Dissertation

Fly Around Orbit and Capture Attitude Control for Space Debris Mitigation

(スペースデブリ除去を実現する為のフライアランド軌道及び捕捉姿勢制御)

2022年3月

Amila Sri Madhushanka Gilimalage

Fly Around Orbit and Capture Attitude Control for Space Debris Mitigation

Amila Sri Madhushanka Gilimalage

Abstract

Utilization of spaces for human activities has significantly expanded over the past several decades. Both private and government sectors are actively involved in improving the commercial and scientific value of space. With an increased number of functional objects populating the orbits around earth, the amount of space debris has also accumulated at an alarming rate. This is hazardous for both the current and future space missions because such objects require sudden evasive maneuvers and continuous monitoring Therefore, space debris mitigation has become an important theme in the sphere of space-related research.

Dealing with this issue is not simple because of the multiple stages of work involved in completing a successful implementation of a mitigation mission. One challenge is that target bodies such as debris and malfunctional satellites can have arbitrary rotational motions. The chaser satellite must then travel along a fly around orbit with close-proximity and formation flying. The velocity and position vectors of both objects should align simultaneously for safer docking opportunities. To keep the chaser at a constant distance from the target, the chaser's thrusters must be fired intelligently. Because resources are limited in space, optimizing fuel consumption while achieving control requirements is a challenge. To overcome this issue, a model predictive control (MPC) based algorithm is developed for a target-chaser rendezvous situation to optimize fuel consumption while considering the thruster constraints and memory usage. It is

compared with several conventional controllers to evaluate the effectiveness of the algorithm.

Once the target and chaser are locked in fixed relative motion, the chaser satellite can move closer to capture the target using different types of grappling mechanisms. Here, another challenge that occurs is in the form of ambiguities that arise in inertial properties while capturing objects. To stabilize the attitude of a spacecraft, a typical control algorithm requires accurate inertia measurements. When the spacecraft captures an unknown object, its body configuration and mass change, leading to changes in its dynamics and inertial properties. This can produce tumbling effects and possible deorbiting scenarios due to the sudden shift in momentum. Thus, the control algorithm needs robustness to cope with these ambiguities and parameter variations. To cope with this situation, an intelligent attitude control algorithm is developed for a satellite with partially known inertial properties. By combining inertia estimation with a neural network-based controller, the objectives of control design can be achieved. A comparison was performed with several other control schemes to evaluate the performance using simulation environments for validation.

Another underlying issue when dealing with space crafts in space is the perturbations that arise due to different reasons. One such reason is the perturbations due to fuel slosh. In general, a spacecraft comprises several flexible and rigid bodies. Fuel, in particular, is contained in a separate enclosure and could hold a significant portion of the total mass of the satellite; however, due to microgravity, it can move freely within its limited space. The motion of fuel can cause disturbances in the translational and rotational control of satellites. To overcome this issue, the sloshing dynamics are analyzed, and a control algorithm is developed considering the sliding mode control together with an error minimizing criterion to suppress the fuel slosh and attitude error of the satellite. This is

compared with several conventional controllers for performance evaluation using computer simulations.

Keywords:

Space debris mitigation, Model predictive control, Neural-networks, Sliding mode control, Attitude control, Orbit control

Contents

CHAPTER 1 INTRODUCTION 13

1.1. Introduction 13

1.2. Contributions in this dissertation 16

1.2.1. Orbit control of a satellite considering a target chaser situation. 16

1.2.2. Attitude control of a satellite when an uncooperative object is attached. 18

1.2.3. Controlling satellites under fuel slosh situation. 19

CHAPTER 2 MPC-BASED ORBIT CONTROLLER DESIGN FOR THE FLY-AROUND SCENARIO CONSIDERING FUEL OPTIMIZATION 21

2.1. Background and Literature Review 21

2.2. Methodology 23

2.2.1. Relative Motion between Two Objects 23

2.2.2. MPC Design 25

2.2.3. Region-Free MPC 29

2.2.4. Linear Quadratic Regulator (LQR) 32

2.2.5. C-W Based Control (Clohessy-Wiltshire) 32

2.2.6. Quantizer Design 33

2.3. Simulation Results 35

2.4. Summary 47

CHAPTER 3 NEURAL NETWORKS BASED ADAPTIVE ATTITUDE CONTROLLER DESIGN FOR A SATELLITE WITH PARTIAL KNOWN INERTIAL PROPERTIES 48

3.1. Background and Literature Review 48

3.2. Methodology 51

3.2.1. Estimating the Inertia 51

3.2.2. Attitude Representation 54

3.2.3. Proportional-Derivative Controller Design (PD) 54

3.2.4. Fuzzy Logic based Controller Design (FBC) 54

3.2.5. Inertia based Controller Design (IBC) 57

3.2.6. Neural based Adaptive Controller Design (NBAC) 58

3.3. Simulation Results 62

3.4. Summary 81

CHAPTER 4 NOVEL SLIDING MODE CONTROL-BASED CONTROLLER DESIGN FOR A SATELLITE WITH LIQUID FUEL SLOSH DISTURBANCES 82

4.1. Background and Literature Review 82

4.2. Methodology 84

4.2.1. Mathematical Model 84

4.2.2. Sliding Mode Controller Design (SMC) 88

4.2.3. Novel Sliding Mode Controller Design (NSMC) 90

4.2.4. Proportional-Integral-Derivative Controller design (PID) 93

4.3. Simulation Results 93

4.4. Summary 103

CHAPTER 5 DISCUSSIONS AND CONCLUSIONS 105

5.1. Discussion 105

5.1.1. Discussion on MPC 105

5.1.2. Discussion on Neural based Adaptive Controller Design 106

5.1.3. Discussion on Sliding Mode Controller Design 107

5.2. Conclusion 108

References 109

Publication List 116

List of Figures

Figure 1.1 Accumulation of space debris over the years : Orbital Space Debris Quarterly News 2021 [1] 14

Figure 1.2 Sequence of the space debris mitigation procedure 15

Figure 1.3 Motion of target and chaser bodies when approaching the capturing process 17

Figure 1.4 Target and chaser bodies soon after the capture process 18

Figure 1.5 Fuel slosh phenomena as seen when moving on a fixed frame 20

Figure 2.1 The target and chaser in LVLH frame with origin at the centre of the Earth 24

Figure 2.2 Satellite control system implementation. 34

Figure 2.3 Comparison of the fly-around orbits with the three controllers. 36

Figure 2.4 Change in fly-around radius with the MPC-based controller. 37

Figure 2.5 Changes in the fly-around radius with the LQR-based controller. 37

Figure 2.6 Changes in the fly-around radius with the C-W based controller. 37

Figure 2.7 Changes in the fly-around angular velocity with the MPC-based controller.38

Figure 2.8 Changes in the fly-around angular velocity with the LQR-based controller. 38

Figure 2.9 Changes in the fly-around angular velocity with the C-W based controller. 39

Figure 2.10 Generated control input for a fly-around scenario with the MPC-based controller in the X direction..... 39

Figure 2.11 Generated control input for a fly-around scenario with the MPC-based controller in the Y direction..... 40

Figure 2.12 Generated control input for a fly-around scenario with the LQR-based controller in the X direction..... 40

Figure 2.13 Generated control input for a fly-around scenario with the LQR-based controller in the Y direction..... 41

Figure 2.14 Generated control input for a fly-around scenario with the CW-based controller in the X direction..... 41
Figure 2.15 Generated control input for a fly-around scenario with the CW-based controller in the Y direction..... 41
Figure 2.16 Change in radius with different configurations..... 44
Figure 2.17 Change in angular velocities with different configurations 45
Figure 2.18 Graphical representation of trajectories with different configurations 47
Figure 3.1 Components of a general fuzzy model..... 55
Figure 3.2 Membership function: attitude error. 56
Figure 3.3 Membership function: rate of change of attitude error 57
Figure 3.4 Membership function: angular velocity. 57
Figure 3.5 Membership function: rate of change of angular velocity. 57
Figure 3.6 Neural based adaptive controller architecture..... 58
Figure 3.7 3-layer neuron perceptron. 59
Figure 3.8 PD controller: variation in quaternions..... 63
Figure 3.9 PD controller: variation in angular velocities. 63
Figure 3.10 PD controller: variation in required torques..... 64
Figure 3.11 FBC: variation in quaternions. 64
Figure 3.12 FBC: variation in angular velocities. 65
Figure 3.13 FBC: variation in required torques..... 65
Figure 3.14 IBC: variation in quaternions. 66
Figure 3.15 IBC: variation in angular velocities. 66
Figure 3.16 IBC: variation in required torques..... 67
Figure 3.17 NBAC: variation in quaternions..... 67
Figure 3.18 NBAC: variation in angular velocities..... 68

Figure 3.19 NBAC: variation in required torques. 68
Figure 3.20 Variation in adaptive gains in NBAC. 69
Figure 3.21 PD control: variation in quaternions. 70
Figure 3.22 PD control: variation in angular velocities..... 71
Figure 3.23 PD control: variation in required torques. 71
Figure 3.24 FBC: variation in quaternions. 72
Figure 3.25 FBC: variation in angular velocities. 72
Figure 3.26 FBC: variation in required torques. 73
Figure 3.27 IBC: variation in quaternions. 73
Figure 3.28 IBC: variation in angular velocities. 74
Figure 3.29 IBC: variation in required torques..... 74
Figure 3.30 NBAC: variation in quaternions..... 75
Figure 3.31 NBAC: variation in angular velocities. 75
Figure 3.32 NBAC: variation in required torques. 76
Figure 3.33 Change in learning rate with quaternions 78
Figure 3.34 Change in learning rate with angular velocities 79
Figure 3.35 Change in quaternion errors with modified learning rate 79
Figure 3.36 Change in angular velocities with modified learning rate 80
Figure 3.37 Change in required torques with modified learning rate 80
Figure 4.1 Satellite model with fuel container 85
Figure 4.2 Particle swarm optimization overview 92
Figure 4.3 Change in attitude angle with time (NSMC) 94
Figure 4.4 Rate of change of attitude angle with time (NSMC) 95
Figure 4.5 Change in slosh angle with time (NSMC) 95
Figure 4.6 Rate of change of slosh angle with time (NSMC) 95

Figure 4.7 Change in control input 1 with time (NSMC)..... 96
Figure 4.8 Change in control input 2 with time (NSMC)..... 96
Figure 4.9 Change in attitude angle with time (SMC) 96
Figure 4.10 Rate of change of attitude angle with time (SMC) 97
Figure 4.11 Change in slosh angle with time (SMC) 97
Figure 4.12 Rate of change of slosh angle with time (SMC) 97
Figure 4.13 Change in control input 1 with time (SMC) 98
Figure 4.14 Change in control input 2 with time (SMC) 98
Figure 4.15 Change in attitude angle with time (PID) 98
Figure 4.16 Rate of change of attitude angle with time (PID) 99
Figure 4.17 Change in slosh angle with time (PID) 99
Figure 4.18 Rate of change of slosh angle with time (PID) 100
Figure 4.19 Change in control input 1 with time (PID) 100
Figure 4.20 Change in control input 2 with time (PID) 101
Figure 4.21 Transverse velocity vs time in the Z direction..... 103
Figure 4.22 Transverse velocity vs time in the X direction..... 103

List of Tables

Table 2.1 Simulation conditions considered in this study. 35

Table 2.2 RMSE values for positions and velocities in the X-Y directions. 42

Table 2.3 ΔU requirement for a total simulation time of 1000 s. 43

Table 2.4 Memory requirements with different control horizons for the MPC controller. 43

Table 2.5 RMSE values and delta V for the considered configurations 46

Table 2.6 Steady state errors with the considered controllers after the simulation time. 46

Table 3.1 Rule table denoting fuzzy model 1. 56

Table 3.2 Rule table denoting fuzzy model 2. 56

Table 3.3 Initial conditions for simulation set 1 62

Table 3.4 Settling times with each controller.. 68

Table 3.5 RMSE of quaternions. 69

Table 3.6 RMSE of angular velocities. 69

Table 3.7 RMSE of required torques. 69

Table 3.8 Initial conditions for simulation set 2 70

Table 3.9 RMSE of quaternions. 76

Table 3.10 RMSE of angular velocities. 76

Table 3.11 RMSE of required torques. 77

Table 3.12 RMSE of estimated inertia components. 77

Table 3.13 Performance with different learning rates 78

Table 3.14 Comparison of data of NBAC with modified learning rate 81

Table 4.1 Parameter values used for the dynamical model 93

Table 4.2 Parameter gains used in simulations 94

Table 4.3 Case 1 when fuel mass 100 kg and inertia 90 kgm^2 101

Table 4.4 Case 2 when fuel mass mf =200 kg, inertia If= 180 kgm^2 102

Table 4.5 Case 3 when fuel mass mf =50 kg, inertia If= 45 kgm_2 102

Table 4.6 Settling times for each case considered..... 102

Table 4.7 Accumulation of control input over the simulation time..... 102

CHAPTER 1
INTRODUCTION

1.1. Introduction

Over the past several decades, utilization of spaces for human activities has rapidly increased. With easier access to space, the number of nations and organizations that are active in space has significantly grown. This is largely through joint research work at the International Space Station (ISS), advances in communication networks, and other space related activities. More than 2000 satellites are currently in operation with mega constellations in development, which will congest the remaining areas. However, as more and more objects are populating space, the amount of space related debris has also gradually increased over the years. As of 2021, the US Space Surveillance Network officially categorized more than 21000 space debris in the earth's orbit [1]. This is excluding the small-scale debris which ranges from 1 to 10 cm and pieces that are less difficult to detect and monitor due to size and large number. Most of these are failed satellites, rocket upper stages, space-mission related parts, and fragments left from collisions between satellites and due to explosions. These cause frequent changes to orbital paths of active satellites and risky environments for commercially valuable orbital planes and increase the probability of creating a massive number of smaller debris via collisions. Thus, the risks that these debris pose for the safety of astronauts and space missions cannot be taken lightly. Furthermore, space debris also shortens the lifespans of active satellites as they must be constantly alert for potential impacts and have to perform evasive manoeuvres when needed. Therefore, active removal of space debris has been identified as a vital component in securing safe access to space for the future space related activities as well as for the sustainable development of space for mankind.

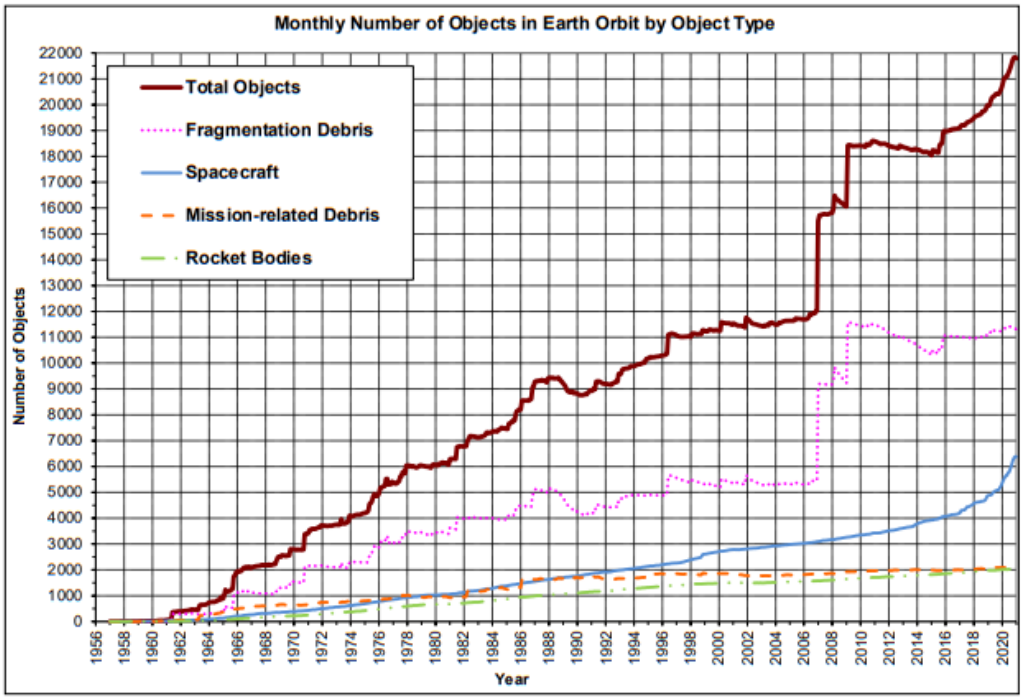

Figure 1.1 Accumulation of space debris over the years : Orbital Space Debris Quarterly News 2021 [1]

The success of space debris mitigation programs depends on collaborative work among state and institutional bodies that govern and oversee the space industry. The primary goals of space debris mitigation can be summarized as follows: minimizing the growth of debris in the future space missions and reducing the space debris currently in orbit. The accumulation of future space debris can be minimized by adopting rigorous regulatory procedures for future space missions. The currently operative satellites can also follow such regulations at the end of their lifespans if the systems on board support such configurations and manoeuvres. The parts that are unable to move on their own, such as difunctional satellites, find it challenging to follow any such schemes. This is where an external mechanism can be applied to capture such objects. This, in particular, has been discussed and researched as a viable solution to regulate the growing amount of space debris. However, it is a process that involves background work both in space and on earth for successful implementation.

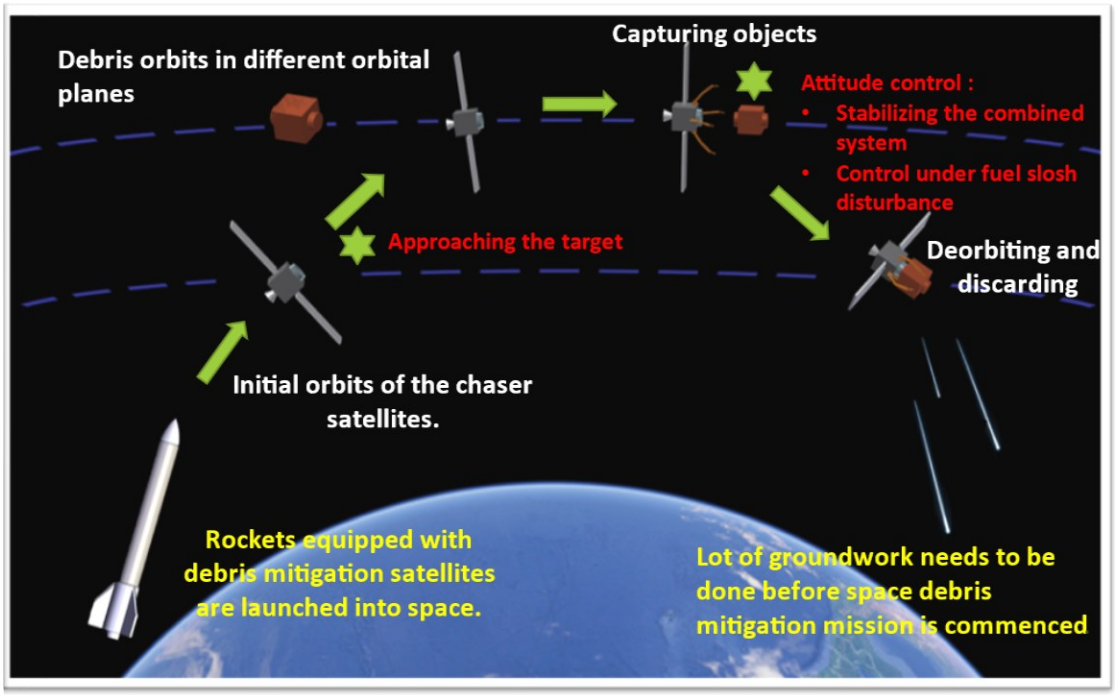

Figure 1.2 Sequence of the space debris mitigation procedure

As demonstrated in Figure 1.2, the process of capturing space debris can be stripped down to a few key points for simplicity. The first part is launching of the shuttle equipped with a spacecraft specifically designed to capture space debris. This comes at the end of groundwork that includes identifying the type of debris that is to be captured, research into the required technology and mechanisms, development of such systems and testing. Once all systems are in place, the chaser satellite can be launched into space and deployed to its initial orbit. In parallel, the target debris must be tracked for preliminary studies. Since these debris can be in space for extended periods of time, their physical properties could be different from their original status. For example, in the case of fuel leakages, the inertial properties could unnoticeably change. Apart from this, these debris do not have any internal mechanisms to control themselves and could be orbiting on a plane with arbitrary rotational motions. Once the capturing satellite settles in its initial orbit, a manoeuvre is performed to change its orbit and move the chaser satellite close to the target body. When both objects are in close proximity, their relative motions are

aligned and an external mechanism such as robotic arms, tethers, and nets could be used to capture the unknown object. After completing this step, the attitude and orientation of this combined body must be brought to a desired value. Then, the final phase of the removal process can be performed by increasing the orbit of the debris to a graveyard orbit or by decreasing the orbit by reducing the speed and discarding it safely by burning with the help of atmospheric drag.

1.2. Contributions in this dissertation

Developing control algorithms for space systems dealing with the above situations is a complex issue which involves several steps including identifying the target using visuals and other means, approaching the target in an efficient manner, deploying a capturing mechanism, and controlling the combined system after the capturing process to safely deorbit the debris. Considering all these aspects, the author has concentrated on developing controllers that can be utilized for a space debris mitigation program considering both orbit and attitude control scenarios. The first objective focuses on the orbit control aspect. The second and third objectives focus on attitude control considering the two situations in which it can be utilized.

- Orbit control of a satellite considering a target chaser situation.
- Attitude control of a satellite when an uncooperative object is attached.
- Controlling satellites under fuel slosh situations.

1.2.1. Orbit control of a satellite considering a target chaser situation.

When capturing an arbitrary object orbiting the earth, it is important to move closer to the object before deploying any grappling mechanism. Generally, these objects do not have any stored energy to control their movement; thus, they can remain in rotational motion

about their principal axes. The chaser satellite must assess the distance between the target and chaser itself and the rotational movement of the target before safely connecting with the target itself.

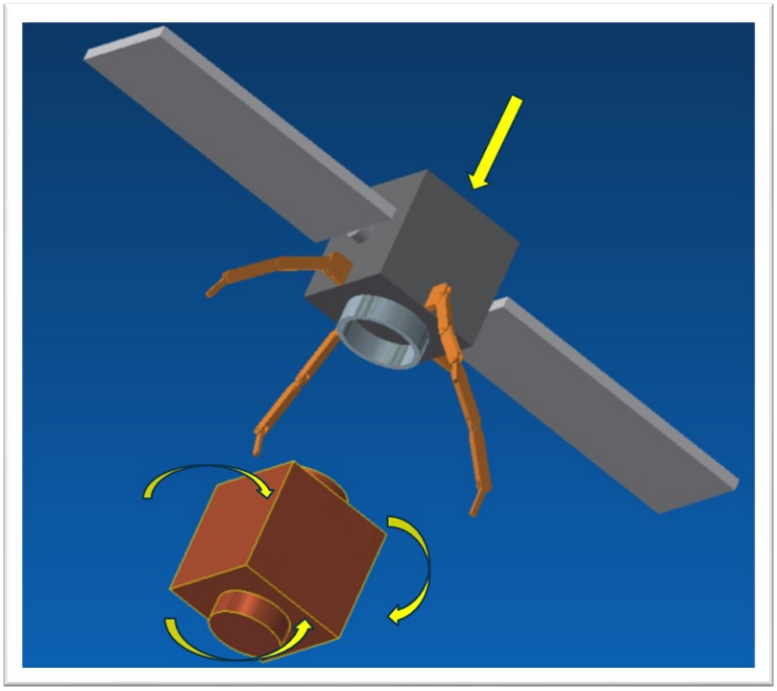

Figure 1.3 Motion of target and chaser bodies when approaching the capturing process

The first part of this research involved identifying the relative motion between objects in space. After that, an analysis of an model predictive control (MPC)-based algorithm was performed to analyse the orbital rendezvous for a target chaser situation with limitations on controller outputs to minimize fuel consumption. In conventional MPC algorithms, high processing power and memory usages are some of the setbacks when limitations are considered. To overcome these issues, a region-free based algorithm is proposed. With this method, the requirement for on board solvers is removed by performing calculations offline, and then, the memory usage is reduced in comparison to general and explicit MPC-based controllers because only the applicable values are selected for storage. The methodology explains the relative motion between objects in space and the development of several controller designs based on the above method, linear quadratic regulator

(LQR)-based methods, and Clohessy-Wiltshire (CW)-based methods for performance comparison.

1.2.2. Attitude control of a satellite when an uncooperative object is attached.

The second objective of this research is developing an attitude-control system for a satellite while capturing uncooperative objects in space. The challenge is that it is not possible to entirely know the actual mass and inertial properties of these objects. To cope with the complexities associated with designing such a system, this research concentrates on two key areas explicitly. The initial system is a combination of two separate parts: the chaser and the target. The chaser (assumed to be a satellite) has known inertial properties, and the target's inertial properties are partially known. When the two objects are combined after the capturing procedure, the new system is expected to have known initial velocity and orientation. The first part of the research was to identify the system's inertial matrix. This is performed using the recursive least squares algorithm combined with satellite velocity and acceleration.

Figure 1.4 Target and chaser bodies soon after the capture process

The next priority in this research is designing a control algorithm to control the attitude of the combined system. This can range from using simple control schemes derived using state feedback to using optimal nonlinear controllers such as linear quadratic regulators and sliding mode controllers as well as artificial intelligence-based controllers like fuzzy logic controllers and neural network-based controllers. Initially, the derived the equations of motion of a rigid body spacecraft in space based on Euler's method are used to implement a simple proportional derivative (PD)-based controller with a feedback system. Then, a fuzzy based controller is implemented to further improve the performance. After that, an adaptive PD-based controller along with estimated inertia is used to control the system. Finally, a neural network-based adaptive controller is introduced as in improvement over these controllers. Although neural network-based controllers could require a high processing power to complete the calculations, it is assumed that this controller can be strategically utilized for the stabilization process of the combined system soon after the capturing procedure is completed.

1.2.3. Controlling satellites under fuel slosh situation.

The third objective of this research is to control the attitude of the satellites under fuel slosh situations. A satellite comprises rigid and flexible components. When the satellite is in transitional or rotational motion, it generates additional forces and moments in these parts. Generally, free moving liquid fuel is stored inside fixed containers inside the satellite. The above-mentioned movements can cause oscillations and arbitrary motion of the liquid. This leads to changes in the centre of mass of the satellite and leads to inaccuracies and disturbances in the stability of the system. Because the satellites can store large amounts of total mass in liquid form to carry out missions in space, the vibrations and disturbances can be considerably large and must be compensated properly

when designing the control algorithm.

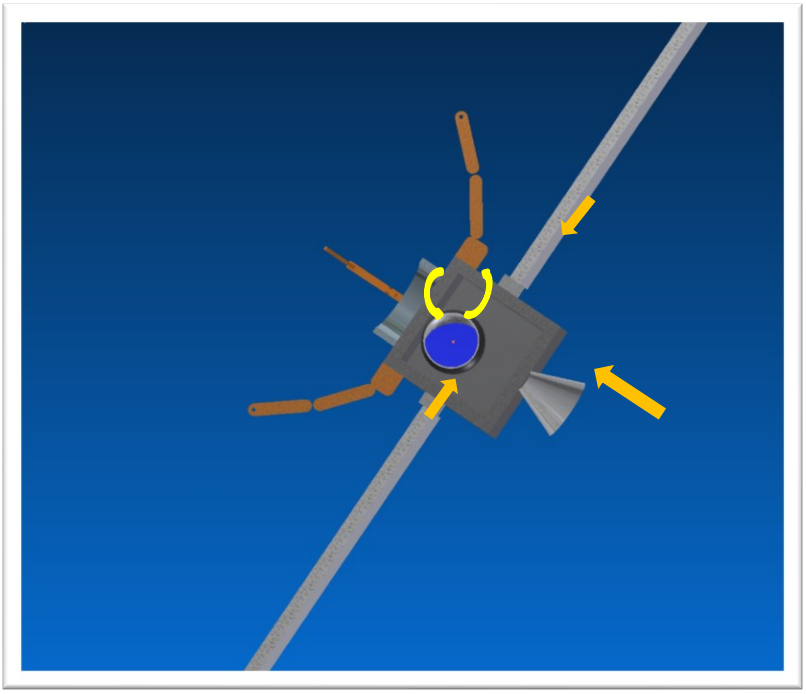

Figure 1.5 Fuel slosh phenomena as seen when moving on a fixed frame

To solve this, a novel sliding mode control-based algorithm is developed to supress fuel slosh while achieving the control goals. Sliding mode control (SMC) is considered a robust control method with the added advantages of resilience toward disturbances and system uncertainties. However, chattering phenomena form a inherent drawback of this method. By incorporating the proposed SMC, this issue can be solved in the control design phase. Furthermore, with the addition of optimal control concepts, which is a systematic approach to solve the required performance criteria, this approach is further expanded. The optimization process is performed offline, thereby reducing the computational requirements for the onboard controllers. The performance analysis is performed using computer-based simulations, and the results are compared with those of conventional controllers in the form of conventional SMC and PID control to verify the effectiveness of the proposed algorithm.

CHAPTER 2
MPC-BASED ORBIT CONTROLLER DESIGN FOR THE FLY-AROUND SCENARIO CONSIDERING FUEL OPTIMIZATION

2.1. Background and Literature Review

Currently, public and private organizations around the world are seeking solutions to tackle the space debris problem. The Japan Aerospace Exploration Agency (JAXA) and the European Space Agency (ESA) are public organizations that have initiated debris removal methods. JAXA proposed deployment of tethers as a solution to debris mitigation in the KITE mission [2] carried out in 2016. ClearSpace-1 [3] is supervised by ESA to conduct space debris removal using a satellite equipped with robotic arms. Private companies such as Kawasaki Heavy Industries and Astroscale are also performing demonstrations to find viable solutions to the space debris problem. The former has a planned demonstration mission, Debris Removal Unprecedented Micro-Satellite, to capture a mock object using an extendable boom as the capture mechanism [4]. Astroscale's ELSA-d is based on magnetic structures placed on the target itself for retrieval through the capture process [5].

In the above situations, a thrust is generated by the satellites' propulsion system to deorbit the debris once it is captured by a grappling mechanism. Debris is an uncooperative object that undergoes arbitrary rotational motion. The chaser satellite should fly in such a way that both the target and the chaser align in a specific way and the relative rotational motion is canceled out. Several studies have been conducted regarding the relative motion between objects in space. Zhenqi et al. [6] described the use of a state-dependent Riccatti equation–based control and linear quadratic regulator (LQR)–based control to maintain a constant distance from a target body while minimizing fuel consumption and settling time. Kumar [7] investigated the continuous thrust Clohessy–

Wiltshire (CW) model with varying initial conditions for low-thrust relative motion. Here, optimization is performed considering state and co-state vectors and is formulated as a two-point value problem.

With the limited amount of resources available for satellite missions in space and restrictions caused by propulsion systems in controller implementation in practical situations, these optimization strategies must be constrained when designing a control system [8]. For such cases, model predictive control (MPC) is an attractive alternative because of its constraint optimization capabilities. The conventional MPC, for example, can be convenient when systems are equipped with resources to compensate for the extensive processing performance. For example [9], an MPC based algorithm was developed for a satellite with the objective of docking with a tumbling object. Here, multiple constraints are considered in relation with control and docking parameters. The objective function is derived to minimize the fuel usage while realizing the required tracking objectives. Another MPC-based controller was proposed in [10] for a spacecraft docking situation. The cost function was based on the unconstrained linear quadratic (LQ) problem with fixed and time varying constraints. By incorporating soft and hard constraints they reduced the computational needs for the MPC problem. This is further expanded to use as an explicit MPC controller to remove the need of an onboard solver. In both situations, MPC is used with linearized constraints, and optimization problems have been solved using quadratic programming methods with high horizon parameter values. In close-proximity rendezvous control, a longer control horizon might be a requirement for adequate system behavior. This is vital when the actuator outputs are very small for delicate maneuvers. When coupled with small sampling rates and high variable count in the optimization problem, this approach can demand extensive computational power from the onboard computer and may require a dedicated solver in the chasing

satellite [11]. One solution is solving the optimization problem explicitly, as realized in [12] and [13], where the control gains are mapped as polyhedral regions and are used as lookup tables in the online calculations to minimize the burden on computational requirements. However, this can also lead to high memory requirements with a high number of states and control inputs [14]. This issue can be addressed with a region-free-based MPC design as studied in [15] and [16], where instead of storing the critical regions, the solutions for locally optimal active sets are calculated offline and stored together with dual variables online.

The rest of the chapter is structured as follows. With the methodology, the controller design based on MPC and other control schemes are discussed. Next, the simulation results pertaining to the derived control algorithms and dynamic equations are provided. A summary of the work is mentioned at the end.

