for rotational control is failed. During this study we will consider two modes of thruster failure. Firstly when one thruster (of the six) becomes inoperable - this leaves only one thruster for control in one rotational sense and the usual two thrusters for control in the opposing direction. Secondly, the case where one thruster continuously fires, leaving the lander with a bias torque unless suitable corrective thrust is applied. Referring to figure 3.1 it will be thruster 1 that will be assumed to fail in both cases. A comparison of the three controllers leads to an another insight into the FDIR capabilities.

Thruster 1 is failed and an investigation similar to that described in the previous section is performed. The same initial conditions and constants are used for both the studies, although only the 10^0 initial attitude case is under investigation here.

Tables 4.8 and 4.9 show a comparison of the impulse and settling times required for this failure. Interesting results appear in the tables. It appears that there is no large

apparent change in the magnitude of impulse required to control the lander, indeed in the cases of all control algorithms there is no significant gain to be made.

Table 4.8: Total impulse for thruster failures

Initial Attitude (10^0)	**Total Impulse** (Ns) PP	**Total Impulse** (Ns) PF	**Total Impulse** (Ns) DAC
No Failure	14.078	5.98	1.7898
Off Failure	14.299	5.84	1.461
On Failure	14.418	6.16	2.005

In table 4.8 the impulse used by the thruster which is continuously firing has been removed so an easier comparison can be drawn. The numerical values for the impulse for the continuously firing thruster in the three simulation are: phase plane control - 8923 [Ns], potential function - 8957 [Ns], direct adaptive control - 8998 [Ns].

Table 4.9: Settling times for thruster failures

Initial Attitude (10^0)	**Settling Time** (s) PP	**Settling Time** (s) PF	**Settling Time** (s) DAC
No Failure	6.9	19.2	820
Off Failure	10.1	21.1	850
On Failure	10.3	21.0	847

An explanation for these slight changes in impulse can be attributed to the model being a freely rotating rigid body. The smaller thrust induced by the failure allows fine-tuning of the attitude to be done more easily, although over a slightly longer time period. Both these trends can be seen on analysis of figure 4.17 that depicts the DAC attitude error. Here the longer settling times are associated with smaller individual error correcting impulses.

The increase in settling time can be seen in figure 4.18 that depicts the attitude error for each of the failure scenarios during the course of the potential function decrease. This increase in settling time in the off failure case is due to only one thruster being available for negative rotation. It is then evident that the on failure case only starts to differ from the no failure case when positive rotation is required. Due to the continuous negative thruster failure the result differs slightly when in the positive rotational sense.

Finally, to explain the slight increase in impulse and settling time in the phase-plane cases figure 4.19 shows the attitude error. The longer time periods spent passing each end of the dead-band accompanied by the longer settling time lead to the total impulse being slightly larger.

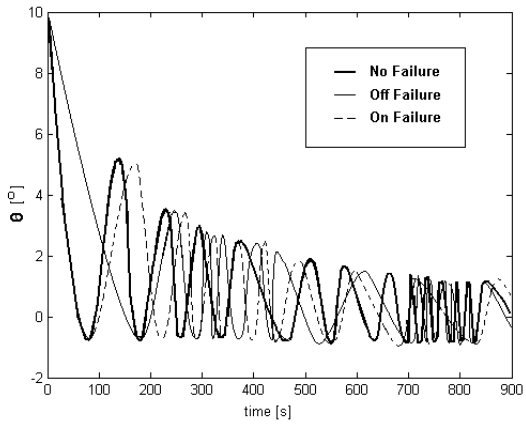

Figure 4.17: Attitude error for direct adaptive control

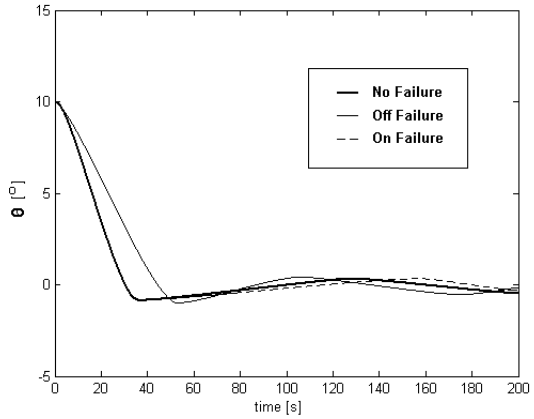

Figure 4.18: Attitude error for potential function

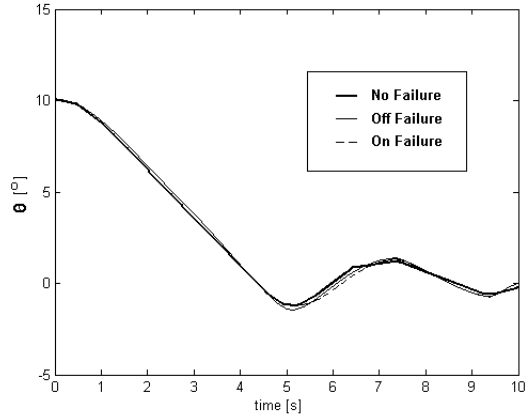

Figure 4.19: Attitude error for phase-plane

4.4 Sampling Study

In addition to the attitude/rotational control and FDIR studies performed above, an additional study has been performed to indicate propellant gains that can be made during the asteroid sampling process by adopting a DAC algorithm. The control force in this instance is intended to be used as a alternative mechanism for maintaining contact with the asteroid during the sampling process. The current proposed method for maintaining contact with the surface is a continuous thrust towards the asteroid using two cold gas thrusters [76].

The aim of this simulation is to stabilise the lander at the stable position the lander rests at after descent. Any deviation from this desired vertical location induces an error into the control law and this error drives the thrusters. A schematic of the system under analysis is given in figure 4.20.

To model the unknown composition of the asteroid a varying reaction drill force is introduced. The maximum reaction force is limited at ten times the weight of the lander base. Also, the reaction force is permitted to vary its form; firstly the reaction force follows a sinusoidal path from maximum to zero and return, then an intermittent pulsing force with discontinuous steps from maximum to zero force is introduced. By doing so most of the possible surface materials that may be encountered are modelled. In these cases: a solid rock (constant force), a porous material (sinusoid force), and a solid with voids (pulsed force).

Visualisation of the control process is demonstrated with a plot of the required reaction force in figure 4.21(a) with a plot of the thrust produced in figure 4.21(b). The impulse required for this simulation is given in table 4.10 alongside the impulse that would be required for the base-line approach with continuous thrusting.

Evidently the impulse required to stabilise the lander is much less using the DAC algorithm compared to the conventional approach. The reason for this becomes apparent on analysis of figure 4.21. Due to the varying type of reaction force that is encountered during the sampling process it is not always necessary to continually counteract the reaction force. The DAC algorithm is capable of selecting the appropriate times when a control thrust is not needed, leading to savings in the total impulse and required propellant mass.

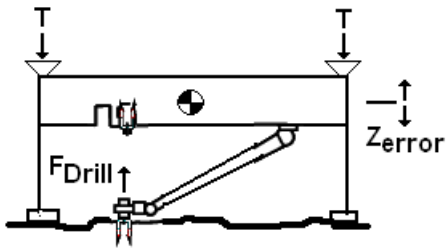

Figure 4.20: Sampling

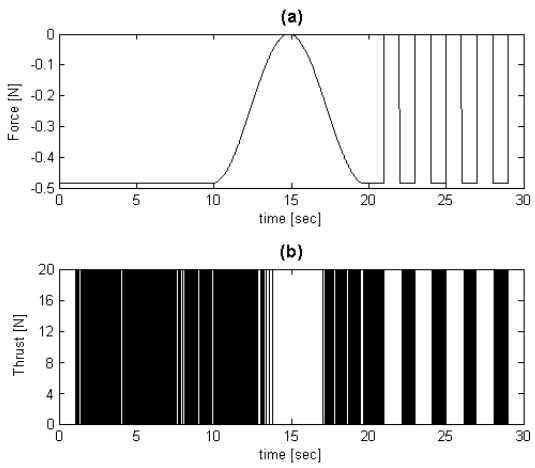

Figure 4.21: Reaction force (a), thrust force (b)

Table 4.10: Total impulse for stabilisation

	Conventional	**DAC**
Total Impulse (Ns)	600	46

4.5 Discussion of Results

In section 4.2 a quantitative comparison of the three controllers implemented is given. It can be seen from the results that the direct adaptive controller performs best, in both rotation and translation. These results are for non-failure cases and demonstrate the larger range of operation that can be acquired through direct adaptive control leads to impulse saving before any beneficial FDIR properties are considered. This larger range of operation, for minimising impulse, came with an associated increase in settling time.

The subsequent FDIR studies conducted in section 4.3 provide more insight into the properties of the three controllers. The first of these studies determined the comparative impulse necessary to recover from failures in the lander legs. Again, the direct adaptive control outperformed the other two controllers in both rotational and translational control. The ability of the direct adaptive control algorithm to vary its control gains means it has larger range of operation and the faults occurring are within this range.

The second FDIR study considered thruster failures. The direct adaptive control algorithm uses the least impulse to recover from the failures but as a consequence takes longer to recover.

A demonstration of another method of incorporating embedded autonomy associated with the direct adaptive control algorithm is provided in the sampling study. A continuous down-force is shown to be unnecessary. Instead, by changing the gains in the controller the occasions when the down-force is needed can be provided. It was shown that direct

adaptive control can decrease the impulse necessary significantly.

The studies conducted in this chapter demonstrate the relative advantages to be gained from adaptive control. In the case of unexpected faults occurring in the system the DAC method of control outperforms the phase plane and potential function controllers. The ability of the controller to react and adapt to uncertainties in the plant dynamics and environment could prove important in cases where there is little or no communication with a ground station or when the communication time-lag with the spacecraft becomes important, such as in deep space missions. This results as a consequence of the autonomy being embedded in the control law and therefore the lander did not have to access any other FDIR algorithms to react to uncertainties.

The asteroid lander control problem demonstrates that an evolutionary process of moving to a 'bottom-up' internally embedded method of autonomous control will have several advantages over conventional systems. First, through propellant mass savings, second, through better tracking by minimising attitude errors, and third, by introducing autonomous control algorithms that can be mathematically verified.

Chapter 5

A State Deficit Approach to Micro-Satellite Action Selection

5.1 Introduction

In this chapter a different application of autonomous control will be investigated in the form of action selection for micro-spacecraft. A range of novel algorithms are developed from dynamical systems theory and applied to an Earth orbiting spacecraft.

First the orbital model that the key control algorithms use is described. Then, the development and stability analysis of the respective control algorithms is presented. Finally, comparisons are drawn between the controllers and a fault recovery study is conducted.

Current research interest on spacecraft systems has a focus on miniaturisation. Concepts such as formation-flying and swarming [77] [78] have demonstrated the advantages that can be gained from having a system comprised of a number of smaller spacecraft as opposed to one large platform. In doing so, and realising the benefits that are to be gained from such a change, introduces new issues that must be addressed. Such micro-spacecraft pose new challenges for on-board autonomy with limited computational overhead.

The key idea for the new algorithms presented here is for the system to adapt itself to changes in its environment during its mission life. Using data from available sensors the algorithm will self-organise into what is the appropriate response that should be realised by the actuators. The actuator function can range from an upload or download of data to battery charging or fault recovery.

Three different concepts for on-board autonomy will be investigated in this chapter. Each of these concepts draws from ideas developed in dynamical systems theory [79] [80], as described in chapter 2.

The first of the three control algorithms is based on the Lotka-Volterra model [81] [82]. The Lotka-Volterra model has been used extensively to analyse inter-species interactions in ecological systems. Differential equations are used to describe the state of each species and the degree they each affect the other. Here, this model has been adapted for the

spacecraft action selection algorithm. Replacing each species with a spacecraft state and permitting the states to compete for available resources (energy and memory) a control algorithm has been developed.

Second, an artificial potential field algorithm [83] [84] is presented. Here each equilibria now represents a local minimum in the potential field. By associating a task to each minimum and permitting the global potential to evolve as a function of the system parameters (battery charge and data stored), the solution to the system of differential equations motivates an action selection.

The third control algorithm develops the idea of Lyapunov stability [85] [86] through direct adaptive control (DAC). Such robust controllers desensitise the system to the uncertainties of the plant by varying the controller gains continuously. Here an attitude control DAC algorithm is presented that attempts to minimise the artificial potential of the system defined by state deficits. The state deficits define the state of the system with respect to the ideal state. As these deficits are inter-related the DAC algorithm must choose an appropriate attitude to bring the deficits closest to their minimum value.

Results from introducing each of the three controllers into a simple model of an autonomous spacecraft in Earth orbit are presented. State-space mappings are presented along with an analysis of the action selection. The results from these models demonstrate that the use of modern dynamical systems theory to generate provable control algorithms appears to be a promising concept for computationally limited micro-spacecraft.

5.2 Orbital and Spacecraft Model

In this section a description of the orbital and spacecraft model used for the micro-spacecraft autonomy development is presented. First the kinematics of the orbital model are defined. Subsequently the subsystems that comprise the spacecraft model will be described and then how the control algorithms are incorporated into the model. The model will follow the analysis of Radice [5].

5.2.1 Orbital Model

Earth orbiting spacecraft commonly adopt the center of mass of the Earth as the reference point to define the inertial reference frame. This system of reference can be assumed to be inertial, even although the Earth orbits the Sun. The Z-axis is the axis of rotation of the Earth. The equatorial plane defines the X-Y plane of the coordinate system, perpendicular to the Earth's rotational axis. The X-axis is defined by the line formed where the Earth's equatorial plane crosses the ecliptic plane. The Y-axis is completed by a right-handed orthogonal rule. Now that the geometry of the coordinate system has been defined, to position the orbit of the spacecraft, three additional parameters must be introduced.

In figure 5.1, the inclination angle, i, of the orbit is defined with respect to the equatorial X-Y plane. By definition, the orbital plane and the equatorial plane intersect at

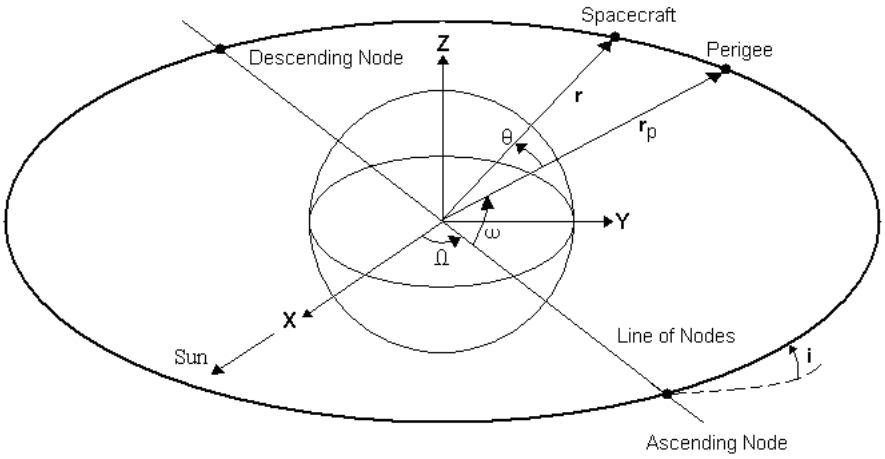

Figure 5.1: Orbit parameters

the line of nodes. The angle that separates the line of nodes from the X-axis in the X-Y plane is the longitude of the ascending node, Ω. The angle between the line of nodes on the ascending side and the orbital perigee is the argument of perigee, ω. Regarding the orbital plane, $\mathbf{r}$ is the position vector of the spacecraft and $\mathbf{r}_p$ is the position vector to the orbit perigee. Combining these three parameters with the eccentricity, e, the semi-major axis, a, and the true anomaly, θ, of the orbit a set of orbital elements are given.

The time varying projection of the spacecraft position vector, $\mathbf{r}$, in the three dimensional inertial frame of reference can be defined. However true anomaly, θ, must be calculated:

$$\theta = M + (2e + \frac{e^3}{4})sinM + \frac{5}{4}e^2 sin2M + \frac{13}{12}e^3 sin3M + 0(e^4) \tag{5.1}$$

Thus knowing e, for $e << 1$, the orbit eccentricity, and M, the mean anomaly, permits the direct calculation of the true anomaly using equation 5.1. Also, as $M = n(t - \tau)$, with the mean angular velocity $n = \sqrt{\mu/a^3}$ where μ is a constant related to the local gravity, and τ is the time of perigee passage, it is possible to express θ as a function of time for small values of eccentricity.

Knowing the true anomaly permits the calculation of the radius vector magnitude $|\mathbf{r}|$ using the equation:

$$|\mathbf{r}| = \frac{p}{1 + ecos\theta} \tag{5.2}$$

where p is the semi-latus rectum of the ellipse. Then the components of the orbital radius on each of the inertial axes can be determined by:

$$r_x = |\mathbf{r}|[cos\Omega cos(\omega + \theta) - cosisin\Omega sin(\omega + \theta)]\mathbf{i} \tag{5.3}$$

$$r_y = |\mathbf{r}|[sin\Omega cos(\omega + \theta) + cosicos\Omega sin(\omega + \theta)]\mathbf{j} \tag{5.4}$$

$$r_z = |\mathbf{r}|[sinisin(\omega + \theta)]\mathbf{k} \tag{5.5}$$

where **i**, **j** and **k** are the unit vectors along the X, Y and Z axis respectively.

5.2.2 Eclipse Model

The time in eclipse is a function of the spacecraft orbital elements. Using the position vector from the orbital model defined in section 4.2.1, a Boolean output is returned, 1 if the spacecraft is in sunlight, and 0 if it is in eclipse as shown in figure 5.1. The Sun-line is defined as being along the positive X-axis, therefore the spacecraft will be in sunlight if $r_x \geq 0$. The spacecraft will also be in sunlight if $r_x < 0$, and a second condition met: $\sqrt{r_y^2 + r_z^2} > R_{Earth}$, as can be seen in figure 5.1.

Two examples are shown in figure 5.2 for two different orbit altitudes but with eccentricity and inclination both zero. The Sun availability will be used later in the development of the action selection algorithms.

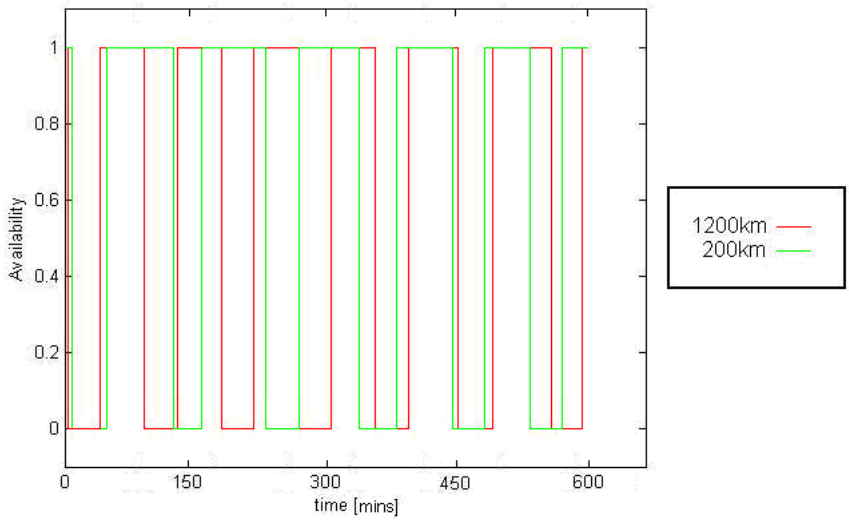

Figure 5.2: Sun availabilities (h = 1200 km - T = 109.5 mins, h = 200 km - T = 88.48 mins)

5.2.3 Spacecraft Model

The focus of this thesis is on action selection control algorithms, therefore a representation of a simple spacecraft is presented here. The spacecraft is defined as having three rotational degrees of freedom that are controlled with reaction wheels. The electrical power system comprises a solar array, battery and electrical loads (which will be defined later). The payload is assumed to be a camera recording data. For data download there is an antenna

for downlink to a ground station. This is accompanied by solid state memory for on-board storage of data. Each of the subsystems of the spacecraft can affect others. For example, data uplink will increase the volume of data stored in the memory and decrease the energy stored in the battery. By choosing to operate the appropriate action the spacecraft must track one of the three objects, the Sun, target and ground station, by activating the reaction wheels.

The spacecraft can therefore perform three different tasks. It may choose to charge its battery by slewing towards the sun. It may uplink data by pointing the payload towards a target. Or, it may downlink data by pointing the antenna towards the ground station. A schematic is shown in figure 5.3. The spacecraft selects the appropriate action by applying one of the control laws to be defined later.

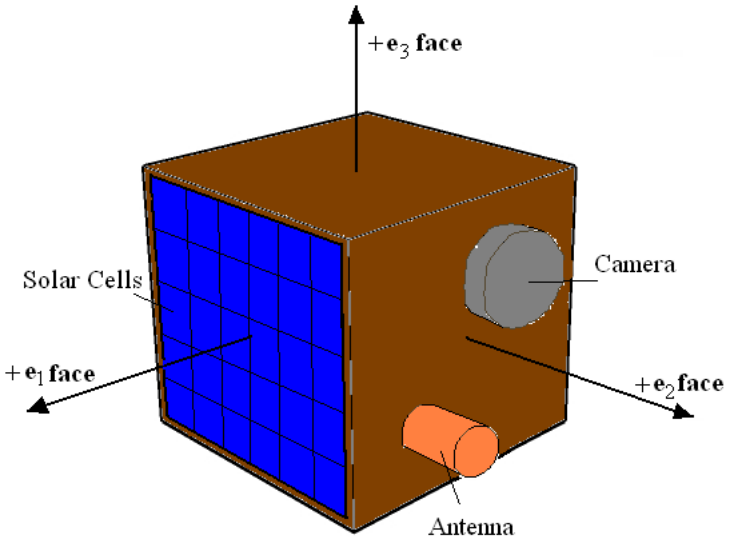

Figure 5.3: Satellite model with body axes e_1, e_2 and e_3

Model Structure

The spacecraft model comprises a range of subsystems, and the functions performed in each subsystem defines the data flow in the model. A block diagram of the data flow in the model can be seen in figure 5.4. In this section each of the subsystems is described. An outline of the values of the physical properties associated with the spacecraft described are given in table 5.1.

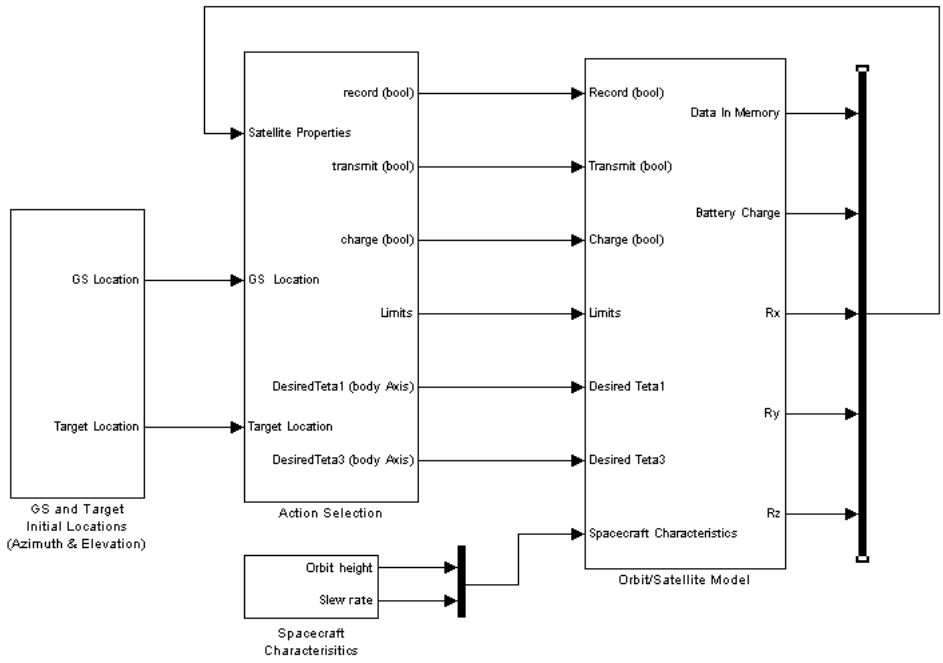

Figure 5.4: Model data flow

Table 5.1: Spacecraft properties

Parameter	**Value**	**Unit**
Camera Data Rate	5	kb/s
Camera Power	0.09	W
Antenna Data Rate	10	kb/s
Antenna Power	2.88	W
Solar Array Power	8	W
Battery Maximum Charge	24	kJ
Battery Minimum Charge	8	kJ
Attitude Slew Power	1.5	W
Attitude Slew Rate	3	$^0/s$
Memory Maximum Data	1	Mb

Payload

For the purposes of the generic satellite model the payload is assumed to be an optical camera. In the camera payload model when the record action is selected by the control law the data rate to the solid state memory of the satellite is 5 kb/s, and a power consumption of 0.09 W acts as a load on the battery [87].

