



Modelling and Simulation of GNC/AOCS Systems for Conceptual Studies

by

Louise Lindblad

July 2013

Master Thesis from

Royal Institute of Technology

Department of Mechanics

SE-100 44 Stockholm, Sweden

and

European Space Technology and Research Center
Guidance, Navigation, and Control Section (TEC-ECN)
Keperlaan 1, 2201 AZ Noordwijk, The Netherlands

Preface

This report was written at the Guidance, Navigation, and Control Systems Division at ESTEC, the European Space Agency, in the Netherlands between February 2013 and July 2013 as a Master thesis for the Department of Mechanics at the Royal Institute of Technology, Stockholm, Sweden.

I would like to thank Dr. Gunnar Tibert for supervising this project from Stockholm and for his contribution to the revision of the report. I would also like to thank Dr. Guillermo Ortega and Ms. Cèlia Yábar Vallès for supervising this project at ESA and for giving me much support and guidance throughout the internship.

Stockholm, July 2013

Louise Lindblad

Abstract

The preliminary design of Guidance, Navigation, and Control (GNC) algorithms and Attitude and Orbital Control Systems (AOCS) for spacecraft plays an important role in the planning of a space mission. Many missions require accurate positioning and attitude control to be able to fulfil the mission objectives. In an early phase of conceptual studies, trade-offs have to be made to the GNC/AOCS subsystem design and compromises with respect to other subsystem designs have to be taken into account. This demands for the possibility of rapid prototyping where design parameters, such as the choice of sensors and actuators, can be changed easily to assess the compliance to the mission requirements. This thesis presents the modelling of GNC/AOCS components for a toolbox created in the MATLAB/Simulink environment. The resulting toolbox is a user-friendly tool, which simplifies the creation of GNC/AOCS system simulations for conceptual studies. A number of complete simulations were constructed to demonstrate the capabilities of the toolbox.

Contents

Preface	iii
Abstract	v
1 Introduction	1
1.1 The European Space Agency	2
1.2 The Concurrent Design Facility	3
1.3 The GNC/AOCS Hardware Database	4
1.4 Aims and Scope	4
2 The GAST Toolbox	7
2.1 Contents of the Toolbox	8
2.2 Building Simulations	10
2.3 Using GAST	11
3 Definitions and Notations	13
3.1 Rotations	13
3.2 Coordinate Frames	15
4 Modelling Sensors and Actuators	17
4.1 Sensors	17
4.1.1 Rate Sensor	18
4.1.2 Laser Range Finder	20
4.1.3 Accelerometer	24

4.2	Actuators	27
4.2.1	Reaction Wheels	28
4.2.2	Control Moment Gyroscope	30
4.2.3	Thrusters	33
5	Simulations using GAST	37
5.1	De-spinning using Quaternion Control	37
5.2	Yaw Steering in Sun-Synchronous Orbit	40
5.2.1	Orbit Definition	41
5.2.2	Solar Panel Coordinate Frame	42
5.2.3	Sun Direction Vector	42
5.2.4	Optimal Yaw Angle	44
5.2.5	Optimal Solar Array Angle	45
5.2.6	GAST Model	46
5.2.7	Results	47
5.3	Control System Design for a Geostationary Satellite	50
5.3.1	Pitch Control Design	51
5.3.2	Roll/Yaw Control Design	52
5.3.3	Simulation Results	53
5.4	Mars Ascent Vehicle	56
5.4.1	Dynamics	57
5.4.2	Environment	57
5.4.3	Aerodynamic Model	58
5.4.4	Propulsion	60
5.4.5	Mass Evolution	62
5.4.6	Inertial Measurement Unit	63
5.5	Active Debris Removal	63
5.5.1	Approach Strategy	63
5.5.2	Relative Dynamics	65