2.2. Methodology

2.2.1. Relative Motion between Two Objects

Let us consider the earth's center as the origin with a target and a chaser orbiting the earth, as denoted in Fig. 2.1. Assuming the target body as the center of a moving reference frame, [x, y, z] completes the right-hand orthogonal coordinate system, with each axis pointing toward the orbital radius, transverse, and angular momentum directions.

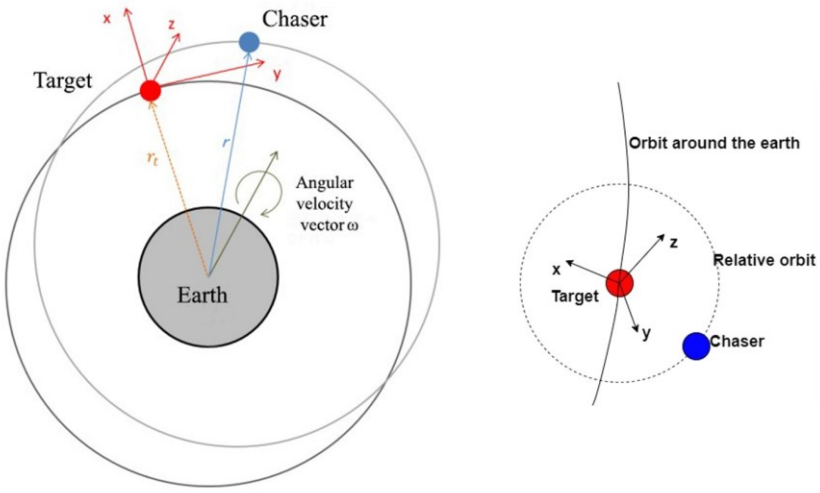

Figure 2.1 The target and chaser in LVLH frame with origin at the centre of the Earth

When the perturbations are assumed to be negligible, the dynamics equations of relative motion between the target and chaser can be derived by solving the corresponding second-order differential equations. When the distance between the chaser and target is relatively small compared to the distance between the target and the center of the earth, and the target's orbit happens to be of circular motion; CW or Hill's equation can be derived as follows [17]:

$$\ddot{x} = 3\omega^2 x + 2\omega\dot{y} + \alpha_x. \tag{1}$$

$$\ddot{y} = -2\omega\dot{x} + \alpha_y. \tag{2}$$

$$\ddot{z} = -\omega^2 z + \alpha_z. \tag{3}$$

$$\omega = \sqrt{\mu/r^3}. \tag{4}$$

where (x, y, z) are the position coordinates, and ($\dot{x}$, $\dot{y}$, $\dot{z}$) correspond to the relative velocities. μ is the geocentric gravitational constant, r is the orbital radius, and

$(\alpha_x, \alpha_y, \alpha_z)$ are acceleration components of the chaser. By using position and velocity coordinates as state variables, the continuous-time state equation can be obtained as follows:

$$\dot{\overline{x}} = A\overline{x} + B\mathrm{u}. \tag{5}$$

With the matrices being

$$\overline{x} = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}. \tag{6}$$

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3\omega^2 & 0 & 0 & 0 & 2\omega & 0 \\ 0 & 0 & 0 & -2\omega & 0 & 0 \\ 0 & 0 & -\omega^2 & 0 & 0 & 0 \end{bmatrix}. \tag{7}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{8}$$

$$u = \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \end{bmatrix}. \tag{9}$$

2.2.2. MPC Design

MPC is a control method in which a short time span is initially considered, and the control input is optimized within that period to bring the predicted system output closer to the

required system output. It is then transferred to the actual system to observe the behavior of the output. The time span is then moved a step forward, and the whole process is repeated to calculate the next control input. This process is performed iteratively to manipulate system behavior at each time step while moving forward. Generally, the method used to determine the manipulation quantity is based on the optimization problem. While solving optimization problems that minimize the evaluation function for a certain interval, the control quantity is expressed in terms of the manipulation quantity using the state equation (dynamic model). The advantage of MPC over other control methods is that, although the state and input constraints are limited to closed set constraints, the target can be controlled by explicitly stating the input. The discrete-time state space model is described below.

$$x_p(k+1) = A_p x_p(k) + B_p u(k). \tag{10}$$

$$y_p(k) = C_p x_p(k). \tag{11}$$

where $x_p(k)$ and $u(k)$ are the state quantity vector and control input vector at time step k, respectively. A, B, and C are the state, input, and output matrices, respectively. Moreover, by defining the difference in the state variable delta and the control input variable delta, we get

$$\Delta x_p(k) = x_p(k) - x_p(k-1). \tag{12}$$

$$\Delta u(k) = u(k) - u(k-1). \tag{13}$$

The increment in the state and output can be written as

$$\Delta x_p(k+1) = A_p \Delta x_p(k) + B_p \Delta u(k). \tag{14}$$

$$y_p(k+1) - y_p(k) = C_p A_p \Delta x(k) + C_p B_p \Delta u(k). \tag{15}$$

Considering a new state variable using

$$x(k) = \begin{bmatrix} \Delta x_p(k)^T \\ y_p(k) \end{bmatrix}^T. \tag{16}$$

(10) and (11) can be rewritten as follows:

$$x(k+1) = Ax(k) + B\Delta u(k). \tag{17}$$

$$y(k) = Cx(k). \tag{18}$$

with the matrices being

$$A = \begin{bmatrix} A_p & {0_p}^T \\ C_p A_p & 1 \end{bmatrix}. \tag{19}$$

$$B = \begin{bmatrix} B_p \\ C_p B_p \end{bmatrix}. \tag{20}$$

$$C = [0_p \quad 1]. \tag{21}$$

where $0_p = [\,0\ 0 \ldots 0]$. The following optimization problem to minimize the evaluation function can then be solved for each time step. Here, $r \in R^n$, $Q \in R^{n \times n}$, and $R \in R^{n \times n}$ are the reference, input, and weight coefficients, respectively. Hp and Hu are the prediction and control horizon, respectively.

$$\min_{\Delta u}\Big(\sum_{i=1}^{Hp}[x(i|k)-r(i|k)]^T Q[x(i|k)-r(i|k)] + \sum_{i=0}^{Hu-1}\Delta u(i|k)^T R\Delta u(i|k)\Big). \tag{22}$$

such that

$$x(k+1) = Ax(k) + B\Delta u(k). \tag{23}$$

$$u_{max} \geq u \geq u_{min} \tag{24}$$

With the control input column Δu denoted by

$$\Delta U(k) = [\Delta u(k), \ldots\ldots, \Delta u(k+Hu-1)]^T. \tag{25}$$

the optimization problem is given by

$$\min_{\Delta U(k)} V(= \tfrac{1}{2}\Delta U^T H\Delta U + \theta^T(k)F\Delta U). \tag{26}$$

such that

$$G\Delta U \leq W + S\theta(k). \tag{27}$$

The variables are $Q = diag[] \in R^{Hp\times Hp}, R = diag[] \in R^{Hu\times Hu}$, $G \in R^{2mHu\times mHu}$, $W \in R^{2mHu\times 1}$, and $S \in R^{2mHu\times(2n+m)}$

$$F = \begin{bmatrix} 2\varphi^T Q\emptyset \\ 0_m \\ -2Q\emptyset \end{bmatrix}, \theta(k) = \begin{bmatrix} x(k) \\ u(k-1) \\ r \end{bmatrix}. \tag{28}$$

$$H = 2R + 2\emptyset^T Q\emptyset. \tag{29}$$

$$\varphi = \begin{bmatrix} CA \\ \vdots \\ CA^{Hp} \end{bmatrix}. \tag{30}$$

$$\emptyset = \begin{bmatrix} CB & 0 & \cdots & 0 \\ CAB & CB & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ CA^{Hu-1}B & CA^{Hu-2}B & \cdots & CB \end{bmatrix}. \tag{31}$$

With Eq. (26), the quadratic problem can be considered as a problem of minimizing V by updating the initial value θ(k) in every time step while taking ΔU(k) as a variable. Here, if the Hessian matrix H is symmetric and semi-positive definite, V can be considered as a convex function. In addition, if the constraint (GΔU ≤ W + Sθ(k)) is a closed set, it is possible to obtain a global optimal solution for this optimization problem using quadratic programming–based solvers.

2.2.3. Region-Free MPC

In Explicit MPC, instead of solving Eq. (26–27) at each time step (hence the online programming), calculations are performed offline for the total range of θ to derive the optimum controller gains, making it a multiparametric quadratic programming problem. In such a case, for $\theta(k)$ and $\Delta U(k)$, the following relationship can be explicitly applied as a piecewise affine state feedback equivalent to MPC:

$$\Delta U(k) = K_j\, \theta(k) + h_j. \tag{32}$$

$$if\ \theta \in X_j$$

where X_j denotes the polyhedral critical regions [18].

In region-free MPC, all possible combinations of constraints, active or inactive, are calculated in advance offline. Next, for each of these combinations, the relationship between the initial value θ and the optimal input ΔU* is stored in the feedback format. For online calculations, the combination to which the initial value θ belongs is then identified, and the optimal input ΔU*is calculated. Both rely on the Karush–Kuhn–Tucker (KKT) conditions to solve the optimization problem [16]. KKT conditions are the conditions that must be satisfied by the optimal ΔU* and dual variable z*.

$$H\Delta U^* + F^T\theta + {G_A}^T z^* = 0. \tag{33.1}$$

$$G_{\mathcal{A}}\Delta U^* = W_{\mathcal{A}} + S_{\mathcal{A}}\theta. \tag{33.2}$$

$$G_N\Delta U^* < W_N + S_N\theta. \tag{33.3}$$

$$z^* > 0. \tag{33.4}$$

$${z^*}^T(G_{\mathcal{A}}\Delta U^* - W_{\mathcal{A}} - S_{\mathcal{A}}\theta) = 0. \tag{33.5}$$

where $\mathcal{A}\ and\ N$ are the sets of active and inactive constraints, respectively. If a solution for the KKT conditions exists, the combination $(\mathcal{A}, N)$ of active and inactive constraints can be considered to exist for a certain value of θ. The number of all combinations can be very large with an increase in the number of constraints. However, the presence of even one inactive constraint makes the combination inactive. The optimal solution for a combination can be obtained using Eq. (33.1).

$$\Delta U^* = -H^{-1}(F^T\theta + {G_{\mathcal{A}}}^T z^*). \tag{34.1}$$

$$\Delta U^* = V_{\mathcal{A}}\theta + v_{\mathcal{A}} z^*. \tag{34.2}$$

$$V_{\mathcal{A}} = -H^{-1}F^T. \tag{34.4}$$

$$v_{\mathcal{A}} = -H^{-1}{G_{\mathcal{A}}}^T. \tag{34.5}$$

Substituting ΔU* in Eq. (33.2) yields

$$z^* = -\left(G_{\mathcal{A}}H^{-1}{G_{\mathcal{A}}}^T\right)^{-1}(W_{\mathcal{A}} + (S_{\mathcal{A}} + G_{\mathcal{A}}H^{-1}F^T)\theta). \tag{35.1}$$

$$z^* = Q_{\mathcal{A}}\theta + q_{\mathcal{A}}. \tag{35.2}$$

$$q_{\mathcal{A}} = -\left(G_{\mathcal{A}}H^{-1}{G_{\mathcal{A}}}^T\right)^{-1}W_{\mathcal{A}}. \tag{35.3}$$

$$Q_{\mathcal{A}} = -\left(G_{\mathcal{A}}H^{-1}{G_{\mathcal{A}}}^T\right)^{-1}(S_{\mathcal{A}} + G_{\mathcal{A}}H^{-1}F^T). \tag{35.4}$$

$V_{\mathcal{A}}$, $v_{\mathcal{A}}$, $Q_{\mathcal{A}}$, and $q_{\mathcal{A}}$ can be generated offline and used in the online implementation using Eq. (34-35) to calculate ΔU^* while fulfilling the conditions below, thereby reducing the amount of memory required in comparison to storing all the combinations offline.

$$z^* > 0 \tag{36}$$

$$G_{\mathcal{A}}\Delta U^{*} - (W_{\mathcal{A}} + S_{\mathcal{A}}\theta\ ) \leq 0 \tag{37}$$

2.2.4. Linear Quadratic Regulator (LQR)

To compare the performance among several controllers, an LQR is initially utilized as a type of optimal control strategy. It uses the state feedback of the system to measure the states and generates a control response while minimizing the cost function V_c. Assume the system is given as

$$\mathrm{X(k+1) = AX(k) + BU(k).} \tag{38}$$

The cost function is

$$V_c = \min \sum_{n=0}^{\infty} x^T Q x + u^T R u. \tag{39}$$

Q and R denote the weighting matrices corresponding to the states and input matrices, respectively. Optimal gain K is calculated by solving the discrete-time algebraic Riccati equation.

$$K = (B^T X B + R)^{-1} B^T X A. \tag{40}$$

$$A^T X A - X - A^T X B (B^T X B R\ )^{-1}\ B^T X A + Q = 0. \tag{41}$$

2.2.5. C-W Based Control (Clohessy-Wiltshire)

Using the CW solution, with a known initial position (x_0, y_0, z_0) and velocity $(\dot{x}_0, \dot{y}_0, \dot{z}_0)$, the change in velocity ΔV required to maneuver to an arbitrary position at

$t = t_0 + \tau$ is calculated using the equation below [19].

$$\begin{bmatrix} \Delta V_x \\ \Delta V_y \\ \Delta V_z \end{bmatrix} = \frac{\omega}{K} \begin{bmatrix} A_1 & A_2 & 0 \\ A_3 & A_4 & 0 \\ 0 & 0 & A_5 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} - \begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{bmatrix}. \tag{42}$$

where

$$A_1 = 4\,sin\,\omega\tau - 3\omega\tau\,cos\,\omega\tau. \tag{43.1}$$

$$A_2 = -2(1 - cos\,\omega\tau). \tag{43.2}$$

$$A_3 = -6\omega\tau\,sin\,\omega\tau + 14(1 - cos\,\omega\tau). \tag{43.3}$$

$$A_4 = sin\,\omega\tau. \tag{43.4}$$

$$A_5 = -\frac{K}{tan\,\omega\tau}. \tag{43.5}$$

$$K = 3\omega\tau\,sin\,\omega\tau - 8(1 - cos\,\omega\tau). \tag{43.6}$$

2.2.6. Quantizer Design

Because propulsion systems have limitations in the activation procedure, they cannot generate smaller incremental outputs similar to those generated from the control system. Hence, a static quantizer is used to simulate limitations on a propulsion system regarding,

- Control period: the minimum time required to go from ignition at thrust ON to burn stop at thrust OFF and vice versa.

- Maximum thrust output: the upper limit of the generated thrust.
- ON and OFF input: the three output values corresponding to the thrust.

The control period is solved by setting the time-step in the discrete state equation used during orbit control. The maximum thrust and ON/OFF constraint is solved using the below equation to turn the generated control input from the control algorithm to one of the following three possible outputs from the quantizer:

$$u = \begin{cases} u_{max} & (u \geq u_{max}/2) \\ 0 & (u_{max}/2 > u > u_{min}/2) \\ u_{min} & (u \leq u_{min}/2) \end{cases} \tag{44}$$

The flow diagram denoting the controller implementation is indicated in Fig. 2.2. The path that the satellite should follow is fed into the controller in the form of a reference input signal. The error between the reference and the satellite's feedback is calculated and used by the controllers to generate control signals, which are fed into the quantizer to simulate the output of the propulsion system. Subsequently, this value is provided into the satellite's dynamic model for simulation, and the generated output is used as feedback for the error calculation in the next control iteration.

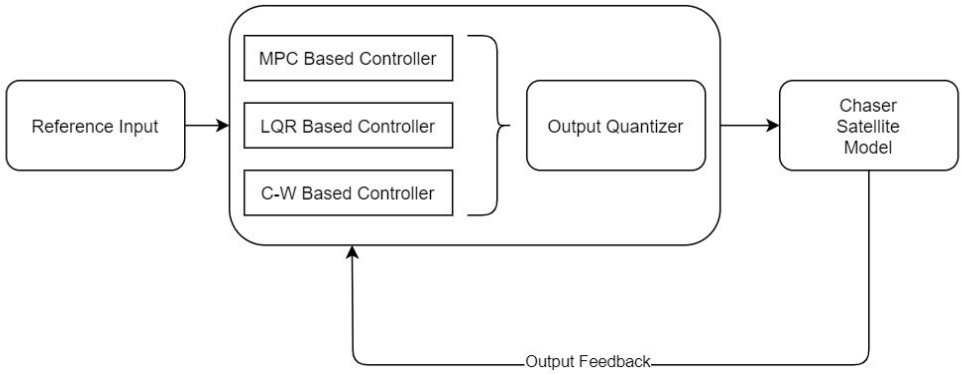

Figure 2.2 Satellite control system implementation.

2.3. Simulation Results

For the fly-around simulation, only the X–Y plane is considered. The parameters and their corresponding values used in the simulation are listed in Table (2.1).

Table 2.1 Simulation conditions considered in this study.

Parameter	Value
Orbit period	400 s
Orbit radius	4m
Initial values $(x_0, y_0, \dot{x}_0, \dot{y}_0)$ [m, ms^{-1}]	(0,-5,0,0)
Input constraints [mms^{-2}]	$\lvert u_x \rvert \leq 2.88$
	$\lvert u_y \rvert \leq 2.88$
Coefficient matrix Q	diag [1,1,1,1]
Coefficient matrix R	diag [1,1]
Prediction horizon	6
Control horizon	3
Discrete time period	1s- CW
	2s- LQR
	2s- MPC

Simulation is performed for a period of 1000 s for each control scheme, and Fig. 2.3 expresses the outputs generated by the satellite model on each occasion. The target and chaser orbit the earth in a circular path, and the relative motion between the two at close proximity is shown in the figure in detail. Here, the target is assumed to be an object situated in the centre of the given figure, with the chaser satellite orbiting it in a circular motion.

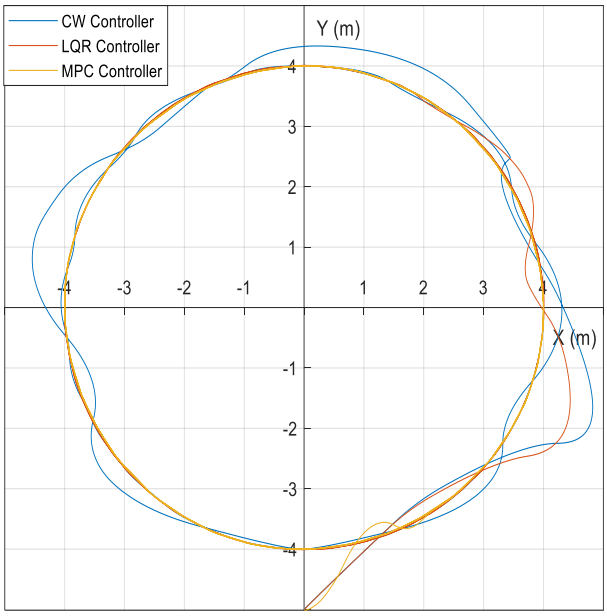

Figure 2.3 Comparison of the fly-around orbits with the three controllers.

Figs. 2.4 to 2.6 show the changes in radius related to the MPC, LQR, and CW-based controllers. The reference input given here is to follow a fixed 4-m radius circle. The MPC-based controller moves from its initial position to the reference position aggressively in a short period, albeit with a slightly unsteady movement along the total simulation time because of the shorter control horizon. LQR-based controllers take a much longer time to settle into the reference input because it must counter the variations due to the quantizer. The CW controller has the least success following the given reference, as it is not a robust controller compared to the other two controllers.

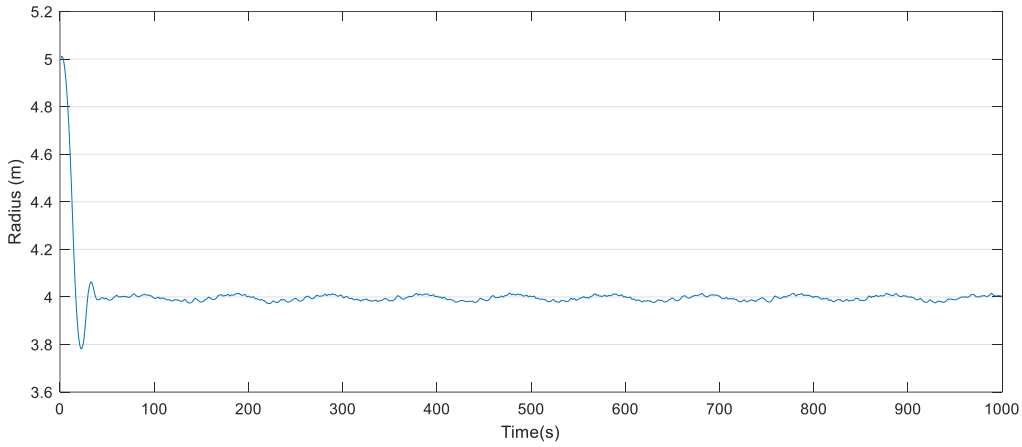

Figure 2.4 Change in fly-around radius with the MPC-based controller.

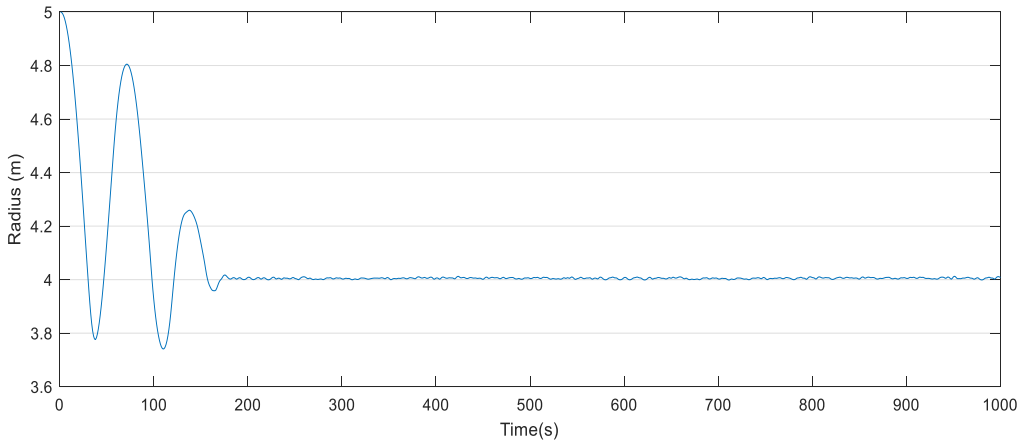

Figure 2.5 Changes in the fly-around radius with the LQR-based controller.

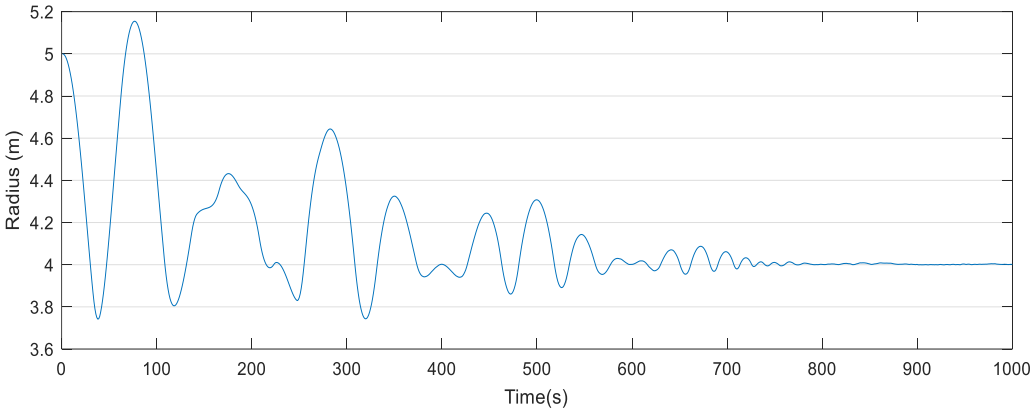

Figure 2.6 Changes in the fly-around radius with the C-W based controller.

Figs. 2.7 to 2.9 denote the changes in angular velocities of the MPC-, LQR- and CW-based controllers when following the circular reference input with a constant angular

velocity. Here, the data again point to faster convergence of the MPC controller toward the required reference value. The LQR method has the least wobbling around the target value of 0.9 deg/s, but it takes almost 10 times as long as the MPC method to reduce the initial speed closer to the target value. The CW method fails to bring the satellite's speed to a respectable value until the last quarter of the simulation time because it is severely affected by the quantizer's restrictions and variations in reference inputs at each time step along the X and Y directions.

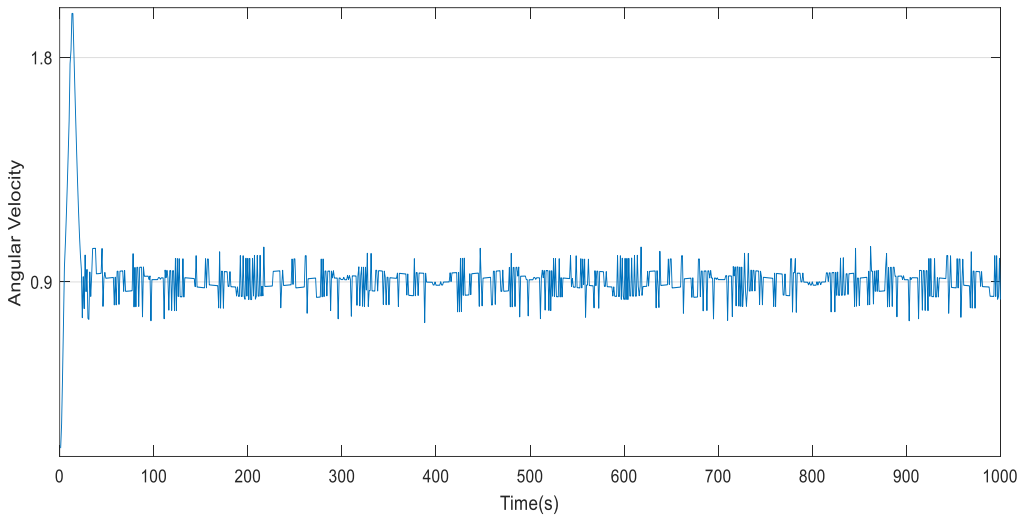

Figure 2.7 Changes in the fly-around angular velocity with the MPC-based controller.

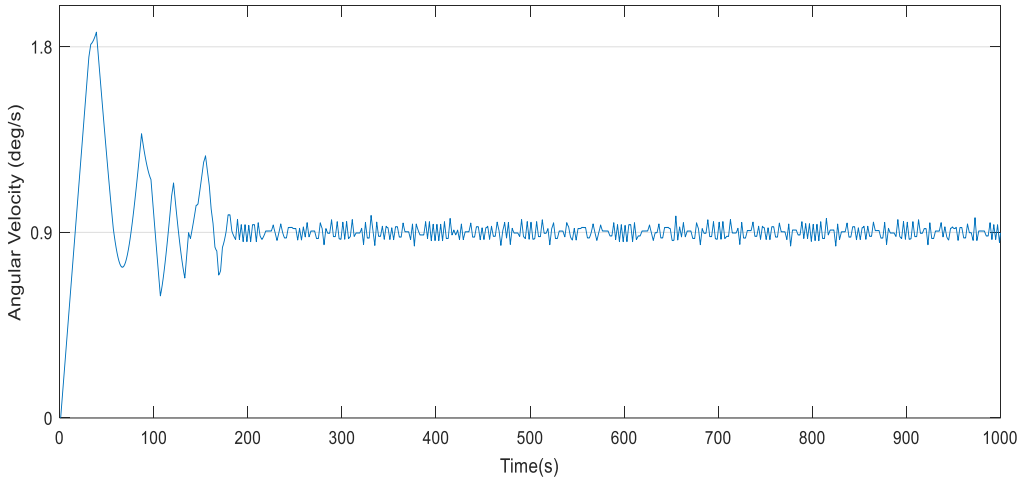

Figure 2.8 Changes in the fly-around angular velocity with the LQR-based controller.

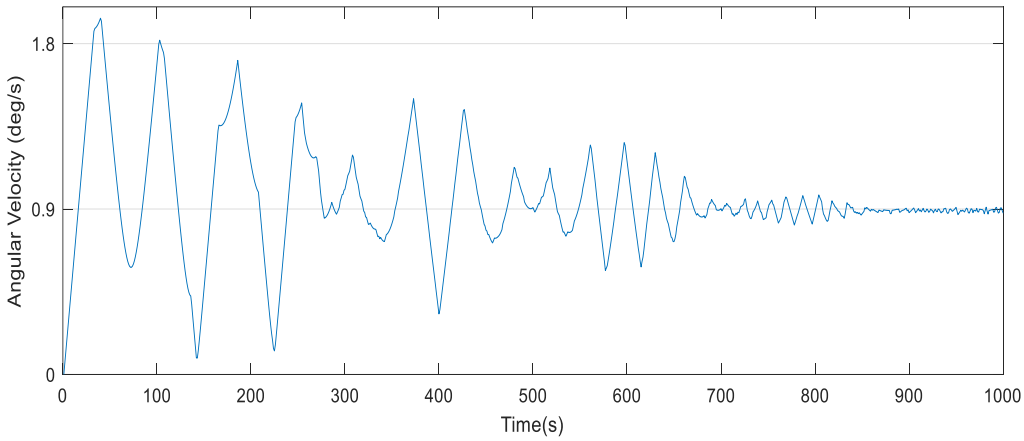

Figure 2.9 Changes in the fly-around angular velocity with the C-W based controller.

The chaser satellite is assumed to have two pairs of thrusters to control the movement of the satellite in each direction. Fig. 2.10 and Fig. 2.11 denote the commanded inputs (generated from the controller) and the control inputs (generated from the quantizer) for each axis with the MPC-based controller. Because the MPC method is based on constrained optimization, the commanded inputs are within the maximum values that can be generated by the thrusters.

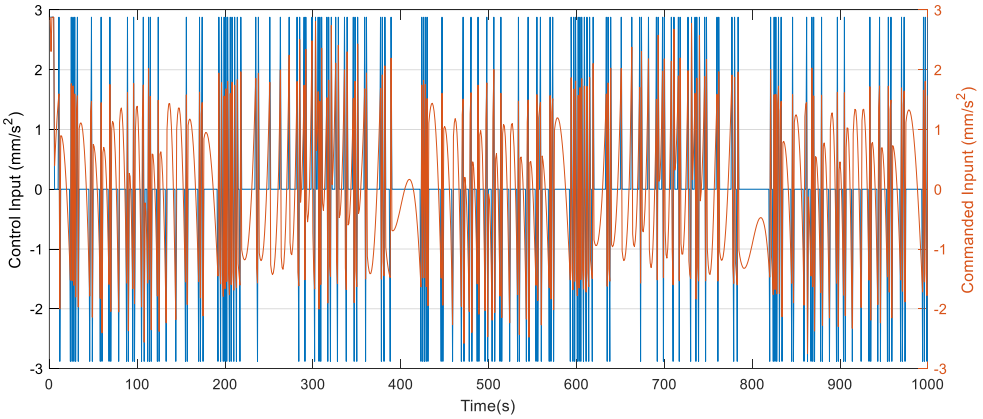

Figure 2.10 Generated control input for a fly-around scenario with the MPC-based controller in the X direction.

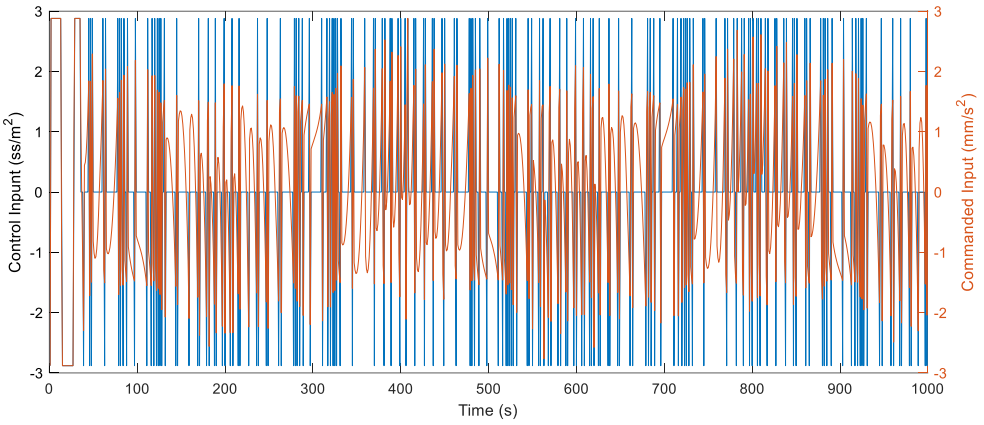

Figure 2.11 Generated control input for a fly-around scenario with the MPC-based controller in the Y direction.

Figs. 2.12 to 2.15 denote the outputs from the LQR and CW methods to the thrusters. On both occasions, the commanded inputs (from the controllers) are much higher than the thrusters' maximum values. Thus, they both fail to bring the satellite's position and angular velocity errors to a minimum value within a short period compared with the MPC method.