It is assumed that the data is digitally stored in the memory of the spacecraft until it can be down-linked to an Earth ground station. The maximum volume of data that can

be stored on-board is assumed to be 1 Mb [88].

Attitude Control

The generic satellite will use electrically powered reaction wheels for three axis control. When operating the wheels represents a battery load of 1.5 W [89] and the maximum slew rate of the spacecraft is assumed to be 3 $^0s^{-1}$ [90].

In order to generate the maximum level of power available, the solar array needs to face the Sun. Therefore a relationship between the spacecraft pointing angles and relative position of the target or ground station and Sun is required. In figure 5.5 the geometrical configuration of the problem is shown.

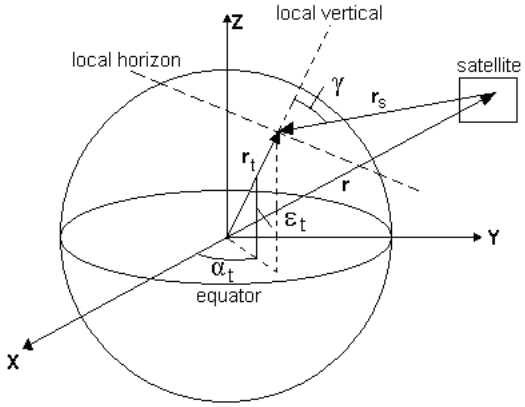

Figure 5.5: Satellite line-of-sight [5]

The optimum position of the satellite with respect to the target or ground station for imaging or downlink respectively is when the satellite is directly above the local vertical of the target or ground station. In the model the availability of the target or ground station is defined by the angle γ:

$$\gamma = cos^{-1}(-\frac{\mathbf{r}_t \cdot \mathbf{r}_s}{|\mathbf{r}_t||\mathbf{r}_s|}) \tag{5.6}$$

where, with reference to figure 5.5, the position of the target is defined by:

$$\mathbf{r}_t = cos\alpha_t cos\epsilon_t \mathbf{i} + sin\alpha_t cos\epsilon_t \mathbf{j} + sin\epsilon_t \mathbf{k} \tag{5.7}$$

where α_t and ϵ_t are the longitude and latitude of the target. Due to the rotation of the Earth the longitude will vary with time as:

$$\alpha_t = \alpha_{t0} + \omega_E(t - t_0) \tag{5.8}$$

with ω_E the angular velocity of the Earth defined as:

$$\omega_E = \frac{2\pi}{T_E} \tag{5.9}$$

where T_E is the Earth's rotational period. The satellite-to-target vector, $\mathbf{r}_s$, is now:

$$\mathbf{r}_s = \frac{\mathbf{r}_t - \mathbf{r}}{|\mathbf{r}_t - \mathbf{r}|} \tag{5.10}$$

The availability will be maximum when the satellite is directly above the target location and zero when the satellite passes over the local horizon plane, so that the availability is determined from $cos\gamma$ $(-\pi/2 \leq \gamma \leq \pi/2)$.

The positions of the target and ground station can be defined for each simulation. This part of the model also receives the three components of the satellite position vector (r_x, r_y and r_z) from the orbital dynamics section of the model. The availability of the target or ground station can then be determined.

Figure 5.6 and 5.7 illustrate availabilities as a function of the spacecraft orbit and the location of the target. In figure 5.6 two equatorial orbits are shown with the resource located at 0^o latitude on the equator. In figure 5.7 a circular polar orbit is shown with the target located on the Arctic circle, at 66.5^o latitude.

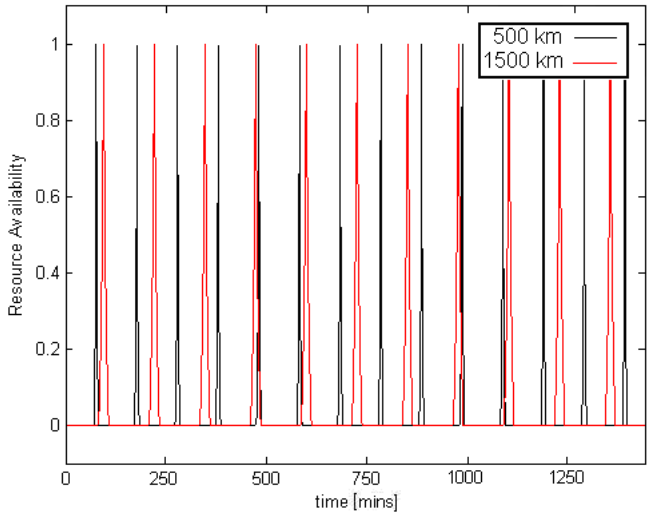

Figure 5.6: Availabilities for two equatorial orbits with target located on Equator

Now that the required pointing direction has been defined it is now necessary to define how the satellite performs the necessary slew manoeuvres.

The direction along which the spacecraft must direct its payload is given by equation 5.10 where $\mathbf{r}$ is obtained from the orbital dynamics and $\mathbf{r}_t$ is the target location. The generic satellite has the camera and antenna placed on the face of the y-body-axis:

$$\mathbf{r}_p^{body} = \mathbf{e}_2 \tag{5.11}$$

where $\mathbf{e}_2$ is the unit vector along the y-axis in the body frame of reference. It is now necessary to define the rotational transformation between the inertial and body axes. This is done by using a transformation matrix $\mathbf{R}$, in this case the 3-1-3 Euler sequence so

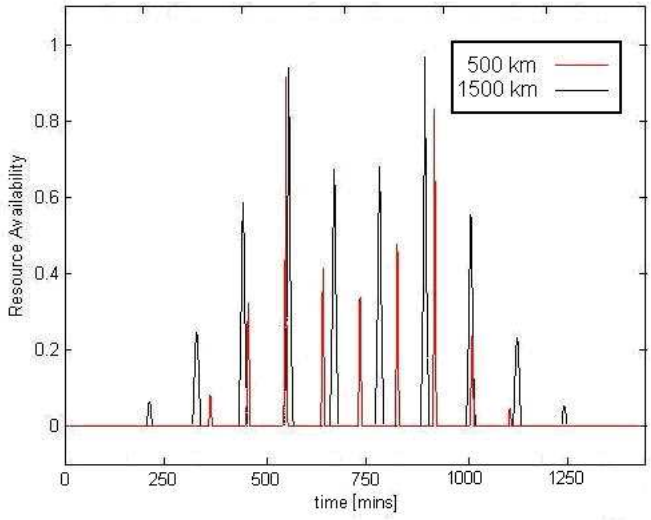

Figure 5.7: Availability for two polar orbits with target located on Arctic circle at 66.5^0 latitude

that:

$$\mathbf{r}_p^{inertial} = \mathbf{R}^T \mathbf{r}_p^{body} \tag{5.12}$$

where the rotation matrix $\mathbf{R}$ is defined as:

$$\mathbf{R} = \begin{bmatrix} c_{\theta_1}c_{\theta_3} - c_{\theta_2}s_{\theta_1}s_{\theta_3} & c_{\theta_3}s_{\theta_1} + c_{\theta_1}c_{\theta_2}s_{\theta_3} & s_{\theta_2}s_{\theta_3} \\ -c_{\theta_2}c_{\theta_3}s_{\theta_1} - c_{\theta_1}s_{\theta_3} & c_{\theta_1}c_{\theta_2}c_{\theta_3} - s_{\theta_1}s_{\theta_3} & c_{\theta_3}s_{\theta_2} \\ s_{\theta_1}s_{\theta_2} & -c_{\theta_1}s_{\theta_2} & c_{\theta_2} \end{bmatrix} \tag{5.13}$$

where c and s denote cos and sin respectively. Therefore:

$$\begin{aligned} \mathbf{r}_p^{inertial} = \quad & (-cos\theta_2 cos\theta_3 sin\theta_1 - cos\theta_1 sin\theta_3)\mathbf{e}_1 \\ & +(cos\theta_1 cos\theta_2 cos\theta_3 - sin\theta_1 sin\theta_3)\mathbf{e}_2 \\ & +(cos\theta_3 sin\theta_2)\mathbf{e}_3 \end{aligned} \tag{5.14}$$

Since the camera and antenna are directed along the $\mathbf{e}_2$ body axis θ_2 is undefined.

This part of the model takes as input the longitude and latitude of either the target or ground station depending on the action selection and the inertial position of the satellite. The output is the desired Euler angles necessary to perform the respective task by slewing toward the required direction.

Telecommunications

When active, the transmitter downloads to the ground station at 10 kb/s and requires 2.88 W of power to operate [91]. The input to this model is the Boolean as to whether this action has been selected. The output is the transmitter data rate, affecting the memory,

and power consumption, affecting the battery load.

Electrical Power
It is assumed that one face of the generic satellite is a solar panel with a 10 cm x 10 cm solar array. Choosing silicon cells that have an energy conversion efficiency of 15% [89], the solar panels produce a constant power of 8 W when in direct sunlight.

The battery model takes as input the Boolean signals as to which loads are selected and the net power from the different subsystems which require energy. Upper and lower lethal limits of the battery are defined. The battery is assumed to have a capacity of 24 kJ and a minimum charge to guarantee the spacecraft survival of 8 kJ. The output from the model is the current battery charge level [89]. By combining the solar panel model and the battery model an electrical power model is produced. The electrical power model provides the battery charge which affects the action selection algorithm.

5.3 Controllers

This section defines the controllers used in the autonomous action selection process. Where applicable, stability analysis of the control laws is provided. The model results from using each of the controllers in the generic satellite model are then given.

5.3.1 Lotka-Volterra Model

The first type of controller developed for use in the autonomous control action selection algorithms is based on the Lotka-Volterra predator-prey equations [81] [82] [92]. The Lotka-Volterra model of species interaction has been used extensively as a means for analyses of ecological systems. Any number of species can be introduced to the system of differential equations that define the propagation of each species as a consequence of competing for resources. The Lotka-Volterra model is presented here. Subsequently a set of similar differential equations will then be developed to represent the internal competition for resources in the generic satellite.

The generalised Lotka-Volterra model is given by:

$$\tau \frac{d}{dt} x_i(t) = x_i(t)[\mu_i - \sum_{j=1}^{n} \phi_{ij} x_j(t)] + x_i(t) \cdot \eta(t), \qquad i = 1, \cdots, n. \tag{5.15}$$

and represents the dynamical analysis of competition among n agents [93]. In the generalised Lotka-Volterra model $x_i \geq 0$ represents each of the competing agents, τ is the time-constant, μ_i represents the resources available to each of the species, ϕ_{ij} is the competition matrix which represents the interaction of the species involved in the competition, and $\eta(t)$ is noise perturbing the model.

The system of equations 5.15 capture the dynamical behaviour of the system being

modelled. One of the dynamical phenomenon present is heteroclinic behaviour, which is a consequence of the competition parameter ϕ_{ij} and the resource parameter μ_i when they are bounded within a certain range [94].

In the following section an amended form of the Lotka-Volterra model, adapted for the generic satellite, is presented alongside a stability analysis of the subsequent cycle.

The Lotka-Volterra Controller

An action selection algorithm based on the Lotka-Volterra model is presented here. This algorithm uses the concept of action 'modes' that are competing for available resources. Each of the action modes are representative of an action that the generic satellite can perform, and the resources available to the satellite in the model are defined by the 'state deficits' of the system. When one of the modes breaks a certain threshold, the associated action can be implemented in the control algorithm. The state deficits [5][95] are given by, in the case of three competing actions:

$$Battery\ Deficit\ d_c = \frac{batt_{max} - batt_{cur}}{batt_{max} - batt_{min}} \tag{5.16}$$

$$Information\ Deficit\ d_r = \frac{mem_{max} - mem_{cur}}{mem_{max}} \tag{5.17}$$

$$Transmission\ Deficit\ d_t = \frac{mem_{cur}}{mem_{max}} \tag{5.18}$$

where *batt* corresponds to the battery charge at current (cur), maximum (max) and minimum (min) values and *mem* corresponds to the data held in the satellite memory at current (cur) and maximum (max) values. The magnitude of d_c increases as energy is consumed. The magnitudes of d_r and d_t hold an inverse relationship. By recording data d_t increases while d_r decreases, while downlinking data causes d_r to increase while d_t decreases.

The three action modes are now labelled x (battery charge), y (record data) and z (downlink data). The amended version of the Lotka-Volterra model that is used as an action selection algorithm, in the case of three action modes (x, y, z) is given by:

$$\dot{x} = x(\sigma_x d_c - \phi_{yx} y - \phi_{zx} z) + x \cdot \eta(t) \tag{5.19}$$

$$\dot{y} = y(\sigma_y d_r - \phi_{xy} x - \phi_{zy} z) + y \cdot \eta(t) \tag{5.20}$$

$$\dot{z} = z(\sigma_x d_t - \phi_{xz} x - \phi_{yz} y) + z \cdot \eta(t) \tag{5.21}$$

where ϕ_{ij} represents the competition between the modes and σ_i is the threshold for each of the modes. The definition of the thresholds define the magnitude each of the modes must reach before the respective action is implemented.

A schematic describing the three dimensional state-space is shown in figure 5.8. Each

axis of the state-space corresponds to one state variable in the above equation set. The dots denote the positions of the equilibrium points that correspond to the thresholds for the respective action.

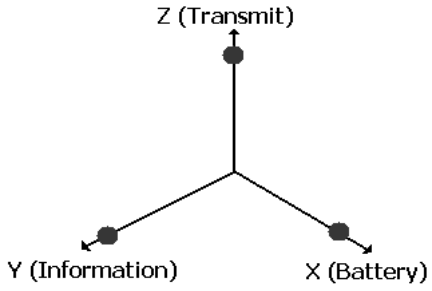

Figure 5.8: Three dimensional state-space with associated actions: x(charge battery), y(record data), z(downlink data)

The set of equations 5.19-5.21 define a permutating cycle occurring as a consequence of the systems stability properties. Changes in the satellite state deficits cause the stability properties of the equations to change, as will be discussed later. The internal state of the generic satellite is modelled through the inclusion of these state deficits. However, it is the dynamical competition between the action modes that define the action selection process. This produces a controller where the resources, mutually available to each action (energy and data), are under competition. The result is a controller that responds to changes in the system state that can result as a consequence of environmental uncertainties or system failures.

Stability Analysis of the Lotka-Volterra Controller

To describe the stability of the system of differential equations represented by the amended Lotka-Volterra controller a simplified case is studied. This is so the stability characteristics of the system can be easily identified. Following this the controller will be applied to the satellite model, where an accurate representation of the generic satellite and internal state interactions are included.

Equations 5.19-5.21 are refined where all the internal state interactions are equal (i.e. $\phi_{ij} = 0.05$). The three mode thresholds σ_x, σ_y and σ_z are 0.1, 0.35 and 0.55 respectively, and a small amount of noise is included ($-0.001 \leq \eta(t) \leq 0.001$):

$$\dot{x} = x(0.1d_c - 0.05y - 0.05z) + x\eta(t) \tag{5.22}$$

$$\dot{y} = y(0.35d_r - 0.05x - 0.05z) + y\eta(t) \tag{5.23}$$

$$\dot{z} = z(0.55d_t - 0.05x - 0.05y) + z\eta(t) \tag{5.24}$$

The ϕ_{ij} numbers are normalised and are set to 0.05 to encourage a reasonable level

of state interaction. Also, the thresholds are selected at three different magnitudes to demonstrate manipulation of the control algorithm.

At any time during model simulation there are four equilibrium points. However, the only point that is stationary is at (0,0,0), the other three points move as a function of the system state deficits. The stability of all the points is found by determining the eigenvalues of the Jacobian, **J**, defined as:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \dot{x}}{\partial x} & \frac{\partial \dot{x}}{\partial y} & \frac{\partial \dot{x}}{\partial z} \\ \frac{\partial \dot{y}}{\partial x} & \frac{\partial \dot{y}}{\partial y} & \frac{\partial \dot{y}}{\partial z} \\ \frac{\partial \dot{z}}{\partial x} & \frac{\partial \dot{z}}{\partial y} & \frac{\partial \dot{z}}{\partial z} \end{bmatrix} \tag{5.25}$$

With respect to the point (0,0,0) all the eigenvalues, λ_i (i = 1-3), are real and positive, although varying in magnitude as a function of the state deficits. This implies that this point is an unstable node/spiral. The exception is when any of three state deficits $d_i = 1$. In this case the corresponding eigenvalue is zero implying a stable center. This is acceptable because the interaction between the state deficits implies that all three deficits cannot be equal to 1 and some evolution of the system will always occur.

The remaining three equilibrium points change their locations depending on the magnitudes of the state deficits. Therefore to analyse the stability of the these equilibrium points it is necessary to choose possible combinations for the state deficits and analyse the resulting eigenvalues. The general solution for the Jacobian with respect to the system of equations 5.22-5.24, assuming the noise terms are negligible, is:

$$\mathbf{J} = \begin{bmatrix} 0.1d_c - 0.05(y+z) & -0.05x & -0.05x \\ -0.05y & 0.35d_r - 0.05(x+z) & -0.05y \\ -0.05z & -0.05z & 0.55d_t - 0.05(x+y) \end{bmatrix} \tag{5.26}$$

From this Jacobian eigenvalues can be calculated at specified locations with specified state deficits. Eigenvalue analysis can be conducted over the range of operating scenarios and examples of the action mode values, state deficits and resultant eigenvalues are given in table 5.2. From the table, the appearance and subsequent disappearance of the stable manifolds illustrates the behaviour of the action selection algorithm. Since only one stable manifold is present the system flows towards the appropriate action mode.

A simulated example of this system running with artificial input state deficits in given in figures 5.9 - 5.11. In figure 5.9 the artificial state deficits are shown. Here each one of the state deficits dominates the others sequentially, leading to a simulation of all the entries in table 5.2. Figure 5.10 shows the magnitudes of all the action modes as the simulation progresses. Finally, figure 5.11 depicts the phase plane of the simulation.

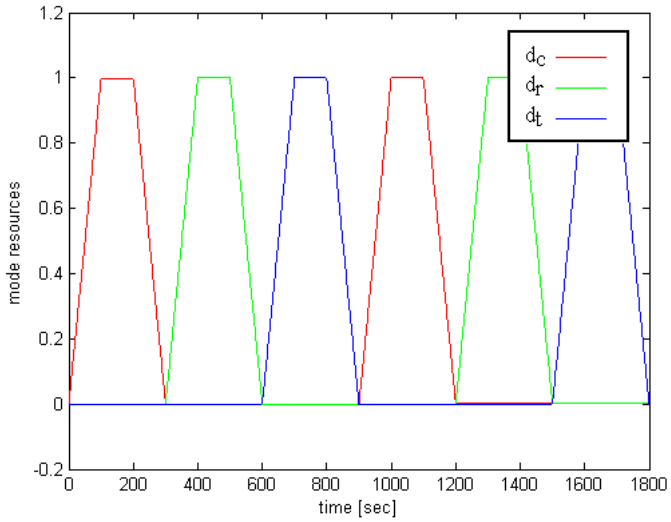

Figure 5.9: Artificial resources for stability analysis

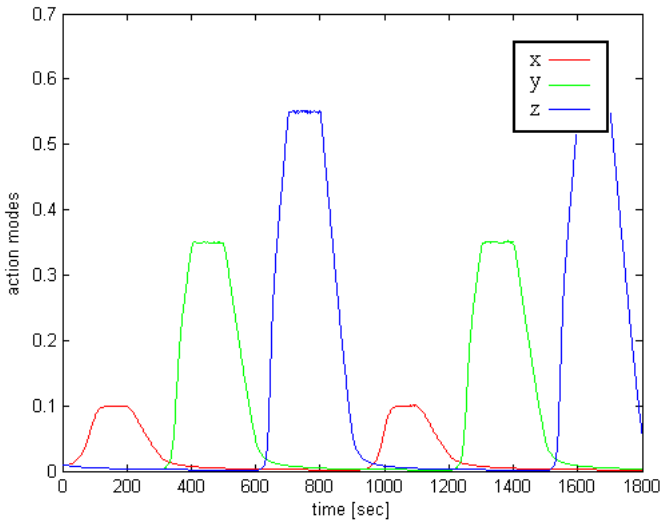

Figure 5.10: Action modes for stability analysis

Table 5.2: Lotka-Volterra stability properties

Action Mode (x, y, z)	**State Deficit** (d_c, d_r, d_t)	**Eigenvalues** $(\lambda_1, \lambda_2, \lambda_3)$
(0.1,0,0)	(0,1,0)	(0,0.345,-0.005)
(0.1,0,0)	(0,0,1)	(0,-0.005,0.545)
(0,0.35,0)	(1,0,0)	(0.0825,0,-0.0018)
(0,0.35,0)	(0,0,1)	(-0.0018,0,0.533)
(0,0,0.55)	(0,1,0)	(-0.0275,0.323,0)
(0,0,0.55)	(1,0,0)	(0.09725,-0.0275,0)

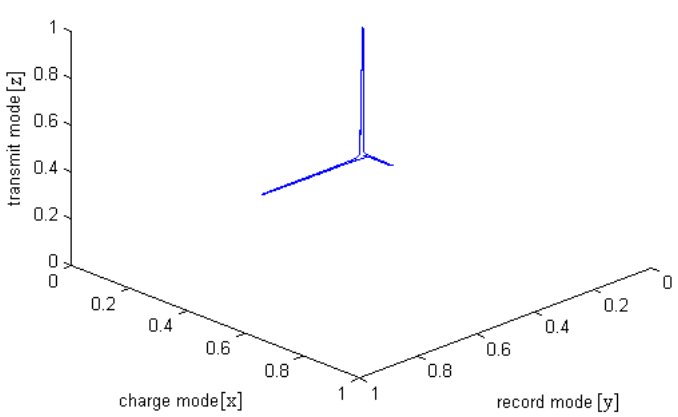

Figure 5.11: State-space for stability analysis

Figures 5.9 - 5.11 show the cyclic behaviour of the system of equations 5.22 - 5.24. The changing magnitudes of the state deficits result in the system evolving in a controlled manner. It is the ability to manipulate and control the cycle that leads to the utility of the equations as a means of controlling the generic satellite.

Lotka-Volterra Orbital Model

For application of the control law to the generic satellite orbital model the parameters in equations 5.19-5.21 are changed. These new parameters for the internal state interactions reflect the change from a idealised environment in the previous example to an orbital model, while still retaining the stability concepts of the control cycle.

The three mode thresholds for the orbital model for σ_x, σ_y and σ_z are maintained at 0.1, 0.35 and 0.55 respectively. The satellite internal state interactions are representative of the satellites battery charge rate, camera data rate and antenna data rate given in table 5.1, but they are normalised to reflect the range of the associated action modes $(0 \leq (x, y, z) \leq 1)$. The normalisation factor with respect to the charge mode is the operational range of the battery $(batt_{max} - batt_{min})$, and for the record and transmit modes the normalisation factor is the maximum data the memory can hold mem_{max}, as defined in table 5.1.