5.5.3	Close Range Rendezvous	65
5.5.4	Forced Motion	67
5.5.5	Out-of-plane Inspection	69
6	Conclusion	71
	Bibliography	73

Chapter 1

Introduction

During the initial phases of planning a space mission, conceptual studies have to be made in order to assess the feasibility of a project. A preliminary design is made for all subsystems and the advantages and drawbacks of different solutions are taken into consideration. An important subsystem to study is the one dealing with Guidance, Navigation, and Control (GNC). To accomplish the goals of a space mission, it is necessary to track the position and attitude of a spacecraft throughout the whole mission and especially during critical manoeuvres. This is done with a navigation system consisting of a combination of sensors and filters to measure the needed spacecraft states. The guidance functions are necessary to calculate the desired position and attitude of a spacecraft and how to achieve this in an optimal way. The control system makes sure that the spacecraft follows the guidance command in a smooth way that ensures stability of the system despite the presence of disturbances and modelling uncertainties. GNC systems are part of every spacecraft to be able to achieve the specific mission goals. The mission goal for a launcher would be to transport a satellite to a specific orbit and the mission goal for an observation satellite would be to constantly point towards a specific target. In both cases the GNC systems play a central role in achieving these goals. The in orbit control of a satellite is managed by the satellite's Attitude and Orbital Control Systems (AOCS), which is used to maintain the desired attitude and orbit. To assess the performance of a GNC/AOCS system during conceptual studies, computer simulations that model the behaviour of the satellite together with the GNC/AOCS system are constructed. From these simulations the subsystem components and guidance, navigation, and control algorithms that are most suitable and comply with the mission requirements can be selected and developed.

A computer simulation of a spacecraft GNC/AOCS system must include the spacecraft state model with the applicable differential equations, GNC functions, and models of sensors and actuators. A general closed loop GNC block diagram found in [8] is shown in Fig. 1.1.

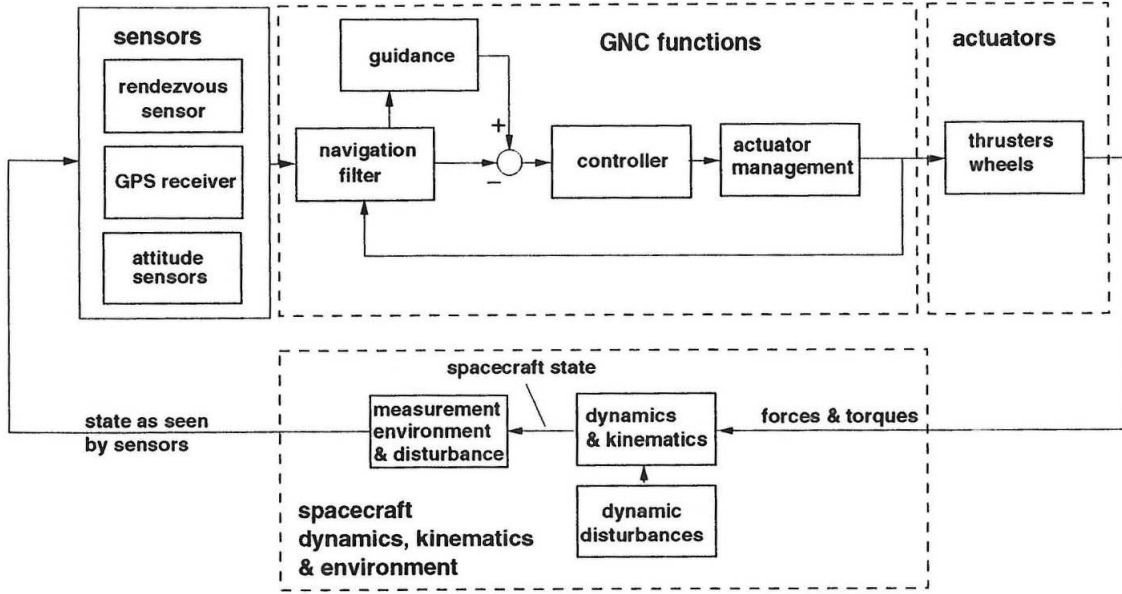


Figure 1.1: Block diagram of a GNC/AOCS system simulation [8].

The spacecraft state model contains the dynamic and kinematic equations, environment models, and disturbance models. The GNC functions includes guidance algorithms, navigation filters, controllers, and possibly actuator management functions. The sensors are modelled to represent the measurements of a real sensor on a spacecraft. The inputs to the sensor models are the states calculated from the dynamic and kinematic equations and the outputs represent the states measured by the sensors with error models. In the actuator models, the inputs are the forces or torques commanded by the controller and the outputs are the realised forces and torques which are then fed back to the spacecraft state model to close the loop.

1.1 The European Space Agency

The work for this thesis was conducted within the scope of an internship at the European Space Agency (ESA) for the Guidance, Navigation, and Control Systems Section. The purpose of ESA, as stated in the ESA Convention, is to “provide for and promote, for exclusively peaceful purposes, cooperation among European states in space research and technology and their applications”. ESA currently has 20 Member States and about 2200 staff in the five establishments in Europe. The European Space Research and Technology Centre (ESTEC) in Noordwijk, the Netherlands, is the technical heart of ESA and the largest of its sites. The Guidance, Navigation, and Control Systems Section (TEC-ECN) is situated at ESTEC and is responsible for project support and technology development of space applications in the areas like AOCS design and implementation (for Earth Observation, Telecom,

Navigation), GNC systems design and implementation (for Planetary exploration missions, Launchers, Ascent and Re-entry vehicles), development of advanced control and estimation techniques and tools, satellite dynamics modelling, trajectory optimisation, and more.