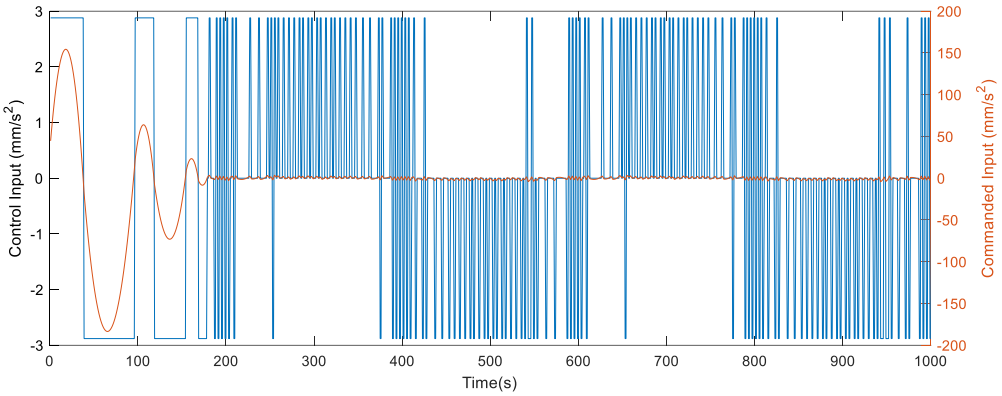

Figure 2.12 Generated control input for a fly-around scenario with the LQR-based controller in the X direction.

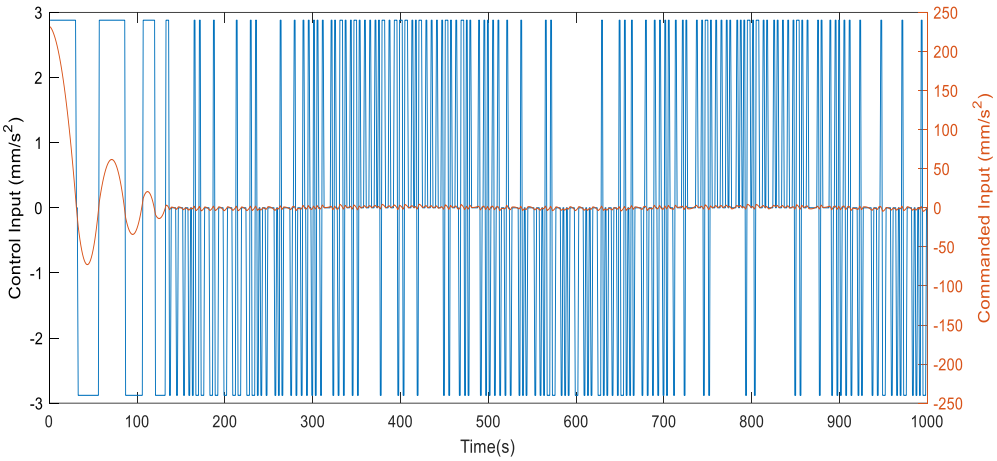

Figure 2.13 Generated control input for a fly-around scenario with the LQR-based controller in the Y direction.

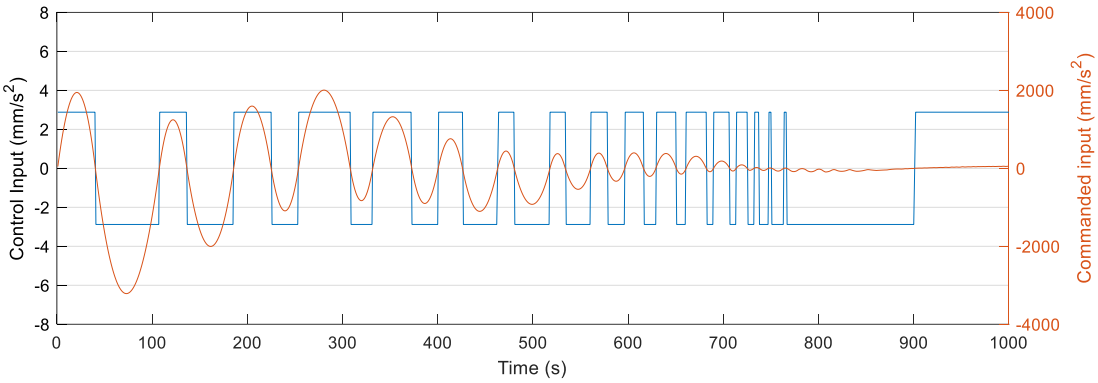

Figure 2.14 Generated control input for a fly-around scenario with the CW-based controller in the X direction.

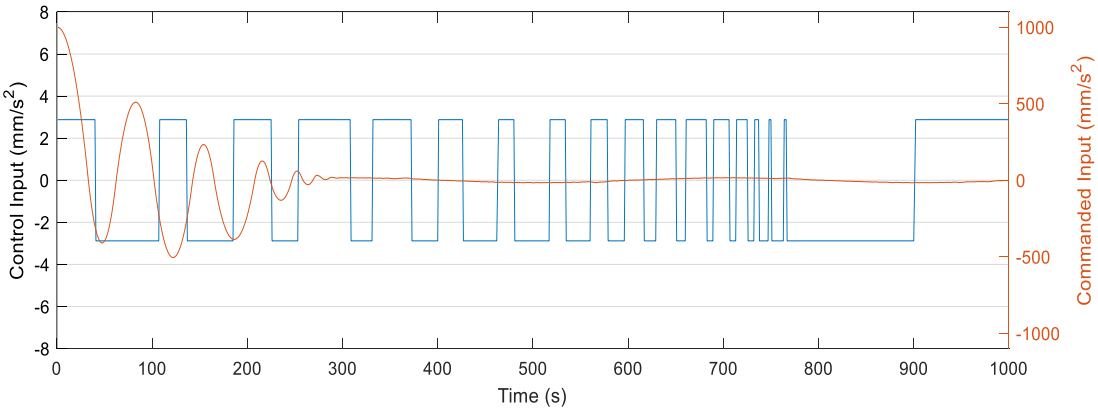

Figure 2.15 Generated control input for a fly-around scenario with the CW-based controller in the Y direction.

Table (2.2) corresponds to the numerical evaluation of the root mean square errors (RMSE) pertaining to the position and angular velocity errors with the three

controllers in the X and Y directions. The MPC-based method achieved better results in position error calculations, whereas the LQR method was superior in the velocity calculations.

Table 2.2 RMSE values for positions and velocities in the X-Y directions.

Controller	Values			
	Position X	Position Y	Velocity X	Velocity Y
C-W	0.314	0.191	0.022	0.014
LQR	0.173	0.143	0.012	0.009
MPC	0.074	0.122	0.022	0.023

In Table (2.3), the total ΔU requirement for the entire simulation is calculated for the corresponding three controllers. Here, ΔU is considered a performance criterion to evaluate the fuel consumption from the start to the end of simulation time T_f, in the following form of:

$$\Delta U = \sum_{i=0}^{T_f} |u_{x,y}(i)| \tag{45}$$

The MPC-based controller has the lowest ΔU requirement because of the constraints used in the optimization. As shown in the previous figures related to the outputs from the controllers and quantizers, when these values are higher than the outputs obtainable from the thrusters, it can take a longer time to switch the thruster directions in the initial phase, which can lead to a larger ΔU requirement to complete the given mission.

Table 2.3 ΔU requirement for a total simulation time of 1000 s.

Controller	ΔU requirement for total simulation time	
	Direction X	Direction Y
C-W	2.877	2.765
LQR	1.549	1.731
MPC	0.821	0.962

In the simulation, a control horizon of 3 was used for the MPC-based algorithm. When the control horizon increases, the required memory capacity also increases by a factor of ~10. A comparison of the memory usage between region-based and region-free MPC is given in Table 2.4.

Table 2.4 Memory requirements with different control horizons for the MPC controller.

Control horizon	Region-free MPC (KB)	Region-based MPC (KB)
1	8.8	18.0
2	120.6	433.8
3	1562.3	10288.4
4	16454.6	230685.6

Even though from the analytical data, we can derive that the MPC-based controller shows better performance compared with the LQR and CW based methods, the graphical data show that the MPC-based controller has some wobbling effects when it tries to follow the required trajectory of 4 m. Because reducing the position error is more important compared with angular velocity error, different combinations of control and prediction horizons were compared to find a suitable solution together with calculation improvement between the conversions of actual control inputs to quantized inputs. The wobbling effects were caused by the time interval of 1 s, which was used for simulations.

Because the target satellite must follow a trajectory with fine margins, the fluctuations that arise within this time interval are higher. This is seen in the Fig. 2.16 that when the prediction horizon has increased, although the settling times reduced, the fluctuations increased.

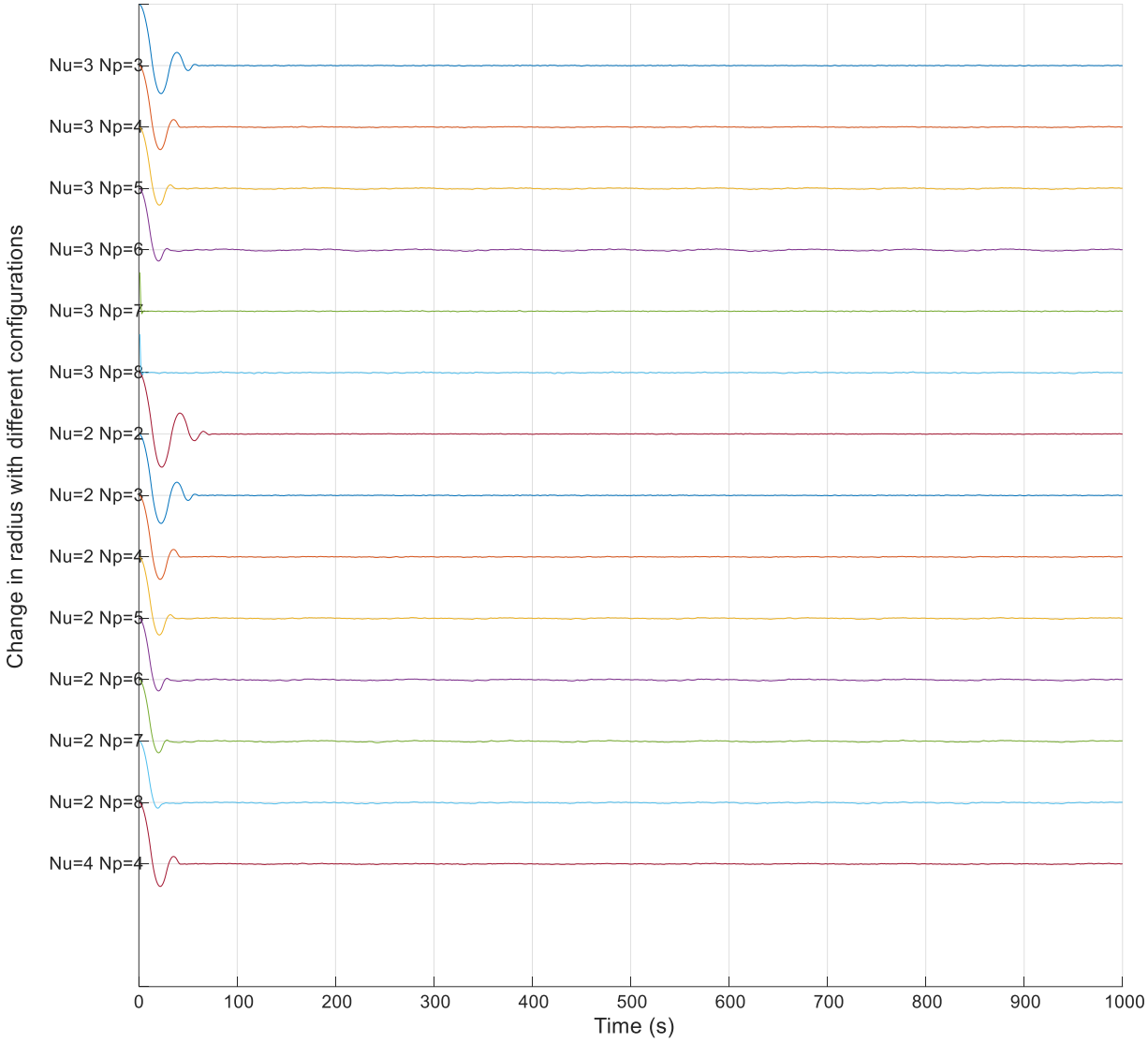

Figure 2.16 Change in radius with different configurations

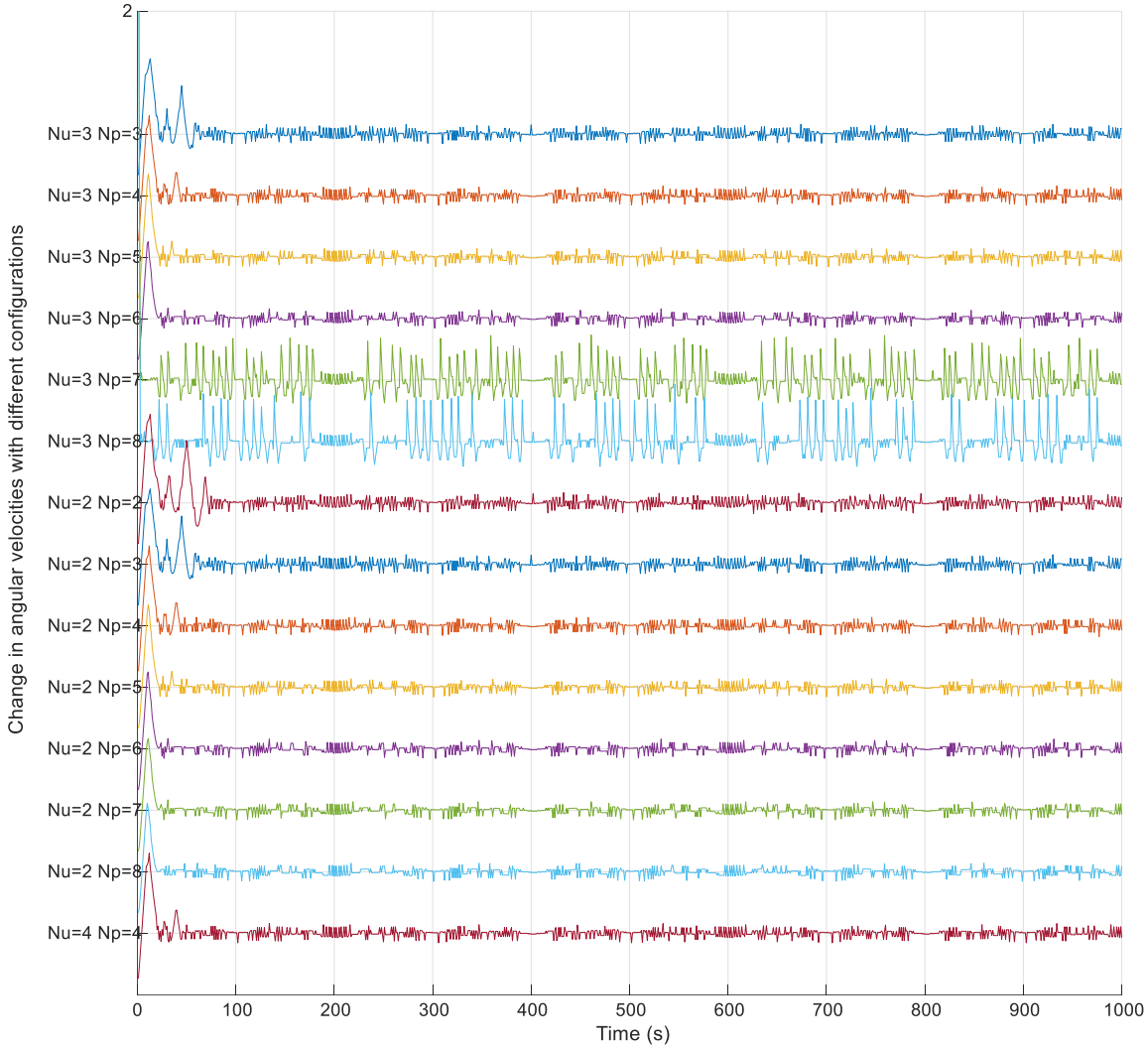

Figure 2.17 Change in angular velocities with different configurations

The RMSE of position errors, angular velocity errors, and total delta V requirement with the different configurations used in the above scenarios are given in Table (2.5). Considering the overall performance in all categories, Nc = 2, Np = 3 and Nc = 2, Np = 4 point to the best results. Fig. 2.18 shows the visualized version of the changes in the trajectory tracking with different combinations. Furthermore, Table (2.6) summarizes the steady state errors which were measured at the end of the simulation time.

Table 2.5 RMSE values and delta V for the considered configurations

Configuration	Position error in x	Position error in y	Velocity error in x	Velocity error in y	Delta V in x	Delta V in y
Nc=3 Np=3	0.0205	0.0366	0.0201	0.0209	1.1117	1.3651
Nc=3 Np=4	0.0404	0.0519	0.0201	0.0207	1.0858	1.2154
Nc=3 Np=5	0.0540	0.0631	0.0211	0.2064	0.9763	1.1002
Nc=3 Np=6	0.0643	0.0717	0.0201	0.0206	0.8986	0.9619
Nc=3 Np=7	0.0349	0.0364	0.2009	0.0206	1.8058	2.4249
Nc=3 Np=8	0.0436	0.0473	0.0201	0.0206	1.5350	2.4797
Nc=2 Np=2	0.0208	0.0418	0.0201	0.0210	1.1808	1.3363
Nc=2 Np=3	0.0205	0.0365	0.0201	0.0209	1.1117	1.3651
Nc=2 Np=4	0.0388	0.0503	0.0201	0.0207	1.0685	1.1520
Nc=2 Np=5	0.0542	0.0632	0.0201	0.0206	0.9965	1.4774
Nc=2 Np=6	0.0649	0.0724	0.0201	0.0206	0.9014	0.9677
Nc=2 Np=7	0.0724	0.0798	0.0201	0.0206	0.8093	0.9734
Nc=2 Np=8	0.0777	0.0839	0.0202	0.0205	0.7949	0.9216
Nc=4 Np=4	0.0404	0.0519	0.0201	0.0207	1.0858	1.2136

Table 2.6 Steady state errors with the considered controllers after the simulation time.

Configuration	Stead state error (m)
Nc=3 Np=3	0.0032
Nc=3 Np=4	0.0022
Nc=3 Np=5	0.0029
Nc=3 Np=6	0.0027
Nc=3 Np=7	0.0021
Nc=3 Np=8	0.0018
Nc=2 Np=2	0.0023
Nc=2 Np=3	0.0032
Nc=2 Np=4	0.0016
Nc=2 Np=5	0.0019
Nc=2 Np=6	0.0030
Nc=2 Np=7	0.0034
Nc=2 Np=8	0.0030
Nc=4 Np=4	0.0004
C-W	0.0014
LQR	0.0018

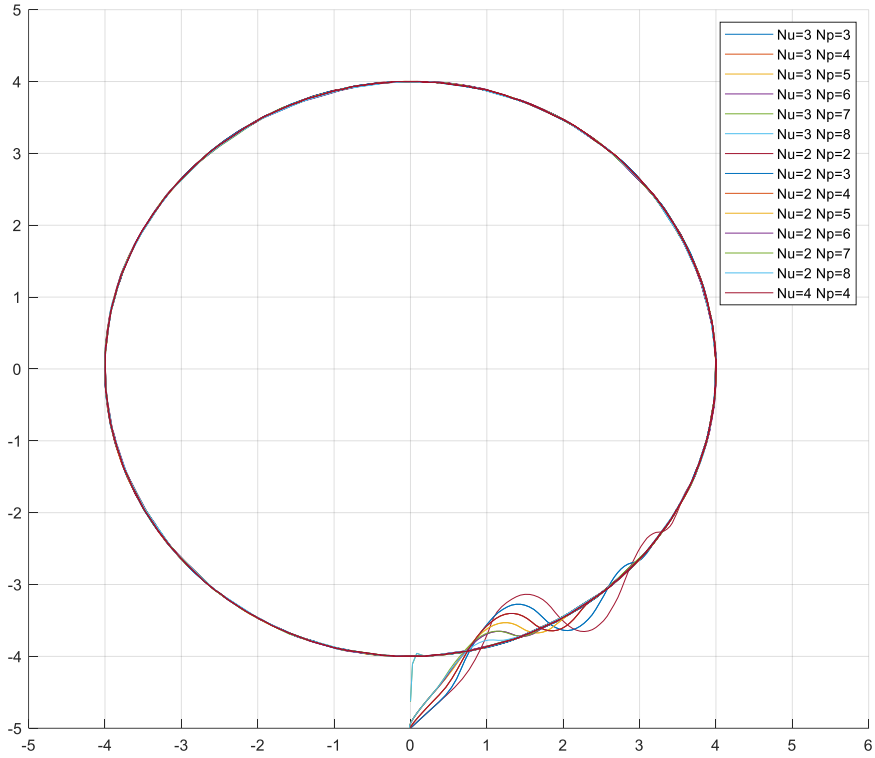

Figure 2.18 Graphical representation of trajectories with different configurations

2.4. Summary

In this chapter, we developed a control algorithm for a fly-around orbit control situation between a target and a chaser satellite with limitations. Initially, the relative motion between two objects orbiting the earth was studied using related equations. To realize fuel optimization while considering the limitations in the storage capabilities of satellites, an MPC-based control algorithm was implemented. The simulation results point to success in the control strategy in minimizing fuel consumption compared to several conventional controls while limiting the memory usage and improving tracking performance.

CHAPTER 3
NEURAL NETWORKS BASED ADAPTIVE ATTITUDE CONTROLLER DESIGN FOR A SATELLITE WITH PARTIAL KNOWN INERTIAL PROPERTIES

3.1. Background and Literature Review

As previously mentioned, a vast majority of research is being conducted to find solutions to this ever growing problem of space debris mitigation [20][21] [2] [3]. However, most debris are uncooperative objects with different sizes and mass properties which require complex design and implementation considerations. Once a debris removal satellite captures such an object with a locking mechanism, the dynamics of the combined system instantly change causing tumbling and spinning effects and hindering its orbital path. Both velocity and orientation of the system should be brought to the desired values for safer maneuvering. Therefore, a robust attitude control algorithm is an integral part in system design. It should be able identify the uncertainties that arise in the mass properties and control the combined system under external torques and perturbations. The main objectives of this research are to estimate the unknown inertial properties of the system and develop a control algorithm that is sufficiently robust to cope with uncertainties in the inertial properties of the system.

When estimating the inertia matrix of a satellite, both moment of inertia and product of inertia must be considered. Considerable research has been conducted in regards to inertia estimation of satellites. Palimaka and Burlton used the weighted least square method to estimate the mass properties [22]. Lee and Wertz proposed an approach for inertia tensor estimation using the least squares method [23]. A A combination of the extended Kalman filter (EKF) and the least squares was proposed in Ref [24] to filter the gryo signals and used batch method to calculate the inertia. A modified law of

conservation of angular momentum was used in Ref [25] to estimate both the moment of inertia and the product of inertia of the STSAT-3 satellite. Yang et al. [26] proposed an improved version for inertia estimation using recursive least squares by combining angular velocity and angular acceleration with filtered angular rates through EKF. Kim et al. [27] focused on the use of extended Kalman and Savitzky-Golay filters for filtering gyro data together with linear least squares method for accurately predicting the inertial matrix.

An extensive amount of research has been conducted to develop attitude control algorithms for satellites. With the probability of fast tumbling effects, large angle maneuvering can be performed in the system. Because rigid body parameters vary in an unpredictable manner, a highly robust controller is required to maneuver the system to a desired orientation. Although nonlinear and robust control theory based algorithms could deliver high performance, it could lead to complexities in the design stage and during implementation [28]. In retrospect, proportional-integral-derivative (PID) controllers are simple and straightforward to implement [29][30][31][32][33].

Another class of controllers includes the fuzzy theory based algorithms that employ the intuition of human expert knowledge in designing complex systems in a simple manner. With the unpredictability of the inertial properties, the control system itself should be able to change its output to compensate for changes in system properties. Intelligent controllers such as those based on fuzzy control are simple to implement and take less burden during computational calculations. With low cost and flexibility for handling nonlinear systems with wider operating conditions, such controllers can perform better than PID controllers [34][35]. Cheng and Shu [36] used two Fuzzy controllers for attitude stabilization and consolidated them to one Fuzzy controller considering that both have similar rule bases and membership functions. Ismail [34] discussed the tuning of

PID gains for steam turbines using the Fuzzy theory. Because PID gains do not guarantee stability under all situations, he used the Fuzzy controller to improve the gain values of the conventional PID controller. Sari et al. [35] expressed his form of an adaptive Fuzzy PID controller for attitude control of a spacecraft. He used two separate Fuzzy inference engines to control each state variable and its priority and then used sliding mode based control to update the gains of the system. Buijtenen [37] focused on the use of reinforcement learning to establish both the critic (evaluation unit) and the adaptive fuzzy controller. A direct adaptive Fuzzy controller is combined with a sliding surface to adequately control the desired orientation of a satellite in the study of Mostafa et al. [38].

When dealing with extreme ambiguities in system parameters intelligent control systems provide more robustness compared to conventional controllers such as PID with improved handling of nonlinearities and adjustments to gains [39]. Neural networks are an efficient control scheme that can deal with the shortcomings of both PID and Fuzzy controllers. Because fuzzy controllers require heuristic information, when the expert knowledge is limited and rule bases are not sufficient, artificial intelligence (AI) based control algorithms such as neural networks have the upper hand because the optimum gain values can be adjusted to meet the expectations of each situation accordingly. Ponce et al. [40] used a neural network based on three layer perceptron, and the output error was considered to adjust the weights of the neural network using a modified backpropagation method. A criteria was given to initialize the parameters while adjusting the learning rate of the system. Yamamoto et al. [41] formulated an adaptive controller based on PID and neural networks. The controller gains for the modeled part were calculated using a self-tuning PID based controller and the unmodelled part gains were derived using a backpropagation based neural network. Hernandez et al. [42] used the backpropagation method to calculate the gains of an auto tuning PID controller type neural network for an

underwater robot. In Ref. [43], Li and Espinosa structured their PID neural network to include both the plant model and the controller model. The first neural network generates the gains for the PID controller, while the second increases the effectiveness of the backpropagation method. Although AI-based control strategies could yield efficient results, they also have the drawback of requiring higher computational power and energy consumption in general. With the ongoing developments of graphical processing unit (GPU)-based onboard controllers for space activities, it will be easier to implement and deploy such controllers for space-related missions [44][45]. This can be further reduced by considering the use of these types of control schemes only for specific mission situations.

The structure of this chapter is as follows. To cope with the complexities of designing such a system, this research concentrated on two areas: inertia estimation and controller design. The system was considered as two separate parts, the chaser and the target. The chaser (assumed to be a satellite) has known inertial properties, and the target's inertial properties are partially known. When the two objects are combined after the capturing procedure, the new system is expected to have a known initial velocity and orientation. The initial process was to identify the system's inertia matrix. Next, the controller system was designed using a conventional PD controller, fuzzy theory-based controller, inertia-based controller, and a neural based adaptive controller. The performance of these controllers was then compared through computer simulations and data analysis.

3.2. Methodology

3.2.1. Estimating the Inertia

Satellites can have different shapes, parts, and physical characteristics, and as a result,

their movement in space can be complex. Let us consider a simple rigid body satellite moving freely in space without any disturbances. The dynamical equation of motion can be expressed in Euler's form [31] with J, ω, u and y being the satellite's inertia matrix, angular velocity, applied torques, and system outputs, respectively.

$$J\dot{\omega} = -\omega \times J\omega + u. \tag{46}$$

It is rearranged in the following form to be used in the regression model.

$$\dot{\omega} = -J^{-1}\omega \times J\omega + J^{-1}u. \tag{47}$$

$$y = \omega. \tag{48}$$

Consider the following expanded matrices for **Ω**, J_E, and cross product of ω.

$$\Omega = \begin{bmatrix} \omega_1 & 0 & 0 & \omega_2 & \omega_3 & 0 \\ 0 & \omega_2 & 0 & \omega_1 & 0 & \omega_3 \\ 0 & 0 & \omega_3 & 0 & \omega_1 & \omega_2 \end{bmatrix}. \tag{49}$$

$$J_E = \begin{bmatrix} J_{xx} & J_{yy} & J_{zz} & J_{xy} & J_{xz} & J_{yz} \end{bmatrix}. \tag{50}$$

$$\omega \times = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \tag{51}$$

The regression model for the linear least squares method and its general solution can be written as follows:

$$z = Hx. \tag{52}$$

$$\hat{x} = (H^T H)^{-1} H^T z. \tag{53}$$

with z, H and x corresponding to the matrices of measurements, regression, and parameters to be estimated. Substitute Eq. (52) with input torques and measurements to solve for the inertia estimation [26][27]. Assume that a constant torque is supplied while inertia is being estimated. Therefore, the regression matrix H, which includes the angular velocity and angular acceleration, is assumed to be a non-zero matrix. Furthermore, because it is a square matrix, the direct inverse can be used to estimate the unknown inertia.

$$\begin{bmatrix} u \\ \int u dt \end{bmatrix} = \begin{bmatrix} \dot{\Omega} + \omega \times \Omega \\ \Omega + \int \omega \times \Omega \, dt \end{bmatrix} J_E. \tag{54}$$

Consider a situation in which the target body is captured by the chaser satellite. The total inertia goes through a sudden change. The inertia matrix of the chaser satellite is taken as follows:

$$\begin{bmatrix} 10000 & 0 & 0 \\ 0 & 9000 & 0 \\ 0 & 0 & 12000 \end{bmatrix} kgm^2 \tag{55}$$

For the target satellite, instead of considering a fixed object, an inertia varying system is considered for simulations. At the maximum variation, the target body is taken as a cube of 300 kg mass with 1 m sides. Then, a sudden change in inertia is assumed to vary between 0 and 5000 kgm^2. Random values between these ranges are generated and added to the satellite inertia, and a constant torque is given to the system for a predefined time. After that, using the above-mentioned regression model, the total inertia is estimated.

3.2.2. Attitude Representation

Different models such as quaternions, Euler-angles, and Rodrigues parameters can be used to represent the orientation of an object in space. Quaternions are used in the calculations in this work because they do not get affected by singularities. Attitude Kinematics are given by the following equation:

$$\dot{q} = \frac{1}{2}\Xi(\omega)\mathrm{q}. \tag{56}$$

With the skew symmetric matrix Ξ and quaternion matrix q being

$$\Xi(\omega) = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix}. \tag{57}$$

$$\mathrm{q} = [\, q_1 \; q_2 \; q_3 \; q_4]'. \tag{58}$$

3.2.3. Proportional-Derivative Controller Design (PD)

Several types of controllers are used for comparison in this research. The first controller is a conventional proportional-derivative (PD) controller in the form of Eq. (59). The control law that is used to drive the system to stability is given as follows:

$$u = k_P q_{1:3} + k_D \omega \; (\; q_{1:3} = [q_1 \; q_2 \; q_3] \;). \tag{59}$$

The selected proportional (k_P) and derivative (k_D) gains are [50,500].

3.2.4. Fuzzy Logic based Controller Design (FBC)

Fuzzy control uses human expertise to develop an intuitive idea regarding how the controller system should work in complex situations. It is then applied to design the

control algorithm in the form of rules. Compared with conventional control, which is based on differential equations, this can yield better results when the dynamical equations are inaccurate and too complex to implement. A general fuzzy model has four components, as mentioned in Fig. 16. Fuzzification aligns inputs to the corresponding rules. The inference mechanism helps in selecting the active rule for each instant. Rule-base is the set of governing rules of the concerned system. Defuzzification deals with converting the decision of the collective model into a control signal. It is then sent to the process or the dynamic model of the system which is to be controlled. The output of the process is then fed back to the fuzzy controller for the next iteration of the control loop.

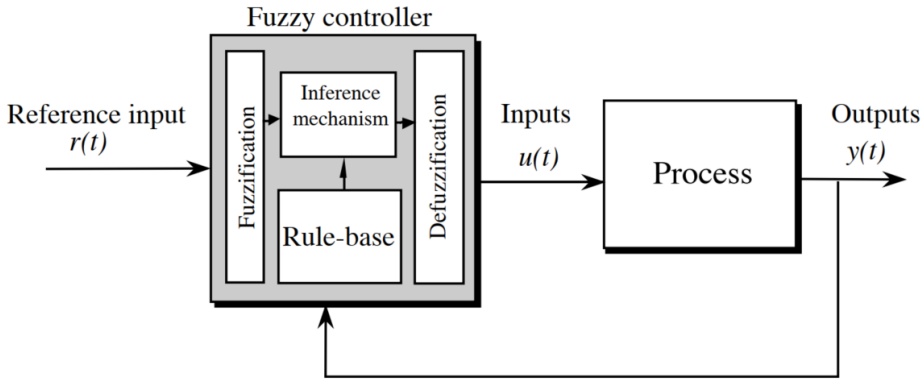

Figure 3.1 Components of a general fuzzy model.