This leads to the system of equations given by:

$$\dot{x} = x(0.1d_{ca} - 5.6 \times 10^{-6}y - 1.8 \times 10^{-4}z) + x\eta(t) \tag{5.27}$$

$$\dot{y} = y(0.35d_{ra} - 0.01z) + y\eta(t) \tag{5.28}$$

$$\dot{z} = z(0.55d_{ta} - 0.005y) + z\eta(t) \tag{5.29}$$

These equations lack an x component in equations 5.28 and 5.29. This is due to the fact that charging the satellite battery does not effect the internal state of the satellite with regards recording or transmitting data. The magnitudes of the ϕ_{ij} terms in equation 5.27 also appear small, this is because these parameters are again normalised, denoting that the record and transmit actions take some time to use the available energy stored in the battery.

Also included in the equation set is a degree of resource availability. This concept was described in section 5.2.2. This implies that an action mode resource is now a function of both the satellite internal state deficit and the resource availability. The new resources are hence defined as:

$$d_{ca} = d_c bool_{sun} \tag{5.30}$$

$$d_{ra} = d_r a_{tar} \tag{5.31}$$

$$d_{ta} = d_r a_{gs} \tag{5.32}$$

where $bool_{sun}$ is the boolean of whether the satellite is in eclipse, a_{tar} is the availability of the target, and a_{gs} is the availability of the ground station.

Since the cycle takes some time to move between actions, some performance could be lost in scenarios where time is important. To overcome this problem an action is presumed to be selected when the associated action mode dominates the others and passes an associated threshold. Therefore, although the action modes take time to increase, the thresholding permits a faster change of actions to be realised.

The time dependant characteristic of this controller is undesirable, but the controller operates to a reasonable level of efficiency. In a more complex controller using some of the same characteristics as the Lotka-Volterra controller, to be discussed in chapter 6, a method for alleviating this issue is presented.

Model Results

To demonstrate the implementation of the method described above, results from the generic satellite model are presented. The first of these is the action potentials and the associated state space mapping. Figure 5.12 shows the individual action selection mode

potentials evolving over a period of 24 hours. Each action mode can be seen dominating respectively and this becomes evident when the same results are illustrated on a three dimensional phase-plane, shown in figure 5.13.

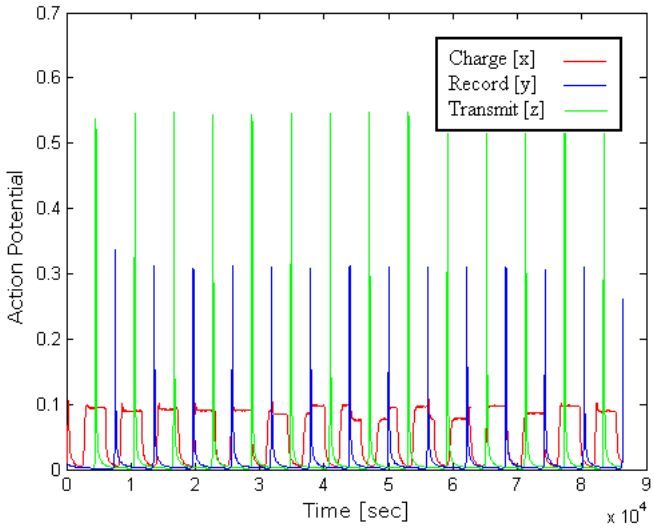

Figure 5.12: Action potentials for Lotka-Volterra model

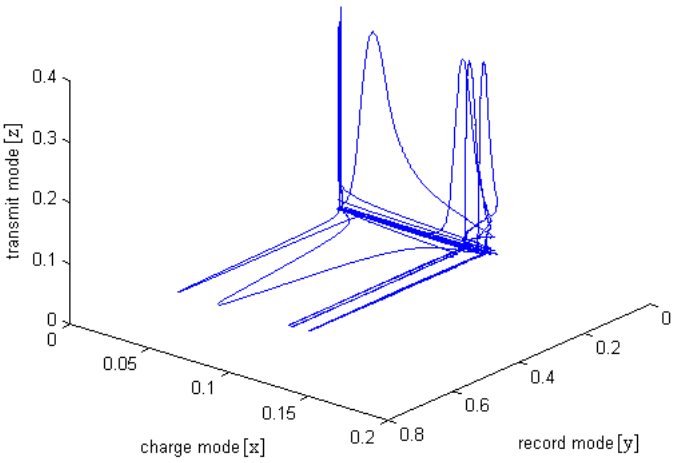

Figure 5.13: Phase plane for Lotka-Volterra model

Figure 5.13 looks significantly different to the result obtained previously in figure 5.11. This is a consequence of the state deficits varying to a wider degree during this simulation and no one dominating deficit causing the respective action. In this simulation the magnitude of the record (y) and transmit (z) modes become the triggering actions at different magnitudes of the charge (x) mode. The phase-plane plot in figure 5.13 therefore looks less like the cycle seen in figure 5.11, but the same stability properties are maintained

leading to a controllable system.

Applying this controller, the results for the internal state of the system give an appreciation of the performance of the controller. The scenario is an equatorial low Earth orbit (500 km) with a target for data recording and ground station for data transmission placed on opposite sides of the Earth. This scenario ensures that the satellite passes in and out of eclipse at the same time as encountering either one or neither of the target and ground station due to the rotation of the Earth.

The results for the 24 hours are given in figure 5.14 - 5.16. Figure 5.14 depicts the different actions being selected during the course of the simulation. The integer '1' indicates the charge action, '2' indicates the record action, and '3' indicates the transmit action. As expected different actions are selected based on the state deficits of the system. There also exists a fourth action denoted 'drift'(4) where none of the other actions can be realised. Figure 5.15 depicts the data stored in the memory of the satellite. A cycle is developed wherein the satellite uploads data until the memory is full when the target is in sight, and then downloads the data until the memory of the satellite is empty when the ground station is in sight. Figure 5.16 depicts the energy in the satellite battery. Here the satellite reaches neither the upper or lower limits of the battery and so can be taken as acceptable.

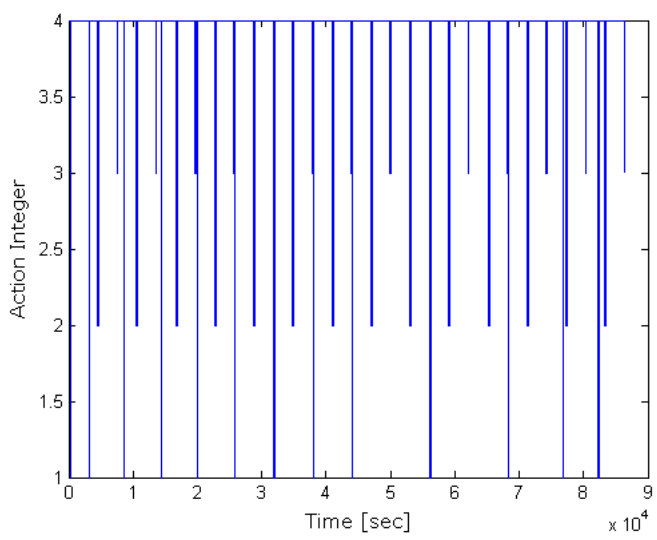

Figure 5.14: Action integers for Lotka-Volterra model

Discussion of Lotka-Volterra Controller

The results show that the control methodology is sufficient for the tasking demanded on the satellite. There are however a number of issues with this controller that make it undesirable and may cause problems if the tasking requirements were more complex or demanding. The main problem for this controller is the time dependance of the cycle.

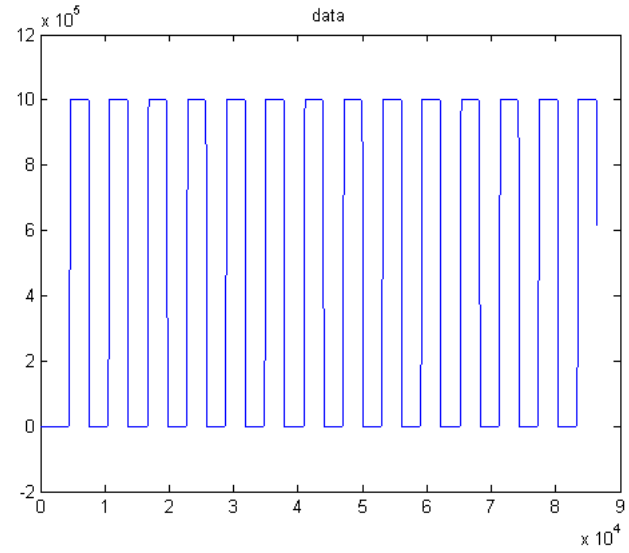

Figure 5.15: Data in memory for Lotka-Volterra model

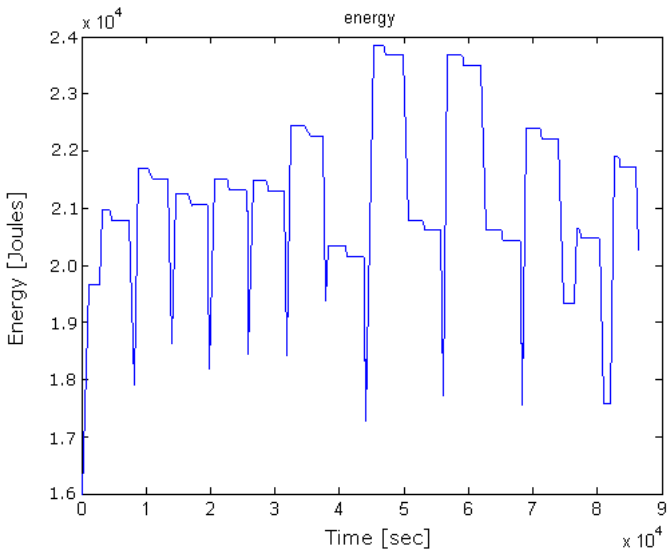

Figure 5.16: Energy in battery for Lotka-Volterra model

This problem is partially overcome by the introduction of thresholding but this still does not address the fundamental characteristic of the cycle taking some time to evolve. This means that as the number of tasks demanded by the system increases the performance of the controller is likely to decrease as a consequence.

The performance of this model varies with the definition of the threshold values. In this section these values have been selected to produce a control cycle where a full upload/download memory sequence is obtained and the battery uses a significant proportion of its operational range. If a failure were to occur in the system, for example a decrease in the solar panel efficiency, then the performance of the system would be reduced, unless the threshold value for the charge (x) mode was changed. This could be included in the controller but in doing so the embedded aspect of the autonomous action selection is lost. These extra values for the thresholds would put an extra burden on the satellite memory and effectively result in a return to a conventional method of fault detection, isolation and recovery.

This controller performs well when the number of task to be performed is small and when the likelihood of encountering failures is low. Stability analysis has also shown that the desired actions are obtained as a consequence of changes in state deficits. However, if the number of tasks to be performed is increased or a fault is encountered then this controller's utility is reduced and use of a different controller, such as the controller described in chapter 6 is advised.

5.3.2 Potential Field Controller

The next control algorithm presented adopts the ideas of the artificial potential field [96] [97]. An artificial potential field can be defined by a system of ordinary differential equations (ODEs) that incorporate state deficits identical to those used in the Lotka-Volterra control method. Solving the system of ODEs, and by manipulation of this predetermined field, can then lead to a range of different control actions being selected.

Each action in this algorithm is defined by a potential field local minimum. This local minimum can be defined to be in any location of the potential field and be a function of a state deficit.

First, consider one local minimum given by an artificial potential V, defined by equation 5.33:

$$V = d_i.e^{-\lambda_i.((X-\rho_x)^2+(Y-\rho_y)^2)} \tag{5.33}$$

where d_i $(i = 1-3)$ is the state deficit, λ_i controls the diameter of the minima, X and Y are the range of artificial potential field and $[\rho_x,\rho_y]$ defines the position of the local minima. A plot of such a field and minima is shown in figure 5.17, where $d_i = 1$, $\lambda = 1$, $-2 \leq X \leq 2$, $-2 \leq Y \leq 2$ and $[\rho_x,\rho_y] = [0,0]$.

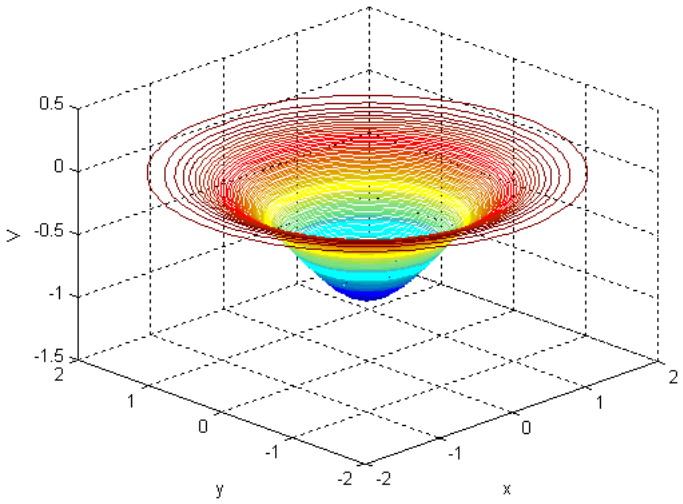

Figure 5.17: Single potential minimum

If the negative derivative of the potential is calculated with respect to each state of the potential field, then the solution to this set of differential equations evolves to [ρ_x,ρ_y]:

$$\dot{x} = -\frac{\partial V}{\partial x} \tag{5.34}$$

$$\dot{y} = -\frac{\partial V}{\partial y} \tag{5.35}$$

The process of adding multiple minima to a potential field permits the creation of a set of differential equations that can be a function of the system (satellite) deficits and physical parameters. This potential field can then be used to develop action selection control algorithms.

Single Spacecraft Model

For the single spacecraft model the same control actions as for the previous model are selected, i.e. charge battery, record, transmit and drift. This pertains to four different local minima that are located on the potential field. If one action is more important than another then it has a more central geometric location and the solver more frequently evolves to this location. For this model each of the three main tasks (charge, record, transmit) are of assumed equal importance and are equally spaced around a circle of radius 1 centered at [0,0]. The ρ_C, ρ_R and ρ_T coordinates that correspond to the local minima locations for charge, record and transmit respectively are therefore:

$$[\rho_C^x, \rho_C^y] = [1, 0]$$

$$[\rho_R^x, \rho_R^y] = [-0.5, cos(\frac{\pi}{6})]$$

$$[\rho_T^x, \rho_T^y] = [-0.5, -cos(\frac{\pi}{6})]$$

$$[\rho_D^x, \rho_D^y] = [0, 0]$$

where, ρ_D is the fourth drift state that the system can choose.

With reference to equation 5.33, the three state deficits that correspond to the three main actions d_{ca}, d_{ra} and d_{ta} (charge, record, transmit respectively) are defined by equations 5.30 - 5.32. The values of λ_i determine the radius of the local minima and in doing so also the gradient of the slope of each minima. To equalise the system the three parameters are set to $\lambda_c = \lambda_r = \lambda_t = 1$, again corresponding to charge, record, and transmit respectively.

Finally, values for the drift minimum are selected. The drift action will only be selected if none of the other three actions are available. To this end the drift minimum must act as an all encompassing safety-net. Here the deficit value is set to $d_d = 0.1$ so it does not influence the selection of any other action when they are available. Also, the drift minimum must cover the entire area of the potential field and to this end the value of λ_d is set to 0.5, which defines it to be much wider than the other minima with a shallower gradient.

The final potential field is therefore defined by:

$$\begin{aligned} V &= d_{ca}.e^{-\lambda_c.((X-\rho_C^x)^2+(Y-\rho_C^y)^2)} \\ &+d_{ra}.e^{-\lambda_r.((X-\rho_R^x)^2+(Y-\rho_R^y)^2)} \\ &+d_{ta}.e^{-\lambda_t.((X-\rho_T^x)^2+(Y-\rho_T^y)^2)} \\ &+d_d.e^{-\lambda_d.((X-\rho_D^x)^2+(Y-\rho_D^y)^2)} \end{aligned} \tag{5.36}$$

Examples of the potential field are given in figure 5.18. In figure 5.18(a) the deficit values are $d_{ca} = 0.9$, $d_{ra} = 0.5$, $d_{ta} = 0.1$, $d_d = 0.1$ and it can be seen that the differential equation solver will evolve to the location of the local minimum for the charge action at [0,1]. In figure 5.18(b) the deficit values are neglected as neither the Sun, record target, or transmit target can be seen, therefore the only minimum on the potential field is the drift action and the differential equation solver tends to [0,0], the location of the drift minimum.

Results for the incorporation of the control algorithm into the satellite model are shown in figures 5.19 - 5.22. The orbital parameters for this simulation are the same as those

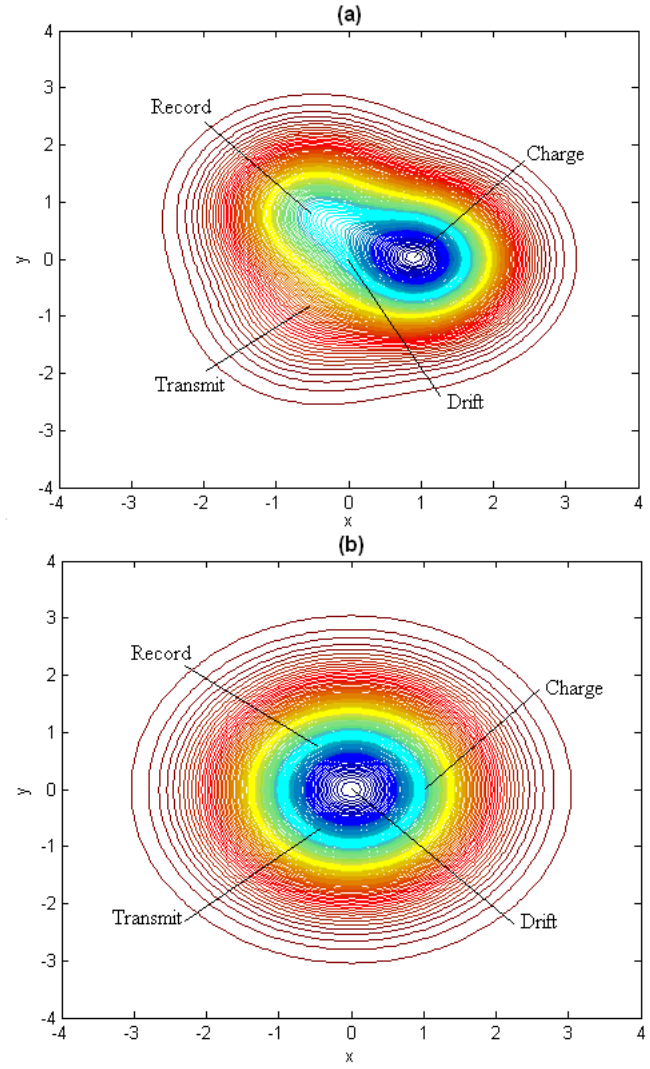

Figure 5.18: Multiple minimum in the potential field

chosen for the simulations in section 5.3.1, i.e. a spacecraft in a circular orbit 500 km above the surface of the Earth, with the ground station and target placed at $\alpha_{gs} = 0^0$, $\epsilon_{gs} = 0^0$, $\alpha_t = 0^0$ and $\epsilon_t = 180^0$. Figure 5.19 shows a plot of the phase plane during simulation. The plot shows the solver moving from the central drift state to the appropriate action locations whenever they become available. Figure 5.20 depicts the action selection process, as the satellite moves in and out of eclipse or has access to the record and transmit targets, the appropriate action can be seen to be selected. The first integer represents the charge action, the second the record action, the third represents the transmit action and the fourth denotes the drift state.

The battery charge and data stored are shown in figure 5.21 and figure 5.22 respectively. The battery can be seen to cycle through a charge-discharge state and the record and transmit actions can be seen as the satellite memory is filled and emptied.

This scenario does not put demand on the algorithm. In addition, the output of the algorithm essentially maps the deficit directly to specific actions. However, for more complex problems the use of the potential field allows a mediation between multiple competing behaviours. In the following section a fourth option will be given to the satellite where it may transfer data to another satellite in the same orbit but further ahead or behind the specific satellite under investigation in order to off-load data and so load-balance across multiple satellites.

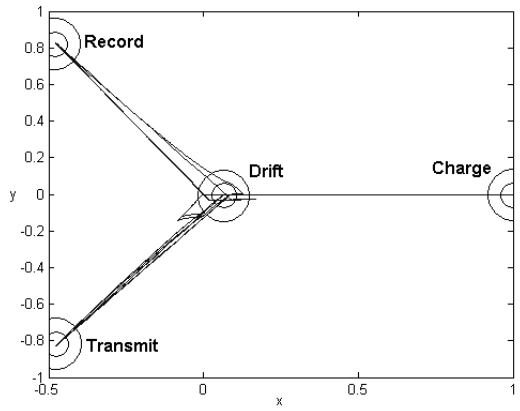

Figure 5.19: Single satellite potential field phase plane

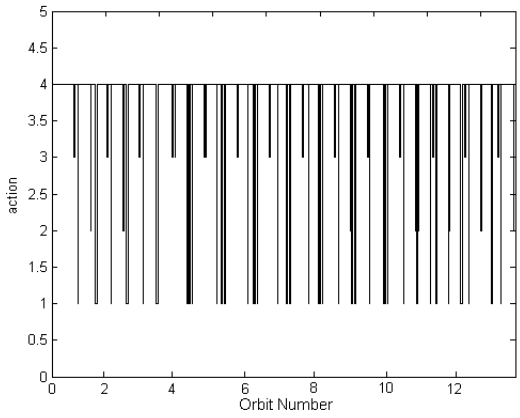

Figure 5.20: Single satellite potential field actions

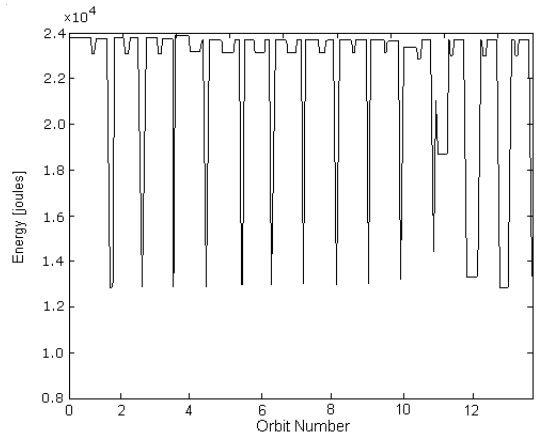

Figure 5.21: Single satellite potential field battery

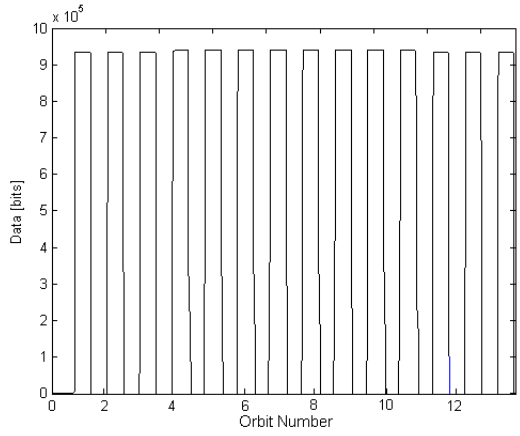

Figure 5.22: Single satellite potential field memory

Multi-Spacecraft Model

One of the main interests in micro-spacecraft is the benefit that can be gained from flying a constellation of satellites [98] [99]. A similar control algorithm to that proposed in the previous section will be implemented where the satellite has the option to transfer data to an identical adjacent satellite which can transfer the data to the ground-station if visible.