1.2 The Concurrent Design Facility

The Concurrent Design Facility (CDF) was established at ESTEC in November 1998 to develop concepts for future space missions. The CDF is a facility equipped with a network of computers, multimedia devices, and software tools, which results in an environment where experts from all subsystem disciplines are able to work together with the system engineers to efficiently design optimal solutions for a given future space mission. The technical and financial feasibility of a future space mission or a new spacecraft concept is studied and evaluated. The studies include, among others, new mission concept assessment, space system trade-offs and options evaluation, and new technology validation at system/mission level.

The GNC engineers in TEC-ECN have supported 51 CDF studies in the past 10 years. Some examples of studies from 2012 are:

- AIM - Asteroid Impactor Mission
- e.Deorbit - Assessment of feasibility for controlled deorbit of Envisat
- AGILE 2 - Extended Advanced Galileo Injection Using Electric Propulsion
- ADI - Asteroid Deflection by Ions Siroco MiCRA
- MMSR A5 - Moons of Mars Sample Return Using Ariane 5 Launcher

The support provided by the TEC-ECN section focuses on the following areas:

- **Navigation:** the process of determining the vehicle's position and velocity or, equivalently, its orbital elements as a function of time
- **Guidance:** the process of defining the optimal state vectors of the vehicle as a function of time
- **Control:** the process of deriving the commands to match the current and optimal vehicle state vectors
- **Fault Detection Isolation and Recovery (FDIR):** the means to detect off-normal conditions, isolate the problem to a specific subsystem/component, and recovery of vehicle systems and capabilities
- **Health Monitoring Subsystems:** capability of detecting a fault as it occurs and identify the faulty component

The mission types supported by GNC engineers include Ascent, Re-entry and Descent, Landing, Rendezvous and Docking, Formation Flying, and Interplanetary missions.

To achieve the goals of a CDF study, models and simulations of the GNC systems are constructed by the TEC-ECN section and the performance is evaluated to find the best solution for a particular study.

1.3 The GNC/AOCS Hardware Database

In CDF studies the choice of sensors and actuators for a mission is important. In the TEC-ECN section, there exists an GNC/AOCS Hardware Database which contains about 150 records of existing and relevant GNC sensors and actuators. The sensor and actuator types in the Database are listed in Table 1.1.

Sensors	Actuators
Star Tracker	Torque Rods
Sun Sensor	Wheels
Earth Sensor	Control Momentum Gyro
Gyrometer	
Accelerometer	
Magnetometer	
GNSS	
Navigation Camera	

Table 1.1: Sensors and actuator types in the Hardware Database.

For each sensor or actuator type there is a list of the ones existing on the market. Each specific hardware component has the characteristics from its data sheet listed in an Excel table. The variables in the Hardware Database can be used in simulations of the GNC/AOCS systems in CDF studies to determine which sensors and actuators are more suitable for a specific mission. The comparison between different sensors and actuators can be facilitated if the parameters from the Hardware Database can be loaded automatically into the simulations.

1.4 Aims and Scope

It was realised in TEC-ECN that there was a need for the engineers supporting CDF studies of having a user-friendly toolbox using MATLAB/Simulink to be able to do fast simulations, even during a CDF session itself. The purpose of the proposed work for this thesis was to support TEC-ECN by developing a AOCS/GNC toolbox through the harmonisation of toolboxes existing in the section and adding simulation models according to the needs for CDF studies.

The resulting GNC and AOCS Simulations Toolbox (GAST) is a MATLAB/Simulink based toolbox which can be used in the support of the CDF studies allowing rapid

prototyping and being supported by the sensors and actuators Hardware Database of TEC-ECN. This thesis shows the development, design, and operations of the GAST toolbox and its components, including demonstrations of its capabilities through several complete simulations of GNC/AOCS systems.

Chapter 2

The GAST Toolbox

In this chapter, the GAST toolbox and its features are described. The GAST toolbox was developed within the scope of this thesis as a simulation tool in MATLAB/Simulink. It was designed to be user-friendly and contain blocks that easily can be put together to perform simple and rapid simulations of GNC/AOCS system designs.