The variations in quaternions and angular velocities are different. As a result, two different fuzzy models are used to minimize each error signal. Each model has two membership functions to map the corresponding error and derivative of error signals. Tables (3.1) and (3.2) denote the rule tables for the two fuzzy models. The ranges of these values are kept similar to those of the PD controller gains used earlier.

Table 3.1 Rule table denoting fuzzy model 1.

Required torque		Rate of change of attitude error		
		n	z	p
Attitude error	n	-50	-25	0
	z	-25	0	25
	p	0	25	50

Table 3.2 Rule table denoting fuzzy model 2.

Required torque		Rate of change of angular velocity				
		nn	n	z	p	pp
Angular velocity	nn	-500	-500	-500	-250	0
	n	-500	-500	-250	0	250
	z	-500	-250	0	250	500
	p	-250	0	250	500	500
	pp	0	250	500	500	500

Membership functions referring to the above mentioned fuzzy models are given in Figs. 3.2 to 3.5 Various types of linear and nonlinear functions can be used for membership functions. These can be combined with input and output scaling to obtain varying results. Here, triangular and trapezoidal functions are used for the membership functions with no scaling. For the horizontal plane values, a combination of uniformly and nonuniformly distributed values have been used after some trials for adequate system performance.

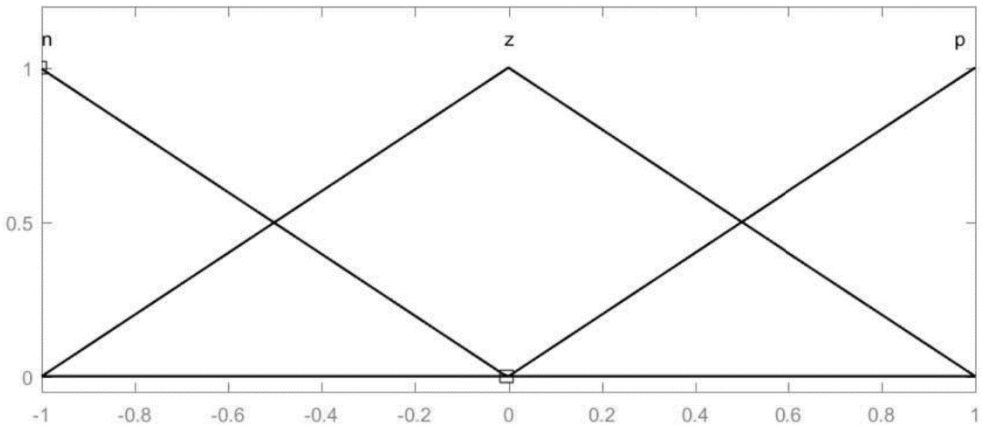

Figure 3.2 Membership function: attitude error.

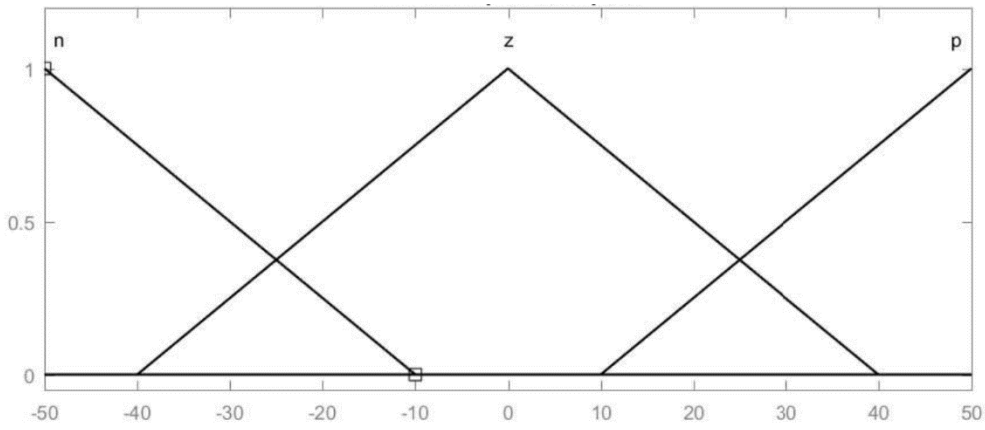

Figure 3.3 Membership function: rate of change of attitude error

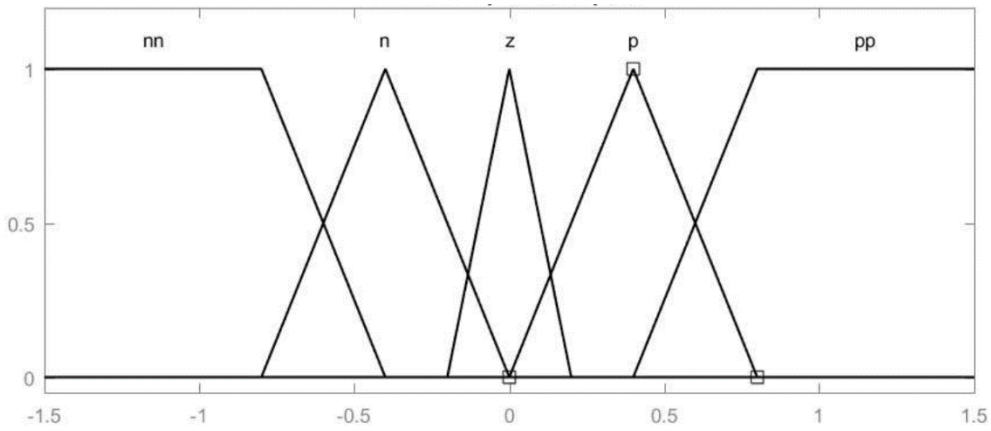

Figure 3.4 Membership function: angular velocity.

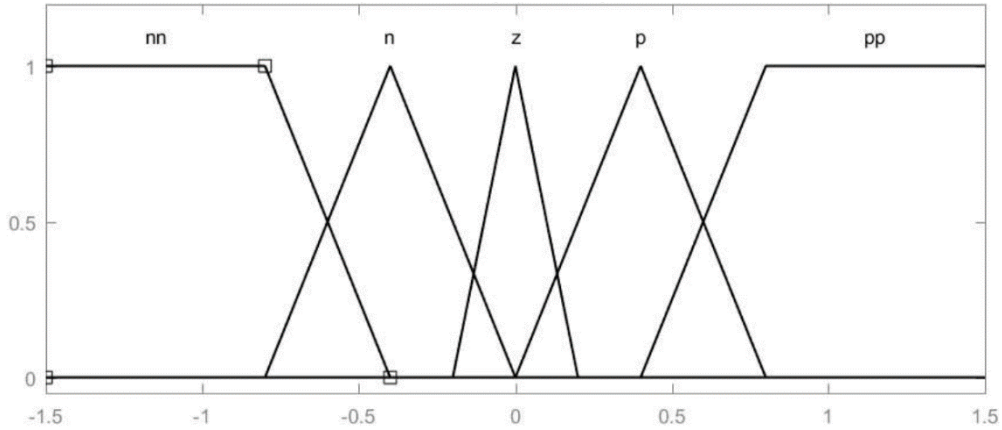

Figure 3.5 Membership function: rate of change of angular velocity.

3.2.5. Inertia based Controller Design (IBC)

A control law based on the estimated inertia [46], along with proportional and derivative gains in the form of Eq. (60), is used as an adaptive controller that utilizes both the estimated inertia matrix and PD equivalent gains to stabilize the system.

$$u = ½p_1 + \psi J_T + kp_2. \tag{60}$$

With constants λ and k and the unknown parameter matrices being

$$p_1 = q_{1:3}. \tag{61}$$

$$p_2 = \lambda p_1 + \omega. \tag{62}$$

$$\psi' = \begin{bmatrix} -a\dot{q}_1 & -\omega_1\omega_3 & \omega_1\omega_2 \\ -a\dot{q}_2 & -\omega_2\omega_3 & {\omega_2}^2 \\ a\dot{q}_3 & -{\omega_3}^2 & \omega_2\omega_3 \\ \omega_1\omega_3 & -a\dot{q}_1 & -{\omega_1}^2 \\ \omega_2\omega_3 & -a\dot{q}_2 & -\omega_1\omega_2 \\ {\omega_3}^2 & -a\dot{q}_3 & -\omega_1\omega_3 \\ -\omega_1\omega_2 & {\omega_1}^2 & -a\dot{q}_1 \\ -{\omega_2}^2 & \omega_1\omega_2 & -a\dot{q}_2 \\ -\omega_2\omega_3 & \omega_1\omega_3 & -a\dot{q}_3 \end{bmatrix}. \tag{63}$$

$$J_T = [J_{xx}\, J_{xy}\, J_{xz}\, J_{yx}\, J_{yy}\, J_{yz}\, J_{zx}\, J_{zy}\, J_{zz}]. \tag{64}$$

3.2.6. Neural based Adaptive Controller Design (NBAC)

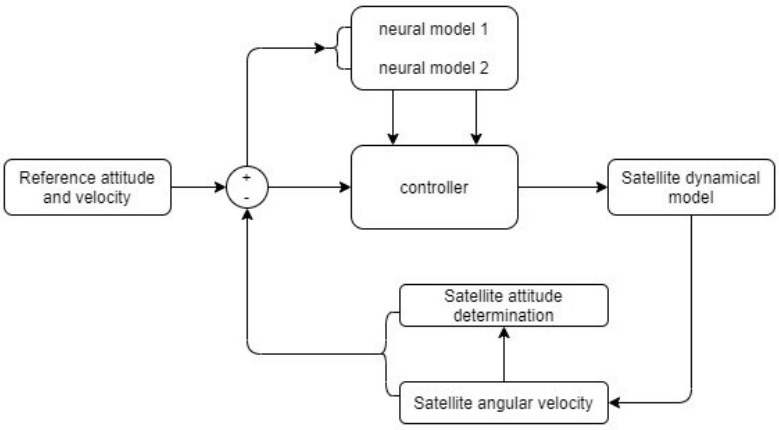

Figure 3.6 Neural based adaptive controller architecture.

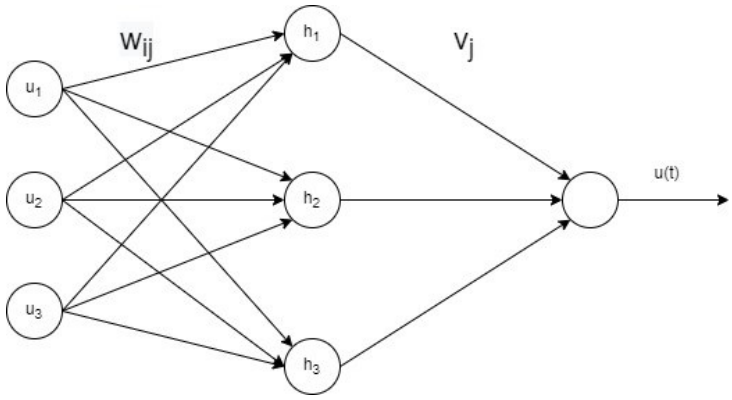

Figure 3.7 3-layer neuron perceptron.

The neural network-based adaptive controller is described in Fig. 3.6 Two identical neural networks are used with the quaternion errors and angular velocity errors to adaptively change the parameter gains of the proposed control algorithm. The neural network comprises a three layer neuron perceptron, as shown in Fig. 3.7 It includes an input layer, a hidden layer, and a output layer. Three inputs considered here are the three quaternion error signals and three angular velocity vectors. w_{ij} and v_j correspond to the weights of the hidden layers and the output layers, respectively. The gradient descent method with the backpropagation algorithm was used to minimize the error function and update the weights at each iteration. As the system propagates through the layers, the outputs from the hidden layers are used in the plant to tune each of the proportional and derivative gains separately and the plant output error is fed back to the neural network for calculating the next iteration.

Elaborating the inner workings of the neural controller, the activation function of the output is represented by Eq. 18. Here, a sigmoid function is used. In general, the calculated weighted sum can have a large disparity. With the sigmoid function, it is compressed within the range of 0 to 1.

$$u(t) = \frac{1}{1 + e^{-m}}. \tag{65}$$

with m being the summation of the multiplications of hidden layer outputs and their weighting coefficients.

$$m = \sum_{j=1}^{3} v_j h_j. \tag{66}$$

The hidden layer output h_j is calculated using another Sigmoid function through

$$h_j = \frac{1}{1 + e_j^{-n}}. \tag{67}$$

Furthermore, the corresponding value of n_j is the summation of multiplications between inputs and their related weight coefficients.

$$n_j = \sum_{i=1}^{3} w_{ji} u_i. \tag{68}$$

Let us consider a cost function in the form of output error. The objective is to minimize the accumulation of the square of the error given by e_y. Let us consider the following function:

$$Err(t) = \frac{1}{2} \sum_{i=1}^{t} e_y(i)^2. \tag{69}$$

The most general method to minimize the cost function is to start at any given output value and identify which way the input should move to decrease the output. By considering the slope of that function at that given time, we can identify whether to move the input to a negative or positive direction to minimize the function. From the concept of gradient descent, the error can be minimized when the gradient of this function moves

in the negative direction with reference to weights' coefficients. The gradient of the error function is given by

$$\nabla Err(t) = \begin{bmatrix} \frac{\partial Err(t)}{\partial v_j} \\ \frac{\partial Err(t)}{\partial w_{ji}} \end{bmatrix}. \tag{70}$$

The partial derivatives can be expanded as follows

$$\frac{\partial Err(t)}{\partial v_j} = \frac{\partial Err(t)}{\partial e_y} * \frac{\partial e_y}{\partial e_u} * \frac{\partial e_u}{\partial u(t)} * \frac{\partial u(t)}{\partial m} * \frac{\partial m}{\partial v_j}. \tag{71}$$

$$\frac{\partial Err(t)}{\partial w_{ji}} = \frac{\partial Err(t)}{\partial e_y} * \frac{\partial e_y}{\partial e_u} * \frac{\partial e_u}{\partial u(t)} * \frac{\partial u(t)}{\partial r} * \frac{\partial r}{\partial h_i} * \frac{\partial h_i}{\partial n_j} * \frac{\partial n_j}{\partial w_{ji}}. \tag{72}$$

Substituting the below notations,

$$\delta^1 = e_y * u(t) * \big(1 - u(t)\big). \tag{73}$$

$$\delta_j^2 = \delta^1 * v_j * h_j * \big(1 - h_j\big). \tag{74}$$

The partial derivatives can be expressed in the following form:

$$\frac{\partial Err(t)}{\partial v_j} = \left(\frac{\partial e_y}{\partial e_u}\right) \delta^1 h_j. \tag{75}$$

$$\frac{\partial Err(t)}{\partial w_{ji}} = -\left(\frac{\partial e_y}{\partial e_u}\right) \delta_j^2 u_i. \tag{76}$$

This is the basis of the backpropagation, where the sensitivity of said coefficients is calculated to reduce the cost function values at each layer of the neural network. Using this, the weighting coefficients can be updated with the following formula:

$$v_j(i+1) = v_j(i) + \left(n\frac{\partial e_y}{\partial e_u}\right)\delta^1 h_j. \tag{77}$$

$$w_{ji}(i+1) = w_{ji}(i) + \left(n\frac{\partial e_y}{\partial e_u}\right)\delta_j^2 u_i. \tag{78}$$

Here, n is the learning rate. $\partial e_y/\partial e_u$ denotes the equivalent gain. ∂e_y is the difference between desired plant output and the actual plant output. Similarly, ∂e_u is the subtraction between the desired neural network output and the actual neural network output. The initial values of the weights are zero, and a learning rate of 0.2 is used in the simulations.

3.3. Simulation Results

To compare performance, two sets of situations are analyzed using computer simulations based on MATLAB/SIMULINK. In the first set, each controller is simulated corresponding to the instant when the inertia variation is assumed to be at the maximum attainable value with the given initial angular velocities and attitude angles denoted in Table (3.3).

Table 3.3 Initial conditions for simulation set 1

Initial Condition	Value
Inertia	$\begin{bmatrix} 15000 & 5000 & 5000 \\ 5000 & 14000 & 5000 \\ 5000 & 5000 & 17000 \end{bmatrix}$ kgm^2
Angular velocity	[-0.6771, 1.1498, -0.0681]
Quaternion angle	[-0.4794, -0.1298, 0.3722, -0.7213]

The simulation time chosen is 500 s, and a saturation limit of ±500 Nm is used for the generated torques. Figs. (3.8)–(3.19) illustrate the change in the quaternion error angles, angular velocities, and the preferred torques for each controller. Tables (3.4)–(3.7)

show the settling duration and root mean square errors (RMSE) of the quaternions, velocities, and the preferred torques. Fig. (3.20) illustrates the variation of adaptive gains with the NBAC controller corresponding to each quaternion and angular velocity errors.

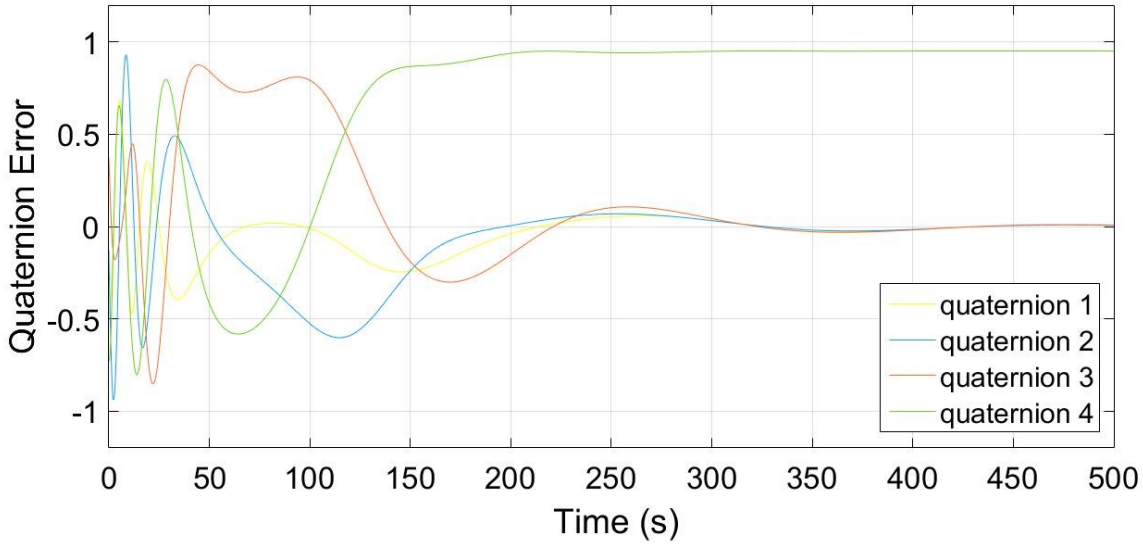

Figure 3.8 PD controller: variation in quaternions.

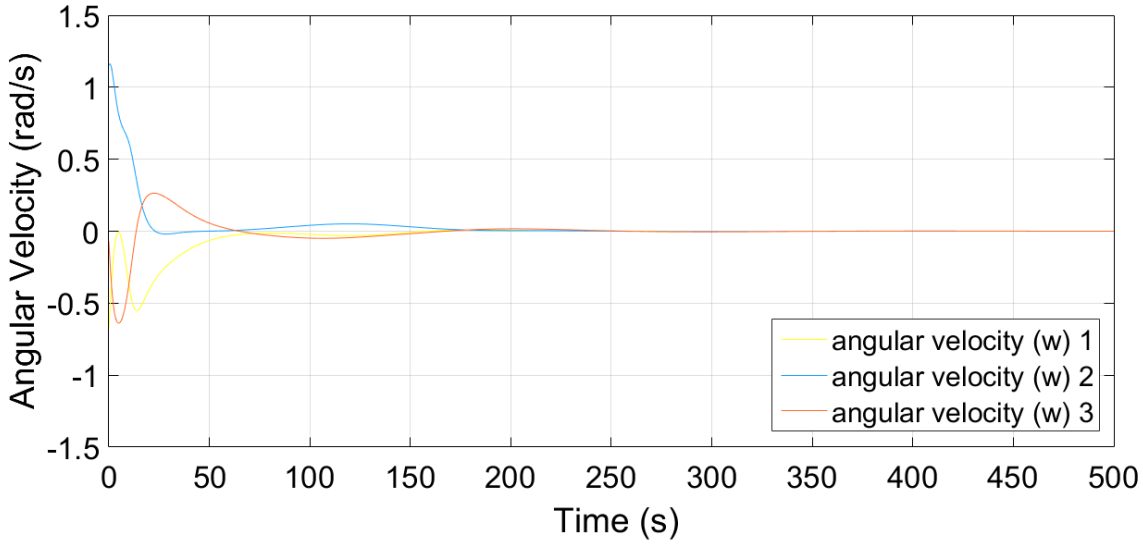

Figure 3.9 PD controller: variation in angular velocities.

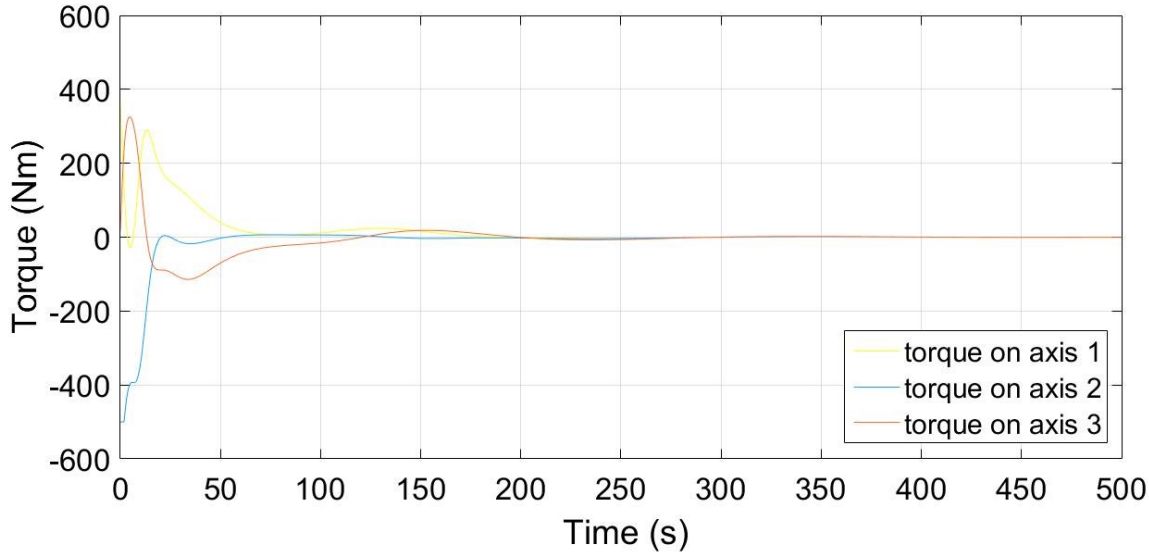

Figure 3.10 PD controller: variation in required torques.

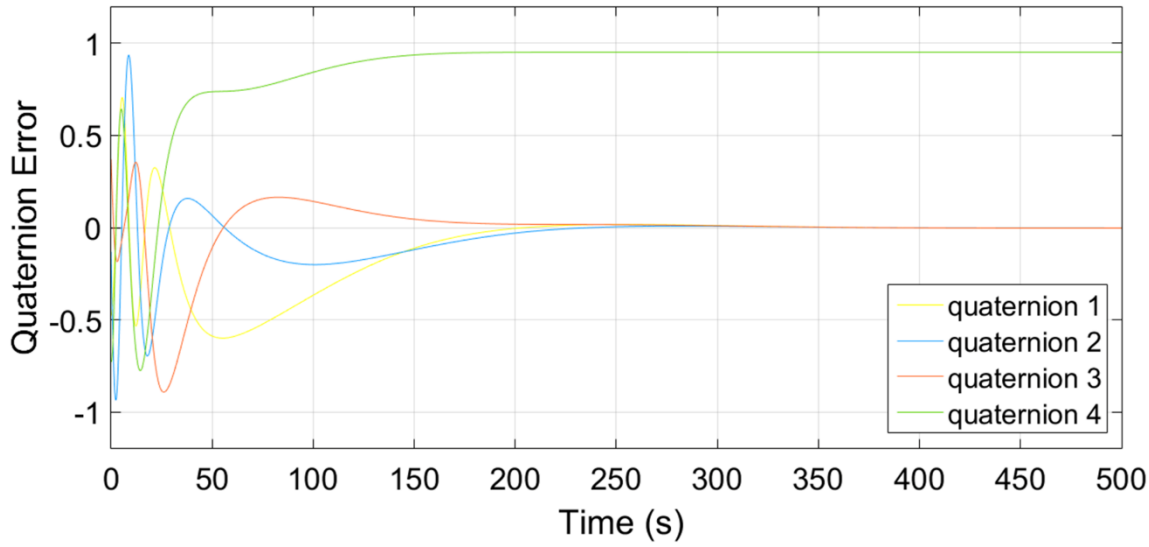

Figure 3.11 FBC: variation in quaternions.

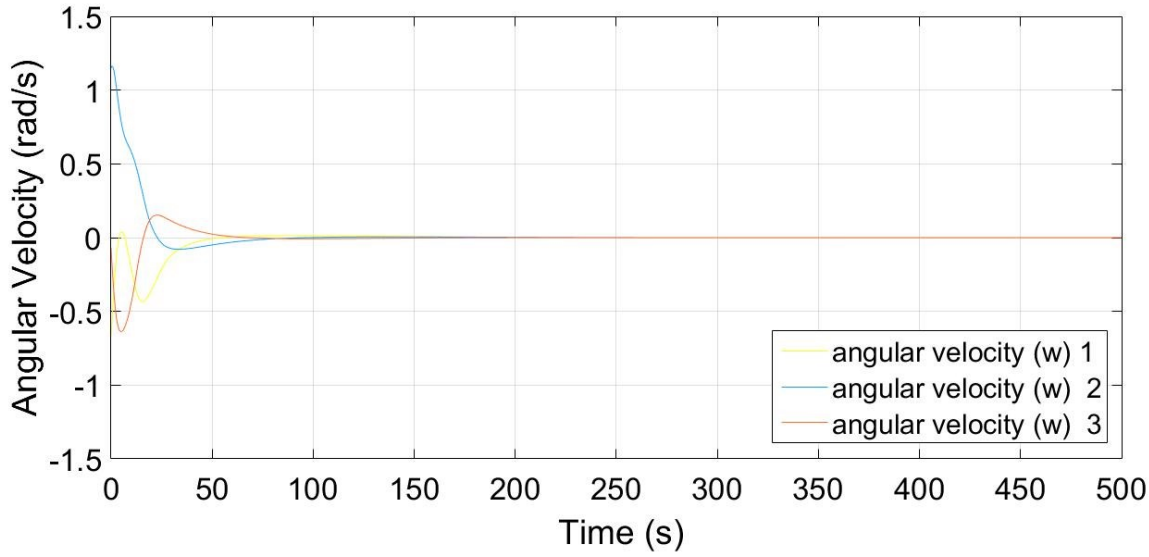

Figure 3.12 FBC: variation in angular velocities.

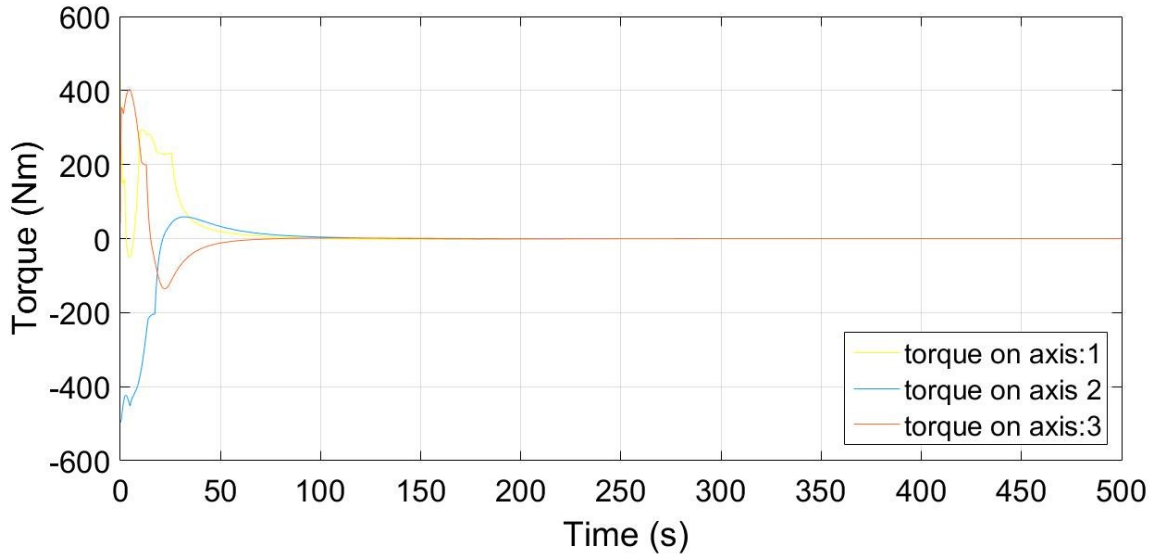

Figure 3.13 FBC: variation in required torques.

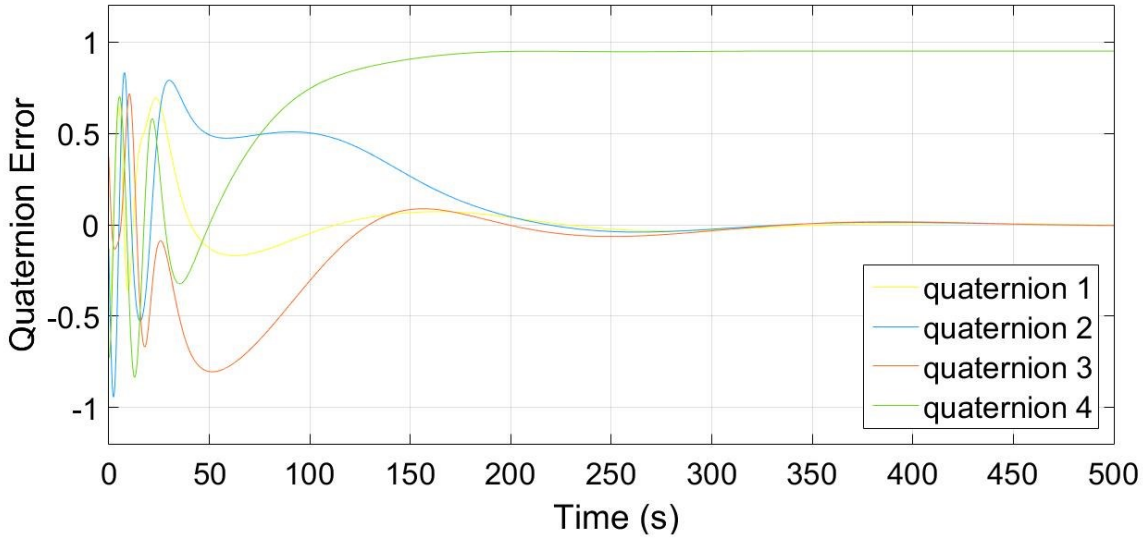

Figure 3.14 IBC: variation in quaternions.

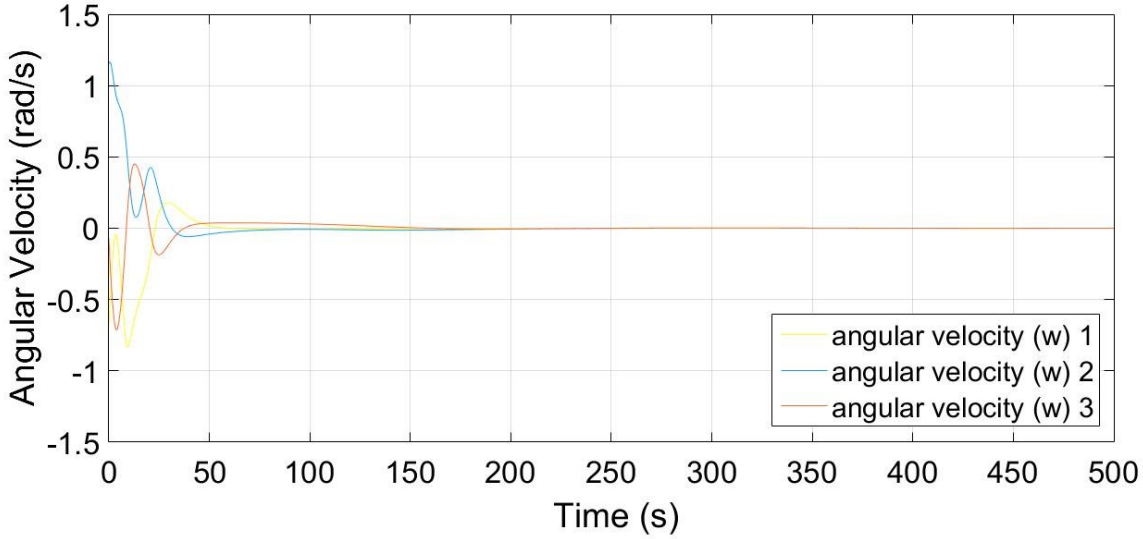

Figure 3.15 IBC: variation in angular velocities.