The local minimum for the potential field is constructed using equation 5.33. Here there are five actions that can be selected: charge, record, transmit to ground station, transmit to a neighboring satellite, and drift. The location of the minima are as follows:

$$[\rho_C^x, \rho_C^y] = [1, 0]$$

$$[\rho_R^x, \rho_R^y] = [0.309, -0.951]$$

$$[\rho_T^x, \rho_T^y] = [0.309, 0.951]$$

$$[\rho_D^x, \rho_D^y] = [0, 0]$$

$$[\rho_{IF}^x, \rho_{IF}^y] = [-0.809, 0.588]$$

$$[\rho_{IB}^x, \rho_{IB}^y] = [-0.809, -0.588]$$

where ρ_{IF} ρ_{IB} are the inter-satellite links for the satellite forward and back respectively. Both satellites can always be seen by the primary satellite, this is guaranteed by selection of an orbital altitude of 450 km. The deficit for the inter-satellite link for the forward satellite is defined as:

$$d_{sf} = 2.mem_{pri}/(mem_{max} + mem_{pri} + mem_{sec}) \tag{5.37}$$

where mem_{pri} is the data in the memory of the primary satellite, mem_{sec} is the data in the memory of the secondary satellite and mem_{max} is the maximum data the satellite can hold. A similar relation is built for the other satellite. The potential field is now defined

as:

$$V = d_{ca}.e^{-\lambda_c.((X-\rho_C^x)^2+(Y-\rho_C^y)^2)} + d_{ra}.e^{-\lambda_r.((X-\rho_R^x)^2+(Y-\rho_R^y)^2)}$$
$$+d_{ta}.e^{-\lambda_t.((X-\rho_T^x)^2+(Y-\rho_T^y)^2)} + d_d.e^{-\lambda_d.((X-\rho_D^x)^2+(Y-\rho_D^y)^2)} \quad (5.38)$$
$$+d_{sf}.e^{-\lambda_{sf}.((X-\rho_{IF}^x)^2+(Y-\rho_{IF}^y)^2)} + d_{sb}.e^{-\lambda_{sb}.((X-\rho_{IB}^x)^2+(Y-\rho_{IB}^y)^2)}$$

An example of this potential field is given in figure 5.23. Here the state deficits are: $d_{ca} = 0.1$, $d_{ra} = 0.1$, $d_{ta} = 0.9$, $d_d = 0.1$, $d_{sf} = 0.5$, and $d_{sb} = 0.5$. In this case the satellite now has two local minimum present and the action selected will depend on the position on the potential field and the local gradients. In this example the deficit corresponding to the transmission action is larger than the deficits corresponding to the inter-satellite links so the probability that the satellite will choose to transmit to the ground is more likely but this also depends on the previous action selected, i.e. corresponding to the prior location on the potential field.

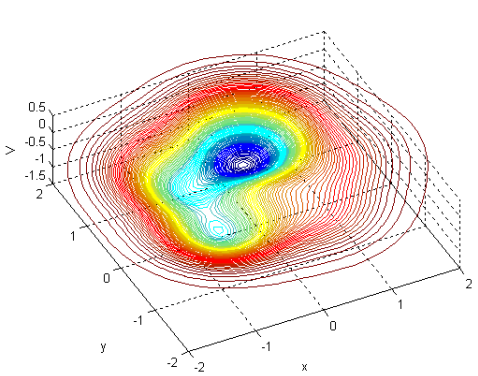

Figure 5.23: Multi-satellite potential field

The inclusion of the extra actions requires two extra spacecraft parameters to be defined. The inter-satellite data rate link is taken to be 1 kb/s and the power required to perform the inter-satellite link is 0.1 W [89].

Results for the inclusion of the enhanced control law into the satellite model are shown in figures 5.24 - 5.27. Figure 5.24 shows the phase-plane for the controller during the simulation period. The associated actions are given in figure 5.25, where the integers 1, 2, 3, 4, 5, and 6 correspond to charge, record, transmit, forward inter-satellite link, backward inter-satellite link, and drift respectively. Battery charge and discharge can be seen in figure 5.26, and the data stored in each satellite is shown in figure 5.27.

With the exception of the initialisation of the satellite internal states and the effect of the rotation of the Earth, the behaviour of each satellite is periodic. Therefore, after an initial settling period the trace each satellite leaves on the phase plane is expected to be similar. This is the result that can be seen in figure 5.24. A similar sequence of internal states for each satellite is mimicked by a similar path on the phase-plane. The differences in the phase plane plots can be attributed to the differences encountered as the model

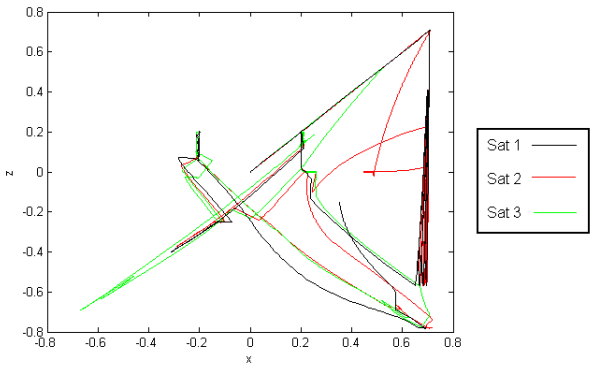

Figure 5.24: Potential field multi-satellite phase plane

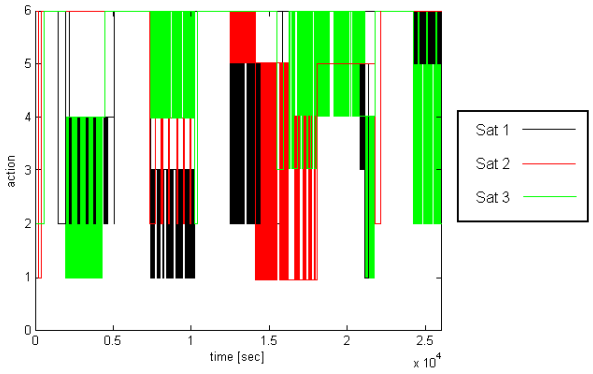

Figure 5.25: Potential field multi-satellite actions

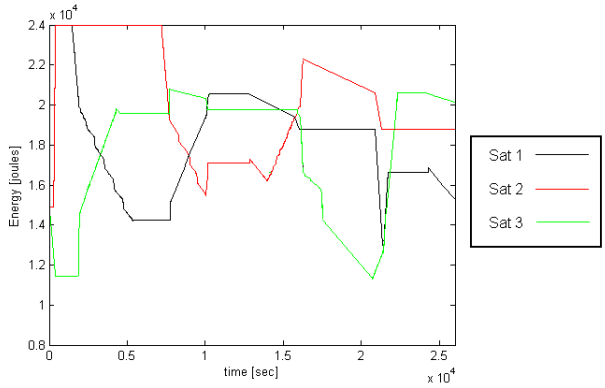

Figure 5.26: Potential field multi-satellite battery

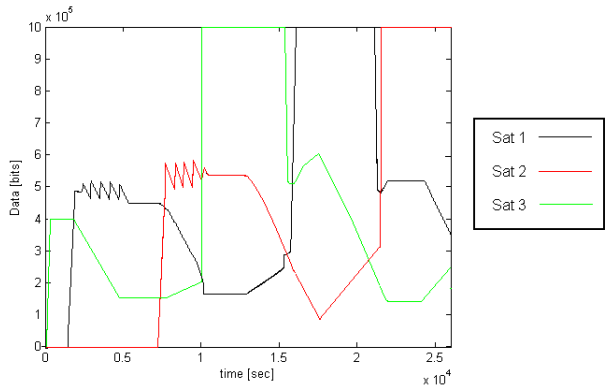

Figure 5.27: Potential field multi-satellite memory

permutates with each subsequent orbit after initialisation.

The results for the individual satellite internal data and energy show a limitation of the potential field method when the number of tasks to be performed increases. Because the number of local minima has been increased, and two (or more) of these minima can have the same depth simultaneously, the solution to equation 5.38 can evolve between these competing actions leading to a degree of indecisiveness in the action selection. For example, in the simulation defined by figures 5.24 - 5.27, for the simulation time between 7,500 s and 10,000 s there exists a continuous change in the action selected for each satellite. This period of the simulation refers to a situation where satellite 1 is switching between the charge and transmit actions, satellite 2 switched between record and interlink with satellite 3, and satellite 3 switched between drift and both passing data to satellite 1 and 2.

The control system, as a consequence of this switching, is not as efficient as with only one satellite. This can been seen from the upload/download sequence of data in figure 5.27 for each of the satellites. With comparison to figure 5.22 for a single satellite, where the data sequence consists of almost filling then emptying the data in the memory, the multi-satellite controller decides to interlink data when it is more important to continue transmitting to the ground station.

Consequently the simulation solver times increase and this results in the controller having a reduced efficiency when multiple satellites are introduced. Its utility can be assigned to the single satellite case unless some improvements are made to the controller. Improvements could be made by assessing different relative positions of the local minimums - moving away from the circular distribution to more intricate patterns. Another possibility is to redefine the inter-satellite state deficits where they take less preference over the primary functions of recording target data and/or transmitting to the ground station. A final suggestion is to introduce artificial repelling potentials to inhibit certain translations across the potential field.

5.3.3 Direct Adaptive Controller

In this section an adaptive control law is presented that provides robust action selection for a micro-spacecraft. Direct adaptive control differs from the previous control algorithms because in the controllers discussed in this chapter any control gains are constant. Here, by utilising stability properties inherent in the controller, the control gains can vary continuously as a function of the error signals based on the state deficits of the satellite defined by equations 5.16 - 5.18. The controller aims to minimise an artificial Lyapunov function defining all the dynamic states of the system. This is achieved by implementing a control signal built around the changing controller gains.

The direct adaptive control (DAC) law is a function of known system parameters while being highly robust to external disturbances. DAC is therefore demonstrated as an attractive option for the micro-spacecraft problem since fault recovery is embedded into the system.

The theory for adaptive control was defined earlier in section 2.4, and can be applied to the micro-spacecraft action selection problem. As the control law, by definition, is applicable to linear systems, all the non-linearities associated with the orbital model make the orbital model applied in the previous sections inappropriate. With this in mind a different model is created for application of the DAC controller.

It will be assumed that the control is the spacecraft pointing direction and that the spacecraft slews between three different directions corresponding to the required action (Sun pointing for battery charging etc).

To implement the control law the system state differential equations, in this case the spacecraft deficits, defined by equations 5.16 -5.18, must be a function of the controlled variable, the spacecraft pointing angle ξ, with the aim of driving all deficits to zero. To build a model capable of being controlled in this manner a linear relationship for the variation of each of these deficits is created. A visualisation of the model is shown in figure 5.28 and the linear differential equations governing the deficit relations are given by equation 5.39 and equation 5.40, for the battery deficit. The equations for the information deficit and transmission deficit are the same although centered on a different angular position as indicated in figure 5.28, and given by equations 5.42 - 5.44. Note that the relationship between the information deficit (record) and transmission deficit (transmit) implies a selection of one action incurs a change in both deficits.

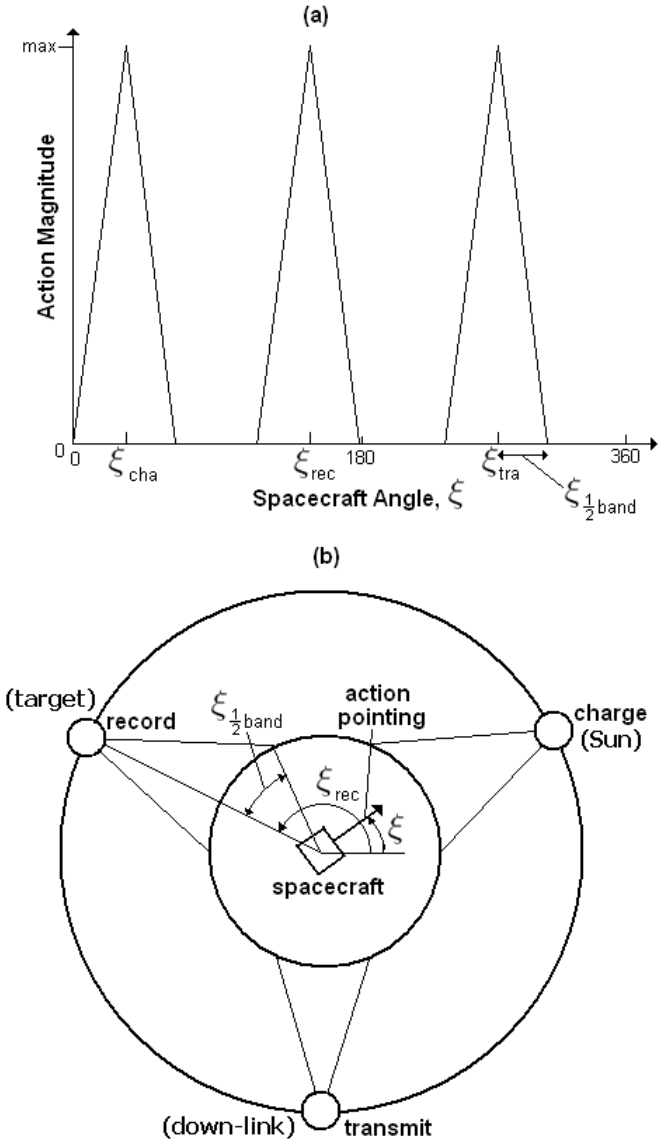

Figure 5.28: DAC action selection model (a)linear action magnitudes (b)spacecraft pointing angles

In figure 5.28(a) the locations of the maximum charge, record, and transmit actions are ξ_{cha}, ξ_{rec} and ξ_{tra}. The lines depicting the triangular regions running off from each of these three points define the linear change in the magnitude of each of the three actions that decrease from a maximum when pointing directly at either of the three actions until zero when at 30^0 ($\xi_{\frac{1}{2}band}$) in a positive or negative sense from any of the maximum locations.

The model is defined as:

$$\begin{aligned}\dot{d}_c &= \dot{e}_{max}(1 + ((\xi/(\xi_{cha} - Nb_{pos}\xi_{\frac{1}{2}band})) \\ \dot{d}_r &= 0 \\ \dot{d}_t &= 0 \end{aligned} \tag{5.39}$$

$$for \quad (\xi - \xi_{\frac{1}{2}band}) \leq \xi \leq \xi_{cha}$$

$$\begin{aligned}\dot{d}_c &= 3\dot{e}_{max}.(1 - ((\xi/(\xi_{cha} + Nb_{neg}\xi_{\frac{1}{2}band})) + (\xi_{cha}/(\xi_{cha} + Nb_{neg}\xi_{\frac{1}{2}band})))) \\ \dot{d}_r &= 0 \\ \dot{d}_t &= 0 \end{aligned} \tag{5.40}$$

$$for \quad \xi_{cha} < \xi \leq (\xi + \xi_{\frac{1}{2}band})$$

$$\begin{aligned}\dot{d}_c &= \dot{e}_{rec} \\ \dot{d}_r &= \dot{r}_{max}(1 + ((\xi/(\xi_{rec} - Ni_{pos}\xi_{\frac{1}{2}band})) - (\xi_{rec}/(\xi_{rec} - Ni_{pos}\xi_{\frac{1}{2}band})))) \\ \dot{d}_t &= \dot{r}_{max}(-1 - ((\xi/(\xi_{rec} - Ni_{pos}\xi_{\frac{1}{2}band})) - (\xi_{rec}/(\xi_{rec} - Ni_{pos}\xi_{\frac{1}{2}band})))) \end{aligned} \tag{5.41}$$

$$for \quad (\xi - \xi_{\frac{1}{2}band}) \leq \xi \leq \xi_{rec}$$

$$\begin{aligned}\dot{d}_c &= \dot{e}_{rec} \\ \dot{d}_r &= 11\dot{r}_{max}(1 - ((\xi/(\xi_{rec} + Ni_{neg}\xi_{\frac{1}{2}band})) + (\xi_{rec}/(\xi_{rec} + Ni_{neg}\xi_{\frac{1}{2}band})))) \\ \dot{d}_t &= 11\dot{r}_{max}(-1 + ((\xi/(\xi_{rec} + Ni_{neg}\xi_{\frac{1}{2}band})) + (\xi_{rec}/(\xi_{rec} + Ni_{neg}\xi_{\frac{1}{2}band})))) \end{aligned} \tag{5.42}$$

$$for \quad \xi_{rec} < \xi \leq (\xi + \xi_{\frac{1}{2}band})$$

$$\begin{aligned}\dot{d}_c &= \dot{e}_{tra} \\ \dot{d}_r &= \dot{t}_{max}(-1 - ((\xi/(\xi_{tra} - Nt_{pos}\xi_{\frac{1}{2}band})) - (\xi_{tra}/(\xi_{tra} - Nt_{pos}\xi_{\frac{1}{2}band})))) \\ \dot{d}_t &= \dot{t}_{max}(1 + ((\xi/(\xi_{tra} - Nt_{pos}\xi_{\frac{1}{2}band})) - (\xi_{tra}/(\xi_{tra} - Nt_{pos}\xi_{\frac{1}{2}band})))) \end{aligned} \tag{5.43}$$

$$for \quad (\xi - \xi_{\frac{1}{2}band}) \leq \xi \leq \xi_{tra}$$

$$\begin{aligned}\dot{d}_c &= \dot{e}_{tra} \\ \dot{d}_r &= 19\dot{t}_{max}(-1 + ((\xi/(\xi_{tra} + Nt_{neg}.\xi_{\frac{1}{2}band})) + (\xi_{tra}/(\xi_{tra} + Nt_{neg}\xi_{\frac{1}{2}band})))) \\ \dot{d}_t &= 19\dot{t}_{max}(1 - ((\xi/(\xi_{tra} + Nt_{neg}.\xi_{\frac{1}{2}band})) + (\xi_{tra}/(\xi_{tra} + Nt_{neg}\xi_{\frac{1}{2}band})))) \end{aligned} \tag{5.44}$$

$$for \quad \xi_{tra} < \xi \leq (\xi + \xi_{\frac{1}{2}band})$$

where $\dot{e}_{max}$ is the normalised maximum charge rate, $\dot{r}_{max}$ is the normalised maximum record rate and $\dot{t}_{max}$ is the normalised maximum transmission rate. $\dot{e}_{rec}$ and $\dot{e}_{tra}$ are the normalised rate of energy used in the record and transmit actions respectively. To work with the state deficits $0 \leq d_i \leq 1$ $(i = 1 - 3)$, it is necessary to normalise these parameters so the deficits change at the correct rate. The normalisation factor for the energy term is the range of battery operation $(batt_{max} - batt_{min})$ and for the data flows the normalisation factor is the maximum data that can be stored mem_{max}. These parameters were defined in table 5.1. ξ_{cha} is the angular position of the maximum charge rate, 30^0, ξ_{rec} is the angular position of the maximum record rate, 150^0, and ξ_{tra} is the angular position of the maximum transmission rate, 270^0. $\xi_{\frac{1}{2}band}$ is half the angular width of the band within which each action occurs, 30^0.

Nb_{pos}, Nb_{neg}, Ni_{pos}, Ni_{neg}, Nt_{pos} and Nt_{neg} are positive integers that define the slope of the linear dependance of each of the associated action rates - defined as 0, 2, 4, 6, 8, and 10 respectively. The three real numbers (3,11,19) are related to the positive integers defining the local positions of the beginning and end of the linear action relationships.

Altogether equations 5.39 - 5.44 define the linear model from which an angle will associate a given action selected by the control algorithm. This is an idealised model giving the control law a synthetic environment to work in. Note that when the record action is selected, for example in the region $(\xi - \xi_{\frac{1}{2}band}) \leq \xi \leq \xi_{rec}$, there is an inverse relationship between the rate of change of the record deficit d_r and the transmit deficit d_t. This is modeled by the associated negative number in the rate of change of the deficits defined in equation set 5.41.

The control objective is to bring each of the systems deficits to zero. Defining the state $\mathbf{x} = [d_c\ d_r\ d_t]^T$, the control $\mathbf{u}(t) = [\xi\ 0\ 0]^T$, and bringing together equations 5.39 - 5.44, a standard set of linear differential equations is obtained:

$$\dot{\mathbf{x}}(t) = \mathbf{A}.\mathbf{x}(t) + \mathbf{B}.\mathbf{u}(t) + \mathbf{d} \tag{5.45}$$

where $\mathbf{A}$ is zero and $\mathbf{B}$ and $\mathbf{d}$ are constructed from the relevant components from equations 5.39 - 5.44.

Definition of Controller

Similar to the control problem defined in section 4.1.3 and with reference to equations 4.16-4.18, the control problem is to assume that there exists a control gain matrix $\mathbf{K}_s$ that satisfies an asymptotically stable representation of the system $\mathbf{A}_s$. As outlined in section 4.1.3, $\mathbf{A}_s$ can be defined arbitrarily as long as there exists $\mathbf{P}$ and $\mathbf{Q}$ that satisfy the algebraic Lyapunov equation $\mathbf{A}_s^T\mathbf{P} + \mathbf{P}\mathbf{A}_s + \mathbf{Q} = 0$, where $\mathbf{P}$ is positive definite and $\mathbf{Q}$ is positive semi-definite. Therefore defining $\mathbf{A}_s$ as:

$$\mathbf{A}_s = \begin{bmatrix} -0.1 & 0 & 0 \\ 0 & -0.1 & 0 \\ 0 & 0 & -0.1 \end{bmatrix} \tag{5.46}$$

The eigenvalues of the $\mathbf{A}_s$ matrix are all negative and real representing a stable node/spiral. Two possible solutions that satisfy the Lyapunov equation are then:

$$\mathbf{Q} = \begin{bmatrix} 0.02 & 0.002 & 0.002 \\ 0.0002 & 0.0002 & 0.0002 \\ 0.0002 & 0.0002 & 0.0002 \end{bmatrix} \tag{5.47}$$

and

$$\mathbf{P} = \begin{bmatrix} 0.1 & 0.01 & 0.01 \\ 0.001 & 0.001 & 0.001 \\ 0.001 & 0.001 & 0.001 \end{bmatrix} \tag{5.48}$$

The last matrix that must be defined is the positive definite constant gain matrix that involves both $\mathbf{\Gamma}$ and $\mathbf{\Lambda}$. Similar to the process described in section 4.1.3, these matrices can be combined:

$$\begin{bmatrix} \lambda_1 & \lambda_{12} & \lambda_{13} \\ \lambda_{21} & \lambda_2 & \lambda_{23} \\ \lambda_{31} & \lambda_{32} & \lambda_3 \end{bmatrix} = (\mathbf{\Gamma}^{-1})\mathbf{\Lambda} \tag{5.49}$$

For simplicity set all non-diagonal elements of the matrix in equation 5.49 equal to zero and defining $\lambda_1 = \lambda_2 = \lambda_3 = 1$, leads to the variable control gains as:

$$\dot{K_1}(t) = -(0.1d_c^2 + 0.01d_c d_r + 0.01d_c d_t) \tag{5.50}$$

$$\dot{K_2}(t) = -(0.001d_r^2 + 0.001d_c d_r + 0.001d_r d_t) \tag{5.51}$$

$$\dot{K_3}(t) = -(0.001d_t^2 + 0.001d_c d_t + 0.001d_r d_t) \tag{5.52}$$

$$\dot{L}(t) = -(0.015d_c + 0.015d_r + 0.015d_t) + \epsilon \tag{5.53}$$

where ϵ is a degree of noise. When the system is in any of the other regions between the action selection regions noise is included in the system through the parameter L.

This leads to the control:

$$u(t) = K_1 d_c + K_2 d_r + K_3 d_t + L \tag{5.54}$$

The advantage of using this control algorithm is due to the fact it can react in real time to any unknown disturbance d. Traditional top-down autonomy is computationally intensive and requires an algorithmic approach to identify and recover from faults.

The results from the simulations of the DAC action selection controller are shown in figures 5.29 - 5.31. It can be seen that the system settles into a limit cycle behaviour with battery charge and discharge to drive the data acquisition from the target and down-link. Figure 5.29 shows the real time changing gains that define the control angle choice. Figure 5.30 shows the corresponding values of ξ that lead to the action selection. Figure 5.31 depicts the deficits of the satellite changing as different actions are selected.