The GAST toolbox is the result of the consolidation of several toolboxes available in TEC-ECN such as the old AOCS Toolbox, the SpaceLAB library, the ViSiLib library, the ATPE simulator, and the PAV simulator. In addition to consolidating these toolboxes, new models were developed for the GAST toolbox according to the needs of the section. Figure 2.1 shows a pictorial representation of the consolidation of the toolboxes of TEC-ECN. This thesis will only present the newly developed models and simulations constructed for the GAST toolbox.

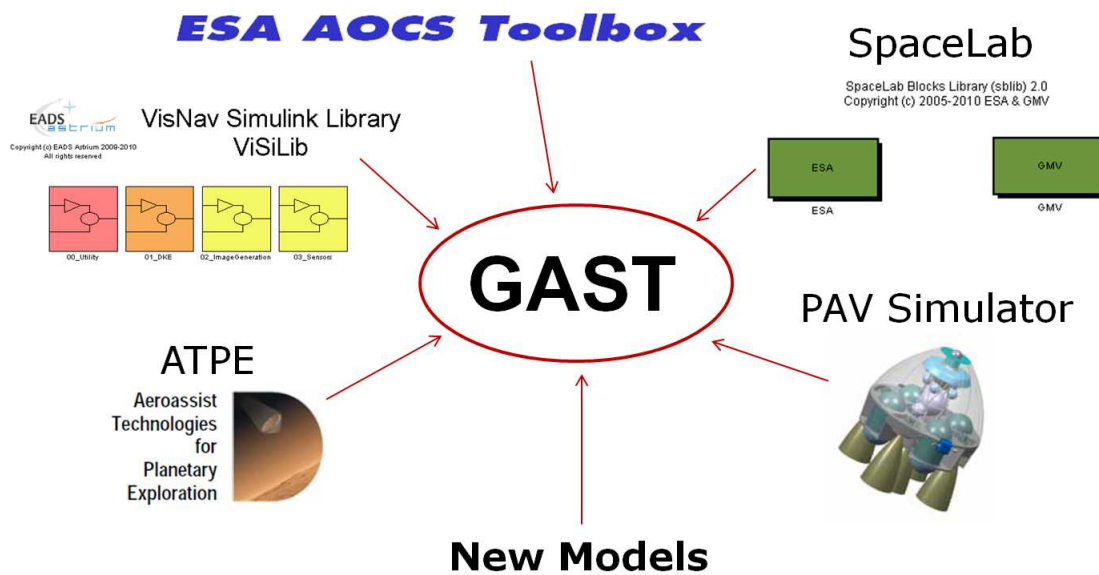


Figure 2.1: Representation of the consolidation of TEC-ECN toolboxes.

The internal software structure of the GAST toolbox is based on the previous ESA AOCS Toolbox created in TEC-ECN in 2008. A variety of simulation blocks have been newly constructed and added to the old toolbox structure, while making sure that all blocks have standard conventions and variable naming. The inputs and outputs of the blocks, the coordinate frame conventions, and the parameter values are all well explained to avoid confusion.

The features inherited from other existing toolboxes include a suit of supporting functions which provides the user with capabilities such as automatic initialisation of the library blocks, and plotting as well as animation tools.

Moreover, all blocks are associated with a “Load” MATLAB function from where sets of parameter values for each block can be retrieved. The Load functions have been synchronised with the Hardware Database available in TEC-ECN to include all the sensors and actuators needed for simulations. The Load function allows the user to easily choose a specific sensor or actuator to be used in a simulation and the parameter values are loaded automatically.

A selection of blocks have an “Auto-test” MATLAB functionality, where several simulation tests are performed in a MATLAB script and the results are compared to analytical or predetermined values. The tests are run automatically when installing the toolbox to ensure that the blocks are up to date and function properly.

All these features are fully integrated into the MATLAB/Simulink environment R2011b. This was achieved within the scope of the development of the previous ESA AOCS Toolbox by integrating the toolbox library into the Simulink library browser, the Toolbox help into the MATLAB help browser as well as adding custom menus into Simulink to access the different features that the library provides.

2.1 Contents of the Toolbox

The structure of GAST and the subcategories of the toolbox are shown in Fig. 2.2.