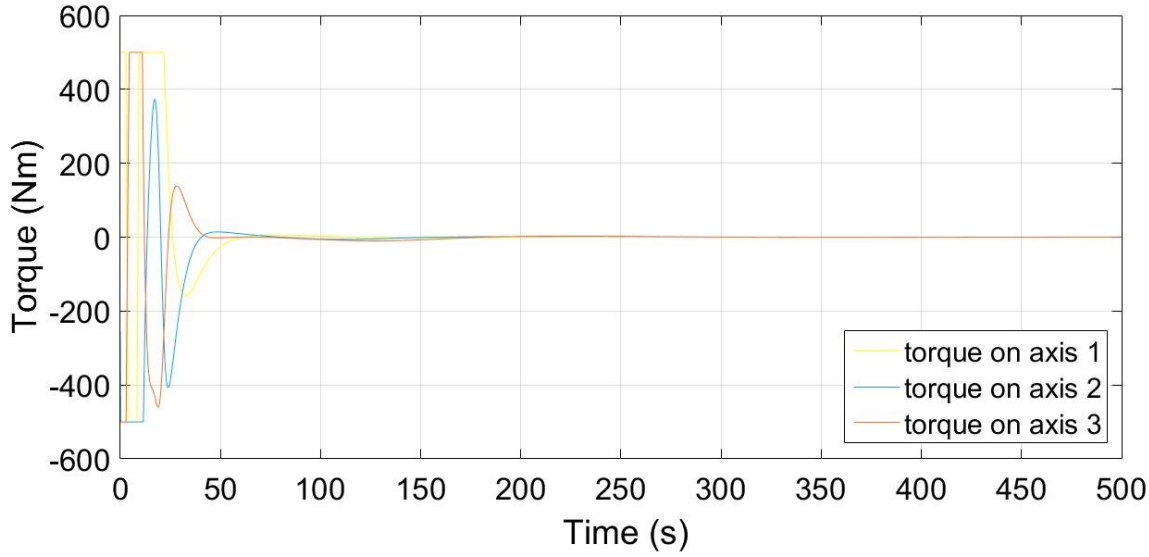

Figure 3.16 IBC: variation in required torques.

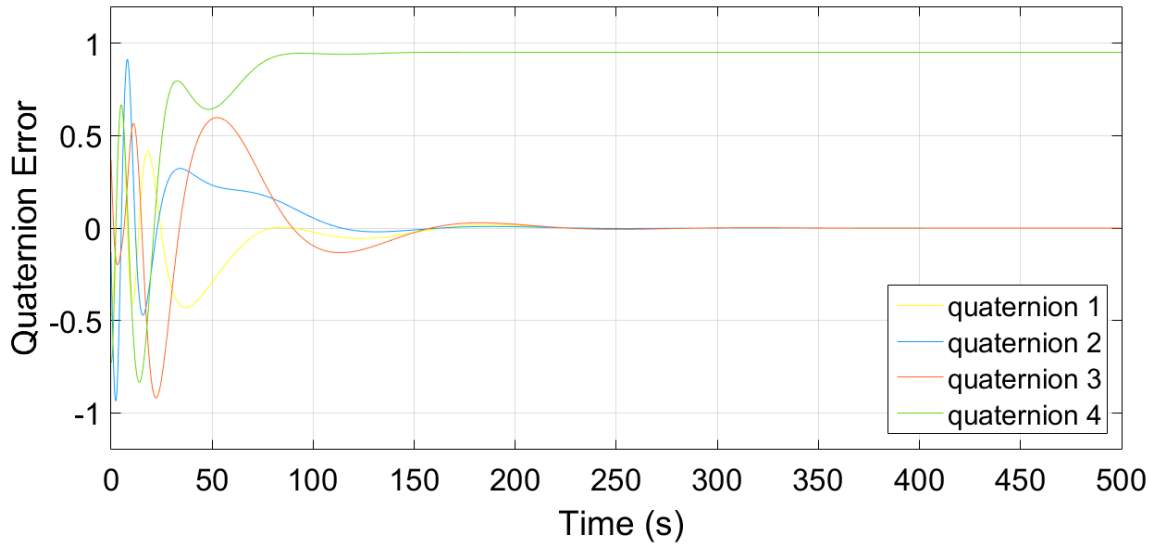

Figure 3.17 NBAC: variation in quaternions.

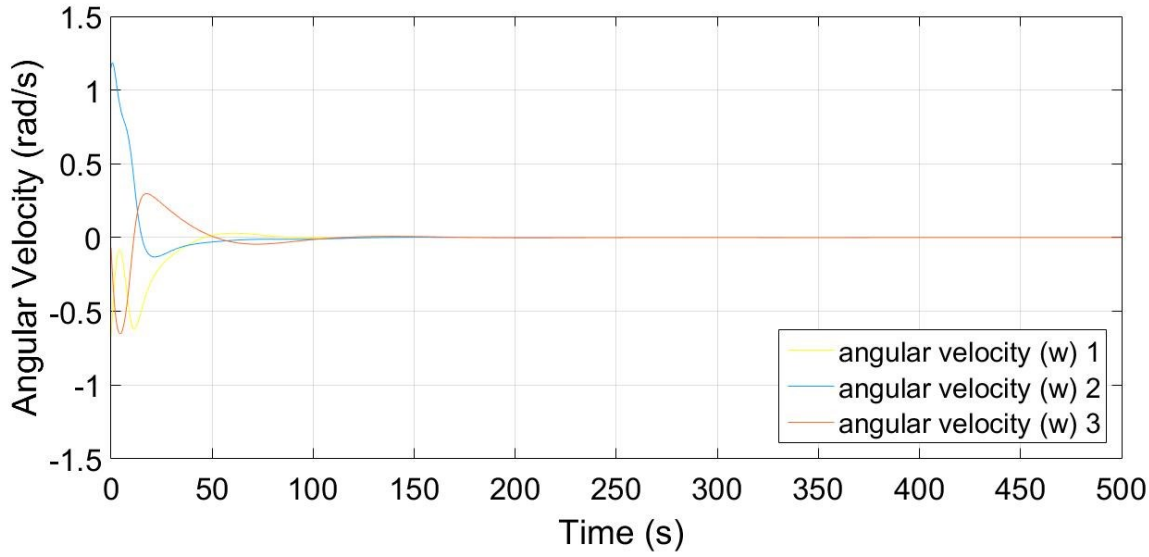

Figure 3.18 NBAC: variation in angular velocities.

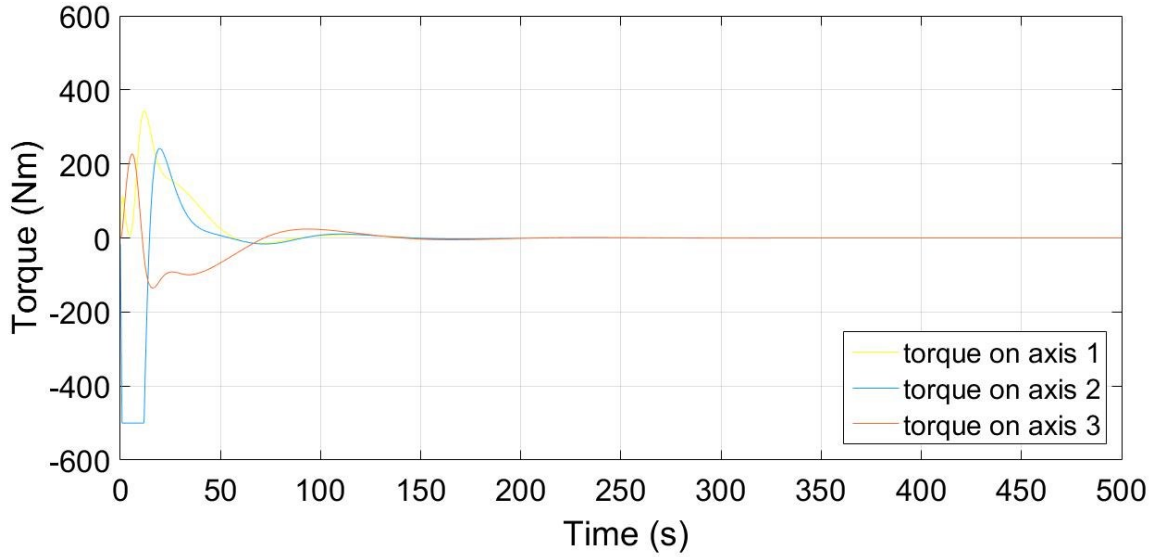

Figure 3.19 NBAC: variation in required torques.

Table 3.4 Settling times with each controller.

Controller	Settling time (s)		
	Quaternion	Angular velocity	Torque
PD	426	223	489
FBC	346	138	259
IBC	454	220	406
NBAC	269	159	263

Table 3.5 RMSE of quaternions.

Controller	Q1	Q2	Q3	Q4
PD	0.1286	0.2418	0.3492	0.5872
FBC	0.2182	0.1552	0.1677	0.3186
IBC	0.1336	0.2790	0.2793	0.4114
NBAC	0.1171	0.1387	0.1941	0.3016

Table 3.6 RMSE of angular velocities.

Controller	Angular velocity 1	Angular velocity 2	Angular velocity 3
PD	0.0953	0.1350	0.0927
FBC	0.0722	0.1344	0.0841
IBC	0.1089	0.1359	0.0873
NBAC	0.0914	0.1344	0.0878

Table 3.7 RMSE of required torques.

Controller	Torque 1	Torque 2	Torque 3
PD	49.6358	65.1253	47.5293
FBC	52.1243	70.2336	55.8923
IBC	106.5149	94.7039	91.6553
NBAC	52.1654	84.5855	36.9084

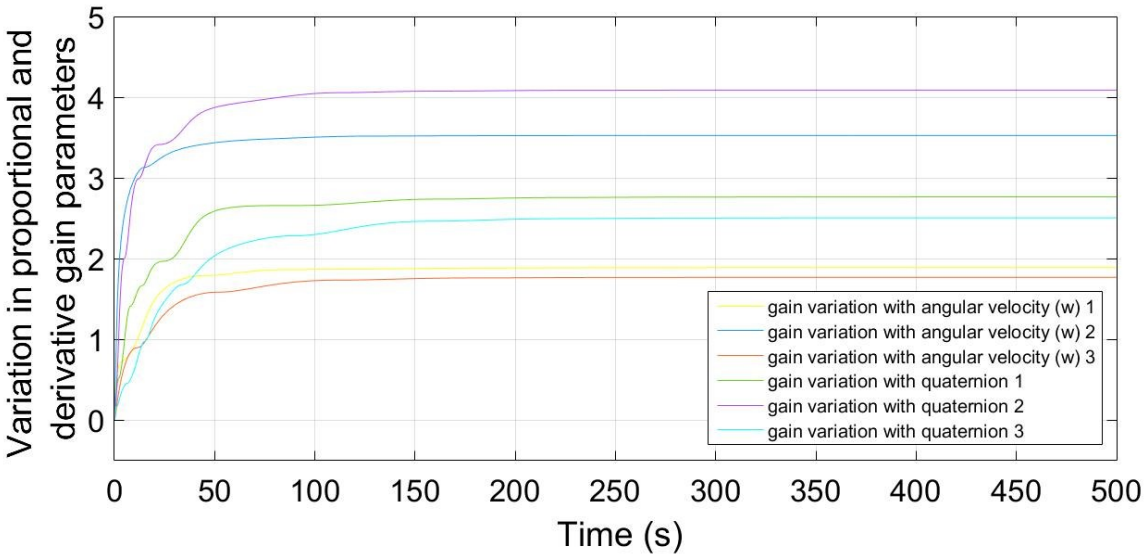

Figure 1. Variation in adaptive gains in NBAC.

The simulation results pertaining to the second situation are illustrated in Figs. (3.21)–(3.32). They demonstrate the fluctuations of quaternion error angles, angular velocity errors, and the required torques with the aforementioned controllers when the

unknown inertia is varied within its bounds. The initial values used in the simulations are given in Table (3.8). An unknown inertia is then added to the system. With the linear squares method, the inertia of the combined system is estimated by applying small constant torques for each axis for a total of 500 s. The estimated inertia is then sent to the dynamical equation of the satellite plant along with the updated angular velocities and attitude angles. The output behaviour is then simulated with each controller for 500 s. This is conducted for 50 different occasions while varying the unknown inertia added to the system.

Table 3.8 Initial conditions for simulation set 2

Condition	Value
Initial inertia	$\begin{bmatrix} 10000 & 0 & 0 \\ 0 & 9000 & 0 \\ 0 & 0 & 12000 \end{bmatrix}$ kgm^2
Inertia variation of (I_{xx}, I_{yy}, I_{zz}, I_{xy}, I_{xz}, I_{yz})	[0 – 5000] kgm^2
Initial angular velocity	[-0.6771, 1.1498, -0.0681]
Initial quaternion angle	[-0.4794, -0.1298, 0.3722, -0.7213]
Constant torque applied	[20 ; 20 ; 10] Nm
Simulation time	500 s
Number of simulations	50

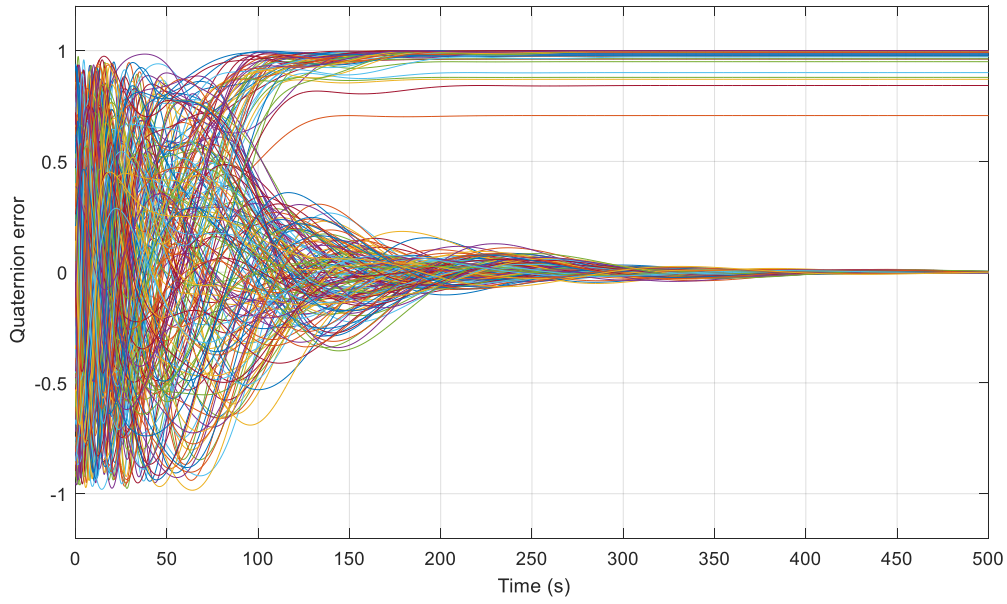

Figure 3.21 PD control: variation in quaternions.

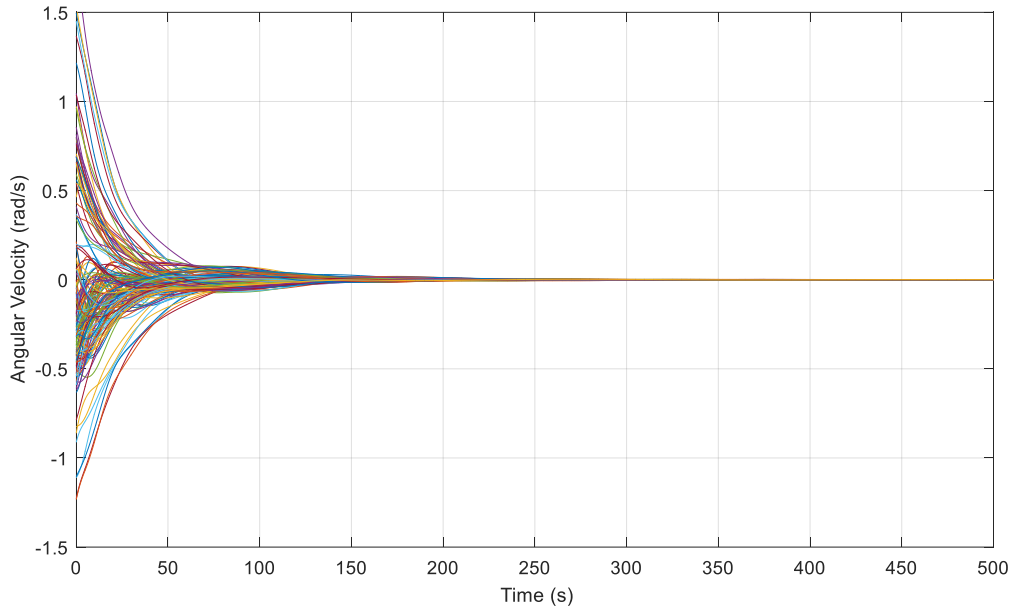

Figure 3.22 PD control: variation in angular velocities.

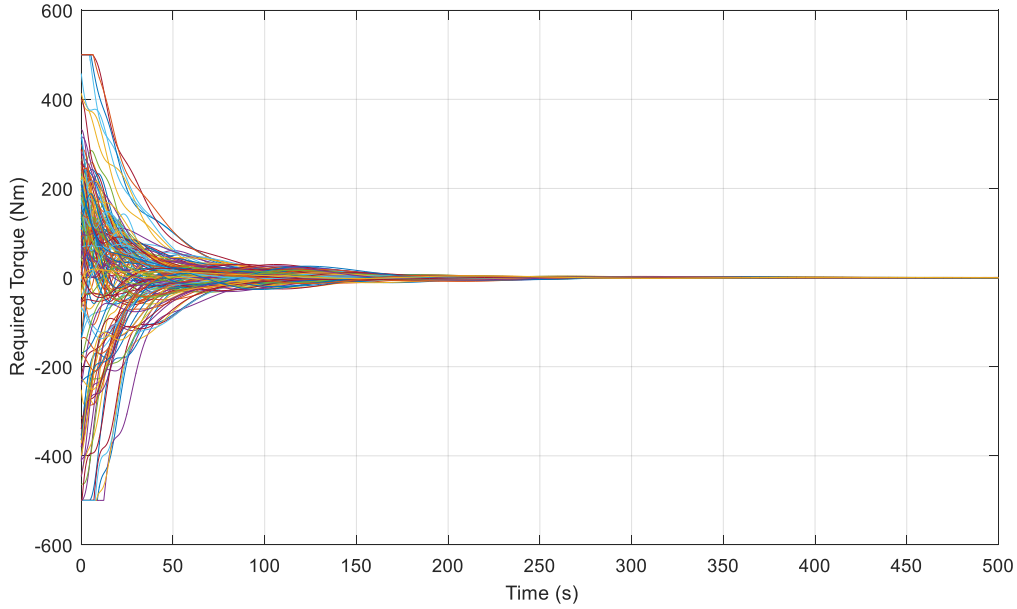

Figure 3.23 PD control: variation in required torques.

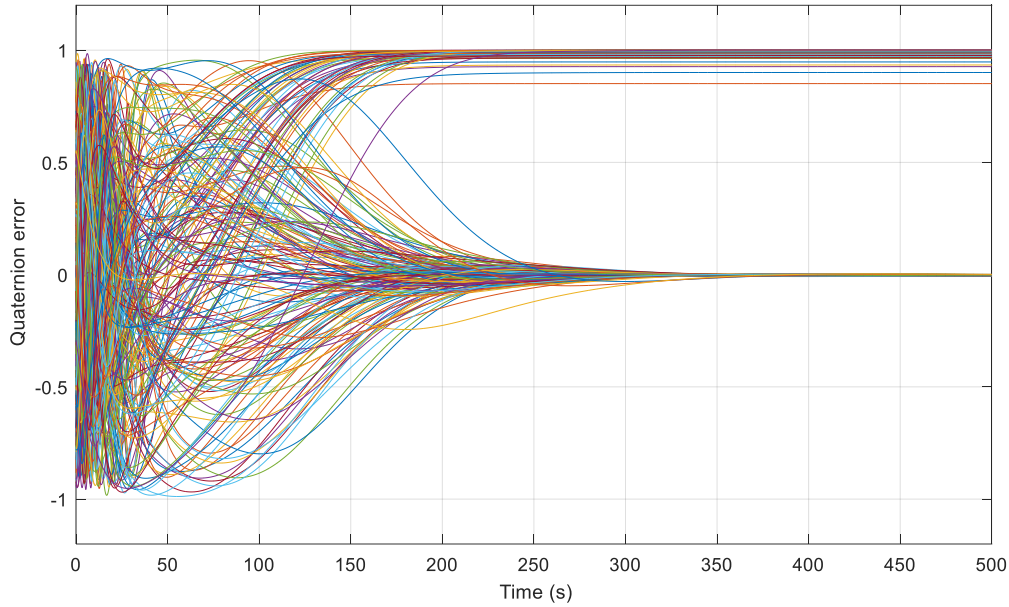

Figure 3.24 FBC: variation in quaternions.

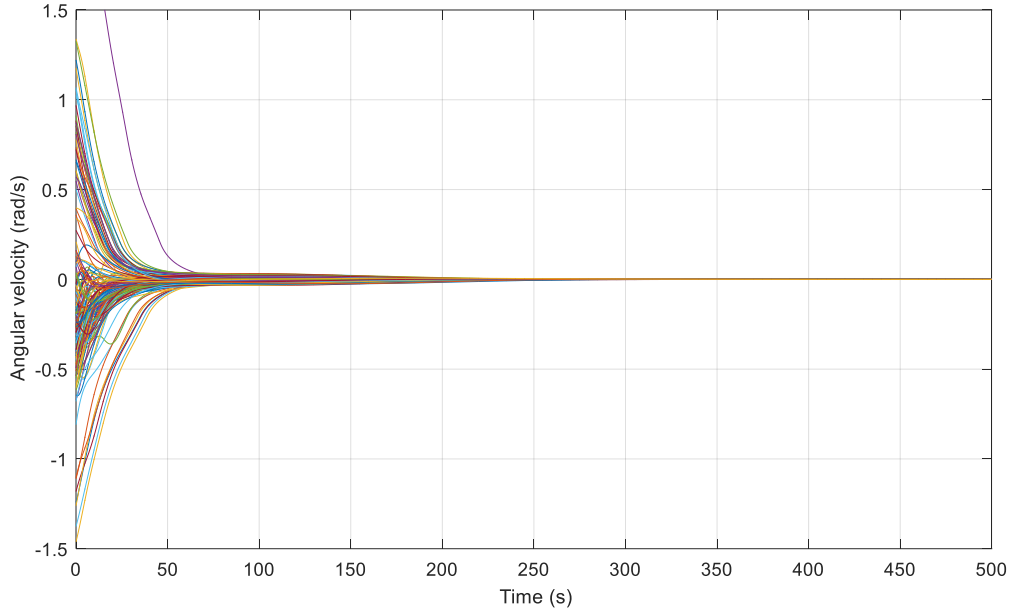

Figure 3.25 FBC: variation in angular velocities.

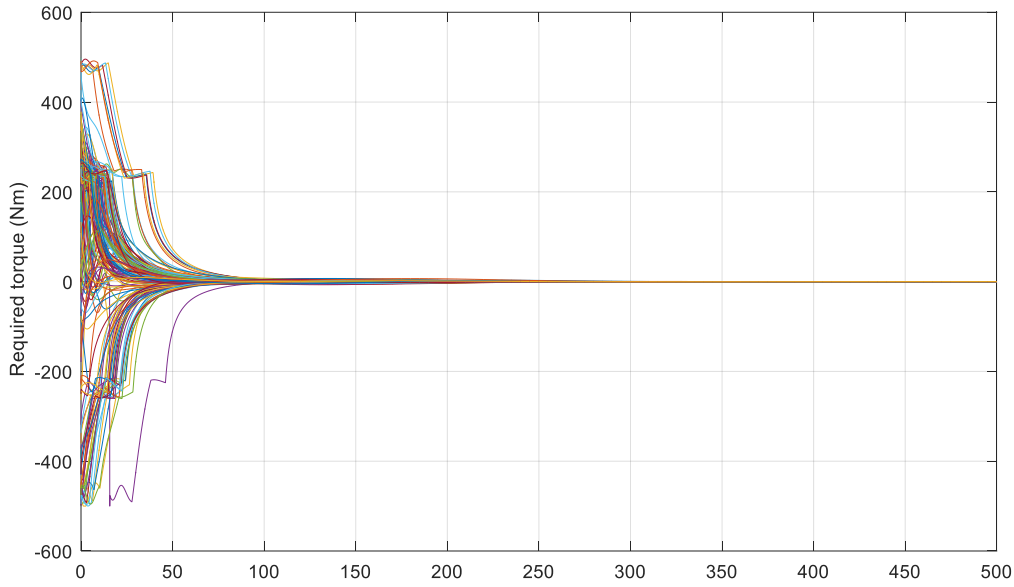

Figure 3.26 FBC: variation in required torques.

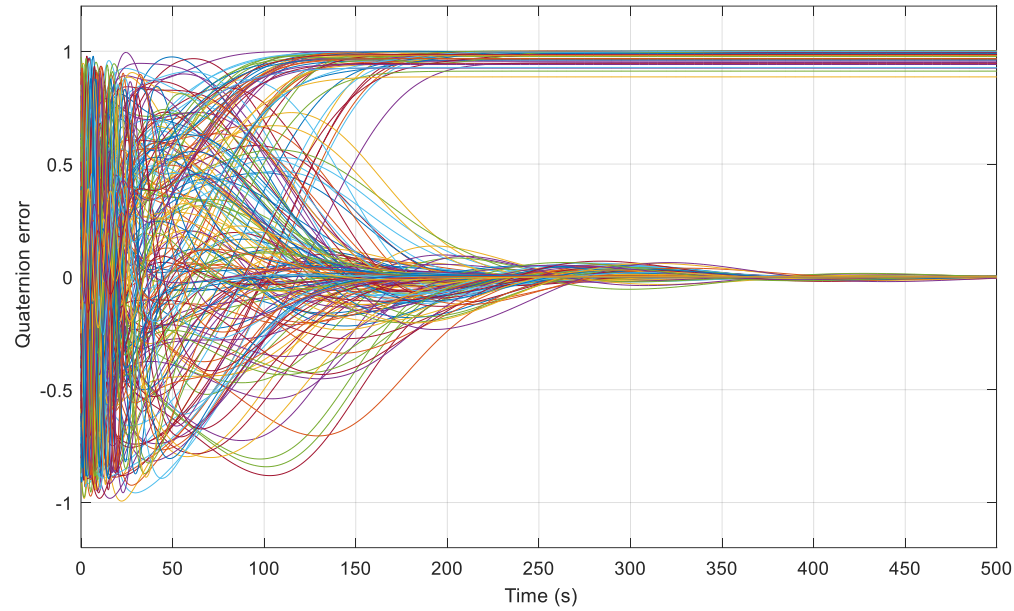

Figure 3.27 IBC: variation in quaternions.

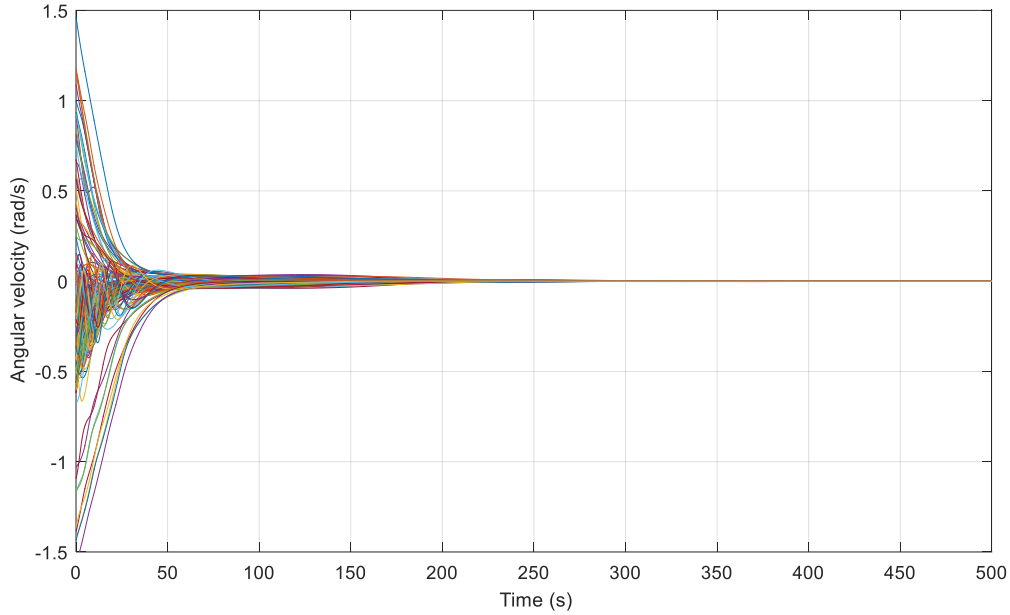

Figure 3.28 IBC: variation in angular velocities.

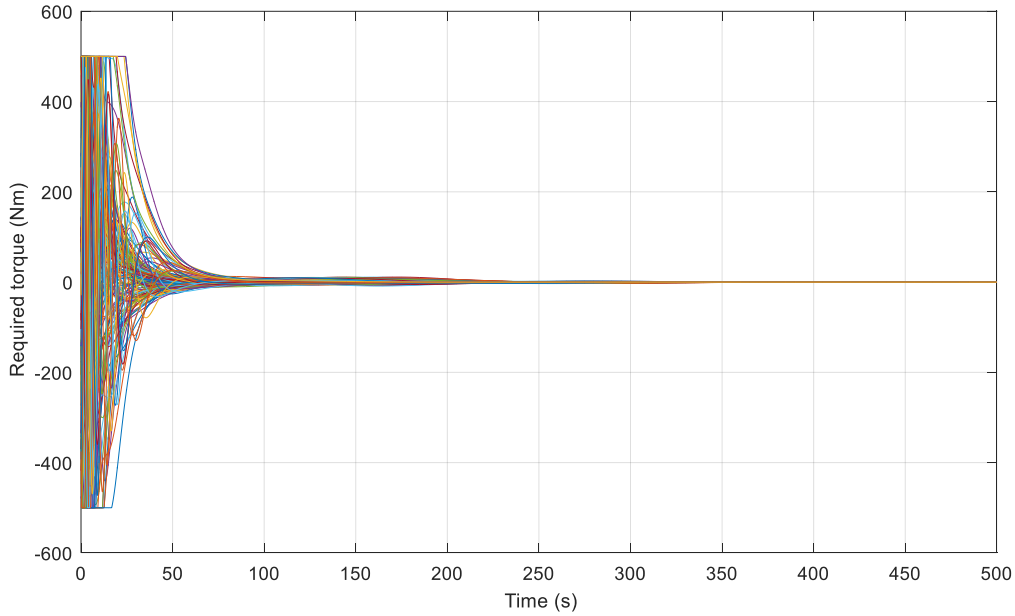

Figure 3.29 IBC: variation in required torques.

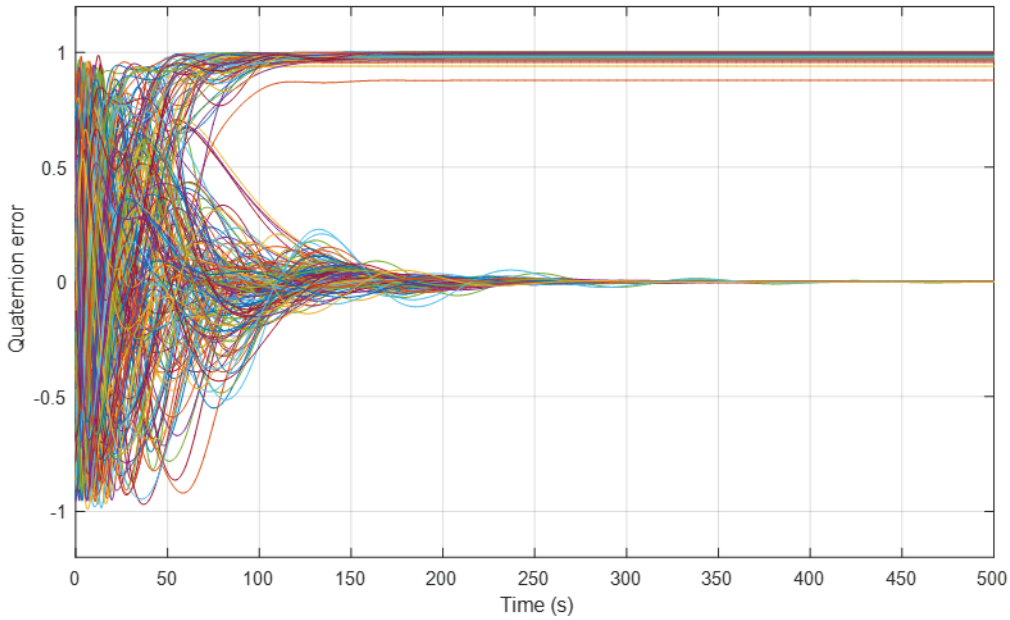

Figure 3.30 NBAC: variation in quaternions.