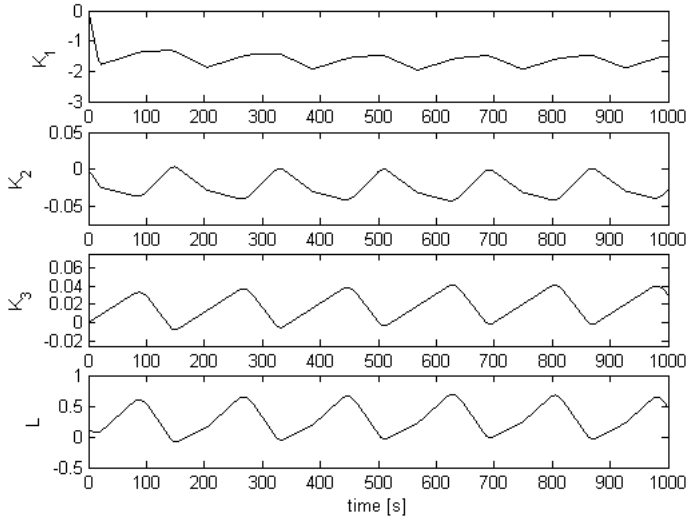

Figure 5.29: DAC gains

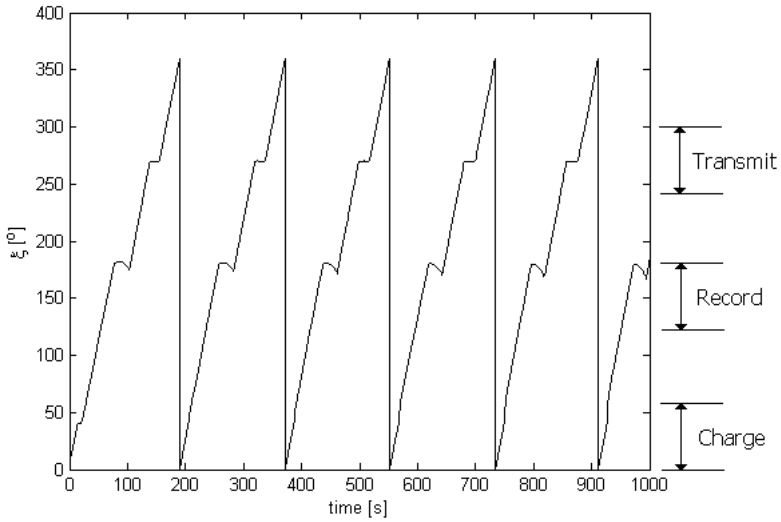

Figure 5.30: Control angle selection

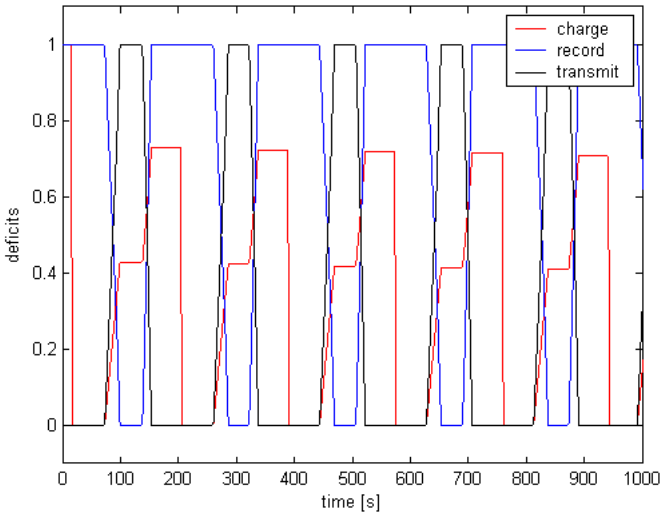

Figure 5.31: Action deficits

It can be seen that it is possible to develop a DAC algorithm for this problem. By permitting the control gains to vary continuously, as can be seen in figure 5.29, the appropriate angle is selected as a consequence, through equation 5.54. This pointing angle then defines the rate of change of the state deficits inherent in the system dynamics. Figure 5.30 shows the angle commanded by the control law, each of the horizontal lines corresponding to an angle where a specific data-related action is being performed, and the length of these lines shows the time taken to upload/download data. The final figure 5.31 shows the continuous process of trying to decrease all the state deficits to zero. As the record and transmit deficits are the inverse of one another, the perfect solution of all deficits reaching zero is impossible. The control therefore continues to cycle between the upload/download sequence while maintaining the battery charge at an acceptable level.

5.4 Fault Recovery Study

Each of the three controllers described above have the ability to select an appropriate action. This however is rarely enough in terms of robustness characteristics of a controller. An extra feature of any controller is its ability to recover from a system failure. This section investigates the response of two of the three controllers (a controller similar to the Lotka-Volterra controller will be assessed in chapter 6) to a failure in the solar array of the spacecraft. In each case, at some specified time, the power output from the solar array is halved. The results of each of these scenarios is presented and a short discussion is included.

First, the potential field (PF) controller, detailed in section 5.3.2 is subject to the solar panel failure. In this case the failure occurs after the start of the fourth orbit (20,000

seconds of simulation time). The results for the failure scenario are given in figures 5.32 and 5.33 - the dashed line in each figure depicts the failure. Figure 5.32 shows the action selections and figure 5.33 shows the battery charge during simulation.

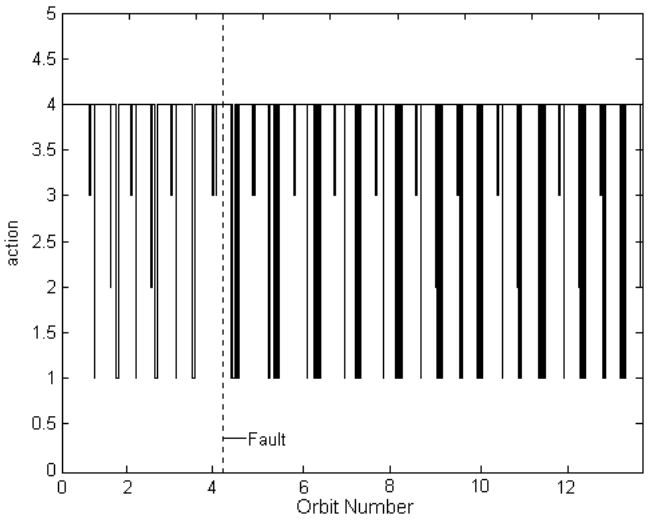

Figure 5.32: Potential field FDIR actions

Figure 5.32 shows that after the failure occurs a longer time is spent charging the battery, depicted by the longer times spent on integer 1. Figure 5.33 shows that after the fault occurs the battery cycle re-stablises in a new cycle. Both of these results show that the algorithm is capable of recovering from such a failure, albeit at a degraded level of performance.

The second case is the same failure applied to the direct adaptive control law, defined in section 5.3.3, in the linear pointing model. Due to this controller being applied to a different satellite model the failure time is different but the failure type is the same. This is simulated by halving $\dot{e}_{max}$. The response of the controller is of interest although no direct comparison can be made between the direct adaptive control algorithm and the potential field algorithm. In this simulation the failure occurs after 500 s. The results for the simulation are shown in figure 5.34 and figure 5.35, and the change in control parameters can easily be seen. Figure 5.34 shows the system deficits changing along with the values of control, ξ, that have caused the action. Figure 5.35 shows the control parameters changing to produce the relevant control angle. In both figures the dashed line represents the case without a failure, and the solid line represents the failed solar panel case.

For the direct adaptive control algorithm the failure of the solar panel is dealt with in real-time by a change in the control gains that define the control angle. Evidently the spacecraft is going to be operating at a less than optimal configuration but the spacecraft autonomously recovers from the failure without explicit fault detection. This is a demonstration of the concept of embedded autonomy.

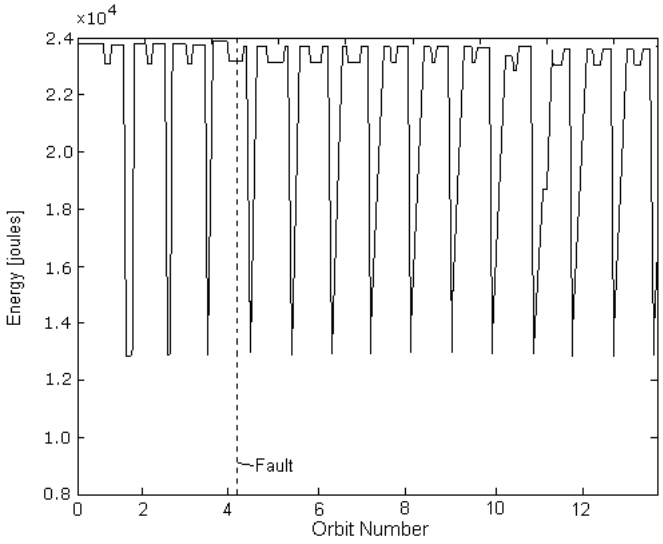

Figure 5.33: Potential field FDIR battery charge

5.4.1 Fault Recovery Evaluation

Although no direct comparison can be made between the direct adaptive controller fault and the potential field controller discussed, the fault recovery merits are worthy of note. The direct adaptive controller is the only controller to implement any change in the control algorithm as a consequence of the fault occurring. Not only does the direct adaptive control algorithm adopt a new sequence of control gains to deal with the fault as it occurs but it does not need any supplementary information about how to deal with the fault. Conventionally, after any fault has been diagnosed by a completely separate piece of software, a new set of control gains is applied to the controller.

The embedded nature of autonomy contained within the direct adaptive control algorithm means that both these conventional FDIR techniques become redundant. Fault diagnosis becomes unnecessary and as the controller gains change autonomously, the memory allocation associated with look-up tables can either be used elsewhere or removed from the system requirements. Also, there may be some time needed by the separate diagnosis software or the ground support team to diagnosis the fault. In the case where mission failure may result due to hesitant or indecisive action in fault diagnosis and resolution, the direct adaptive control algorithm provides an efficient and rapid means of recovery.

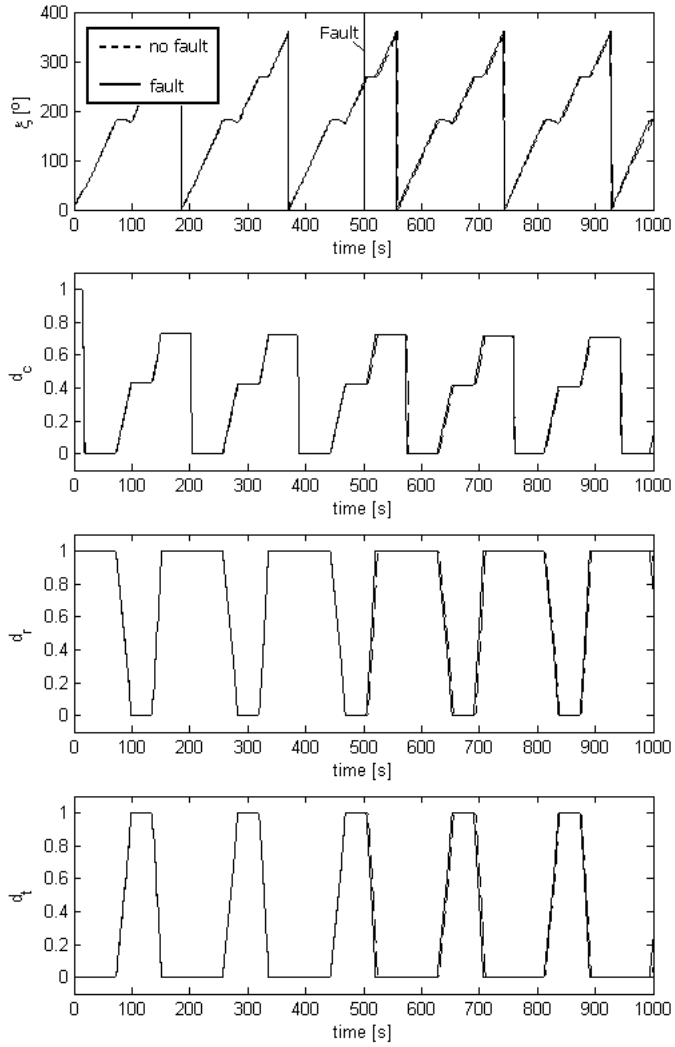

Figure 5.34: Direct adaptive control FDIR system

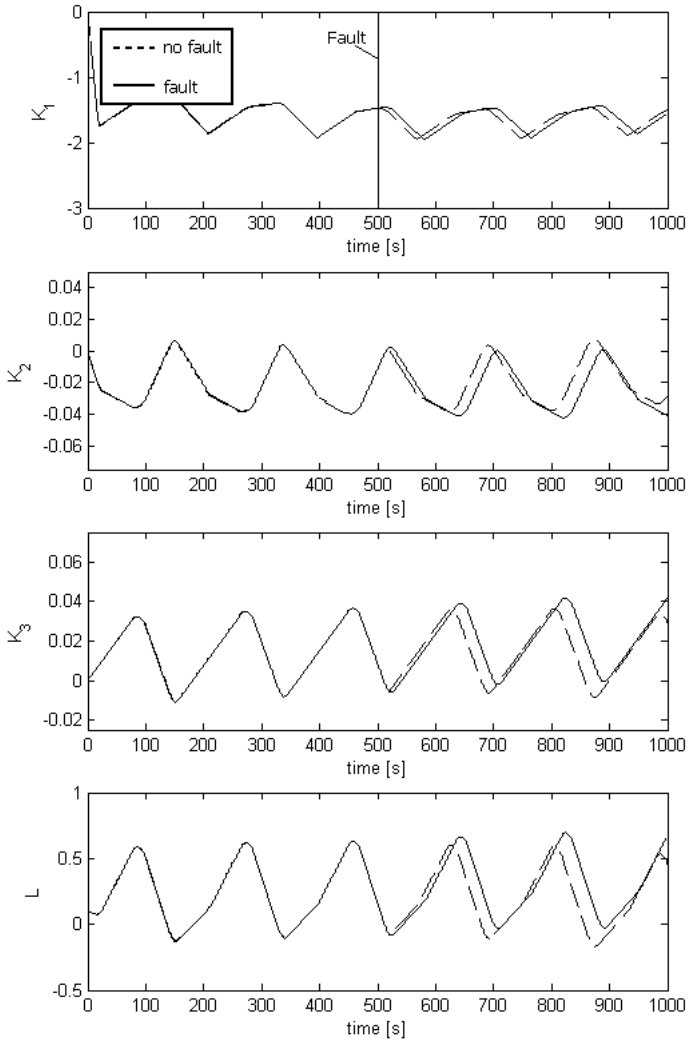

Figure 5.35: Direct adaptive control FDIR gains

Chapter 6

A Practical Application of Micro-Satellite Autonomy

6.1 Introduction

The control algorithms considered in chapter 5 are evaluated in simulation. The stability of the control laws were analysed, applied to an orbital and satellite model, and some scenarios were investigated. In this chapter another control algorithm will be developed, analysed, and simulated in the same manner, but this controller will be taken a step further. The controller defined in this chapter will be developed by an analogue electronic circuit. As the requirement of the micro-satellite controllers was to reduce computational burden, by embedding the autonomy onto a circuit board the need for a processor is completely removed. In this way mission critical functions can be hard-wired into the spacecraft, while other less critical functions can be embedded in software.

In this chapter the control algorithm will be developed and the stability analysis of the controller performed. Results from incorporating the controller into the same orbital and satellite models defined in section 4.2 are presented, alongside a fault recovery study into the response of the controller to a solar panel failure. Finally, the process involved with developing an analogue electronic representation of the control law for application in a real system is described.

The control algorithm presented here uses heteroclinic connections [100] [101], but in addition also uses bifurcation theory [102] [103]. Bifurcation theory captures changes in the qualitative or topological structure of a given family of differential equations, as discussed in chapter 2. Combining these concepts permits a more complex, yet more controllable autonomous action selection law to be defined.

6.2 Controller

The control law used for simulation and subsequently developed electronically is described in this section. Stability analysis is performed before and after any bifurcation has taken place to ensure the operation of the controller.

6.2.1 Bifurcating Heteroclinic-cycle Controller

The concept presented here is an action-selection algorithm built around the Guckenheimer-Holmes (GH) heteroclinic cycle [104]. This is an example of a heteroclinic cycle involving three equilibrium solutions of a system of ordinary differential equations proposed by Guckenheimer and Holmes, equations 6.1-6.3, with state variables (x,y,z):

$$\dot{x} = x - (Ax^2 + By^2 + Cz^2)x + \epsilon y \tag{6.1}$$

$$\dot{y} = y - (Cx^2 + Ay^2 + Bz^2)y + \epsilon z \tag{6.2}$$

$$\dot{z} = z - (Bx^2 + Cy^2 + Az^2)z + \epsilon x \tag{6.3}$$

From any initial condition set, cycling behaviour results between three equilibrium positions. It is these three equilibrium positions that will constitute the three actions of the algorithm. To generate the action selection algorithm a new form of the above equation set, revised Guckenheimer-Holmes (RGH), is proposed that permits different behaviour. The RGH equation set is shown in equations 6.4-6.6:

$$\dot{x} = \Omega x - (A_1x^2 + By^2 + Cz^2)x + \epsilon y \tag{6.4}$$

$$\dot{y} = \Omega y - (Cx^2 + A_2y^2 + Bz^2)y + \epsilon z \tag{6.5}$$

$$\dot{z} = \Omega z - (Bx^2 + Cy^2 + A_3z^2)z + \epsilon x \tag{6.6}$$

The RGH equations permit the differential equations to stop cycling by introducing three individual parameters A_i ($i = 1 - 3$) that correspond to the three state deficits, defined by equations 5.30 - 5.32. Included in these control parameters is the degree of 'availability', where the control deficit is weighted by a function of the cosine of the angle between the satellite and target/ground station location, i.e. a function which represents the visibility of the target or ground station, as discussed in section 5.2.3. Whenever one of these parameters becomes dominant the corresponding attractor does not permit the cycling to continue and an action is determined according to the location in state space. Similarly, the availability of battery charging is unity in sunlight and zero in eclipse. The inclusion of the parameter Ω increases the frequency of the cycle so there is less time lag between changes in action selection.

This type of controller has advantages. No set sequence of events must occur before an action is selected and the time lag between actions is small. Some similarities can be

drawn between this controller and a neural network. The idea of a threshold being broken for an action to occur is comparable to the threshold signals between neurons that induce an output. Aspects of neural network design could therefore be used to improve on the efficiency of the heteroclinic controller. An additional benefit is the stability of the set of differential equations can be proven by analysis.

Stability Analysis of GH Equations

The stability of the equilibria in equations 6.1-6.3 can be determined by linear stability analysis. For example setting $A = -1$, $B = -1.5$ and $C = -0.6$ an attracting heteroclinic cycle is produced. Equations 6.1-6.3 have four equilibrium points at (0,0,0), ($\sqrt{3}$,0,0), (0,$\sqrt{3}$,0), and (0,0,$\sqrt{3}$). The stability of these points is then found by finding the eigenvalues of the Jacobian, **J**, defined by equation 5.25.

The results of the analysis, and the stability of the points can be quantitatively described by computing the eigenvalues at the respective points. For point (0,0,0) all the eigenvalues are real and positive, defining an unstable node/spiral.

$$\lambda_1 = \lambda_2 = \lambda_3 = 1 \tag{6.7}$$

where λ_i $(i = 1 - 3)$ represents an eigenvalues of the Jacobian J.

For all other equilibrium points the eigenvalues are real, but for each equilibrium point one of the eigenvalues is positive while the others are negative. This is the condition necessary for the appearance of the three stable saddle nodes described above. A sketch of the state-space can be seen in figure 6.1.

$$Point(\sqrt{3}, 0, 0) : \lambda_1 = -2 \;\; \lambda_2 = -0.5 \;\; \lambda_3 = 0.45 \tag{6.8}$$

$$Point(0, \sqrt{3}, 0) : \lambda_1 = 0.45 \;\; \lambda_2 = -2 \;\; \lambda_3 = -0.5 \tag{6.9}$$

$$Point(0, 0, \sqrt{3}) : \lambda_1 = -0.5 \;\; \lambda_2 = 0.45 \;\; \lambda_3 = -2 \tag{6.10}$$

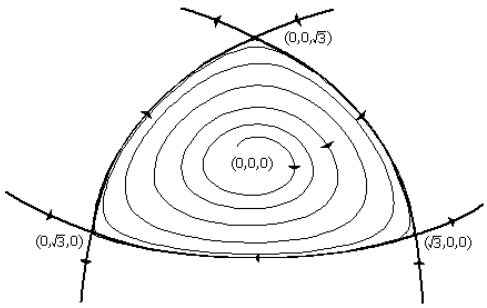

Figure 6.1: GH state-space

Consider the set of equations defined by equations 6.1-6.3, and denote the state vector as $\mathbf{X}$=$(\dot{x}, \dot{y}, \dot{z})$, the stability of the cycle will now be assessed.

Consider a sphere S defined by $x^2 + y^2 + z^2 - 3 = G(x, y, z) = 0$. It can be seen that the flow of the sphere S, falls onto the cycle as:

$$\frac{1}{2}(\nabla G).\mathbf{X} = (x^2 + y^2 + z^2) + A(x^2 + y^2 + z^2)^2 \\ -(1 + 3A)(x^2y^2 + y^2z^2 + x^2z^2)$$

and since

$$(x^2y^2 + y^2z^2 + x^2z^2) \leq \frac{1}{3}(x^2 + y^2 + z^2)^2$$

therefore

$$\frac{1}{2}(\nabla G).\mathbf{X} \geq -\frac{G}{3}(G + 3) \tag{6.11}$$

It can be deduced that for trajectories inside S that G will increase implying all trajectories beginning inside S will flow to the surface of S. Consider figure 6.2 and the above result, equation 6.11. Applying equation 6.11, if -3 $< G <$ 0, i.e. inside the sphere, then $\nabla G.\mathbf{X} > 0$ and the flow is outwards towards the sphere. If $0 < G < \infty$, i.e. outside the sphere, then $\nabla G.\mathbf{X} < 0$ and the flow is inwards towards the sphere. Finally if $G = 0$ then $\nabla G.\mathbf{X} = 0$ implies that there is flow on the surface of the sphere.

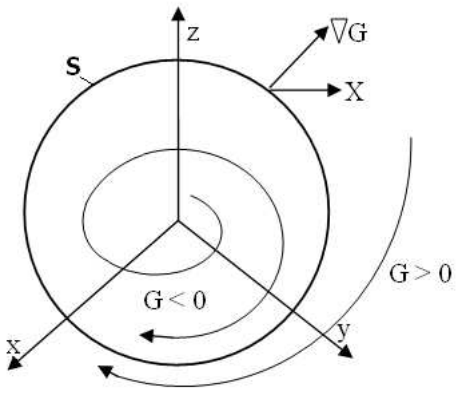

Figure 6.2: Flow to and from Sphere

The Poincare-Bendixson theorem [105] implies that all trajectories in a coordinate plane approach an equilibrium point or periodic orbit. This holds true here because movement between saddle points is restricted to one plane at a time. No periodic orbit

exists because there are no equilibria in the coordinate planes off the coordinate axes and the coordinate axes are invariant. In the (x,y) plane, the equilibria at $\pm\sqrt{(-1/A)}$ are stable (see above). Therefore, there will be a heteroclinic cycle formed from trajectories in the coordinate planes that make a circuit from (0, $\sqrt{(-1/A)}$,0) to ($\sqrt{(-1/A)}$,0,0) to (0,0, $\sqrt{(-1/A)}$) to (0, $\sqrt{(-1/A)}$,0). This cycle forms the limit cycle attractor of all trajectories lying off the invariant lines $x = \pm y = \pm z$ and the coordinate planes.

All features described are maintained for evolutions of equation set **X**. There will always be a spherical annulus [106] that attracts the trajectories and the trajectories in this region approach an equilibrium on one of the lines $x = \pm y = \pm z$, or they approach the coordinate planes. Since the coordinate planes are invariant, and the flows within the coordinate plane are structurally stable, all evolutions of equation set **X** have stable heteroclinic cycles.