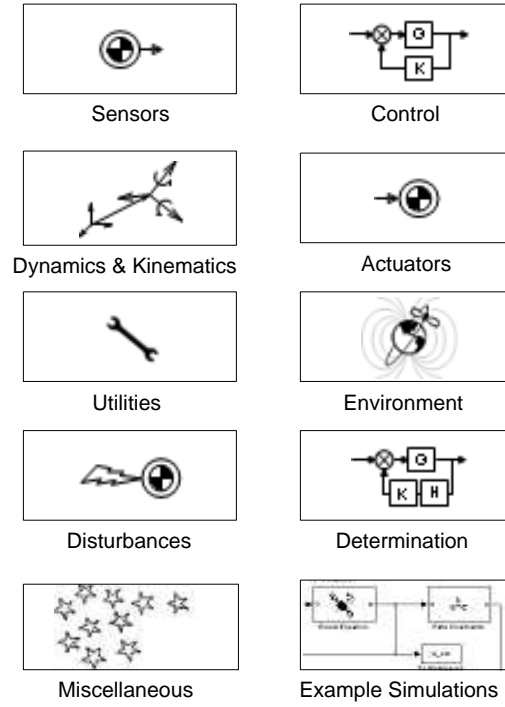


Figure 2.2: Block categories of the GAST Toolbox.

The blocks available in GAST, sorted by category, are:

- **Actuators:** Reaction Wheels, Reaction Wheels Simple, Magnetorquers, Thrusters, Simple Thruster, Thrust Management Function, Engine Management Differential Thrust, CMG 1-axis, CMG 3-axis
- **Control:** PID, Quaternion Control, PWM, Thrust Distribution, Bang Bang Controller, Lead Lag Compensator, Thrust Vector Control, Actuator Control System
- **Determination:** GyroStellar
- **Disturbances:** Gravity Gradient, Aerodynamic Torque, Aerodynamic Forces and Torques, Solar Pressure Torque, Dynamic Pressure
- **Dynamics & Kinematics:** Attitude Dynamics, Eulers Equation, Rate Kinematics, Linear Attitude Model, Attitude Dynamics and Euler Kinematics, Attitude Dynamics and Quaternion Kinematics, Translational Equations of Motion, Rotational Equations of Motion, Orbit2Polar, Launcher Dynamics 6 DoF, Clohessy Wiltshire Equations 2D, Clohessy Wiltshire Equations 3D.
- **Environment:** Orbital Propagator, Magnetic Dipole, Sun Direction, Earth Direction, Right Ascension of Zero Meridian, Eclipse, Orbital Frame, Gravity Goddard Earth Model, Zonal Harmonics 4th Order Gravity Model, Mars Atmosphere
- **Miscellaneous:** Attitude Sensor Misalignment, Rate Sensor Misalignment, Angle of Attack and Sideslip, Propellant Consumption, Thruster Fuel Con-

sumption, CoG and Inertia Matrix Evolution, Mass Releases, Total Mass of 2-stage Launch vehicle

- **Sensors:** Rate Sensor Simple, Rate Sensor Complex, Star Tracker Quaternion, Star Tracker Euler Angles, Magnetometer, GPS, Sun Sensor, CESS, Accelerometer, Laser Range Finder, LIDAR, Camera Sensor, Radar Altimeter
- **Utilities:** Cross Product, Unit Vector, Error Quaternion, Quaternion Multiplication, Euler Angles to Quaternion, Quaternion to Euler Angles, Quaternion Conjugation, Angle Between Vectors, Rotation Matrix from Quaternion, Quaternion Frame Transformation, Quaternion from Rotation Matrix, Positive Scalar Quaternion, Quaternion from Vector and Angle, Negative Scalar Quaternion, Smooth Quaternion, Cartesian to Spherical, Vectornorm, Quaternion Conversion scalar first to last, Quaternion Conversion scalar last to first, Euler 313 Rotation Matrix, Euler 321 Rotation Matrix, Frame Rotation x-axis, Frame Rotation y-axis, Frame Rotation z-axis, PCPF to LLA, LLA to PCPF, PCI to PCPF, PCPF to PCI

The toolbox includes a total number of 100 simulation blocks.

All blocks of the toolbox have a Mask in Simulink which contains all the changeable parameters, so that the user does not have to consider the contents of the block in detail. The Mask also contains a detailed description of the inputs and outputs of the block, with for example dimensions and coordinate frame convention.

All blocks have a help file explaining the contents of the block and presenting the used equations and assumptions in detail. The help files together build up a searchable help database, which makes it easy for the user to get started with using the toolbox.

For all new blocks added to the toolbox, the associated block features, such as Mask, Load function, and Help documentation, were created in MATLAB/Simulink. The modelling of several new components added to the GAST toolbox is described in detail in Chapters 4 and 5.

2.2 Building Simulations

A number of example simulations were developed for the toolbox that allow the user to quickly run simulations for a variety of common scenarios and spacecraft configurations. These example simulations will be presented in more detail in Chapter 5.