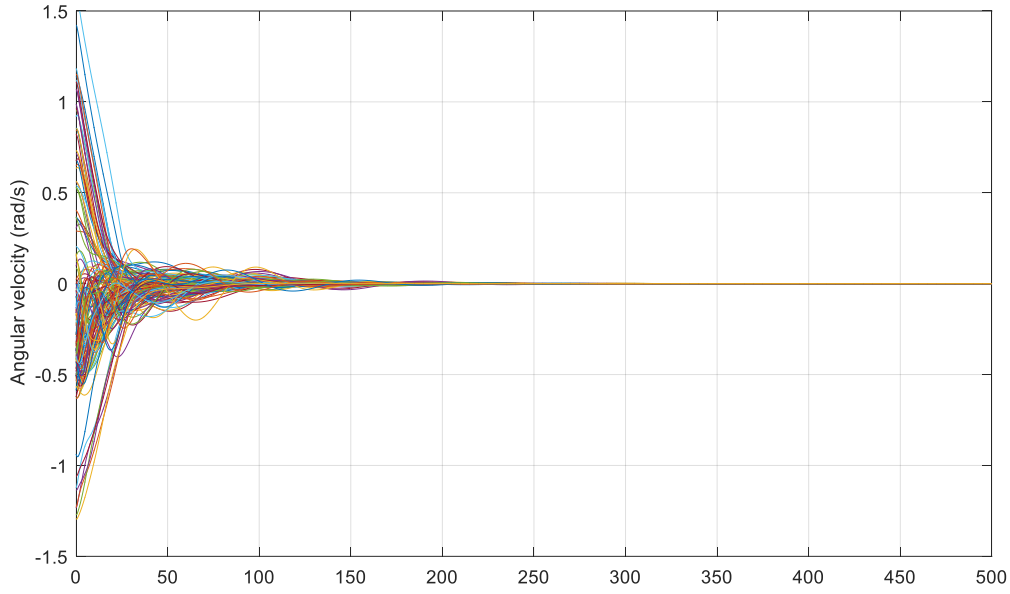

Figure 3.31 NBAC: variation in angular velocities.

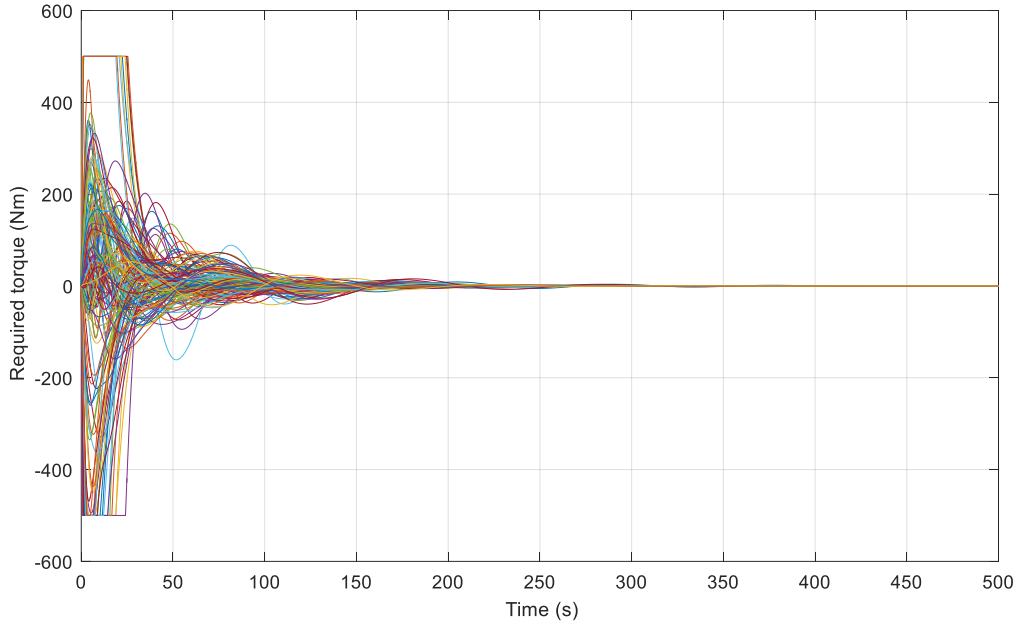

Figure 3.32 NBAC: variation in required torques.

Tables (3.9)–(3.11) present the RMSE of the quaternion angle errors, the angular velocity errors, and the preferred torques pertaining to the second set of simulations illustrated in Figs. (3.21)–(3.32). Table (3.12) presents the standard deviation of the mean squared errors of the six components of the estimated inertias, as calculated in the above simulations.

Table 3.9 RMSE of quaternions.

Controller	Q1	Q2	Q3	Q4
PD	0.2127	0.2553	0.1889	0.3653
FBC	0.2289	0.2720	0.1851	0.4229
IBC	0.2298	0.2489	0.1946	0.4174
NBAC	0.1496	0.1813	0.1678	0.2908

Table 3.10 RMSE of angular velocities.

Controller	Angular velocity 1	Angular velocity 2	Angular velocity 3
PD	0.0576	0.0927	0.0856
FBC	0.0418	0.0832	0.0714
IBC	0.0453	0.0748	0.0927
NBAC	0.0599	0.0846	0.0778

Table 3.11 RMSE of required torques.

Controller	Torque 1	Torque 2	Torque 3
PD	29.4499	47.0033	42.1701
FBC	34.0388	48.2003	45.9032
IBC	52.2904	60.8533	60.4307
NBAC	34.8662	60.0285	51.5018

Table 3.12 RMSE of estimated inertia components.

Inertia estimate	Controller			
	PD	FBC	IBC	NBAC
I_{xx}	260.23	205.11	177.86	196.90
I_{yy}	317.22	162.61	144.04	186.86
I_{zz}	332.48	327.68	270.01	254.83
I_{xy}	168.42	80.94	119.08	104.99
I_{xz}	157.52	75.22	131.53	103.01
I_{yz}	112.89	77.34	141.65	112.82

Considering both single event and multiple event situations observed above in the RMSE of quaternions, angular velocities, and required torques, NBAC controllers demonstrated the best overall performance with their adaptive abilities. As the attitude of the satellite is the most important factor considering this type of system, NBAC showed that it requires the least amount of time to stabilize quaternion errors under both situations. The second place is taken by FBC controller with its abilities to use the intuition of the expertise knowledge of the design criteria. The third place is taken by the PD controller with its easy design and implementation processes. The fourth place is awarded to the IBC controller as it also incorporates some of the estimated inertia values into its calculations which can have some disparity from the true inertia values.

Although the overall performance of the NBAC is convincing, the amount of control torque used by the NBAC controller is higher and could be reduced. This could be done by changing the learning rate of the NBAC controller. A performance comparison of different learning rates is given in Table (3.13). The data suggest that with increased

learning rates, the RMSE values of quaternion errors and angular velocity errors show reducing trends. However, the amount of torque required is increased.

Table 3.13 Performance with different learning rates

Learning rate	RMSE values pertaining to quaternions, angular velocities and torques									
	Q1	Q2	Q3	Q4	V1	V2	V3	T1	T2	T3
0.01	0.2498	0.2711	0.2182	0.5090	0.1549	0.2000	0.1258	32.8026	65.9142	23.7563
0.05	0.2815	0.1536	0.1974	0.4968	0.1145	0.1433	0.1050	40.9284	78.3369	44.8007
0.1	0.1332	0.1209	0.2714	0.3256	0.0994	0.1379	0.0949	45.1283	82.9581	42.8693
0.2	0.1171	0.1387	0.1941	0.3016	0.0915	0.1345	0.0878	52.1655	84.5855	36.9084
0.3	0.1033	0.1388	0.1744	0.3075	0.0872	0.1330	0.0853	56.1088	85.6184	33.8936
0.4	0.0949	0.1359	0.1707	0.3155	0.0843	0.1322	0.0839	58.7242	86.3572	32.2636
0.5	0.0917	0.1332	0.1716	0.3232	0.0822	0.1315	0.0832	60.7162	86.9008	31.3316
1	0.1055	0.1249	0.1793	0.3536	0.0759	0.1298	0.0816	66.8594	88.3371	30.4164

To compensate for this, an adaptive learning rate is added to the system in the following form of:

$$u_{lr} = u_0 + \alpha k \; ; \; k = \sum |error|. \tag{79}$$

The change in the learning rate with the conditions in simulation set 1 is given below.

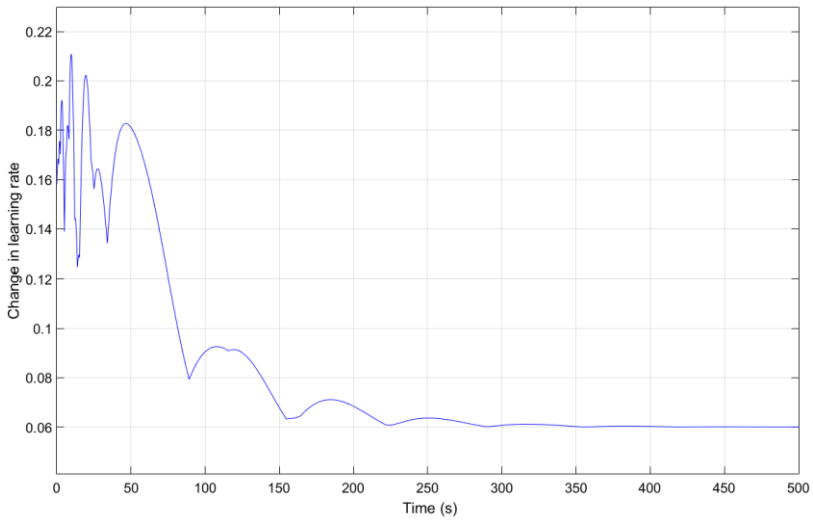

Figure 3.33 Change in learning rate with quaternions

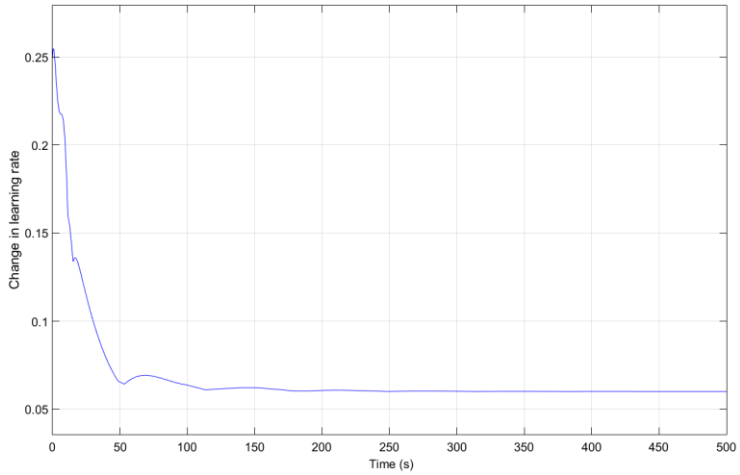

Figure 3.34 Change in learning rate with angular velocities

The modified learning rate method with the simulation set 2 yields the following results in the 50 simulation criterion with quaternions, angular velocities, and generated torques as shown in Figs. (3.35)–(3.37).

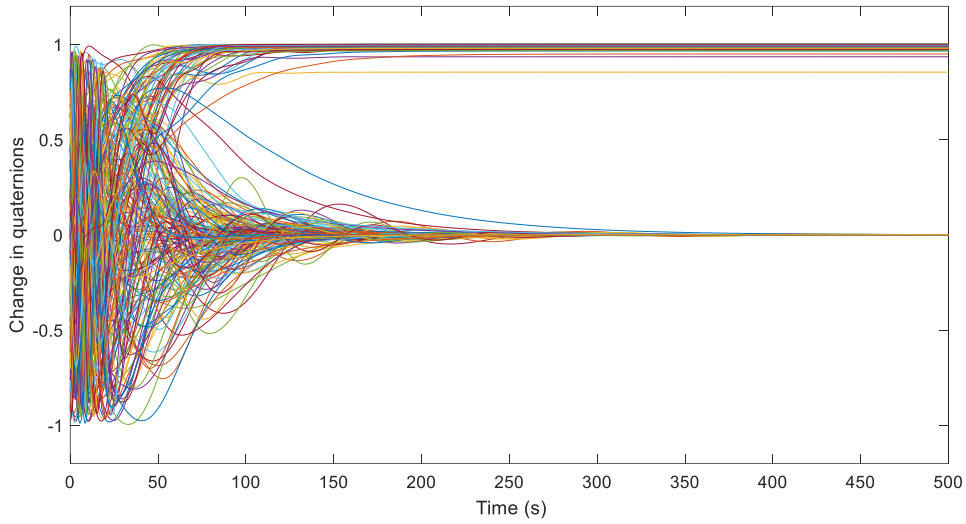

Figure 3.35 Change in quaternion errors with modified learning rate

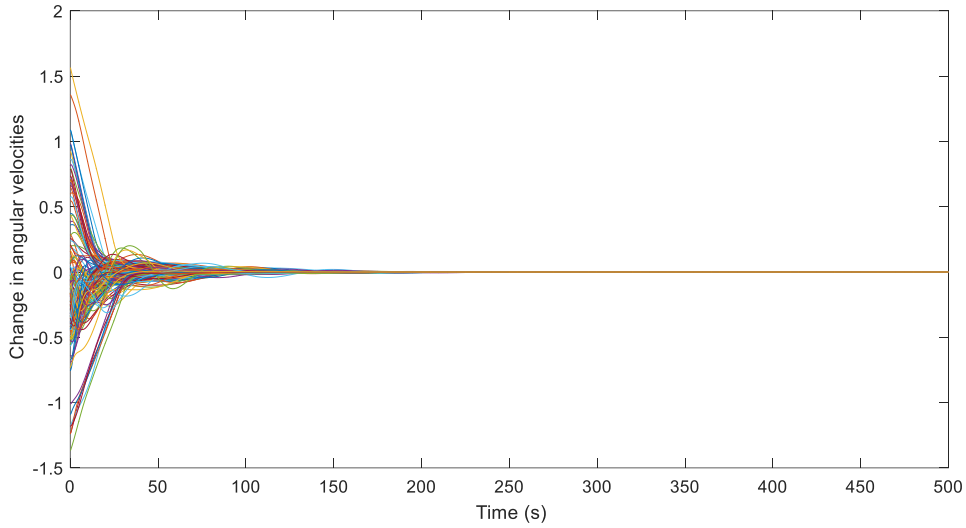

Figure 3.36 Change in angular velocities with modified learning rate

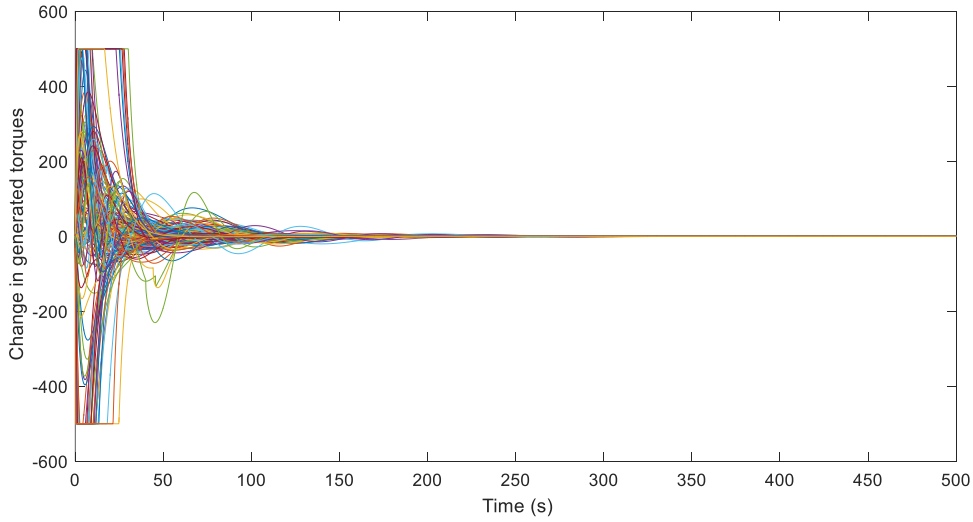

Figure 3.37 Change in required torques with modified learning rate

A qualitative comparison between the NBAC and NBAC with modified learning rate is given in Table (3.14). With an adaptive learning rate, the controller showed more improvements in simulation set 2 compared with simulation set 1. Although in set 1, the adaptive version is slightly behind the original version, it is still better than other controllers that have been used here. The advantage is that this adaptive version requires less control torques in comparison with the original NBAC controller in both sets of simulations.

Table 3.14 Comparison of data of NBAC with modified learning rate

Parameter	Comparison with NBAC single event scenario				Comparison with NBAC multiple events scenario			
	Value	Better	Worse	Overall rank	Value			Overall rank
Quaternion 1	0.1195		B	2	0.1590		B	2
Quaternion 2	0.1452		B	2	0.1783	G		1
Quaternion 3	0.2154		B	3	0.1399	G		1
Quaternion 4	0.3107		B	2	0.2603	G		1
Ang. Vel. 1	0.0903	G		2	0.0454	G		2
Ang. Vel. 2	0.1341	G		1	0.0674	G		1
Ang. Vel. 3	0.0886		B	4	0.0533	G		1
Torque 1	51.6816	G		2	38.183		B	4
Torque 2	85.8925		B	4	51.311	G		3
Torque 3	36.221	G		1	35.425	G		1

G= Good, B = Bad

3.4. Summary

In this chapter, an approach to solve the issue of controlling the attitude of a satellite with partially known inertial properties is introduced by combining both inertia estimation and controller implementation. The method considers an object with partially known inertial properties. Initially, the inertia of an unknown object is estimated using the linear least squares method, and an intelligent controller based on neural networks is implemented to drive the system to stability. From the results obtained from the simulations, it can be concluded that the neural network-based adaptive controller outperforms all the other controllers considered in this study for comparison. It demonstrates an appropriate performance in both the single- and multi-event situations analyzed in the simulations.

CHAPTER 4
NOVEL SLIDING MODE CONTROL-BASED CONTROLLER DESIGN FOR A SATELLITE WITH LIQUID FUEL SLOSH DISTURBANCES

4.1. Background and Literature Review

With the involvement of the private sector in space development, competition to build advanced and cost-effective rockets has seen major improvements. Thus, space has become more crowded in the recent times with satellites conducting evasive manoeuvres becoming a common sighting in the space related news. Therefore, they need to be on alert for possible collision situations and manoeuvre themselves safely in such scenarios.

Various control algorithms are being used to control satellites in space, ranging from simple feedback controllers to complex AI-based adaptive controllers. The development of such control algorithms occurred due to different challenges they had encountered over the years. In general, a satellite is a collection of rigid and flexible bodies moving in space. However, these parts must be changed sometimes to realign the solar panels and antennas to improve the conditions of the satellites. These could change the internal dynamics of the system to some extent. Similarly, the change in fuel mass can also alter satellite dynamics. This is due to the percentage of fuel a satellite carries onboard. Because the lifespan of a satellite depends on the onboard fuel, close to 40% of the satellite mass could be filled with fuel mass. As the fuel is in liquid state, it can move freely in space inside its container. Furthermore, this motion can cause issues in the control system itself when attempting to align the satellite to a particular orientation.

Considerable research has been conducted to implement different types of controllers to suppress the fuel slosh [47]. Yu [48] used a conventional sliding mode controller with direct physical meaning to the control inputs to stabilize rotational motion. Reyhanoglu [49] proposed a LQR and a Lyapunov-based control scheme to suppress fuel

slosh and transverse motion of the satellite considering a fixed axis motion. Souza [50] used LQG control for flexible spacecraft control considering the fuel slosh while estimating the slosh and flexible parameters in the system using the Kalman filter for practical situations. Thompson [51] proposed a wave-based controller in the form of a spring-mass system to suppress the fuel slosh in a satellite and compare it with a bang-bang controller for superior controllability. Hussein [52] further expanded sloshing suppression by incorporating vibrations of flexible appendages into a dynamical model.

In most of these cases, SMC is used to suppress and control the attitude and slosh angles. However, the settling time and control input values could be higher in many cases. In this research, we attempt to implement a novel sliding mode controller to mitigate chattering effects as well as the attitude and slosh angles originating from the satellite motion. To optimize the control parameters, a particle swarm optimization-based algorithm is used, considering error minimization to improve the controller parameters. The optimization calculations are performed offline, and only the acquired parameter values are used in the implementation of the algorithm in an online scenario. As such, the bulk of the computationally heavy work is performed not on the onboard control system of a satellite. This is to ensure that these types of controllers do not require excessive amount of energy and memory usage while being used in space under limited resources.

The rest of the chapter is arranged as follows. The methodology explains the mathematical model of the system and controller design. The results section summarizes the performance with simulations and data analysis of several controllers. This is followed by the conclusions drawn from this research.

4.2. Methodology

For simplicity, let us consider only the motion of a satellite in a fixed frame, and the gravity is assumed to be negligible. The goal is to stabilize the satellite model given in Fig. 1 about its Y axis. Two translational forces are applied to the satellite in the X and Z directions. This can also be represented as a gimbal stabilized system with a small deflection angle. The dynamical equations are derived considering the rigid satellite and the free-floating fuel mass inside the container. This fuel mass can be represented by both a spring-mass system or a pendulum system.

4.2.1. Mathematical Model

The equations of motion for the satellite with sloshing dynamics can be derived using Lagrange's equations [53]. Let us first consider the following satellite model with a fuel container for simplicity. Here, [x, z] denotes the coordinate system in the satellite body frame. [X, Y] denotes the coordinate system in the global frame of reference with its origin at O. The attitude angle of the satellite is given by θ , while the slosh angle is given by ψ. The fuel mass is m_f, and the satellite mass without fuel is given by m. The fuel mass is assumed to be moving in a pendulum motion with a pendulum length of a. The length between the fixed end of the pendulum and the centre of mass of the satellite without fuel mass is b. F and f correspond to the transverse forces applied in the X and Z directions. Finally, the pitch moment is given by M.

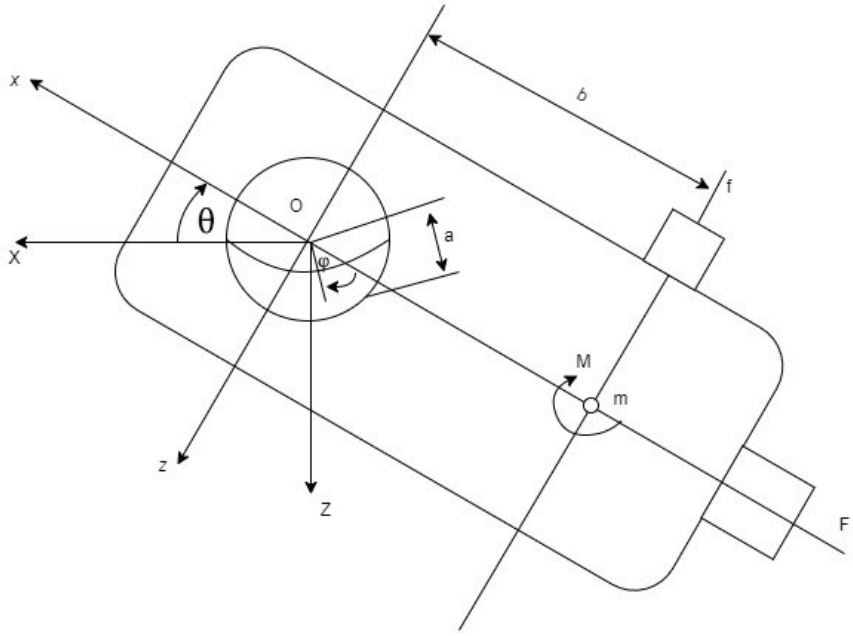

Figure 4.1 Satellite model with fuel container

Using Kirchhoff's and Lagrange equations, the equations of motion for this model can be derived. Here, $L, \bar{V}, \bar{\Omega}$, and R denote the Lagrangian of the system, translational velocity vector, angular velocity vector, and Rayleigh dissipation function, respectively. $\bar{\tau}_t$ and $\bar{\tau}_r$ are the external and internal torque vectors acting on the system, respectively. $\hat{V}$ and $\hat{\Omega}$ correspond to the skew symmetric matrices of the translational and angular velocities, respectively.

$$\frac{d}{dt}\frac{\partial L}{\partial \bar{V}} + \hat{\Omega} \times \frac{\partial L}{\partial \bar{V}} = \bar{\tau}_t. \tag{80}$$

$$\frac{d}{dt}\frac{\partial L}{\partial \bar{\Omega}} + \hat{\Omega} \times \frac{\partial L}{\partial \bar{\Omega}} + \hat{V} \times \frac{\partial L}{\partial \bar{V}} = \bar{\tau}_r. \tag{81}$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\psi}} - \frac{\partial L}{\partial \psi} + \frac{\partial R}{\partial \dot{\psi}} = 0. \tag{82}$$

Because we are only considering the motion in a fixed plane, the following conditions are applied when solving the above equations.

$$R = \frac{1}{2}\varepsilon\dot{\psi}^2, \bar{V} = \begin{bmatrix} v_x \\ 0 \\ v_z \end{bmatrix}, \bar{\Omega} = \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix}, \bar{\tau_t} = \begin{bmatrix} F \\ 0 \\ f \end{bmatrix}, \bar{\tau_r} = \begin{bmatrix} 0 \\ M + bf \\ 0 \end{bmatrix}.$$

With the calculation of the total kinetic energy considering both the translational and rotational motion, we can arrive at the following nonlinear equations. The additional parameters ε, I and I_f denote the dissipation coefficient and inertias of the satellite mass and fuel mass, respectively.

$$(m + m_f)\left(\dot{v}_x + v_z\dot{\theta}\right) + mb\dot{\theta}^2 + m_f a\left(\ddot{\theta} + \ddot{\psi}\right)\sin(\psi) + m_f a(\dot{\theta} + \dot{\psi})^2 cos(\psi) = F. \tag{83}$$

$$\left(m + m_f\right)\left(\dot{v}_z - v_x\dot{\theta}\right) + m_f a\left(\ddot{\theta} + \ddot{\psi}\right)cos(\psi) - m_f a(\dot{\theta} + \dot{\psi})^2 sin(\psi) + mb\ddot{\theta} = f. \tag{84}$$

$$(I + mb^2)\ddot{\theta} + mb(\dot{v}_z - v_x\dot{\theta}) - \varepsilon\dot{\psi} = M + bf. \tag{85}$$

$$(m_f a^2 + I_f)(\ddot{\theta} + \ddot{\psi}) + m_f a((\dot{v}_x + v_z\dot{\theta})\sin(\psi) + (\dot{v}_z - v_x\dot{\theta})cos(\psi)) + \varepsilon\dot{\psi} = 0. \tag{86}$$

Eq. (82) can be simplified due to negligible perturbations from slosh and attitude angular accelerations as well as angular velocities toward forward translational motion.

$$(\dot{v}_x + v_z\dot{\theta}) = \frac{F}{(m + m_f)}. \tag{87}$$

Equations (82) to (85) can be simplified by removing $(\dot{v}_x + v_z\dot{\theta})$ and $(\dot{v}_z - v_x\dot{\theta})$. The transverse force acting perpendicular to the fixed motion can also be replaced by considering a gimbal stabilized system where force f is replaced with gimbal angle δ and transverse force F.

$$[I + m^*(b^2 - abcos(\psi)]\ddot{\theta} - m^*ab\ddot{\psi}cos(\psi) + m^*ab(\dot{\theta} + \dot{\psi})^2 sin(\psi) - \varepsilon\dot{\psi} = M + b^*F\delta. \tag{88}$$

$$[I_f + m^*(a^2 - abcos(\psi)]\ddot{\theta} - (I_f + m^*a^2)\ddot{\psi}^2 + (a^*F - m^*ab\dot{\theta}^2)sin(\psi) + \varepsilon\dot{\psi} = -a^*F\delta cos(\psi). \tag{89}$$

With the substitutions being

$$m^* = \frac{mm_f}{(m + m_f)}. \tag{90.1}$$

$$a^* = \frac{am_f}{(m + m_f)}. \tag{90.2}$$

$$b^* = \frac{bm_f}{(m + m_f)} + d. \tag{90.3}$$

$$d = \frac{Fc}{(m + m_f)}. \tag{90.4}$$

$$c = \frac{m_f a}{(m_f a^2 + I_f)}. \tag{90.5}$$

The nonlinear equations are linearized considering small angle approximation as follows for further simplicity.

$$I_1\ddot{\theta} - I_2\ddot{\psi} - \varepsilon\dot{\psi} = M + b^*F\delta. \tag{91}$$

$$I_3\ddot{\theta} - I_4\ddot{\psi}^2 + a^*F\psi + \varepsilon\dot{\psi} = -a^*F\delta. \tag{92}$$

with the notations being

$$I_1 = I + m^*(b^2 - ab). \tag{93.1}$$

$$I_2 = m^*ab. \tag{93.2}$$

$$I_3 = I_f + m^*(a^2 - ab). \tag{93.3}$$

$$I_4 = I_f + m^*a^2. \tag{93.4}$$

4.2.2. Sliding Mode Controller Design (SMC)

The conventional sliding mode controller (SMC) is designed with a combination of equivalent control and switching control components in the following form [54] [55]

$$u = u_{eq} + u_{sw}. \tag{94}$$

It is derived by considering sliding surfaces in the following form:

$$s_1 = \dot{\theta} + z_1\theta + z_2^2 \int \theta_{err}\, dt. \tag{95}$$

$$s_2 = \dot{\psi} + z_1\psi + z_2^2 \int \psi_{err}\, dt. \tag{96}$$

The derivatives of the sliding surfaces are expressed as follows:

$$\dot{s}_1 = \ddot{\theta} + z_1\dot{\theta} + z_2^2\theta. \tag{97}$$

$$\dot{s}_1 = \ddot{\psi} + z_1\dot{\psi} + z_2^2\psi. \tag{98}$$

The equality control u_{eq} can be calculated by equaling the derivative of the sliding surface to 0 after substituting angle accelerations to the equations from Eqns. (97) and (98). Switching control u_{sw} is derived from the sliding surface.