Stability Analysis of RGH Equations

The stability analysis performed on the GH equations is pertinent with respect to the RGH equations. As stated above the two main differences between the equation sets are the inclusion of the control parameters, A_i (i=1-3) and a cycle frequency Ω. A_1 and A_2 correspond to the control parameters for the record and transmit actions of the system. These are the same parameters used in the previous controller in section 4.3.1, but for comparison with the GH cycle their names are changed, and defined by equations 5.30 - 5.32:

$$A_1 = f(\theta_{sat}, \alpha_{tar}, M) = d_{ra}$$

and

$$A_2 = g(\theta_{sat}, \alpha_{gs}, M) = d_{ta}$$

where θ_{sat} is the true anomaly of satellite, α_{tar} is the longitude of target, α_{gs} is the longitude of ground station and M is the data in the memory of the satellite. A_3 has a slightly different relationship as it corresponds to the charge in the satellite battery:

$$A_3 = h(\theta_{sat}, Bool_{sun}, E) = d_{ca}$$

where $Bool_{sun}$ is the boolean of the sun availability for charging the battery and E is the energy stored in the satellite battery. The positions θ_{sat}, α_{tar} and α_{gs} are defined by the dynamics of the satellite relative to a rotating Earth and are functions of parameters including the satellite orbital elements and the relative positions of the satellite, target and ground station.

The Jacobian for the eigenvalue analysis is defined in the method above by equation 5.25, although it is derived with respect to the system of equations 6.4-6.6. When values

for Ω, A_i, B and C are selected similarly to the values for the GH equations the response of the system matches that discussed above, depicted in figure 6.1.

To increase the cycling frequency/decrease the orbit periodicity of the cycle the value of Ω is increased. The first effect of this is to move the position of the stable saddle points accordingly. The new locations of the saddle points are now ($\sqrt{-\Omega/A}$,0,0), (0, $\sqrt{-\Omega/A}$,0), and (0,0,$\sqrt{-\Omega/A}$), where $A < 0$. Eigenvalue analysis at these points draws the same conclusions as previously found for equations 6.1-6.3, i.e. three stable saddle points.

Interesting results occur when A_i are varied individually. Setting the requirement that $|A_1| \gg |A_{2,3}|$ for example, changes the stability of the equilibria. By setting this condition, i.e. A_1 high, the corresponding responsive axis (x in this case) now develops a stable attracting node as opposed to the previous stable saddle. This ensures that although a cycle may start, and pass through the two other saddle points, it will terminate and tend to a stationary position at the newly developed attracting node. It should be noted that the remaining nodes (i=2,3) will maintain their position on the state-space at the respective positions of $\sqrt{\Omega/A_i}$ corresponding to the appropriate axis. The newly developed attracting node has a slightly different position on the state-space, with the corresponding non-zero value being higher than the two equivalent saddle point values.

Figure 6.3 describes this phenomenon with a schematic of three differing initial conditions leading to convergence of the trajectories to the same point on the z-axis. The three differing initial conditions display how the trajectories can join the cycle at three locations, 1, 2 or 3, and still tend to the same end point (0, 0, z_e). Here A_3 would be the dominant control parameter.

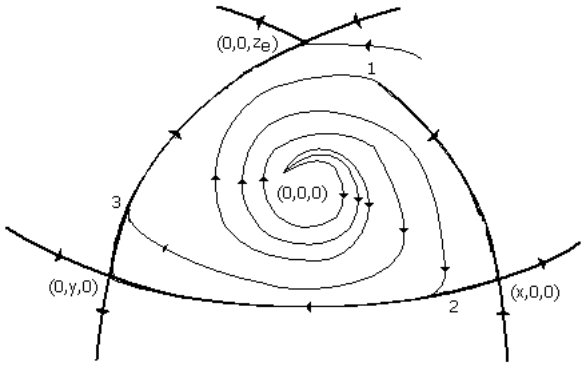

Figure 6.3: RGH state-space after bifurcation

6.2.2 Model results

In order to demonstrate the practicality of the method described, results from the model are presented. The first of these is the state space mappings. A comparison can be

drawn between the GH equation set and the RGH equation set from figure 6.4 and 6.5 respectively. In figure 6.4 the GH cycle, starting from some arbitrary point, is evident in a continual cycle moving between the three equilibrium positions. Alternatively, in the RGH equation set, with one control parameter set to dominate, figure 6.5 depicts the cycle being broken and the state space coming to rest at a particular equilibrium position.

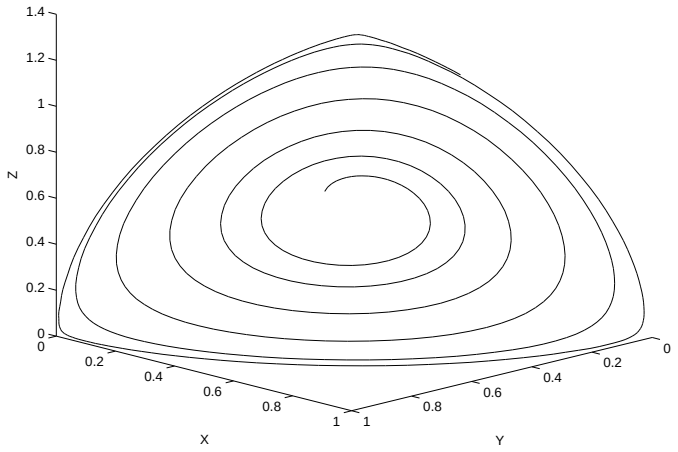

Figure 6.4: Demonstration of GH cycle

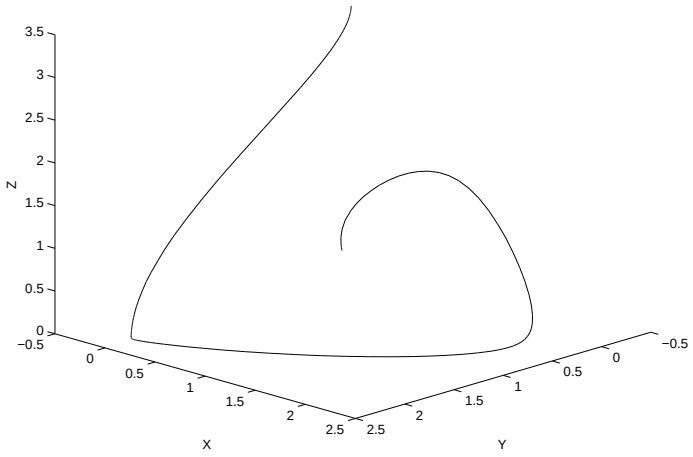

Figure 6.5: Demonstration of RGH cycle

Secondly an analysis of the action selection output is provided. The same model, as described in section 5.2, is setup to run for 24 hours. This permits the satellite to pass over both the target and the ground station several times and also pass in and out of the Earth's shadow with battery charge and discharge. Over the one day simulation the satellite's deficits vary along with the availabilities, leading to a control parameter or 'action potential' becoming dominant. When the magnitude of this potential passes

a certain threshold the model selects the appropriate action. This process can be seen in figure 6.6 and 6.7, where the different peaks in figure 6.6 correspond to the different action integers seen in figure 6.7. The first integer represents the charge action, the second integer the record action and the third integer represents the transmit action. The fourth integer denotes a state of 'drift' where none of the three actions may be realised. Figure 6.8 depicts the data in the satellite memory as the simulation progresses and figure 6.9 depicts the corresponding charge of the battery.

The scenario was chosen for simplicity and to allow the different prerequisites for each action to occur during each orbit. This can be seen in figure 6.7 when the control algorithm selects an approximate sequence of record-transmit-charge, with drift periods in between, during each orbit. This comes as a consequence of each of the respective bifurcations taking place and leading to an appropriate potential, corresponding to the respective action, becoming dominant, as is seen in figure 6.6. The efficiency of the algorithm is detailed in the memory and battery charge of the satellite, given in figure 6.8 - 6.9, the volume of memory present in the memory over a cycle is approximately five percent from being at full and empty capacity implying the control algorithm is making full use of most of the memory available, and similarly there is a large discharge depth in the battery.

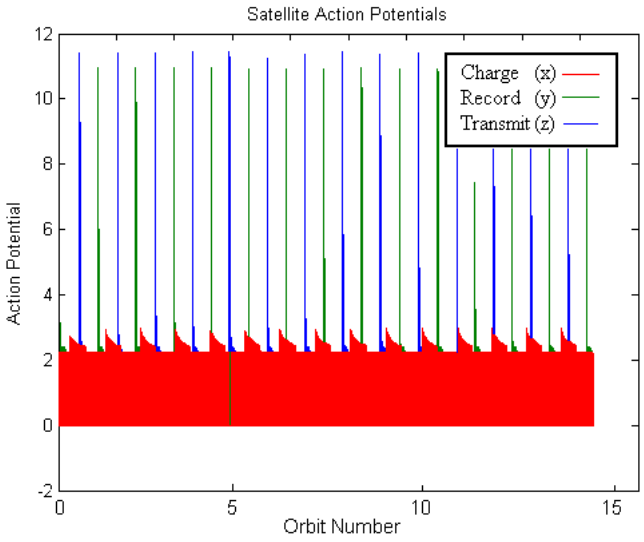

Figure 6.6: Model output for action potentials

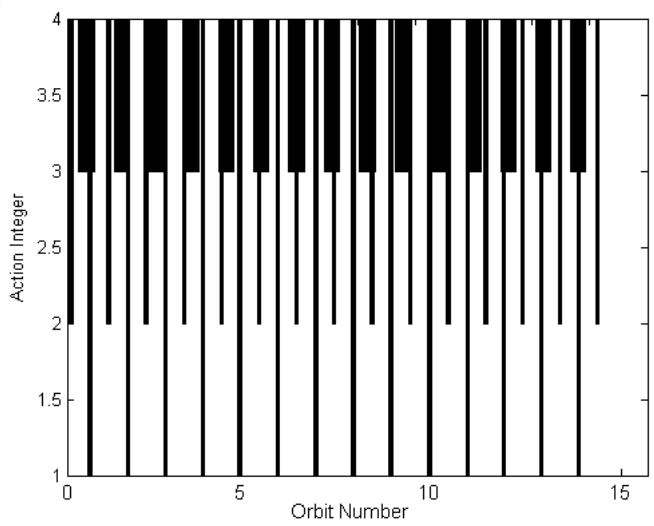

Figure 6.7: Model output for action selection

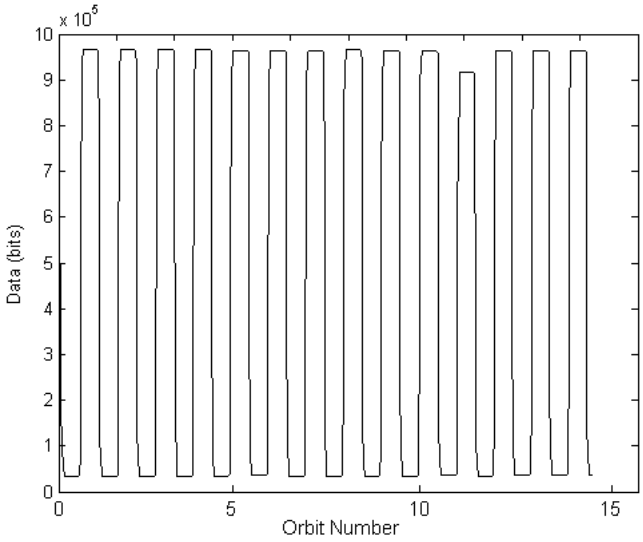

Figure 6.8: Model output for data in satellite memory

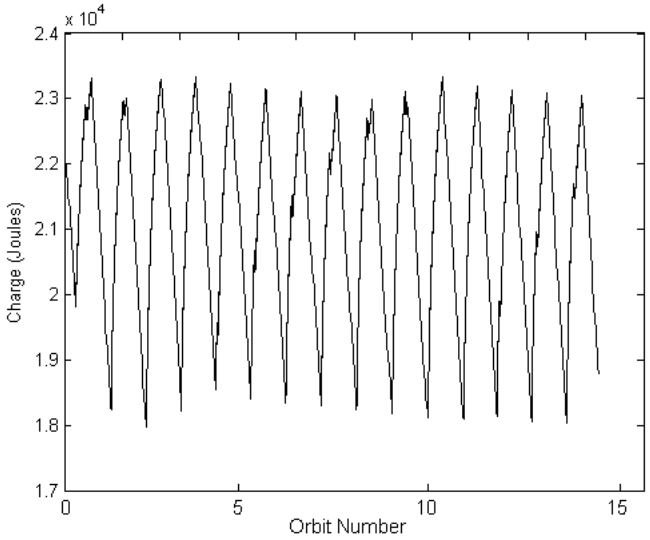

Figure 6.9: Model output for satellite battery charge

6.2.3 Fault Recovery Study

This section investigates the response of the controller to the same failure as described in section 5.4, i.e. the power output from the solar array is halved. The results for the failure are presented and a short discussion is included.

The failure is applied to the heteroclinic-cycle bifurcation controller, detailed in section 6.2. Again the failure occurs after the start of the fourth orbit. The results given in figure 6.10 and figure 6.11 show the reaction to the failure. In figure 6.10 the action selections are depicted and figure 6.11 shows the corresponding battery charge for the same simulation.

Here the failure encourages the RGH cycle to evolve at a slower rate when the charge action is selected. The solution to the bifurcating heteroclinic cycle problem therefore spends more time in the local attractor corresponding to the charge action. This is evident in both plots, as is a new lower discharge level that occurs as a result of there not being enough time out of eclipse to fully recharge the battery with the new, lower, solar power input.

For this controller the main point to note is the increased time spent on the action defined by integer 3 (charge) after the failure has occurred, as is evident when analysing the battery charge plot. The failure here incurs a new battery cycle where a lower battery depth of discharge is realised. Eventually the battery is discharged to such an extent (orbit number 11) that the controller chooses to drift for a short period of time and then charge again to bring the battery level back to an acceptable level.

Fault Recovery Evaluation

A direct comparison between the fault recovery characteristics of the potential field and revised Guckenheimer-Holmes controllers can be established. For the potential field con-

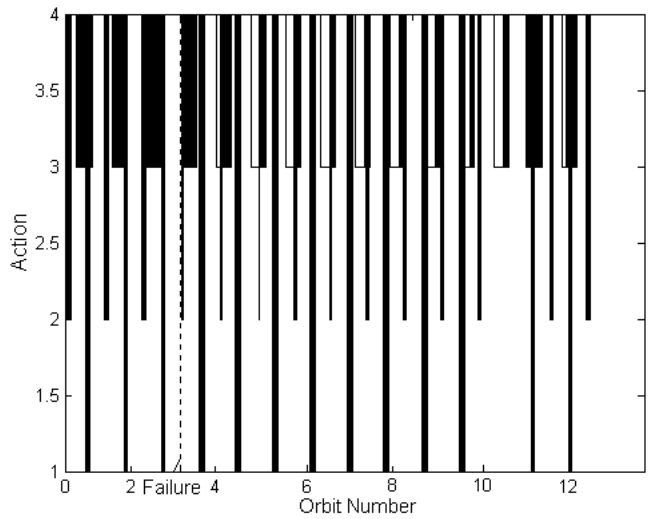

Figure 6.10: Revised Guckenheimer-Holmes fault recovery actions

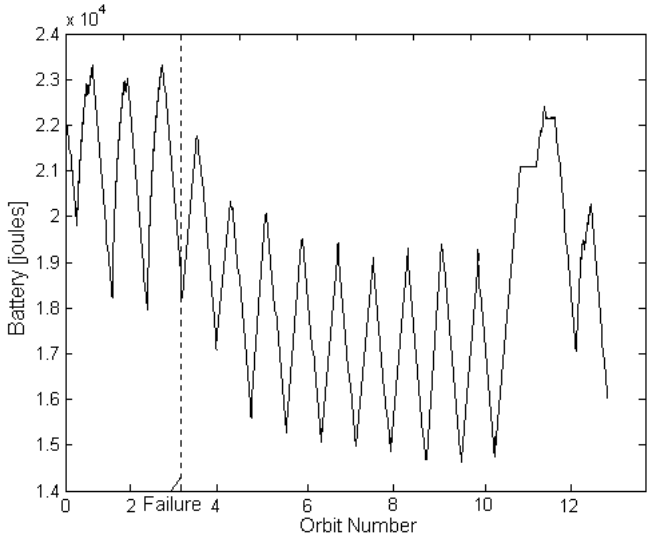

Figure 6.11: Revised Guckenheimer-Holmes fault recovery battery charge

troller, described in section 5.3.2, when the fault is encountered the only qualitative change in the evolution of the potential field is the rate at which the local minimum corresponding to the charge action changes. The controller parameters defining the shape and position of the local minimum on the potential field could be changed to offset the effects of the failure but the evolution of the minimum always remains the same.

As a consequence, the rate of battery charge in figure 5.33 can be seen to change but the depth of discharge of the battery does not. This is a positive aspect of the controller as although a fault is encountered and performance is reduced, the safe operation of the battery is not jeopardised.

For the revised Guckenheimer-Holmes controller there is one performance parameter permanently associated with the algorithm. This is the action threshold that must be broken in any particular action potential before the action is performed. The threshold that leads to the charge action being selected is therefore only crossed after a greater degree of discharge occurring in the satellite. This is evident when analysing figure 6.11 where after the failure occurs a new lower discharge depth is realised in any subsequent charge cycle.

For this simulation the maximum discharge depth is not a problem as it lies within the operational boundaries of the satellite. If a satellite system had a more constrained battery range this would be a problem where too much energy could be expended and the satellite would be unable to recover.

6.3 Analogue Electronic Circuit Equivalent

The analogue electronic circuit developed in this section has the aim of providing a cheap, robust method of autonomous control for action selection on-board a micro-satellite. By synthetically reproducing the artificial potentials present in a set of mathematical differential equations with the electronic voltage in an analogue electronic circuit the need for a processor is removed. Here the method for developing such an electronic circuit will be derived from the original differential equations.

6.3.1 Matlab Baseline Model

It is necessary to define a baseline for comparison between the mathematical model described here and the electronic model that follows. Due to the fact the electronic model will not be incorporated into an orbital model it is necessary that the mathematical baseline model equivalent is similarly not driven using an orbital model. Here, the satellite state control parameters are artificially changed so the same control parameters can be incorporated into the electronic model.

To produce the bifurcating phenomenon described in section 6.2, it is required that one control parameter be dominant at specified instants. The time evolution for the three control gains is shown in figure 6.12. Each control parameter changes with time but only

one is ever dominant. This has the effect on the transient response of the system potentials as shown in figure 6.13. When one parameter dominates the bifurcation can be seen and the respective potential dominates.

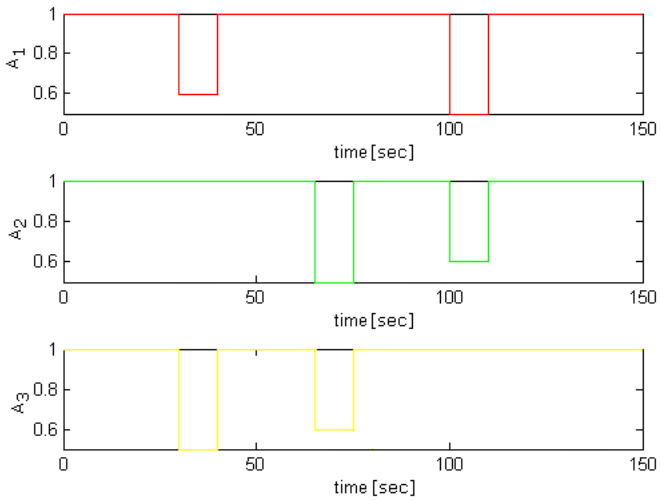

Figure 6.12: Artificial control parameters

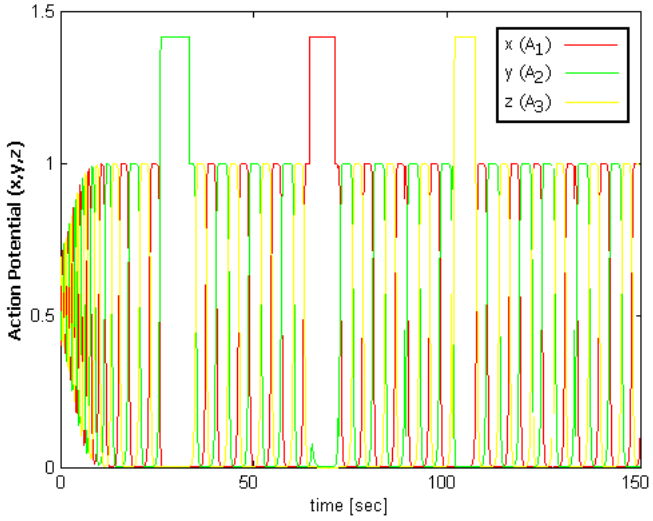

Figure 6.13: Transient response of mathematical model

6.3.2 Electronic PSPICE Model

Many mathematical operations can be synthesized in analogue electronic circuitry. For the majority of these mathematical operations the electronic 'work-horse' is the operational amplifier. Operational amplifiers (op-amp) have been used extensively through the history of space systems [107] [108], some have space grade qualifications including OP484 [109], and it is these Op-Amps that will be assumed for this controller.

PSPICE (Personal Simulation Program with Integrated Circuit Emphasis) is analog circuit and digital logic simulation software, and calculates circuit voltages and currents much like how Matlab calculates mathematical procedures. What follows here are the circuit diagrams in PSPICE containing the voltage equivalents to the action potentials defined by the revised Guckenheimer-Holmes control law, given by equations 6.4-6.6. Subsequently the circuits will be driven by the same state deficits that were used in section 6.3.1 to drive the Matlab model and the results are compared.

Revised Guckenheimer-Holmes model

The requirement is to build an electronic voltage equivalent of the Guckenheimer-Holmes equations stated in equations 6.1-6.3. To accomplish this a general understanding of electronic engineering principles alongside the use of the mathematical operators available must be implemented. The final circuit constructed must represent a printable circuit board (PCB) that can be fabricated and integrated with a micro-satellite.

The procedure adopted for constructing such a PSPICE model will now be described. Reference will be made to figure 6.14, which depicts the final model for the equivalent 'x' voltage potential. Due to the equations for the 'y' and 'z' voltage potentials being almost identical, only the 'x' circuit will be described here.

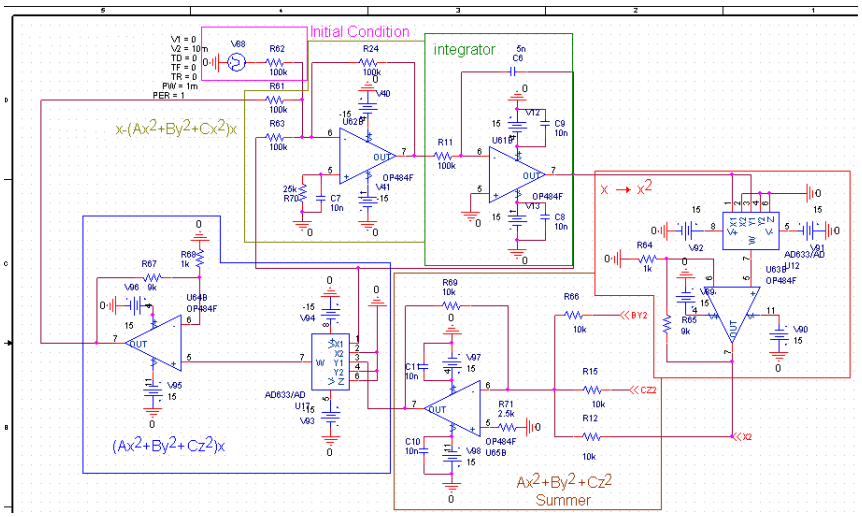

Figure 6.14: Guckenheimer-Holmes PSPICE model

As with any time dependant model the best place to start is with the integrator, this is depicted by the green box section in figure 6.14. Using the op-amp configuration defining the integrator function and the voltage relationship defined by equation:

$$i = C\frac{dV}{dt} \tag{6.12}$$

where i is the current in the op-amp, C is the value of capacitance of the negative feedback capacitor, and the $\frac{dV}{dt}$ fraction is a calculus expression representing the rate of voltage

change over time, the output of the op-amp is the integral of the op-amp input.