A new simulation is created by dragging and dropping blocks from GAST to a new model file in a similar manner as building a standard Simulink model.

Before running a simulation, the initialisation method can be chosen by clicking the

custom menu “ESA GNC/AOCS SIMULATIONS” in Simulink. The user is then able to choose Database, Numeric, or Manual Initialisation as shown in Fig. 2.3.

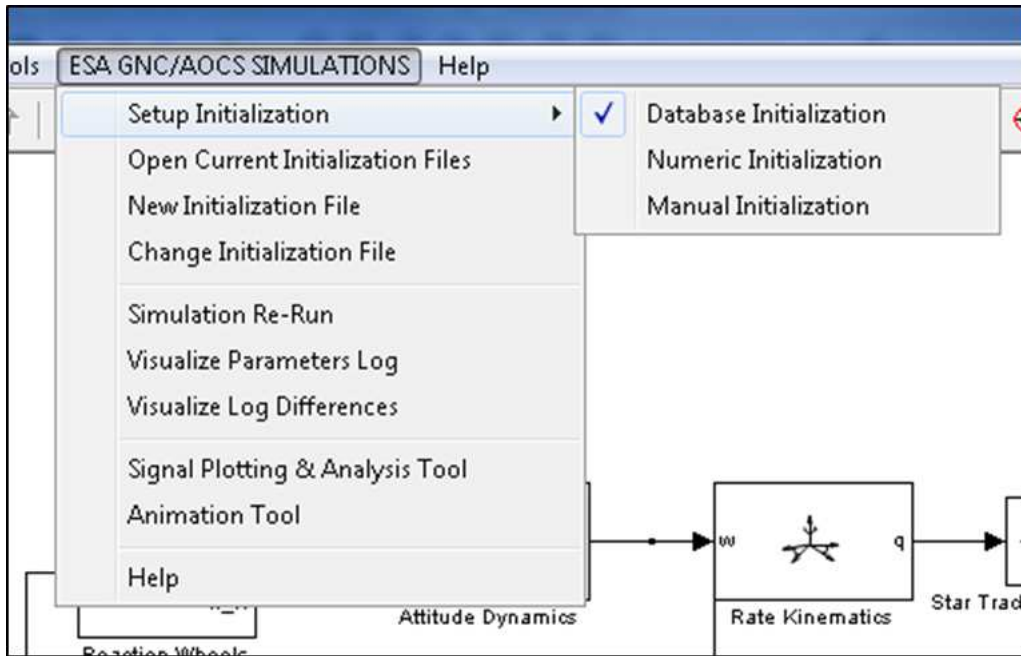


Figure 2.3: Initialisation choices of GAST simulations.

The Manual Initialisation implies that the parameters are initialised by entering values directly in the block Mask. The values can be numerical or variables initialised into the workspace before the simulation is run.

The Numeric Initialisation automatically generates an initialisation file for a model with GAST blocks. This file initialises each of the block’s Mask parameters using the variable names that appears in the block’s Mask at the time of saving the model. The desired values of the parameters have to be specified in the initialisation file.

The Database Initialisation also generates an automatic initialisation file for the GAST model simulation. This initialisation file allows the user to load a set of values for the block parameters from the Load function database, containing sensors and actuators from the TEC-ECN Hardware Database. The file also allows for the user to overload any of the loaded values from the database, and assign a custom value.

2.3 Using GAST

The goal of the GAST toolbox is to facilitate the development of more functional and less complex simulation models. This fulfils the requirements of the CDF and facilitates the set up of a complete simulation. With all helping features of the blocks

and the easy initialisation methods, the GAST toolbox provides the user with an easy way to simulate simple GNC/AOCS scenarios for conceptual studies.

Chapter 3

Definitions and Notations

In this chapter, the definitions and notations used in the GAST toolbox are described.

3.1 Rotations

The orientation of a satellite is always described as a rotation with respect to a certain reference frame. The relation between the two coordinate systems can be described by a 3×3 rotation matrix. The rotation matrix \mathbf{C} from frame A to frame B is denoted \mathbf{C}_{BA} and a vector described in frame A is written in frame B as

$$\mathbf{v}_B = \mathbf{C}_{BA} \mathbf{v}_A \quad (3.1)$$

To describe the attitude of the satellite with respect to a reference frame it is common to use Euler angles or unit quaternions.