Therefore, the total control input u is given as follows:

$$u = G^{-1}\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} - k\, sign \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}. \tag{99}$$

where

$$G = \begin{bmatrix} g_{21} & g_{22} \\ g_{31} & g_{32} \end{bmatrix}. \tag{100.1}$$

$$g_{21} = (I_4 b^* F - a^* I_2 F)/delta. \tag{100.2}$$

$$g_{22} = I_4/delta. \tag{100.3}$$

$$g_{31} = -(I_1 a^* F + I_3 b^* F)/delta. \tag{100.4}$$

$$g_{32} = -I_3/delta. \tag{100.5}$$

$$delta = I_1 I_4 + I_2 I_3. \tag{100.6}$$

$$w_1 = z_1 \dot{\theta} + z_2^2 \theta + f_1. \tag{100.7}$$

$$w_1 = z_1 \dot{\psi} + z_2^2 \psi + f_2. \tag{100.8}$$

$$f_1 = -(I_2 a^* F\psi + (I_2 \varepsilon - I_4 \varepsilon)\dot{\psi})/delta. \tag{100.9}$$

$$f_2 = -(I_1 a^* F\psi + \varepsilon(I_1 + I_3)\dot{\psi})/delta. \tag{100.10}$$

One of the drawbacks of the conventional SMC is the chattering phenomena; to reduce this effect, the control law is slightly modified in the form of

$$k\, sign \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} = kS_{1,2}. \tag{101}$$

where

$$S = \begin{cases} sign(s), s > 1 \\ s = s \\ sign(s), s < -1 \end{cases} \tag{102}$$

4.2.3. Novel Sliding Mode Controller Design (NSMC)

To design the sliding mode controller (NSMC), let us use the sliding surfaces as follows:

$$p_1 = \theta. \tag{103}$$

$$p_2 = \psi. \tag{104}$$

The first derivatives of the sliding surfaces are

$$p_{11} = \dot{\theta}. \tag{105}$$

$$p_{21} = \dot{\psi}. \tag{106}$$

The second derivative is then taken as

$$p_{12} = f_1 + g_{21}u_1 + g_{22}u_2. \tag{107}$$

$$p_{22} = f_2 + g_{31}u_1 + g_{32}u_2. \tag{108}$$

The third derivative is

$$p_{13} = \dot{f}_1 + g_{21}v_1 + g_{22}v_2. \tag{109}$$

$$p_{23} = \dot{f}_2 + g_{31}v_1 + g_{32}v_2. \tag{110}$$

with v being

$$v = G^{-1}\begin{bmatrix} -\dot{f}_1 - r_{11}p_1 - r_{12}p_{12} - r_{13}p_{13} - r_{14}sign(p_{13}) \\ -\dot{f}_2 - r_{21}p_2 - r_{22}p_{22} - r_{23}p_{23} - r_{24}sign(p_{23}) \end{bmatrix}. \tag{111}$$

and

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \int \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} dt. \tag{112}$$

Particle swarm optimization (PSO), which is a simple, yet effective optimization algorithm, uses a population of particles resembling candidate solutions to solve any optimization problem [56]. Here, we take the objective function as an error minimization in the form of

$$min \int_0^T (\theta_{err} + \dot{\theta}_{err} + \psi_{err} + \dot{\psi}_{err})\, dt. \tag{113}$$

PSO works in a given workspace to find the best solution to a given problem as shown in Fig. (4.2)

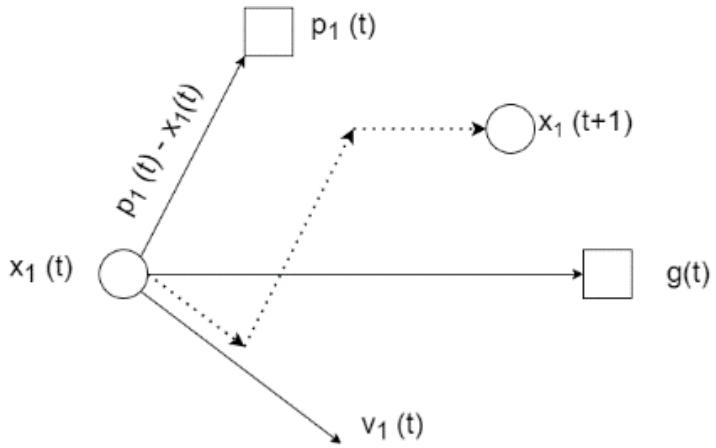

Figure 4.2 Particle swarm optimization overview

Considering a single particle from a swarm of particles, the position and velocity can be described by $n_i(t)$ and $v_i(t)$ in a given search space. The personal best of the performance criteria of the particle i is given by $p_i(t)$ and the best performance among the concerned swarm of particles is given by $g(t)$. Using this knowledge, updating the next position and velocity of the particle i is given by [57]

$$v_i(t+1) = \lambda v_i(t) + \mu_1 c_1 (p_i(t) - x_i(t)) + \mu_2 c_2 (g_j(t) - x_i(t)). \tag{114}$$

$$x_i(t+1) = x_i(t) + v_i(t+1). \tag{115}$$

For the simulation, a swarm size of 50 particles is used with an iteration count of 30. λ is selected as 1, and c_1 and c_2 are 2.

4.2.4. Proportional-Integral-Derivative Controller design (PID)

A PID-based controller was also implemented to compare the performance in the form of [58], $u = [u_1 : u_1 + u_2]$

$$u_1 = k_{p1}\theta + k_{d1}\dot{\theta} + k_{i1}\int \theta\, dt. \tag{116}$$

$$u_2 = k_{p2}\psi + k_{d2}\dot{\psi} + k_{i2}\int \psi\, dt. \tag{117}$$

4.3. ***Simulation Results***

To conduct the simulations, the following parameter values were used for dynamical equations.

Table 4.1 Parameter values used for the dynamical model

Parameter	Value
m	600 kg
m_f	100 kg
I	720 kgm^2
I_f	90 kgm^2
a	0.32 m
b	0.25 m
ε	0.19
F	2300 N
v_z at t=0	120 ms^{-1}
v_x at t=0	1500 ms^{-1}
θ at t=0	0.0349 rad
$\dot{\theta}$ at t=0	0.0099 rad
ψ at t=0	0.0873 rad
$\dot{\psi}$ at t=0	0.0087 rad

Simulation results pertaining to the change in the attitude angle, rate change of attitude angle, slosh angle, rate change of slosh angle, and control inputs are given from Figs. (4.3) to (4.20) for each of the NSMC, SMC and, PID controllers. The following values were used for the gain parameters with each of the control algorithm, as given in Table (4.2).

Table 4.2 Parameter gains used in simulations

Controller	Gains
NSMC	$r_{11} = 700$, $r_{21} = 700$ $r_{12} = 346.284$, $r_{22} = 353.533$ $r_{13} = 5.361$, $r_{23} = 11.365$ $r_{14} = 0.0001$, $r_{23} = 0.0001$
SMC	$z_1 = 16$, $z_2 = 64$, $k_1 = 18$, $k_2 = 6.0244e4$
PID	$k_{p1} = 50$, $k_{d1} = 50$, $k_{i1} = 0$ $k_{p2} = 10^4$, $k_{d2} = 10^4$, $k_{i2} = 0$

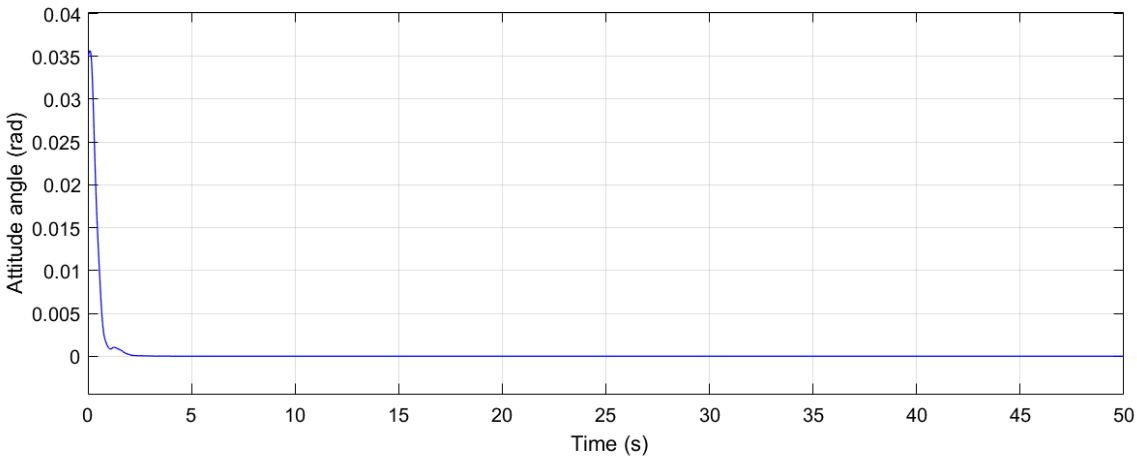

Figure 4.3 Change in attitude angle with time (NSMC)

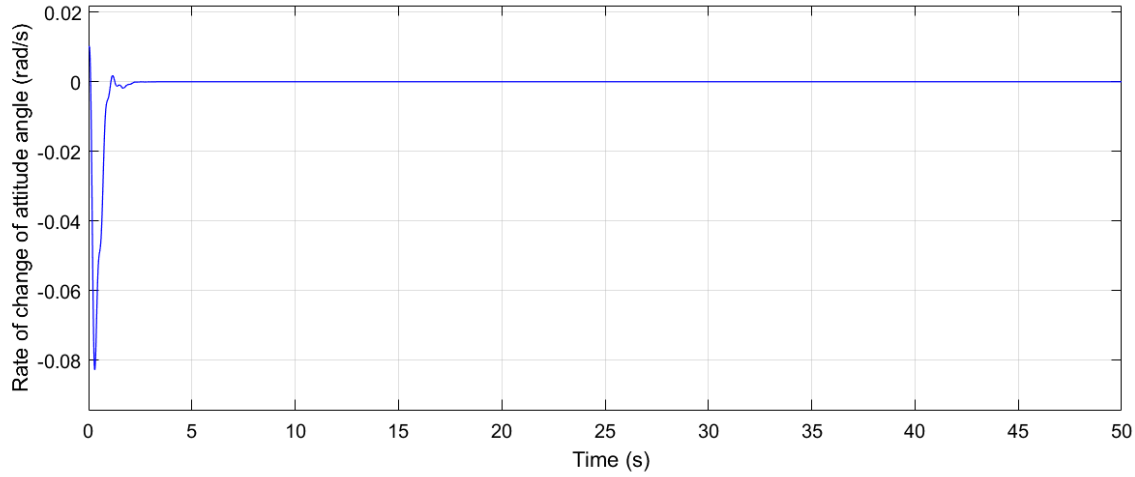

Figure 4.4 Rate of change of attitude angle with time (NSMC)

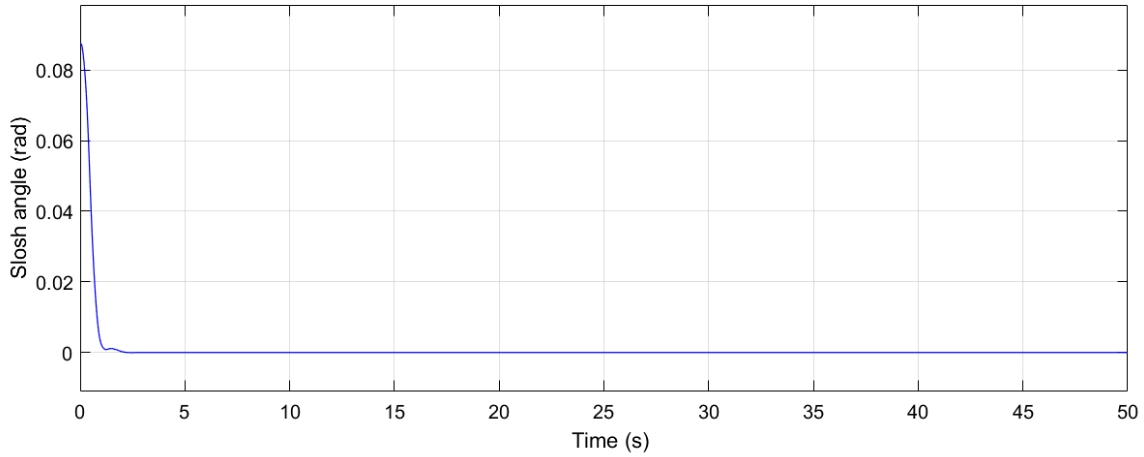

Figure 4.5 Change in slosh angle with time (NSMC)

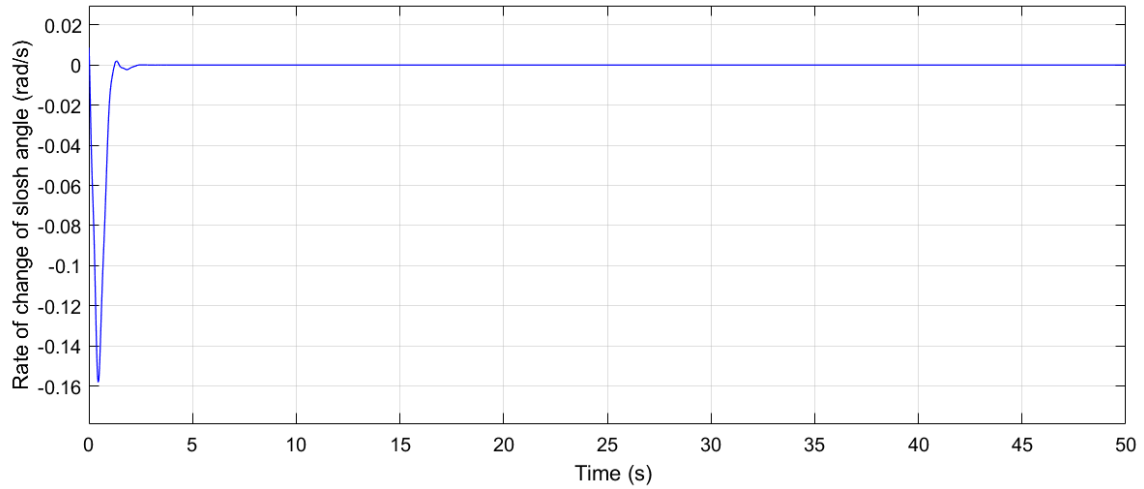

Figure 4.6 Rate of change of slosh angle with time (NSMC)

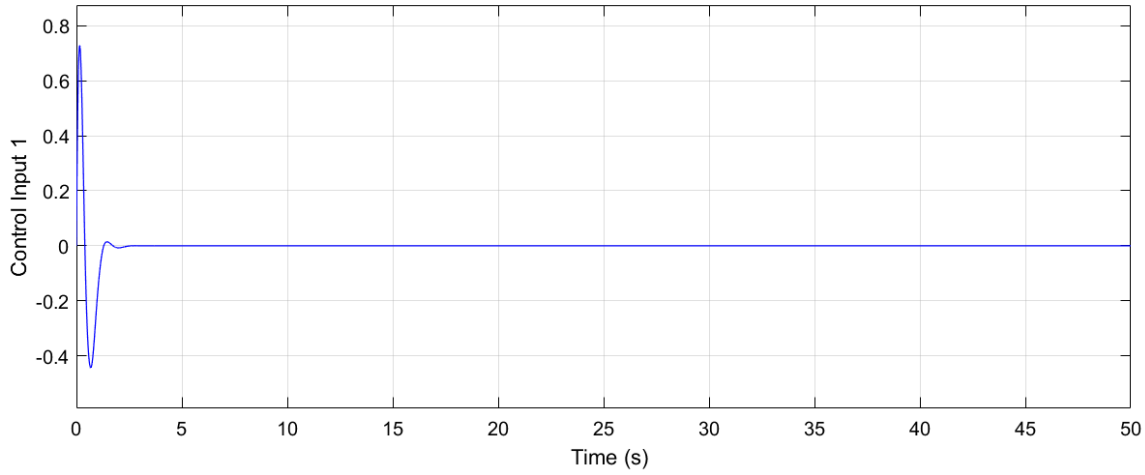

Figure 4.7 Change in control input 1 with time (NSMC)

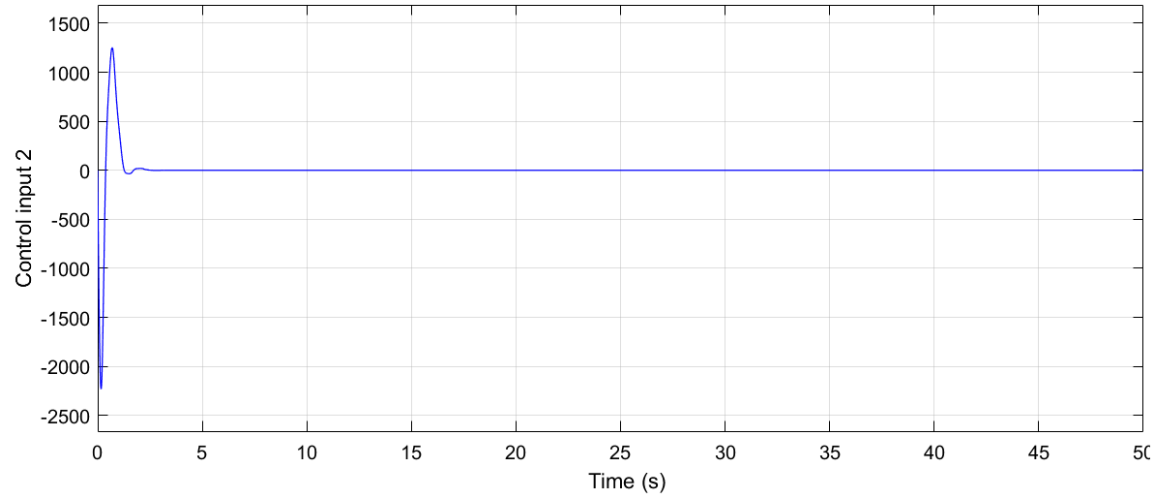

Figure 4.8 Change in control input 2 with time (NSMC)

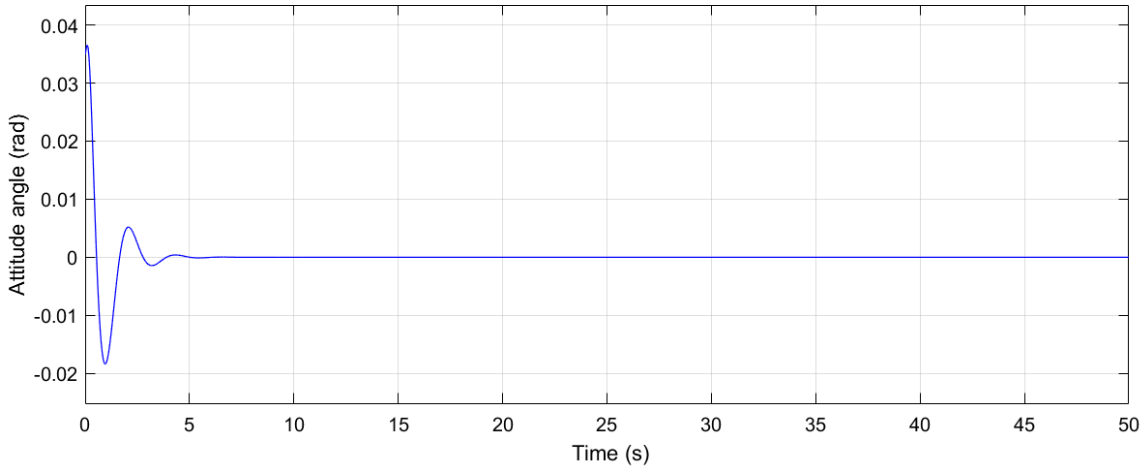

Figure 4.9 Change in attitude angle with time (SMC)

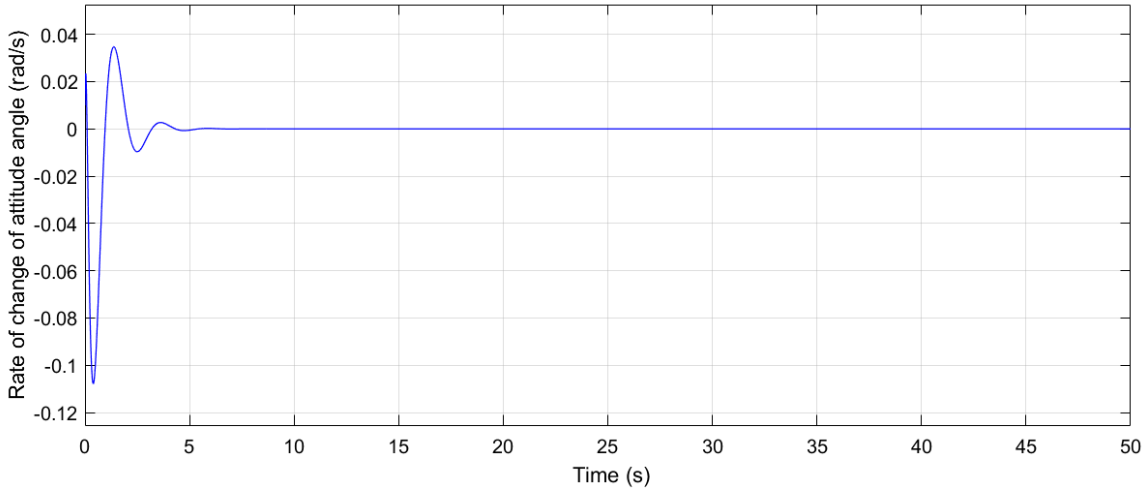

Figure 4.10 Rate of change of attitude angle with time (SMC)

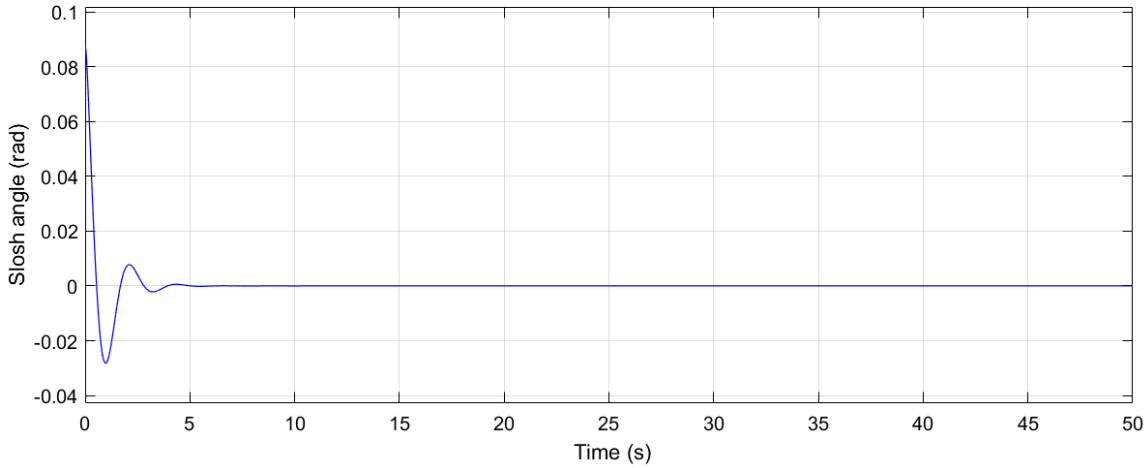

Figure 4.11 Change in slosh angle with time (SMC)

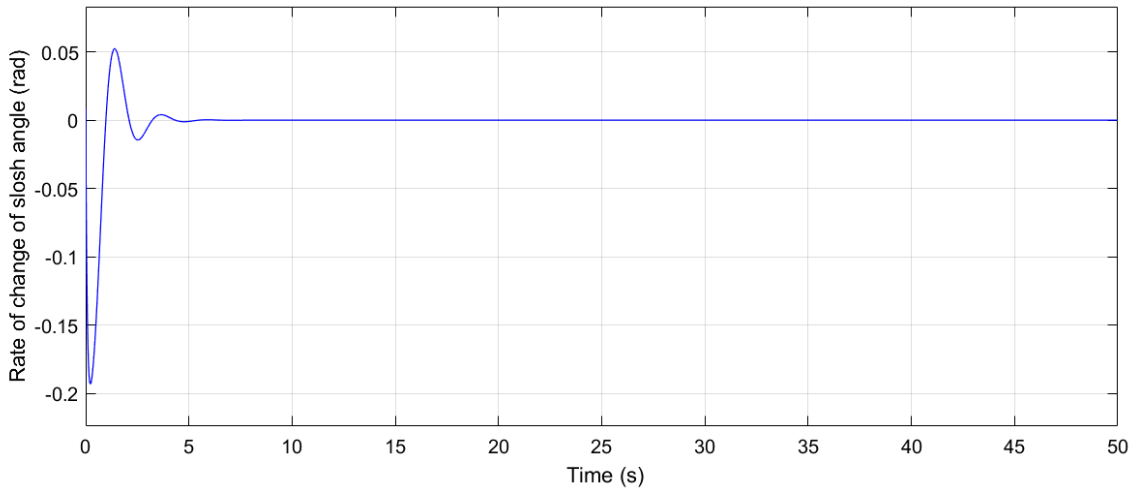

Figure 4.12 Rate of change of slosh angle with time (SMC)

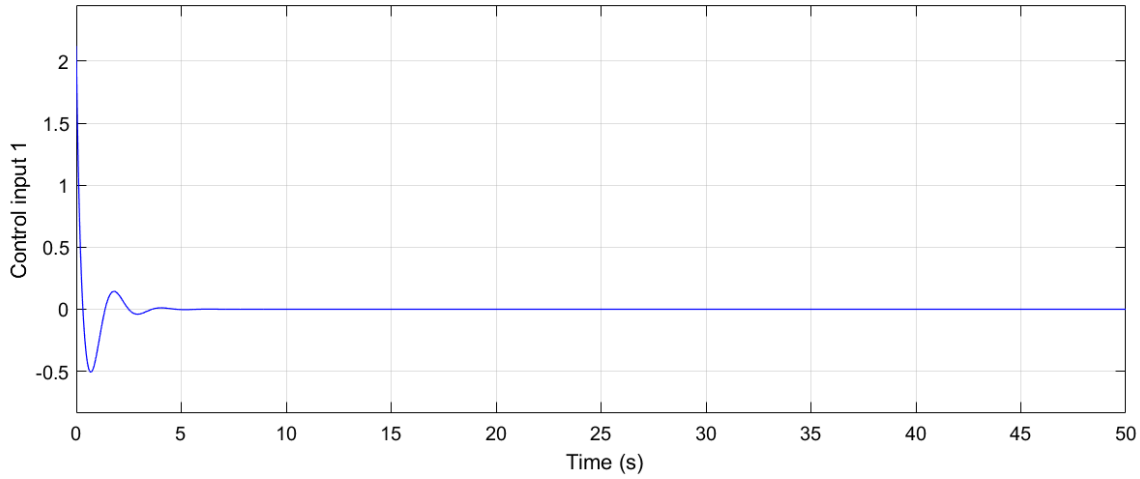

Figure 4.13 Change in control input 1 with time (SMC)

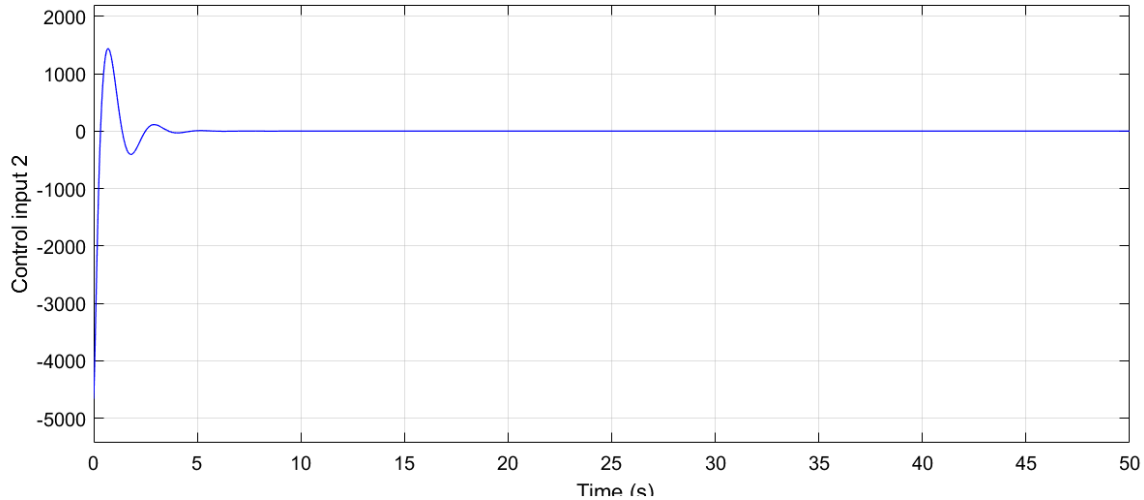

Figure 4.14 Change in control input 2 with time (SMC)

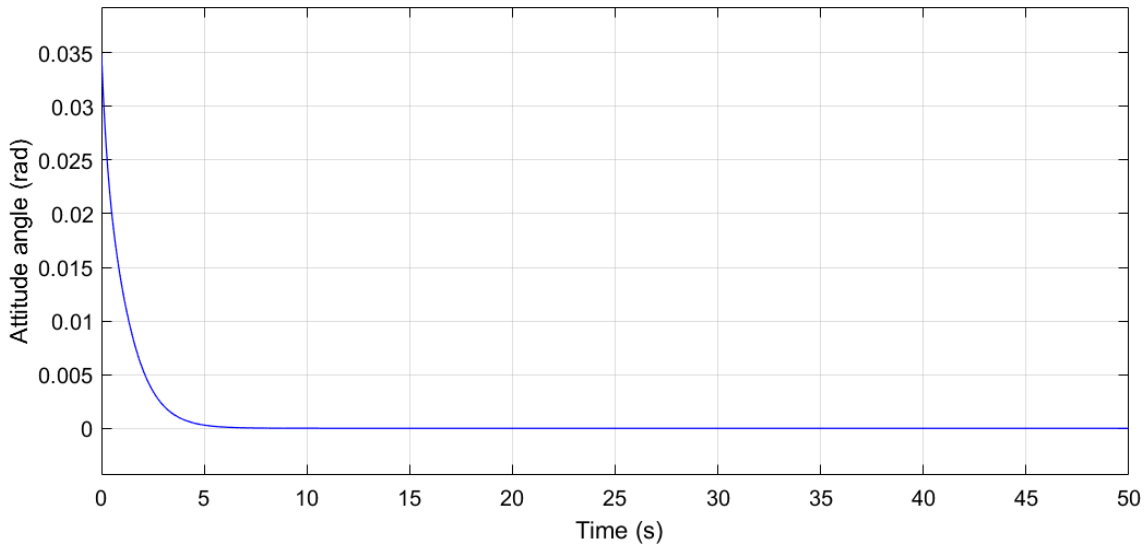

Figure 4.15 Change in attitude angle with time (PID)

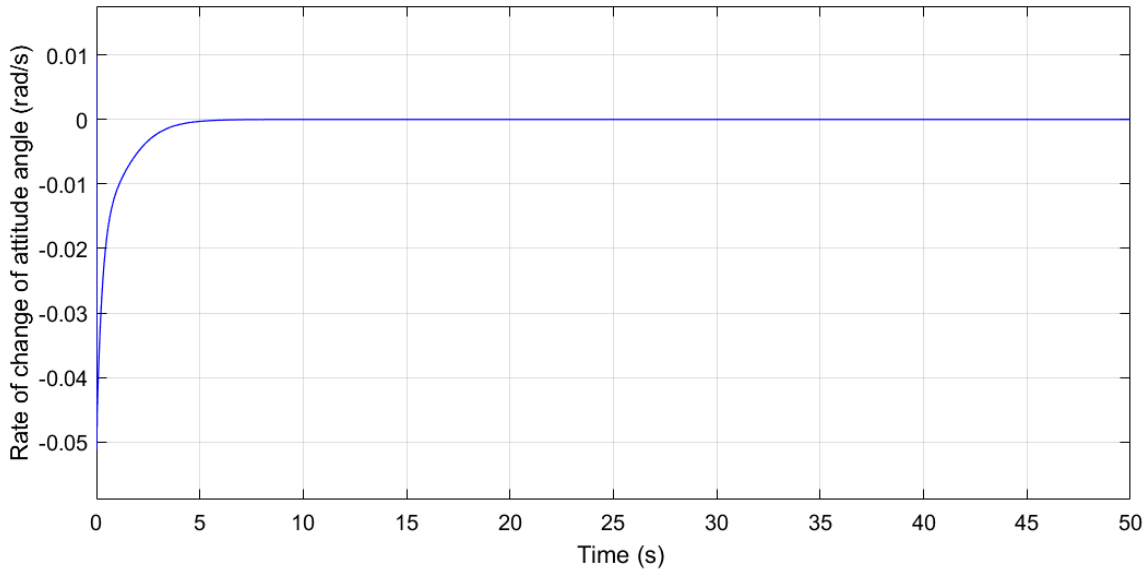

Figure 4.16 Rate of change of attitude angle with time (PID)

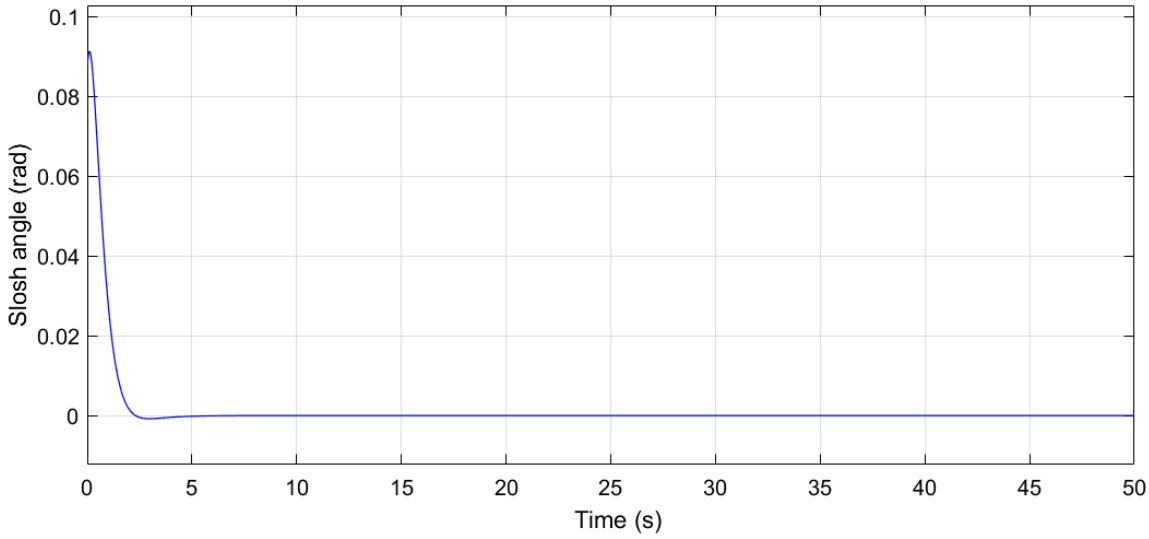

Figure 4.17 Change in slosh angle with time (PID)