The next sequential step in the Guckenheimer-Holmes loop is to square the voltage potential that is the output of the op-amp integrator. This operation is seen in figure 6.14 in the red box. Here the AD633 analogue multiplier [110] is used, where the voltage operation is described by:

$$V_{out} = \frac{(X_1 - X_2)(Y_1 - Y_2)}{10} + Z \tag{6.13}$$

where X_1, X_2, Y_1, Y_2 and Z are all input voltages.

The circuit under construction takes its input for X_1 and Y_1 to be the voltage that is the output from the integrator. With reference to equation 6.13, it is necessary to boost the output voltage from the analogue multiplier by a factor of 10 to maintain the correct relative relationship for the rest of the circuit. This gain of 10 is performed by the subsequent op-amp depicted in the red box after the analogue multiplier in figure 6.14.

It is at this point in each of the three circuits, corresponding to the voltage potentials (x, y and z), that the squared voltage potentials are multiplied by three fixed gains: A, B and C (see equations 6.1-6.3). To retain a degree of easy visualisation these operations are performed on a separate PSPICE diagram, although in reality they would exist on the same circuit board. Information is passed freely in PSPICE between different circuit diagrams and the points where the flow is passed between diagrams are indicated by either the '<<' or '>>' symbols. This can easily be seen in figures 6.14 and 6.15. Figure 6.15 depicts these three gains being performed on each of the three voltage flows. The three gained voltages are then feed back into the main circuits as required.

The three gained voltage components from each of the three voltage potentials must now be added together. This is conducted through an op-amp summing configuration, indicated by the brown box in figure 6.14. It should be noted at this point that the format for successfully summing the three components requires that the magnitude of the resistor in the op-amp negative feedback loop be equal to the magnitude for each of the input parallel voltage paths, in this case the magnitude for the four resistors is 10 kΩ.

Next the output from the summing op-amp (brown box - figure 6.14) is to be multiplied by the linear x potential, which is the output from the original integrator op-amp configuration. Due to the fact that voltages in parallel have the same magnitude (see Kirkchoff's laws [111]) a direct circuit potential voltage line can be drawn from the output of the integrator op-amp and used as one of the inputs to the analogue multiplier depicted by the blue box in figure 6.14. The linear x potential is therefore multiplied by the output from the summing block. Finally, this voltage must be amplified by a factor of 10 to maintain the correct magnitude of the flow (see equation 6.13).

The last stage in constructing the looped voltage flow is to sum the output from the previous analogue multiplier/gain combination in the blue box with another linear x component of voltage. This is depicted by the hazel coloured box in figure 6.14. Again

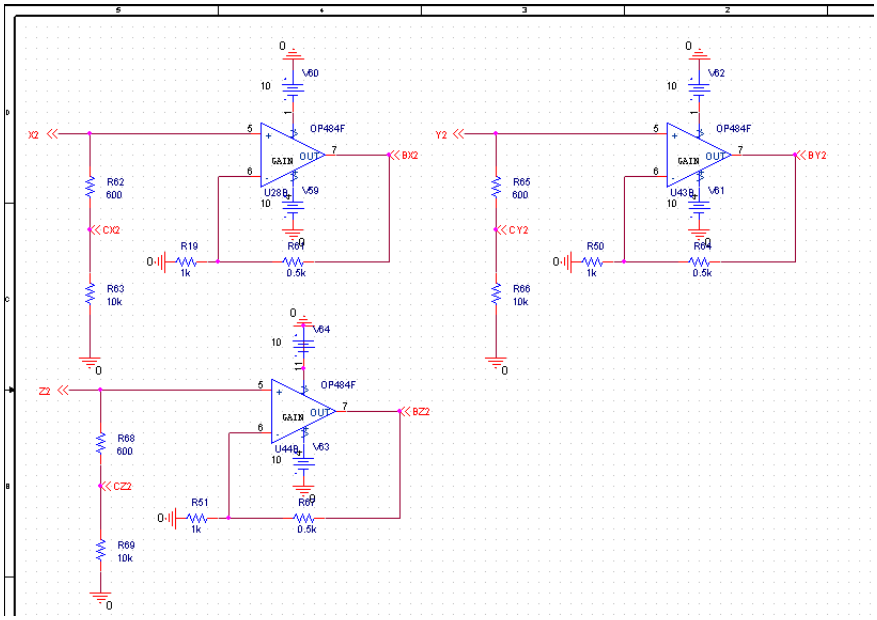

Figure 6.15: Voltage flow gains

the negative feedback voltage must be equal in magnitude to each of the parallel input voltages and this is set at 100 kΩ.

With reference to figure 6.14 (and the pink box) there is one final aspect of the circuit that must be described. Any set of differential equations must have initial conditions. Mathematically this is easy to specify but for this voltage loop the same implications must be addressed by another means. The initial conditions for the voltage loop are incorporated through the inclusion of another parallel part to the summing op-amp (hazel box, figure 6.14). Here an instantaneously desirable magnitude of voltage is pulsed into the circuit through the summer. The initial conditions for the x, y and z voltages are 10 mV, 20 mV and 30 mV respectively, a depiction of this is given in figure 6.16.

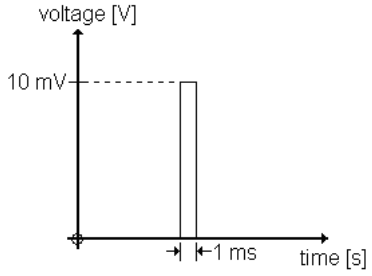

Figure 6.16: Initial condition pulse

The above described method of constructing an electronic equivalent of the Guckenheimer-Holmes cycle can now be put through a transient response and the time response of each of the voltages measured. A simulation depicting the evolution of the voltages in the circuit over a period of 100 ms is shown in figure 6.17. The cycle can be seen in the three voltage

responses with only one voltage becoming dominant at any one time. After a settling period initiated by the initial condition the cycle can be seen to relax into a heteroclinic cycle after a period of approximately 50 ms. Quantitatively the magnitude of the upper and lower bounds of the electronic Guckenheimer-Holmes cycle are not the same as is seen in the mathematical equivalent but all the qualitative dynamic phenomenon are present and this will be re-enforced when we look into the revised Guckenheimer-Holmes cycle.

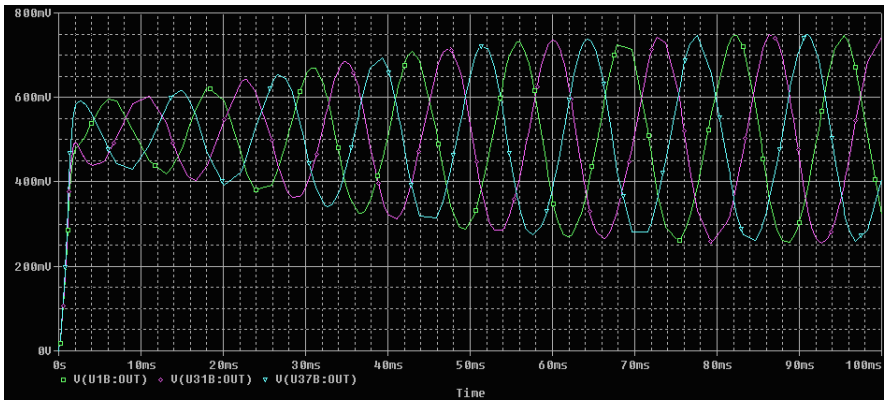

Figure 6.17: Electronic Guckenheimer-Holmes cycle

A few changes to the circuit diagram of figure 6.14 must be made in order to develop the revised Guckenheimer-Holmes cycle. These changes are described here and references to figure 6.18 are made during this process. Figure 6.18 appears similar to figure 6.14, however, there is one important addition.

With reference to equations 6.4-6.6, there are two new features of the revised Guckenheimer-Holmes equations that must be implemented in the new revised Guckenheimer-Holmes circuit diagram. The first of these is the cyclic permutation frequency Ω. It is possible to include this parameter in the circuit, with a gain op-amp added to the linear potential feedback component, but it is not necessary in this case. Due to the magnitude of the reference voltages being in mV and temporal response of the op-amps being in ms, one complete heteroclinic cycle period is in the order of approximately 15 ms. This is within the range of response necessary for the scenario under investigation so the gain term is omitted.

The important inclusion to the circuit is the addition of the control parameters A_i $(i = 1-3)$. This is shown in the green box in figure 6.18, where the squared potential term is multiplied by a new input voltage which represents the respective control parameter. The three control parameters represent the three state deficits of the spacecraft system and are given in equations 5.16-5.18.

In the real satellite system these control signal parameters come from the satellite battery, in the form of the voltage potential that represents the charge action selection. For the record and transmit actions the signal would come from the satellite memory bus. These signals would be digital in nature due to the digital nature of the modern components

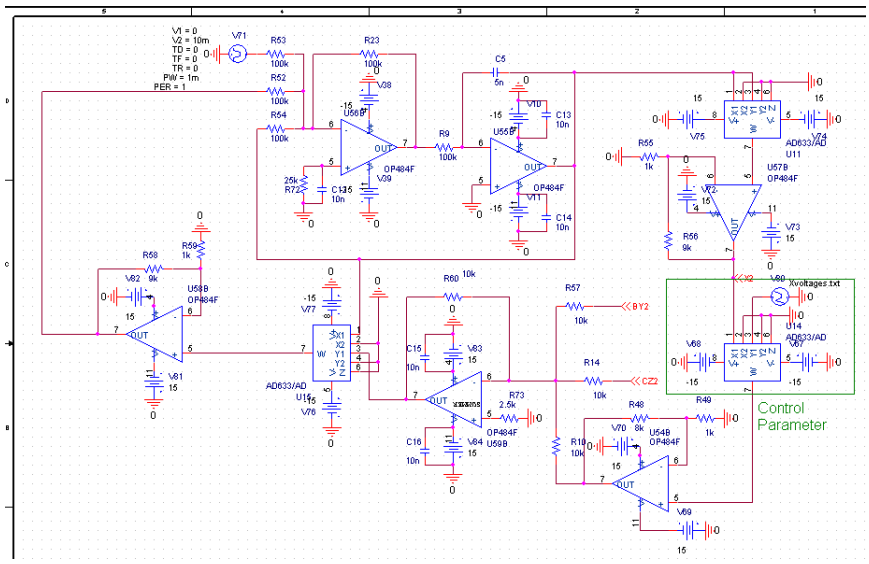

Figure 6.18: Revised Guckenheimer-Holmes PSPICE Model

that compose the battery and memory. It would therefore be necessary to convert these signals to analogue for use in this circuit. Fortunately such digital to analogue converters exist and a typical converter converts the abstract digital numbers into a sequence of impulses that are then processed by a reconstruction filter using some form of interpolation to fill in data between the impulses.

For this circuit the process is assumed to already have been performed and the input control parameter voltages that represent the parameters A_i are introduced synthetically. A similar input to that described by figure 6.12 is created only this time the time scales are different where each control parameter becomes dominant at 100 ms, 200 ms and 300 ms for the x, y and z potentials respectively.

A plot of the resulting voltage potentials can be seen in figure 6.19. After an initial brief period of stabilisation of approximately 25 ms the revised Guckenheimer-Holmes cycle settles into the heteroclinic cycle. At the times of 100 ms, 200 ms and 300 ms each of the control parameters corresponding to the x, y and z voltage potentials respectively are artificially held dominant for 20 ms. For each of these three 20 ms time periods the response of the system is a definite change in the qualitative structure of the electronic voltages. A bifurcation takes place in the system dynamics and the relative voltage potential can be seen to rise. This is the phenomenon that was required for the circuit to produce a desired action selection, and is evident in figure 6.19.

6.4 Conclusions

The aim of this chapter was to develop an electronic equivalent to a mathematical control law. By de-constructing the mathematical operations necessary into their individual parts and replacing them with electronic components capable of exactly the same principles such

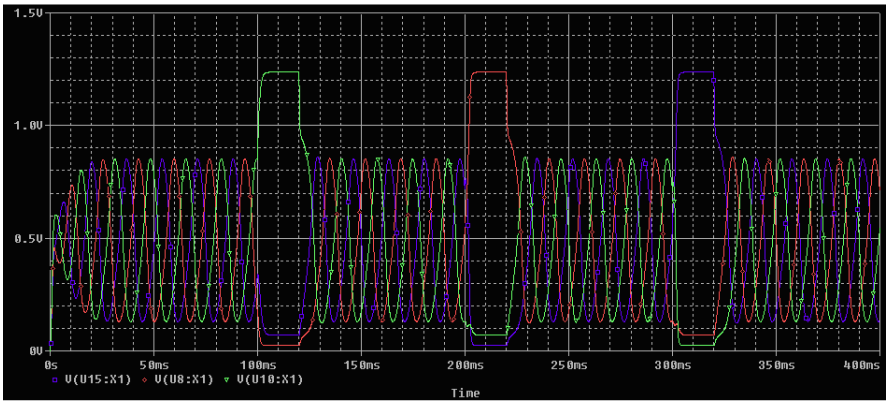

Figure 6.19: Electronic revised Guckenheimer-Holmes cycle

a system is possible.

It has been demonstrated here that analogue electronic components maintain robust useful characteristics in control law application. In this example they have been selected due to their space-qualification grading and known reliability. A single op-amp when manufactured in mass can be low in cost, providing all the dynamical properties necessary to develop an autonomous control algorithm. An expensive, radiation hardened, processor can therefore be by-passed.

Adversely there is the fact that an analogue electronic circuit is not re-programmable. Once an analogue circuit is in orbit doing its prescribed task it can never be changed to a different task. Also, the mating between analogue and digital components is not always sufficient for complex tasks. If the mission requirement is for low computational overhead, performing relatively simple tasks, then this method has been shown to be of interest. Again, it can be envisaged that mission critical tasks would be embedded in analogue electronics, hard-wired for robustness, while other overlayed, embedded in software.

Chapter 7

Conclusions

7.1 Remarks on Embedded Autonomy for a Deep Space Asteroid Lander

In chapters 3 and 4 one of the two main challenges of the thesis was presented, embedded autonomy for a deep space mission. Attitude and translational control of an asteroid lander considered. Here three differing controller types (phase plane, potential function, and direct adaptive control) were implemented into a model and the comparative merits of the controllers analysed. A range of fault recovery studies were conducted to highlight the benefits of the adaptive algorithm. The direct adaptive controller was also implemented into a separate study where the controller had to control the translational movement of the lander during a sampling process on the surface of the asteroid.

All of the control algorithms used in these chapters (phase plane, potential function and direct adaptive control) are deemed autonomous because they do not require any input during operation. One of these controller types (phase plane) also has extensive industrial heritage. The aim was to introduce newer autonomous control concepts and highlight the advantages that can be gained from appropriate selection and implementation of these controllers.

The first outcome from the asteroid lander study was a quantitative baseline comparison study, between the controller types, of the impulse required to correct an initial attitude error on the spacecraft body while simultaneously having to deal with the lateral translational effects associated with the rotation of the asteroid. It was found that the adaptive controller is capable of substantial savings in impulse over the other types of controllers. This is due to the degree to which the adaptive controller can self-optimise to the descent scenario in question. The consequence of this optimisation becomes apparent however when the transient settling time of the controllers is assessed. The reason that the adaptive controller could decrease the impulse required to alleviate the attitude error was because the settling time for the scenario was increased by a similar margin. In the case of the lander problem this is not a significant problem as the micro-gravity conditions

leave a longer descent time during which the control action is implemented. Of course this may not always be a desired effect but the ability to tune the controller to a given situation gives the engineer using the adaptive control technique the choice during the controller design phase.

Subsequently a range of fault recovery studies were conducted on the controllers. These took the form of artificially failing one of the lander legs and also failing one of the thrusters used for control. The first outcome of these studies was the positive ability of all the controllers to react and return the lander rotational and translational errors to an acceptable level. The degree to which the individual controllers could achieve this control is the topic of interest.

Regarding the magnitudes of the impulse required to compensate for the lander leg failures, the adaptive controller uses the least impulse but the degree of saving, with respect to the non-failure cases, is much less. It is only when the maximum attitude errors are investigated that the full advantage of the adaptive controller becomes apparent. If the sensors on-board the spacecraft are taken to have the usual operational range requirements, then for the radar altimetry (for example) to operate efficiently the underside of the lander must be pointing towards the surface of the asteroid. The only controller capable of maintaining this maximum angle to a reasonable degree is the adaptive controller. The reason the adaptive controller is capable of such a degree of controllability lies in the adaptive nature of the controller itself. Control gains driving the controller change in real time with respect to the failure as it occurs unlike the other controllers where the gains are fixed. This illustrates the concept of embedded autonomy which can replace traditional FDIR approaches in mission critical applications such as safe landing on an unknown surface.

Traditional methods of FDIR do allow for controllers to change their control gains. This is done through fault diagnosis where the type of fault incurred would be defined and then the spacecraft would react by changing the control gains after consultation of a 'look-up' table. The problem here lies in the fact that most probably the time taken to diagnose the problem then implement the new control gains may be too long, particularly for the lander problem. In this time frame the spacecraft may have reached an unacceptable level of failure that may be impossible to recover from. Also, if the diagnosis algorithms and look-up tables are stored onboard the spacecraft this requires more memory allocation to the process than may be available.

In the case of the thruster failure study a similar result to the baseline attitude control problem is found. The adaptive controller uses less impulse but takes a comparatively longer time to settle to the required attitude. Here a negative effect of the adaptive controller results - due to the longer settling time associated with the adaptive controller and the fact the failure could occur at any instant during the descent phase, the lander attitude may not be within sufficient error tolerances when coming in contact with the asteroid surface. This can be taken as an acceptable secondary effect however as the result

of such a dynamic situation would be a bounce and the error would be rectified during that period. The advantages gained from the adaptive control out-weight the drawbacks.

The short sampling study that was conducted again demonstrated the ability of the adaptive control to cope with uncertain environmental conditions that could be encountered. Conventionally, to deal with the problem of the reaction force produced as a consequence of drilling on the surface of the asteroid, a constant force from the thrusters would be implemented to hold the lander on the surface. Although this method would work, the impulse required for such a process far exceeds what is necessary to perform the task. By introducing the adaptive control method the impulse necessary is reduced by an order of magnitude. Although this is not large, the weight savings made could be put to use somewhere else on the lander system. Also a reduction in burn time would reduce the possibility of contamination of the sampling site.

The general conclusion that can be drawn is that by adopting a bottom-up approach to autonomous control, with the autonomous control algorithms operating at an embedded systems level with no external guidance mission, performance can be increased. By permitting the control algorithm direct real time access to the associated control torques and forces, the time taken for the spacecraft to react to any change in system or environmental state is decreased. This comes alongside a reduction in the computation necessary for implementation of the control law as no secondary FDIR algorithms are necessary with their associated look-up tables. The one drawback from implementation of the adaptive control is the additional development needed, in comparison to the more conventional algorithms, which can be developed and implemented much quicker.

7.2 Remarks on the State Deficit Approach to Micro-Satellite Action Selection

In chapters 5 and 6 the second of the two main parts of the thesis was presented. Novel embedded, mathematically verifiable control algorithms for action selection on resource limited micro-spacecraft was investigated. Three novel controllers were developed and incorporated into a generic satellite model. A fourth controller, again utilising adaptive control techniques, was developed although this controller was incorporated somewhat differently.

The first action selection algorithm developed was based on the Lotka-Volterra predator-prey model, originally designed to simulate species interactions in an ecosystem, but more generally contains the phenomenon known as a heteroclinic channel. The beginning and end of this heteroclinic channel can be defined as the same point and this leads to a heteroclinic cycle being created. It is using this cycle that the control algorithm was developed.

By drawing a comparison between an inter-species competing ecosystem and a similar competition regime where it is the internal states of the satellite system that are com-

peting, it is possible to create a novel control algorithm. The species interactions are replaced by action consequences and natural resources are replaced by state deficits. This is represented by a set of differential equations and the integrated time solution to which gives the required action selection. Also, as all the dynamic phenomenon exist explicitly in the mathematics of the differential equations, the stability of the cycle was analysed and the stable regimes of operation were defined.

This method of action selection worked with a degree of efficiency, but this could be attributed to the low dimensionality of the control problem (only three actions to be selected). As the heteroclinic channel, developed into a cycle, must permutate at a limited rate, there is a limitation in the time between actions. This limitation was overcome to a degree by introduction of thresholds, much like synaptic thresholding in neural networks, but this still leaves some degree of ambiguity as to what these thresholds should be.

It can be concluded that the Lotka-Volterra controller can be implemented as an action selection controller but it falls short of the goal to define a relatively computationally cheap control that can be verified and validated. The idea and inspiration behind the utilisation of the heteroclinic cycle is of interest but can be elaborated on further.

To reach the research goal, the controller developed in chapter 6 continued with the concept of the heteroclinic cycle but the additional concept of bifurcation theory was introduced to remove the need for thresholding. In this context a bifurcation is taken as a change in the qualitative structure of the dynamical system under investigation.

The heteroclinic cycle used in this controller is the structurally stable Guckenheimer-Holmes (GH) cycle. This cycle is a three-dimensional cycle with proven stability characteristics. By amending the GH cycle to suit the needs of the controller, i.e. by introducing the controller parameters, the revised GH cycle becomes an effective dynamical system for use as an action selection controller. The bifurcations occur in the cycle when one of the control parameters (state deficits) dominates the others. In this dynamic state the heteroclinic cycle ceases to exist and a singular global attracting node prevails. It is then this attracting node that defines the action selection process depending on which of the three possible bifurcations occur.

This heteroclinic cycle controller works along similar principles as the former controller; with the cycle permutating as a function of the three artificial action modes corresponding to the three actions that can be selected. The cycle evolves in a defined order (1-2-3-1), however, this order is not significant as an action is defined as occurring when a bifurcation occurs and this order only applies to the artificial mode order. Thresholding is no longer required as when the bifurcation occurs the location, on the phase space, of the new attracting node is unique to the individual action.

The new characteristics introduced to the heteroclinic cycle controller as a consequence of the bifurcation phenomenon make this controller more attractive than the previously discussed Lotka-Volterra model. By increasing the frequency of the cycle permutation, instantaneous changes between any of the actions can occur. Also, as the need for thresh-

olding has been removed the computational expense of the controller has also been decreased.

Another type of controller developed for action selection was a potential field controller. This controller type works with the premise of defining local potential minima (or 'wells') on a global potential field. Each of these local potential minima represent an individual control option and each minima changes depth as a function of the associated state deficit.

Characteristics of this type of controller are defined by the designer of the controller. Almost all the physical properties that will define the controller and hence its behaviour are prescribed during the controller design phase. This gives the control engineer authority as to the response of the controller, but as a consequence there is no inherent stability characteristics present.

Modifying the three actions for a single spacecraft to five actions for multiple spacecraft, entails placing either three or five local minima on a global field. As this work is mainly concerned with the development of the controllers, a simple ring configuration for the minima is chosen where each of the minima are evenly distributed around the circumference of a circle of specified diameter. Then, depending on the solution to the differential equations that represent the potential field, which in turn defines the action, causes the depths of the minima to change leading to a new action being selected.

The general conclusion to this type of controller is that it gives excellent authority during the design phase. However, when the number of tasks that can be selected by the control algorithm increases, interference between the local minima becomes an issue. Increasing the geometric distances between the minima to reduce this interference can lead to singularities in the differential equation solutions and decreasing the gradients of the local minima to accommodate this leads to time lags between action changes. Therefore, although an efficient control design can be implemented, this control technique is not desirable for complex action selection algorithms.