The Euler angles, ϕ , θ , ψ , represent the rotations about the x-, y-, and z-axis of the satellite body respectively. The angles are also denoted roll, pitch, and yaw. The rotation matrices are given as

$$\mathbf{C}_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \quad (3.2)$$

$$\mathbf{C}_\theta = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (3.3)$$

$$\mathbf{C}_\psi = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

The Euler rotations are given as a sequence of three consecutive rotations around the roll, pitch, or yaw axes, where no two successive rotation are about the same axis. One example is the Euler 3-1-3 sequence, where the rotation is first around the yaw axis, thereafter the roll axis, and lastly around the yaw axis again. The rotation matrix is written as

$$\mathbf{C}_{Euler313} = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

The Euler 3-2-1 rotation sequence is commonly used in attitude description and is written as

$$\mathbf{C}_{Euler321} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

Euler angles present a relatively intuitive description of the attitude, however, this representation also introduces some undesired singularities. Therefore, the indirect method of measuring attitude using unit quaternions is often used. According to Euler's eigenaxis rotation theorem, an arbitrary rotation of a rigid-body can be determined by a rotation about a fixed eigenaxis \mathbf{e} with an angle ϕ . Based on this theorem, the attitude quaternion is defined as

$$\bar{q} = \begin{bmatrix} e_1 \sin(\phi/2) \\ e_2 \sin(\phi/2) \\ e_3 \sin(\phi/2) \\ \cos(\phi/2) \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ q_0 \end{bmatrix} \quad (3.7)$$

where $\mathbf{q} = [q_1, q_2, q_3]$ is the vector part and q_0 is the scalar part of the quaternion. The GAST toolbox uses this convention, where the scalar term is last in the quaternion vector, in all simulation blocks. The notation \bar{q}_{BA} is used to describe a rotation from frame A to frame B so that a vector described in frame A is written in frame B as

$$\mathbf{v}_B = \bar{q}_{BA}^* \begin{bmatrix} \mathbf{v}_A \\ 0 \end{bmatrix} \bar{q}_{BA} \quad (3.8)$$

where \bar{q}^* is the conjugate quaternion defined as $\bar{q}^* = [-\mathbf{q} \ q_0]^T$.

3.2 Coordinate Frames

In this section, the coordinate reference frames used throughout the thesis are defined.

The Inertial frame (I) is fixed in space and is defined with the unit vectors $(\hat{\mathbf{i}}_1, \hat{\mathbf{i}}_2, \hat{\mathbf{i}}_3)$, where $\hat{\mathbf{i}}_1$ points to the Vernal Equinox, $\hat{\mathbf{i}}_2$ is the normal of the orbital plane of the Earth around the sun and $\hat{\mathbf{i}}_3$ is completing the set of right-handed, orthogonal unit vectors.

The Earth Centered Inertial frame (ECI) is fixed in space with the origin at the center of the planet. The coordinate system is tilted around $\hat{\mathbf{i}}_1$ to coincide with the rotation axis of the Earth. The rotation matrix C_{EI} from frame I to the ECI frame is

$$C_{EI} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(-23.5^\circ) & \sin(-23.5^\circ) \\ 0 & -\sin(-23.5^\circ) & \cos(-23.5^\circ) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.9171 & -0.3987 \\ 0 & 0.3987 & 0.9171 \end{bmatrix} \quad (3.9)$$

so that a vector in the Inertial frame can be described in the ECI frame as

$$\mathbf{v}_E = C_{EI}\mathbf{v}_I \quad (3.10)$$

For other planets than the Earth, the Planet Centered Inertial Frame (PCI) is defined in a similar way, with the origin at the center and the third axis along the rotational velocity vector of the planet.

The Planet Centered Planet Fixed (PCPF) reference frame also has the origin located at the center of the planet. However, the x- and y-axes rotate with the planet relative to the PCI frame. The rotation rate equals the planet rotational rate around the z-axis.

The Local Orbital frame (LO) is defined with an Euler 3-1-3 rotation of the ECI frame with angles Ω - i - u , where Ω is the right ascension of the ascending node, i is the orbit inclination, and u is the argument of latitude, which is time dependant $u = \omega_0 t$, if the satellite is in a near circular orbit. The transformation from ECI to LO is described with the rotation matrix

$$C_{LE} = \begin{bmatrix} c(u)c(\Omega) - s(u)c(i)s(\Omega) & c(u)s(\Omega) + s(u)c(i)c(\Omega) & s(u)s(i) \\ -s(u)c(\Omega) - c(u)c(i)s(\Omega) & -s(u)s(\Omega) + c(u)c(i)c(\Omega) & c(u)s(i) \\ s(i)s(\Omega) & -s(i)c(\Omega) & c(i) \end{bmatrix} \quad (3.11)$$

where $s(\cdot)$ and $c(\cdot)$ are abbreviations of $\sin(\cdot)$ and $\cos(\cdot)$ respectively.