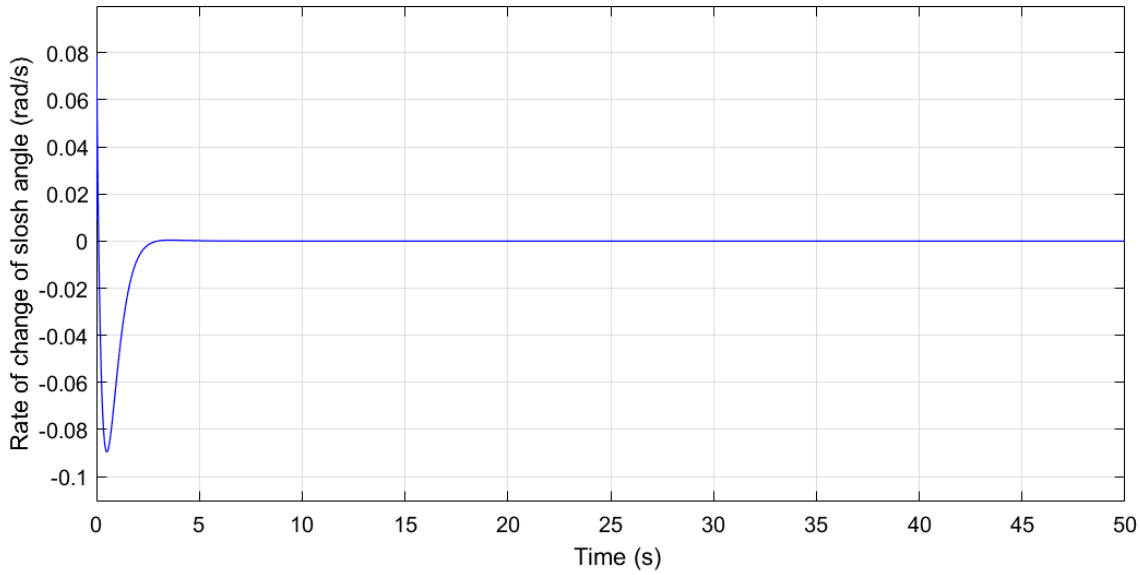

Figure 4.18 Rate of change of slosh angle with time (PID)

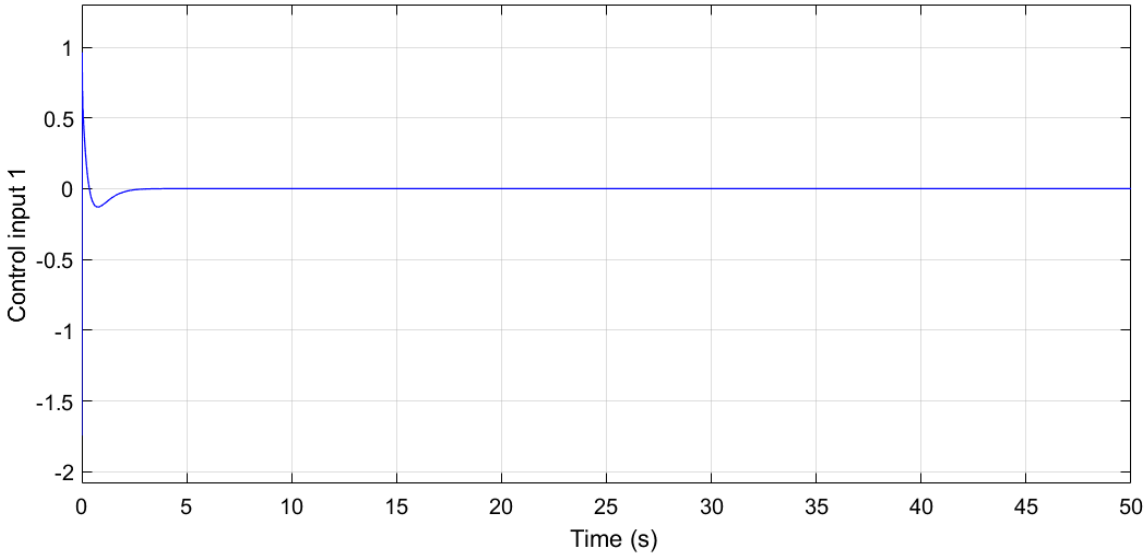

Figure 4.19 Change in control input 1 with time (PID)

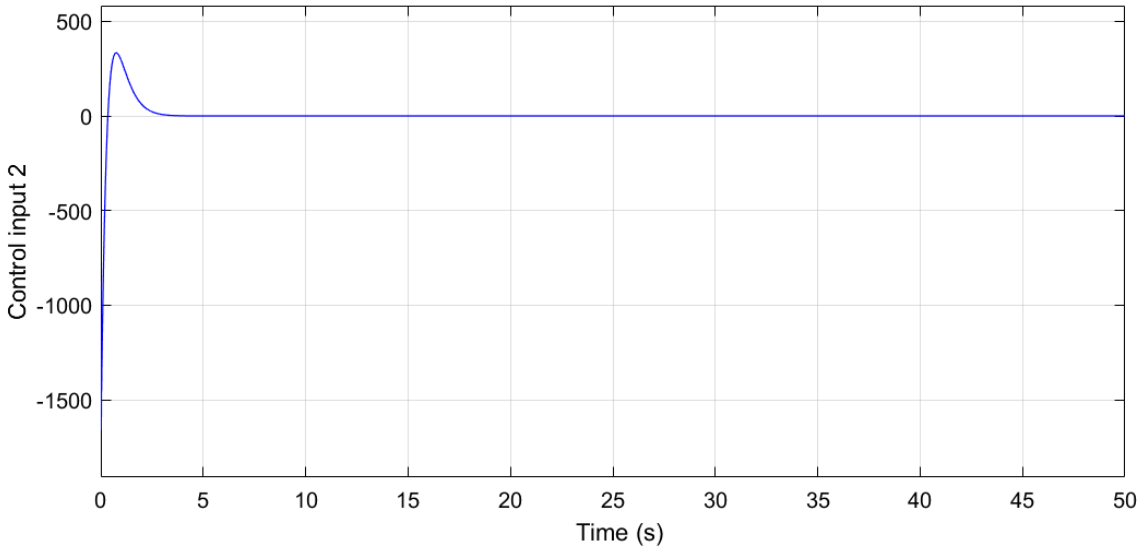

Figure 4.20 Change in control input 2 with time (PID)

To numerically compare the performances, we compared the RMSE values along with settling times when the fuel mass and its inertia were varied for the considered controllers. As the fuel mass can change significantly over the span of the satellite's lifetime, these three cases could mimic the satellite's condition in different stages of its lifespan. Tables (4.3)–(4.5) describes these conditions. The acquired data show that the PID controller has a consistent RMSE output across the three scenarios analysed. However, it takes a longer settling time in comparison. The SMC controller has much better settling times and demonstrates even better RMSE values on few occasions; however, it requires a significant control input compared to both the PID and NSMC. The proposed NSMC had the shortest settling time and overall better RMSE values in the simulations.

Table 4.3 Case 1 when fuel mass 100 kg and inertia 90 kgm^2

	Attitude angle	Rate change of attitude angle	Slosh angle	Rate change of slosh angle
NSMC	4.1582e-05	8.8477e-05	1.1011e-04	1.9456e-04
SMC	4.8753e-05	1.4493e-04	9.0887e-05	2.7529e-04
PID	4.8859e-05	5.0948e-05	1.3863e-04	1.5402e-04

Table 4.4 Case 2 when fuel mass mf =200 kg, inertia If= 180 kgm²

	Attitude angle	Rate change of attitude angle	Slosh angle	Rate change of slosh angle
NSMC	4.4397e-05	1.6089e-04	9.6246e-05	1.8310e-04
SMC	5.9135e-05	2.3973e-04	6.2464e-05	3.1238e-04
PID	4.8656e-05	5.1350e-05	1.4021e-04	1.5351e-04

Table 4.5 Case 3 when fuel mass mf =50 kg, inertia If= 45 kgm2

	Attitude angle	Rate change of attitude angle	Slosh angle	Rate change of slosh angle
NSMC	3.9797e-05	7.6495e-05	1.3524e-04	2.1607e-04
SMC	3.7387e-05	9.9721e-05	1.3907e-04	2.9892e-04
PID	4.8957e-05	5.0760e-05	1.3777e-04	1.5439e-04

Table 4.6 Settling times for each case considered

	Error angle			Slosh angle		
	Case 1	Case 2	Case 3	Case 1	Case 2	Case 3
NSMC	1.0	1.0	1.4	1.6	2.0	3.1
SMC	3.5	2.0	4.1	3.4	1.2	6.1
PID	3.7	3.7	3.7	3.2	2.6	3.0

Table 4.7 Accumulation of control input over the simulation time

Controller	Case 1		Case 2		Case 3	
	Control input 1	Control input 2	Control input 1	Control input 2	Control input 1	Control input 2
NSMC	0.4047	1114	0.7405	2048	0.3133	908.4
SMC	0.725	2028	0.9594	2476	0.5473	1556
PID	0.2487	591.4	0.2767	594.8	0.2343	590.5

The changes in transverse velocities in the z and x directions are described in Figs. (4.21) and (4.22). In both occasions, the accelerations in the transverse directions have been reduced to zero. However, the transverse velocity in the X direction has not been reduced to zero due to the inherent design of the controller and could be further studied

and improved. The rather high value of the controller input requirement is due to the fact that all three controllers work aggressively to stabilize the angle and fuel slosh errors within a small span of time. However, in a practical scenario, not only the time required to stabilize the system but also the controller input limitations need to be considered.

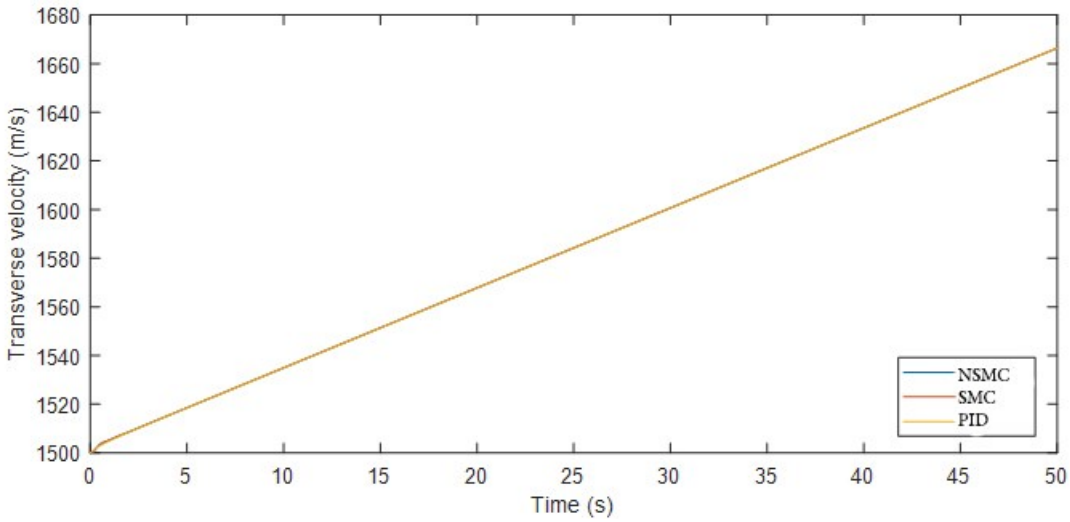

Figure 4.21 Transverse velocity vs time in the Z direction

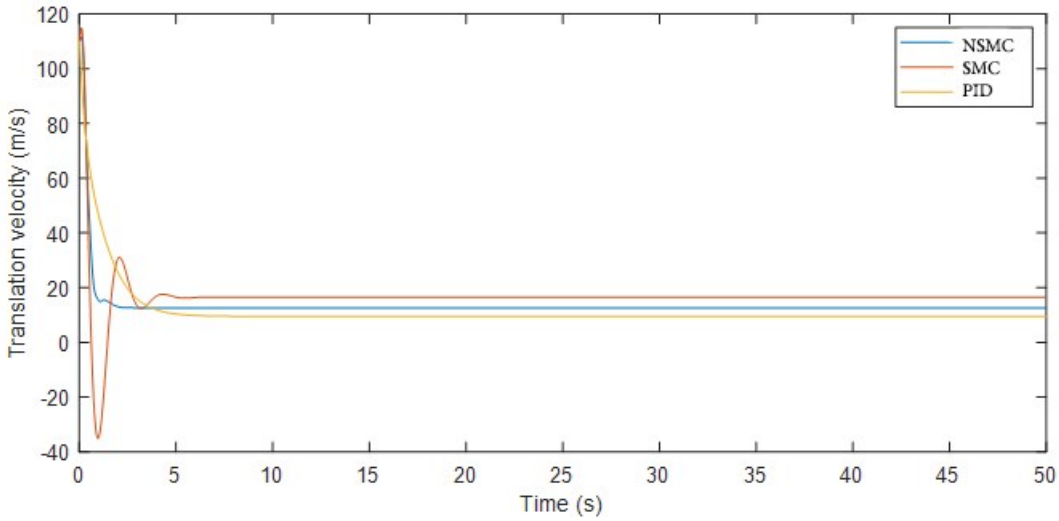

Figure 4.22 Transverse velocity vs time in the X direction

4.4. Summary

Space-related activities have considerably increased in the past several decades. With the total number of spacecrafts orbiting the earth increasing every year, the onboard control

algorithms also need to be improved to compensate for the potential risks involved in these complex systems. In this chapter, the irregularities due to fuel slosh are studied for a satellite in a fixed frame movement. A novel sliding mode controller was implemented to reduce the sloshing and attitude angle errors that arise in such situations. Through simulation data, we showed that this controller has a better performance compared with the other conventional controllers used in the comparison.

CHAPTER 5
DISCUSSIONS AND CONCLUSIONS

5.1. Discussion

The discussion is span over three points pertaining to the three chapters discussed above. It also includes the possible expansions on further research using these research topics as a guideline.

5.1.1. Discussion on MPC

One major focus for this research was to identify the implementation of MPC with region-free control to minimize the memory usage while realizing the required control objectives. Hence, we considered 2-dimensional movement. However, in practice, this is insufficient and both attitude and orbital control should be combined for precise manoeuvring in a 3-dimensional plane. This can increase the number of state variables and complexities of calculations. This must be considered in future work for system design and implementation.

Another hurdle in the implementation is the higher prediction horizon and the control horizon requirements for much smoother tracking control. In this research, these parameters were chosen as small values to limit calculation complexities. Among the three controllers considered, the MPC-based controller showed the best results against position tracking errors. However, the error in velocity tracking was compromised because of former considerations. Nevertheless, it is within a respectable margin of error, and given the fuel usage, it has shown much better results in the total simulation time. To further investigate the fuel minimization, different performance criterions can be analyzed while changing the coefficient matrices related to control inputs and considering different initial positions. This result could also be analyzed in future work to improve tracking performance while maintaining a small control horizon to suppress memory

requirements. It can be further extended with the usage of different solver algorithms to compare the processing times and memory usage since onboards microprocessor systems have hardware limitations.

5.1.2. Discussion on Neural based Adaptive Controller Design

A large amount of waste exists in space, some of which is still intact such as old difunctional satellites, rockets upper stages, and inactive payloads. These parts may have gone through corrosion, mechanical failures, and fuel loss, but the dynamics of the total systems can be approximated to a certain range. As a result, this type of control approach can be applicable to maneuvering such systems. As the attitude stability is more important than the time taken for the actual maneuver, a longer settling time is not a concern in this approach. Due to the high computational cost, a neural network-based controller can be problematic to implement. However, this type of algorithm can primarily be used for the stabilization part when capturing objects due to its adaptive qualities. In a similar manner, this can be utilized when releasing the captured object from a combined system in a controlled deorbiting situation.

This research theme can be further improved in few other areas as well. One such area is the consideration of the estimated inertia as a supplement to the neural controller itself as inputs. Another area could be the constraints and delays which arise when activating the propulsion systems similar to the case study in the orbit control scenario that can distort the control behavior. With different classes of satellite constellations active in space, considering such an object could considering improve this type of system for practical purposes. The calculation cost should also be considered since neural network-based controllers are inherently complex in design. With the development of

GPUs based on board computing units for space activities, such systems can be utilized and tested for performance cost analysis.

5.1.3. Discussion on Sliding Mode Controller Design

Another hindrance to designing control algorithms in general for spacecrafts is the fuel slosh situation. As satellites are made of several components, fuel itself is stored in a separate enclosure inside the satellite body. However, due to the inherent microgravity conditions in space, the liquid fuel can move freely inside the container, creating disturbances to the motion of the satellite. Here, in this research, disturbances in a fixed frame were explored; however, in a practical scenario, the satellite could be moving in 3-dimensional space and the motion in all directions, including the transverse and rotational directions, will have to be analysed for adequate controller design.

This could be further complicated if we consider the motion of flexible parts of the satellite such as the robotic arms and solar panels, and a general framework for such a system would help for both learning perspectives as well as for implementing different control algorithms to see how they perform.

Another area of improvement could be the use of higher order sliding mode control to achieve finite time control for this type of a system considering theoretical stabilization analysis, which can be extended to both fixed plane and multi plane situations.

5.2. Conclusion

As increasingly more debris clogs the valuable orbital space areas, it is a timely requirement to seek methods to actively remove space debris for the safety of future space missions. This process requires extensive research into many different fields. Analyzing such processes is both rigorous and time consuming. This dissertation is a small attempt to explore some of the issues space debris mitigation satellites face in space while performing orbital and attitude maneuvers. In the first chapter, an introduction is given regarding the bases this dissertation is going to cover. In the second chapter, an MPC based control algorithm is developed for a fly-around orbit control situation between a target and a chaser satellite to realize fuel optimization while considering limitations in the storage capabilities of satellites. In the next chapter, the development of an adaptive neural network-based control algorithm for an attitude control system while capturing objects with partially known inertial properties is detailed. In the final chapter, the sloshing situation is analysed while controlling a satellite in space and a SMC based control algorithm is developed to compensate for such situations. For all these situations, computer-based simulations have been conducted to compare the performance with those of several other conventional controllers to validate the developed control algorithms.

References

[1] E. Stansbery-Meter, Orbital Debris *Orbital Debris Quaterly News*, *Office*, vol. 25, no. 1. NASA Orbit Debris Program, 2021.

[2] B. K. I. Ki *et al.*, "Expected On-Orbit Tether Deployment Dynamics on the KITE Mission," *Jpn. Soc. Aeronaut. SP Sci.*, vol. 14, pp. 9–18, 2016.

[3] J. Mullick, S. Srinivasa, Y. Sahu, and A. Sata, "A Comprehensive Study on Space Debis, Threats Posed by Space Debris, and Removal Techniques," *SSRN Electron. J.*, 2019.

[4] H. Nakamoto, T. Maruyama, and Y. Sugawara, "Key Technology Demonstration for Active Debris Removal by Microsat 'DRUMS,'" in 1st International Orbital Debris Conf., 2019.

[5] H. Brettle, J. Forshaw, and J. Auburn, "Towards a Future Debris Removal Service: Evolution of an ADR Business Model,", no. October 21–25 in 70th International Astronaut. Congr. (IAC), 2019.

[6] Z. Ke, H. Zhenqi, L. Meibo, and O., Access, Study on Maintaining Formations During Satellite Formation Flying Based on SDRE and LQR, *Open Phys.*, vol. 2, pp. 394–399, 2017.

[7] Y. Kumar 'Applicability of Clohessy-Wiltshire Model for Optimal Low-Thrust Rendezvous,' *10.13140/RG.2.2.20998.83520*, no. February, 2019.

[8] A. Weiss, M. Baldwin, R. S. Erwin, and I. Kolmanovsky, "Model Predictive Control for Spacecraft Rendezvous and Docking: Strategies for Handling Constraints and Case Studies," *IEEE Trans. Control Syst. Technol.*, pp. 1–10, 2015.

[9] Q. Li, J. Yuan, B. Zhang, and C. Gao, “Model Predictive Control for Autonomous Rendezvous and Docking with a Tumbling Target,” *Aerosp. Sci. Technol.*, vol. 69, 700–711, 2017.

[10] H. Park, S. D. Cairano, I. Kolmanovsky, Model Predictive Control of Spacecraft Docking with a Non-Rotating Platform, *IFAC Proceedings Volumes*, vol. 44, no. 1, 8485–8490, 2011.

[11] L. Sauter and P. Palmer, “Analytic Model Predictive Controller for Collision-Free Relative Motion Reconfiguration,” *J. Guid. Control Dyn.*, vol. 35, no. 4, 1069–1079, 2012.

[12] E. Rogers and S. B. Gabriel, “Explicit Model Predictive Control Approach for Low-Thrust Spacecraft Proximity Operations,” *J. Guid. Control Dyn.*, vol. 37, no. 6, 2014.

[13] S. Di Cairano, H. Park, and I. Kolmanovsky, “Model Predictive Control Approach for Guidance of Spacecraft Rendezvous and Proximity Maneuvering,” *Int. J. Robust Nonlinear Control*, vol. 22, no. 12, 1398–1427, 2012.

[14] Y. Wang and S. Boyd, “Fast Model Predictive Control Using Online Optimization” in *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 2, pp. 267–278, 2010.

[15] J. Theunissen *et al.*, “Regionless Explicit Model Predictive Control of Active Suspension Systems with Preview,” *IEEE Trans. Ind. Electron.*, vol. 67, no. 6, pp. 4877–4888, 2020.

[16] M. Klauco, J. Drgona, F. Janecek, and M. Kvasnica, “Optimal Control of a Laboratory Binary Distillation Column via Regionless Explicit MPC,” *Comput. Chem. Eng.*, vol. 96, pp. 139–148, 2016.

[17] W. H. Clohessy and R. S. Wiltshire, “Terminal Guidance System for Satellite Rendezvous,” *J. Aerosp. Sci.*, vol. 27, no. 9, pp. 653–658, 1960.

[18] D. Cairano, J. Holaza, B. Takacs, and M. Kvasnica, “On Region-Free Explicit Model Predictive Control,” IEEE Conf. Decis. Control., pp. 3669–3674, 2015.

[19] S. D. Straight “Maneuver Design for Fast Satellite Circumnavigation,”, 2004, Thesis and Dissertations. 3941. https://scholar.afit.edu/etd/3941.

[20] M. A. C. Silva, M. Shan, A. Cervone, and E. Gill, “Fuzzy Control Allocation of Microthrusters for Space Debris Removal Using CubeSats,” *Eng. Appl. Artif. Intell.*, vol. 81, no. February, pp. 145–156, 2019.

[21] M. Shan, J. Guo, and E. Gill, “Review and Comparison of Active Space Debris Capturing and Removal Methods,” *Prog. Aerosp. Sci.*, vol. 80, pp. 18–32, 2016.

[22] J. Palimaka and B. Burlton, “Estimation of Spacecraft Mass Properties Using Angular Rate Gyro Data” in Guidance, Navigation and Control Conference, 1992.

[23] A. Y. Lee and J. A. Wertz, “In-Flight Estimation of the Cassini Spacecraft’s Inertia Tensor,” *J. Spacecr Rocket. - J Spacecr*, vol. 39, no. 1, pp. 153–155, 2002.

[24] D. H. Kim, S. Yang, D. I. Cheon, S. Lee, and H. S. Oh, “Combined Estimation Method for Inertia Properties of STSAT-3,” *J. Mech. Sci. Technol.*, vol. 24, no. 8, pp. 1737–1741, 2010.

[25] D. H. Kim, D. G. Choi, and H. S. Oh, “Inertia Estimation of Spacecraft Based on Modified Law of Conservation of Angular Momentum,” *J. Astron. SP Sci.*, vol. 27, no. 4, pp. 353–357, 2010.

[26] S. Yang, S. Lee, J. H. Lee, and H. S. Oh, “New Real-Time Estimation Method for Inertia Properties of STSAT-3 Using Gyro Data,” *Trans. Jpn. Soc. Aeronaut. Space Sci.*, vol. 58, no. 4, pp. 247–249, 2015.

[27] D. Kim, S. Yang, and S. Lee, “Rigid Body Inertia Estimation Using Extended Kalman and Savitzky-Golay Filters,” *Math. Probl. Eng.*, vol. 2016, 1–7, 2016.

[28] L. L. S. Show, J. C. Juang, C. T. Lin, and Y. W. Jan, “Spacecraft Robust Attitude Tracking Design: PID Control Approach” Proc. Am. Control. Conf., 2, pp. 1360–1365, 2002.

[29] Q. Wang, J. Yuan, and Z. Zhu, “The Application of Error Quaternion and PID Control Method in Earth Observation Satellite’s Attitude Control System,” 2006 1st Int. Symp. Syst. Control. Aerosp. Astronaut., pp. 128–131, 2006.

[30] J. Li, M. Post, T. Wright, and R. Lee, “Design of Attitude Control Systems for CubeSat-Class Nanosatellite,” *J. Control Sci. Eng.*, vol. 2013, 1–15, 2013.

[31] B. Wie and J. Lu, “Feedback Control Logic for Spacecraft Eigenaxis Rotations Under Slew Rate and Control Constraints,” *J. Guid. Control Dyn.*, vol. 18, no. 6, pp. 1372–1379, 1995.

[32] Y. Li and D. Ye, “Robust PID Controller for Flexible Satellite Attitude Control Under Angular Velocity and Control Torque Constraint,” *Asian J. Control*, vol. September 2018, pp. 1–18, 2019.

[33] B. Wie and P. M. Barba, “Quaternion Feedback for Spacecraft Large Angle Maneuvers,” *J. Guid. Control Dyn.*, vol. 8, no. 3, pp. 360–365, 1985.

[34] M. M. Ismail, “Adaptation of PID Controller Using AI Technique for Speed Control of Isolated Steam Turbine” Proc. 2012 Jpn.-Egypt Conf. Electron. Commun. Comput. JEC-ECC 2012, pp. 85–90, 2012.

[35] N. Najafizadeh Sari, H. Jahanshahi, and M. Fakoor, “Adaptive Fuzzy PID Control Strategy for Spacecraft Attitude Control,” *Int. J. Fuzzy Syst.*, vol. 21, no. 3, pp. 769–781, 2019.

[36] C. H. Cheng and S. L. Shu, “Application of Fuzzy Controllers for Spacecraft Attitude Control,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 45, no. 2, pp. 761–765, 2009.

[37] W. M. Van Buijtenen, G. Schram, R. Babuška, and H. B. Verbruggen, “Adaptive Fuzzy Control of Satellite Attitude by Reinforcement Learning,” *IEEE Trans. Fuzzy Syst.*, vol. 6, no. 2, pp. 185–194, 1998.

[38] M. R. Dizadji, A. Yousefi-Koma, and Z. Gharehnazifam, “Direct Fuzzy Controller” in IcRoM 6th RSI Intl Conf. on Robot. and Mechatron. (IcRoM), pp. 1–5, 2018, no.

[39] M. Suzuki, T. Yamamoto, and T. Tsuji, “A Design of Neural-Net Based PID Controllers with Evolutionary Computation,” *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E87-a, no. 10, pp. 2761–2768, 2004.

[40] A. N. Ponce, A. A. Behar, A. O. Hernández, and V. R. Sitar, “Neural Networks for Self-Tuning Control Systems,” *Acta Polytech.*, vol. 44, no. 1, pp. 49–52, 2004.

[41] T. Yamamoto, M. Kaneda, and T. Oki, “A Self-Tuning PID Controller Fused Artificial Neural Networks,” *IFAC Proc.*, vol. 29, no. 1, pp. 5066–5071, 1996.

[42] R. Hernández-Alvarado, L. G. García-Valdovinos, T. Salgado-Jiménez, A. Gómez-Espinosa, and F. Fonseca-Navarro, “Neural Network-Based Self-Tuning PID Control for Underwater Vehicles,” *Sensors (Basel)*, vol. 16, no. 9, p. 1429, 2016.

[43] J. Li and A. Gomez-Espinosa, “Improving PID Control Based on Neural Network” Proc. −2018 Int. Conf. Mechatron., Electron. *Automot. Eng. ICMEAE*, vol. 2018, pp. 186–191, 2018.

[44] B. S. L. Ee, B. L. Ee, and J. K. Im, “A Strategy to Determine Whether to Use GPU for a Satellite Mission Scheduling Algorithm,” *Jpn. Soc. Aeronaut. SP Sci.*, vol. 55, no. 3, pp. 166–174, 2012.

[45] F. C. Bruhn, N. Tsog, F. Kunkel, O. Flordal, and I. Troxel, “Enabling Radiation Tolerant Heterogeneous GPU - Based Onboard Data Processing in Space,” *CEAS Space J.*, vol. 12, no. 4, pp. 551–564, 2020.

[46] G. Kang, J. Wu, C. Jin, and X. Chen, “Adaptive Controller Design for Satellite Attached by Non-Cooperative Object,” *Chin. J. Aeronaut.*, vol. 33, no. 3, pp. 1006–1015, 2020.

[47] H. Zhang and Z. Wang, “Attitude Control and Sloshing Suppression for Liquid-Filled Spacecraft in the Presence of Sinusoidal Disturbance,” *J. Sound Vib.*, vol. 383, pp. 64–75, 2016.

[48] S.X. Yu and Q. R. Yun Using Sliding Mode Control Method to Suppress Fuel Sloshing of a Liquid-Filled Spacecraft,“*Proc*, 2015, no. 61374116 27th Chin. Control. Decis. Conf. CCDC, pp. 1268–1273, 2015.

[49] M. Reyhanoglu, Modeling and Control of Space Vehicles with Fuel, *InTech*, 2011.

[50] A. G. De Souza and L. C. G. De Souza, “Satellite Attitude Control System Design Taking into Account the Fuel Slosh and Flexible Dynamics,” *Math. Probl. Eng.*, vol. 2014, 1–8, 2014.

[51] J. W. Thompson and W. J. O’Connor, “Wave-Based Attitude Control of Spacecraft with Fuel Sloshing Dynamics” Proc. ECCOMAS Themat. Conf. Multibody Dyn, vol. 2015, pp. 968–977, 2015*, Multibody Dyn.*

[52] M. Hussein and L. Qian, “Dynamic Modeling and Attitude Control System of Underactuated Fuel Slosh with Flexible Appendages,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 631, no. 3, 2019.

[53] M. Navabi and A. Davoodi, “3D Multi-Pendulum Model of Slosh Dynamics in a Spacecraft” Proc. 8th Int. Conf. Recent. Adv. Sp. Technol. RAST 2017, pp. 259–262, 2017.

[54] W. Perruquetti and J. P. Barbot Sliding Mode Control in Engineering. CRC Press, 2002.

[55] F. Dinuzzo, A. Ferrara, and S. Member, “Higher Order Sliding Mode Controllers with Optimal Reaching,” *IEEE Trans. Autom. Contr.*, vol. 54, no. 9, pp. 2126–2136, 2009.

[56] R. Poli, J. Kennedy, and T. Blackwell, “Particle Swarm Optimization: An Overview,” *Swarm Intell.*, vol. 1, no. 1, pp. 33–57, 2007.

[57] M. R. Soltanpour and M. H. Khooban, “A Particle Swarm Optimization Approach for Fuzzy Sliding Mode Control for Tracking the Robot Manipulator,” *Nonlinear Dyn.*, vol. 74, no. 1–2, pp. 467–478, 2013.

[58] N. Coulter ‘Design of an Attitude Control System for a Spacecraft with Propellant Slosh Dynamics,’ PhD Dissertations and Master’s Theses. https://commons.erau.edu/edt/424, 424, 2018.

Publication List

Papers that constitute the dissertation.

(1) Model Predictive Control-based Control Algorithm for a Target-Chaser Maneuvering Situation

Amila Sri Madhushanka Gilimalage, Shinichi Kimura

Advanced Robotics (2021), 35:21-22, 1265-1276

DOI:10.1080/01691864.2021.1955002

(2) Development of a Mode-Predictive-Control-based Algorithm for a Target-Chaser Rendezvous Maneuvering Scenario

Amila Sri Madhushanka Gilimalage, Shinichi Kimura

The International Symposium on Artificial Intelligence, Robotics and Automation in Space 2020, 5018（i-SAIRAS 2020）

(3) Attitude Control Algorithm for Satellites with Partially Known Inertial Properties

Amila Sri Madhushanka Gilimalage, Shinichi Kimura, Manukid Parnichkun

Proceedings of the 71st International Astronautical Congress (IAC 2020)

(4) Development of a Novel Sliding Mode based Control Algorithm for a Satellite with Sloshing Dynamics

Amila Sri Madhushanka Gilimalage, Shinichi Kimura

(Accepted and in press) 33rd International Symposium on Space Technology and Science (ISTS 2022)