The last type of controller developed as an action selection algorithm is a modification of the adaptive controller used earlier. Here the same principles used for the attitude controller are incorporated in the action selection controller but now the action comes as a consequence of the spacecraft pointing angle. The reason for the different orbital model is that the mathematical conditions under which the controller must operate dictate a linear system and the orbital model used for the other models contains non-linearities.

Adaptive controllers of this type operate with the aim for minimising an artificial potential that is representative of the system state deficits through a Lyapunov equation. Due to the fact that the individual internal system states of the spacecraft under investigation here are interconnected then there does not exist a solution that drives all the system deficits to zero. Therefore the adaptive controller continuously tries to find a varying solution, in the form of the spacecraft pointing angle, that brings the system deficits as close to zero as possible.

As with the attitude controller, this is done by permitting the controller gains to vary

in real time. These controller gains then specify the angle that the spacecraft should point at any instant through a relation that maintains the stability of the control law. This real time variation in the control gains gives the same advantages as were inherent in the attitude controller.

One additional feature of autonomous control algorithms that is of importance is the ability of the control algorithms to recover from system failures. One such failure study was conducted with regards comparing the performance of the controllers. At a specified simulation time the solar array efficiency is presumed to half and the response of the controllers investigated.

The general conclusion to the study is that the controllers all posses a good 'bottom-up' means of recovering from the systemic failure. As none of the controllers use conventional fault identification followed by implementation of a different control law, then the real time changes in the control dynamics, defined by the system state, which influence the controller can work faster. The solar array failure leads to a degradation in performance in all the controllers but it is the adaptive controller that sees the least degradation. This is due to the continual change in the controller gains that occurs in this method only - this is effectively an continual optimisation process that leads to the most efficient means of dealing with the fault that has occurred.

7.3 Future Work

Adaptive control techniques are becoming more widely used in many areas of control engineering where plant and environmental dynamics are susceptible to error or change. More complex adaptive control algorithms to that developed and implemented here exist for use in controlling nonlinear dynamical systems. However, alongside this applicability to nonlinear systems come more complex and involved procedures for development of the algorithms.

In chapter 4 the reason that the adaptive controller could be applied to the asteroid lander control problem was due to the model having only one rotational degree of freedom. For a six degrees of freedom model of the asteroid lander there would be three rotational degrees of freedom and nonlinearities become present in the dynamics of the control problem. Developing a nonlinear adaptive controller should be one of the main aims of any continued work in this field. These controllers would be a much better representation of the true controlled systems that would be encountered.

With development of any nonlinear adaptive controller an associated increase in the fidelity of the models of the asteroid lander or spacecraft to be controlled should accompany this. Including the extra degrees of freedom of the asteroid lander model will increase the understanding of the real lander system. More accurate representations of thruster layout, lander leg reaction forces/moments, and lander interactions with a surface of unknown composition could then be simulated. These models could then be put through a more

rigorous FDIR investigation.

In chapters 5 and 6, for both the purposes of ease of visualisation and proof of concept, the dimensionality of the action selection problem was limited to three (four if the 'drift' action is counted). As most task scheduling procedures require the selection of a greater number of individual tasks than this attempts should be made to realise heteroclinic cycles that contain greater numbers of saddle nodes, ideally having any number, specifiable by the user.

This task is trivial when building a heteroclinic cycle based on the Lotka-Volterra model as the heteroclinic channel can have any number of saddles. The problem occurs when the same bifurcating phenomenon is desired, such as with the Guckenheimer-Holmes cycle, in the dynamics of the differential equations. A comprehensive study of known heteroclinic cycles could be conducted with their dimensionality and dynamical characteristics noted. Alternatively, attempts could be made at creating a system of differential equations that exist with the desired characteristics but can be extrapolated to a higher dimensionality.

In chapter 4 the only controller that was extended to simulate multiple satellites was the potential field method. With the development of heteroclinic cycles with higher dimensionality then multi-satellite models can be created, that require the extra dimensions, and a comparison can be drawn between the controllers while they control extra actions. More multi-satellite modeling will also give a better appreciation for the advantages to be gained from micro-satellite constellations involved in task sharing during formation flight for example.

Also, for the potential field controller there is the possibility of increasing the effectiveness of the action selection algorithm. By including local 'repellers' in the field by inclusion of negative potentials certain undesirable trajectories could be removed. There could also be an investigation into the placement of the attractors and repellers to discover if more efficient fields can be realised. A problem with the potential field controller is when the algorithm chooses to inter-link data instead of performing a primary action (record or transmit). Therefore by introducing a biasing factor, the inter-link actions would only be performed when the primary tasks could not be realised.

Similarly to the work proposed for attitude control there is a case for building a nonlinear adaptive controller for use in the action selection process. This was the main hinderance in the comparison of the controllers. A nonlinear action selection controller would primarily permit a direct comparison of the controllers by permitting the inclusion of the adaptive controller into the orbital model. Also, due to the fact the nonlinear controller could work with nonlinearities it would be capable of dealing with more complex action selection processes. For example by pointing in the correct direction the spacecraft may be capable of conducting two tasks simultaneously if the payload camera and antenna were on different faces of the spacecraft.

Once all the control algorithms could be applied to the same orbital model it would then be of interest to have a quantitative comparison of the CPU requirements of each

controller. Not only could a comparison between these controllers be made but also a comparison with other forms of autonomous action selection could be made, i.e. neural networks. The degree to which these controllers are more efficient could be quantified and a better case for continued research or integration into real systems for these techniques could be made.

Bibliography

[1] http://space.weber.edu/harbor/systemScience.shtml. month accessed - April 2010.

[2] Anderson C. Connerton R. Doyle R. Hoffman M. Man R. Fesq L., Aljabri A. *Spacecraft Autonomy in the New Millennium.* Breckenridge, Colorado, USA, 1996.

[3] Smith P. Jordan D. *Nonlinear Ordinary Differential Equations: An Introduction to Dynamical Systems.* Oxford University Press, Oxford, 1999.

[4] Sobel K. Kaufman H., Barkana I. *Direct Adaptive Control Algorithms: Theory and Application.* Springer, New York, 1994.

[5] Radice Gianmarco. *Behavior Based Autonomy for Single and Multiple Spacecraft.* PhD thesis, University of Glasgow, 2002.

[6] www.spaceavionics.com. month accessed - April 2010.

[7] http://en.wikipedia.org/wiki/Autonomy. month accessed - April 2010.

[8] Sweeting M. Underwood C. *Small Satellite Missions in Spacecraft Systems Engineering.* Wiley, Blackwell, 2002.

[9] Hubner H. Wimmer W., Ferri P. *On-Board Autonomy: Eureca Experience and Requirements for Future Space Missions.* Control Engineering Practice, 4:12:1715-1725, 1996.

[10] Gamble E. Kanefsky B. Kurien J. Man G. Millar W. Muscettola N. Nayak P. Rajan K. Rouquette N. Smith B. Taylor W. Tung Y. Bernard D., Dorais G. *Spacecraft Autonomy Flight Experience: The DS1 Remote Agent Experiment.* Proceedings of the AIAA Sapce Technology Conference and Exposition, Sept 28-30, Albuquerque, USA, AIAA-99-4512 1999.

[11] Doyle R. *Spacecraft Autonomy and the Missions of Exploration.* IEEE Journal of Intelligent Systems and their Applications, 13:5:277-296, 1998.

[12] Internal technical document EADS Astrium Ltd. *The Marco-Polo mission: Executive summary report.* EADS Astrium, AP.ASU.EU, 2009.

[13] http://adams.dm.unipi.it/cgibin/neodys/neoibo.objects:1999JU3;main. month accessed - April 2010.

[14] Faye F. Villefrache P., Evans J. *Rosetta: The ESA Comet Rendevous Mission.* Acta Astronautica, 40:871-877, 1997.

[15] http://www.esa.int/SPECIALS/Mars_Express/index.html. month accessed - April 2010.

[16] http://www.esa.int/SPECIALS/Venus_Express/index.html. month accessed - April 2010.

[17] Soffer A. Samet H. *MAGELLAN: Map Acquisition of GEographic Labels by Legend ANalysis.* International Journal on Document Analysis and Recognition, 1:89-101, 1998.

[18] Frapard B. Flandin G., Polle B. *Vision based navigation for planetary exploration.* Journal of Guidance and Control, 133:277-296, 2009.

[19] Hashimoto T. Yoshimitsu T., Kawaguchi J. *Hayabusa-final autonomous descent and landing based on target marker tracking.* Acta Asronautica, 65:657-665, 2009.

[20] Gaines D. Chouinard C. Schaffer S. Nesnas I. Estlin T., Fisher F. *Continuous Planning and Execution for an Autonomous Rover.* Proceedings of the Third International NASA Workshop on Planning and Scheduling for Space, October 27-29, Houston, USA, 2002.

[21] Stechert A. Sherwood R. Rabideau G. Chien S., Knight R. *Using Iterative Repair to Increase the Responsiveness of Planning and Scheduling for Autonomous Spacecraft.* IJCAI99, Workshop on Scheduling and Planning meets Real-time Monitoring in a Dynamic and Uncertain World, August 2-4, Stockholm, Sweden, 1999.

[22] Apfelbaum D. Simmons R. *A Task Description Language for Robot Control.* Proceedings of the IEEE Intelligent Robots and Systems Conference, October 13-18, Vancouver, Canada, 1998.

[23] http://nmp.jpl.nasa.gov/ds1/tech/autora.html. month accessed - April 2010.

[24] Bermyn J. Mellab K. Teston F., Creasey R. *PROBA: ESA's Autonomy and Technology Demonstrator Mission.* IAC-04-P.5.05, 48th International Astronuatical Congress, October 6-10, Turin, Italy, 1997.

[25] Vuilleumier P. Van der Braembussche P. Mellab K. de LaFontaine J., Buijs J. *Development of the PROBA Attitude Control and Navigation Software.* ESA-SP-425, ESA 4th International Conference on Spacecraft Guidance, Navigation and Control Systems, October 18-21, Noordwijk, Netherlands, 1999.

[26] http://space.skyrocket.de/doc_sdat/hawksat 1.htm. month accessed - April 2010.

[27] http://space.skyrocket.de/doc_sdat/aerocube 1.htm. month accessed - April 2010.

[28] McEntire R. Sibeck D. *Multiple satellite observations of leakage of particles from the magnetosphere.* Advances in Space Research, 8:9-10:201-216, 2002.

[29] Marshall M. Cameron G. *Exploring the Practical Limits of Operations Autonomy.* Paper ID: 1b020, Proceedings of the Fifth SpaceOps Symposium, 1-5 June, Tokyo, Japan, 1998.

[30] Nguyen T. Stone H. Cooper B. Wilcox B. Mishkin A., Morrison J. *Experiences with Operations and Autonomy of the Mars Pathfinder Microrover.* 2A008, Proceedings of the IEEE Aerospace Conference, October 27-29, Montana, USA, 1998.

[31] Matthies L. Helmick D, Angelova A. Terrain adaptive navigation for planetary rovers. *Journal of Field Robotics*, 26:391–410, 2009.

[32] Doyle R. J. *Spacecraft Autonomy and the Missions of Exploration.* IEEE Intelligent Systems, Guest Editors Introduction, Special Issue on Autonomous Space Vehicles, 13(5): 36-44, September/October, 1998.

[33] Swinard G. Forescue P., Stark J. *Spacecraft Systems Engineering.* John Wiley, Hoboken, 2003.

[34] Brown C. *Element of Spacecraft Design.* AIAA Education Series, Reston, 2003.

[35] McCall J. *Autonomous techniques for space exploration.* PhD thesis, Cranfield University, 2000.

[36] Xibin C. Xingrui M. Jiajun Y., Guodong C. *A fuzzy expert system architecture implementing onboard planning and scheduling for autonomous small satellites.* SSC98-VII-1.4, 12th Annual AIAA/ Utah State University Conference on Small Satellites, August 31 - Sept 3, Logan, Utah, USA, 1998.

[37] Bermyn J. *PROBA Project for On-Board Autonomy.* Air and Space Europe, 2:7-76, 2000.

[38] Chien S. *Automated Planning and Scheduling for Goal-Based Autonomous Spacecraft.* Jet Propulsion Laboratory, Technical Report, 1094-7167/9, 1998.

[39] Ericson C. *Fault tree analysis - a history.* In Proceedings of the 17th international system safety conference, pp. 8796., August 16-21 , Orlando, USA, 1999.

[40] ZHANG l. Hao X. *Standard Sneak Circuit Analysis and its Application in China Space Reliability Engineering.* ICRMS.09.5270233, In Proceeding of the 8th Internation Conference on Reliability, Maintainability and Safety, July 20-24, Chengdu, China, 2009.

[41] Author Unknown. *Apollo Reliability and Quality Assurance Office, Procedure for Failure Mode, Effects and Criticality Analysis (FMECA).* NASA, RA0060131A, 1966.

[42] Bornschlegl E. *(On-Board) Autonomy Objective and Technology Support.* ESA SP-509, Proceedings of DASIA Conference, May 13-16, Dublin, Ireland, 2002.

[43] Norvig P. Russel S. *Artificial Intelligence: A Modern Approach.* Prentice Hall, Upper Saddle River, 2003.

[44] Rolston D. *Principles of Artificial Intelligence and Expert Systems.* McGraw-Hill, Maidenhead, 1988.

[45] Schneider S. *Fuzzy Expert.* John Wiley, Hoboken, 1996.

[46] Xiaofei X. Rui X., Pingyuan C. *Realization of multi-agent planning system for autonomous spacecraft.* Advances in Engineering Software, 36:4:266-272, 2009.

[47] Beale M. Hagan M., Demuth H. *Neural Network Design.* PWS, Pacific Grove, 1996.

[48] Kosko B. *Neural Network and Fuzzy Systems: A dynamical Systems Approach to Machine Intelligence.* Prentice Hall, Upper Saddle River, 1992.

[49] Parthasarathy K. Narendra K. *Identification and Control of Dynamical Systems using Neural Networks.* IEEE Transactions on Neural Networks, 1:4-27, 1990.

[50] Good S. Conkey D., Dell G. *Telerobotic Operations Testing in Neural Buoyancy Simulation.* IEEE Explore, 1063-7117/92, 1992.

[51] Good S. Conkey D., Dell G. *Experiences with Operations and Autonomy of the Mars Pathfinder Microrover.* IEEE Explore, 0-7803-5846-5/00, 2000.

[52] Shortliffe E. *Computer-Based Medical Consultation: MYCIN.* Elsevier, San Francisco, 1976.

[53] Shah R. *JET-X: Jet Engine Troubleshooting Expert System.* Proceedings of the International Workshop on Artificial Intelligence for Industrial Application, August, New Jersey, USA, 1988.

[54] Hbler A. Lscher E. Maurer M. Meinke P. Eberl W., Kuchler M. *Analytical representation of stroboscopic maps of ordinary nonlinear differential equations.* Zeitschrift fur Physik B Condensed Matter, 68:253-258, 1987.

[55] Shaw C. Abraham R. *Dynamics: The Geometry of Behavior.* Addison-Wesley, Reading, 1992.

[56] Bodson M. Sastry S. *Adaptive Control: Stability, Convergence, and Robustness.* Prentice-Hall, Upper Saddle River, 1989.

[57] Wittenmark B. Astrom K. *Adaptive Control.* Addison Wesley, Reading, 1989.

[58] Maciejowski J. *Multivariable Feedback Design.* Addison Wesley, Reading, 1989.

[59] Doyle J. C. Zhou K. *Essentials of Robust Control.* Prentice-Hall, Upper Saddle River, 1998.

[60] Erzberger H. *On the use of algebraic methods in the anaylsis and design of model following control systems.* NASA, Tech. Rep. D-4663, 1963.

[61] LaSalle J. *Stability of non-autonomous systems.* Non-linear Analysis Theory Methods and Applications, 1:87-97, 1981.

[62] Lyapunov A. *Stability of motion.* Academic Press, New York, 1966.

[63] Khalil H. *Nonlinear Systems.* Prentice Hall, Upper Saddle River, 1992.

[64] Michael R. Michael F., AHearn C. *Deep Impact at Comet Tempel 1.* ICARUS, 0019-1035, 2007.

[65] Chawla S. Meyers M. *Mechanical behavior of Materials.* Prentice Hall, Upper Saddle River, 1999.

[66] Homann K. Kallay N. Kuchitsu K. Mills I., Cvitas T. *IUPAC Quantities, Units and Symbols in Physical Chemistry.* Blackwell Scientific Publications, Oxford, 1993.

[67] Barr M. *Pulse Width Modulation.* Embedded Systems Programming, Addison-Wesley Longman Publishing, Boston, 2001.

[68] McDowall R. *Fundamentals of HVAC Control Systems.* Elsevier, Maidenhead, 2009.

[69] Elster K. *Modern Mathematical Methods of Optimization.* Wiley VCH, Cologne, 1993.

[70] Widnall W. S. *Lunar Module Digital Autopilot.* Journal of Spacecraft and Rockets, 8:56-60, 1970.

[71] Bertram R. E. Kalman R. E. *Control system analysis and design via the second method of Lyapunov part I: Continuous systems.* ASME, 83:371-393, 1960.

[72] Bertram R. E. Kalman R. E. *Control system analysis and design via the second method of Lyapunov part II: Discrete systems.* ASME, 83:394-400, 1960.

[73] Bernstein D. S. Hong J. *Adaptive stabilization of non-linear oscillators using direct adaptive control.* Internation Journal of Control, 74:432-444, 2000.

[74] Yoshikawa M. Michel P. Barucci M. *NASA Guidebook for Safety Critical Software - Analysis and Development.* NASA-GB-8719.13, 1995.

[75] Padovan S. Casotto S. *Detecting Body Tides and Librations of Europa With an Altimetric Exploration Mission.* AIAA 2008-7200, AIAA/AAS Astrodynamics Specialist Conference and Exhibit, August 18-21, Honolulu, 2008.

[76] Agnolon D. *Study Overview of the Near Earth Asteroid Sample Return.* ESA, SCI-PA/2007/004/PA, 2007.

[77] Baoyin H. Gong S., Li J. *Solar radition pressure used for formation flying around Sun-Earth libration point.* Applied Mathematics and Mechanics, 30:1009-1016, 2009.

[78] Scharf D.P., Ploen S.R., and Hadaegh F.Y. *A survey of spacecraft formation flying guidance and control (Part II): Control.* In Proceedings of the American Control Conference, Vol.4 pp 2976-2985, June 4-6, Colorado, 2003.

[79] Devaney R. L. Hirsch M. W., Smale S. *Differential Equations, Dynamical Systems and An Introduction to Chaos.* Elsevier, Maidenhead, 2004.

[80] Agiza H. El-Dessoky M. *Global stabilization of some chaotic dynamical systems.* Chaos Solitions and Fractals, 42:1584-1598, 2009.

[81] Lotka A. *Contribution to the Theory of Periodic Reaction.* Journal of Physics and Chemistry, 14:3:271-274, 1910.

[82] Berryman A. *The Origins and Evolution of Predator-Prey Theory.* Ecology, 75:5:1530-1535, 1992.

[83] Hutchinson S. Choset H., Lynch K. *Principles of Robot Motion.* Cambridge Press, Cambridge, 2005.

[84] Zeng J. Xue Z. *Circle Formation Control of Large-Scale Intelligent Swarm Systems in a Distributed Fashion.* 6th International Symposium on Neural Networks, May, Wuhan, 2009.

[85] Chellaboina V. Haddad W. *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach.* Princeton University Press, Princeton, 2008.

[86] Momeni H. Boroujeni E., Dadras S. *An adaptive Controller for a Class of Non-linear Systems: An Application to Ship Roll Motion.* M3C.3, 41st Southeastern Symposium on System Theory, March 15-17, Tullahoma, 2009.

[87] Ravine M. Soulanille T. Malin M., Danielson G. *Design and Development of the Mars Observer Camera.* International Journal of Imaging Systems and Technology, Vol. 3, 76-91, 1991.

[88] NASA. *http : //sunland.gsfc.nasa.gov/smex/sampex/.* month accessed - April 2010.

[89] Larson W. Wertz J. *Space Mission Analysis and Design.* Kluwer Academic Publishers, 1999.

[90] Underwood C. Lappas V., Steyn W. *Attitude control for small satellites using control moment gyros.* Chaos, 51:1-9:101-111, 2002.

[91] Michel P. Morgan-Owen R. Winton A., Gerner J. *The Transponder - A Key Element in ESA Spacecraft TTC Systems.* ESA Bulletin, No. 86, pp 95-104, 1996.

[92] Muezzinoglu M. Rabinovich M. *Mutual Emotion-Cognition Dynamics.* Journal of the Association for the Advancement of Artificial Intelligence, 0909.1144v1, 2009.

[93] Huerta R. Rabinovich M. Afraimovich V., Tristan I. *Winnerless competition principle and prediction of the transient dynamics in a Lotka-Volterra model.* Chaos, 18:043103, 2008.

[94] Rabinovich M. Afraimovich V., Zhigulin V. *On the origin of reproducible sequential activity in neural circuits.* Chaos, 14:1123-1129, 2004.

[95] McInnes C. Gilles E., Johnston A. *Action Selection Algorithms for Autonomous Microspacecraft.* Journal of Guidance Dynamics and Control, 22:6:914-916, 1999.

[96] Xie L. Chen H. *A novel artificial potential field based reinforcement learning for mobile robitics in ambient intelligence.* International Journal of Robotics and Automation, 24:245-254, 2009.

[97] Lugo-Gonzalez E. Ramirez-Gordillo J., Merchan-Cruz E. *The Laplacian Artificial Potential Field for the Path Finding of Robotic Manipulators.* CERMA.2008.66, 5th Electronics, Robotics and Automotive Mechanics Conference, September 30 - October 3, Cuernavaca, 2008.

[98] Leger A. Cockell C., Herbst T. *Darwin - An experimental astronomy mission to search for extrasolar planets.* Experimental Astronomy, 23:435-461, 2009.

[99] Jennrich O. *LISA technology and instrumentation.* Classical and Quantum Gravity, 26:153-162, 2009.

[100] Holmes P. Guckenheimer J. *Structurally Stable Heteroclinic Cycles.* Mathematical Proceeding to the Cambridge Philosophical Society, 103:189-192, 1988.

[101] Li Z. *Chaos induced by heteroclinic cycles connecting repellers and saddles in locally compact metic space.* Nonlinear Analysis-Theory Methods and Applications, 71:1379-1388, 2009.

[102] Kuznetsov Y. *Elements of Applied Bifurcation Theory.* Springer, New York, 2004.

[103] Celik C. *Hopf bifurcation of a ratio-dependent predator-prey system with time delay.* Chaos Solitions and Fractals, 42:1471-1484, 2009.

[104] Holmes P. Guckenheimer J. *Structurally Stable Heteroclinic Cycles.* Mathematical Poceeding to the Cambridge Philosophical Society, 103:189-192, 1988.

[105] Smith P. Jordan D. W. *Nonlinear Ordinary Differential Equations.* Oxford University Press, Oxford, 1999.

[106] Kreyszig E. *Advanced Engineering Mathematics.* John Wiley, Hoboken, 1999.

[107] Zebulum R. Mojarradi M. Katkoori S. Daud T. Stoica A., Keymeulen D. *Adaptive and Evolvable Analog Electronics for Space Applications.* Springer, Berlin-Heidelberg, 2007.

[108] Stoica A. Keymeulen D. Daud T. Zebulum R., Mojarradi M. *Self-Reconfigurable Analog Arrays: Off-The Shelf Adaptive Electronics for Space Applications.* AHS.2007.96, Second NASA/ESA Conference on Adaptive Hardware and Systems, August 5-8, Edinburgh, 2007.

[109] OP184/OP284/OP484. *www.analog.com/static/imported−files/data_sheets/OP184_284_484.pdf - datasheet.* Analog Devices Inc, 1996.

[110] AD663. *www.analog.com/en/analog-to-digital-converters/products - datasheet.* Analog Devices Inc, 2002.

[111] Clayton P. *Fundamentals of Electric Circuit Analysis.* John Wiley, Hoboken, 2001.