The Local Vertical/Local Horizontal reference frame (LVLH) has its origin at the center of mass of the orbiting satellite. It is defined with the unit vectors $(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}})$ where $\hat{\mathbf{z}}$ is pointing towards the center of the planet, $\hat{\mathbf{y}}$ points in the direction of the negative orbital plane normal, and $\hat{\mathbf{x}}$ completes the base (pointing along the velocity vector if the orbit is circular). The rotation matrix from LO to LVLH is

$$\mathbf{C}_{LVLH,LO} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix} \quad (3.12)$$

The Body Fixed frame (B) also has its origin at the center of mass of the satellite with unit vectors $(\hat{\mathbf{x}}_b, \hat{\mathbf{y}}_b, \hat{\mathbf{z}}_b)$. It is fixed in the satellite body and rotating with it. The attitude of a satellite is often defined as the rotation of the body fixed frame with respect to the LVLH frame.

Chapter 4

Modelling Sensors and Actuators

In this chapter the models of several sensors and actuators modelled for the GAST toolbox are described.

4.1 Sensors

Sensors are necessary equipment on satellites, as they track the absolute or relative position and attitude of the satellite and provide both the observers on the ground and the onboard computer with valuable information. However, no sensors are perfect in a sense that instead of measuring the true spacecraft states, they are affected by measurement errors such as bias, noise, random walk, drift, scale factor, and time delay. In sensor simulations, these errors are modelled to simulate the real sensor measurements in order to evaluate the performance of the GNC functions.

Bias is a measurement error defined as the output signal of a sensor when all inputs are zero. The bias is added to the output of the sensor.

Noise is a random deviation of the measurement signal varying with time. The variance of the noise describes how far the deviations lie from the mean value and is calculated as the expected value of the squared deviation from the mean, $\mu = E[X]$, as

$$Var(X) = E[(X - \mu)^2] \quad (4.1)$$

where $E[X]$ denotes the expected value of the measurement X . The standard deviation, σ , is the square root of the variance. The noise in the sensor models is simulated by a random variable with mean μ and standard deviation, σ . If the mean is zero, the disturbance is called white noise.

Random walk in statistics describes a situation where the output of a system is driven by a succession of random steps. The random walk is modelled as the integral

of a random variable with zero mean.

Angle random walk (ARW) is a common specification of rate sensors, which describes the average deviation that occurs when integrating the output signal to get the angle changes [15]. The noise in the rate measurements contribute to an angle error that will increase with time. The ARW can be calculated as in [15] as

$$ARW = \frac{\sigma}{\sqrt{BW}} \quad (4.2)$$

where σ is the standard deviation of the white noise on the rate measurements and BW is the effective bandwidth of the sensor in Hz. This type of random walk error occurs in all integrations of sensor measurements where noise is present.

Drift is the change in a measured value when it is measured under the same conditions after a period of time. The drift is modelled as a ramp and added to the output.

The **resolution** of the measurements states how small deviations the sensor can detect. The resolution is modelled as a discretisation of the true measurement.

All sensors have a measurement limit, which is denoted **range**. The sensor cannot measure values that lie outside of the limits of the range. The range limit is modelled by a saturation of the input to the specified maximum and minimum output values.

The **scale factor** is a constant which scales the true signal to the measured signal. It is defined in [18] as the ratio of a change in output to a change in the input intended to be measured. The scale factor is generally evaluated as the slope of the straight line of input-output data.

A **time delay** is added to the model, as sensors do not change output state immediately when an input parameter is changed. Rather, it will change to the new state over a period of time, so that the output of the sensor is shifted in comparison to the real states.

The following sections show the modelling of some sensors for the GAST toolbox. The models are made as general as possible to be able to use a large range of different components from different manufacturers in the simulations.

4.1.1 Rate Sensor

A rate sensor is an inertial sensor that measures angular rotation with respect to inertial space about its input axes [18]. Different types of rate sensors include mechanical devices, optical devices, and Micro Electro Mechanical Systems (MEMS). According to the error descriptions above, the output $y(t)$ of the rate sensor is modelled as

$$y(t) = Sw(t - \tau) + n(t - \tau) + d(t - \tau) + r(t - \tau) + b \quad (4.3)$$

where $w(t)$ is the true rate signal, S is the scale factor, $n(t)$ is the noise term, $d(t)$ is the drift consisting of the time integral of a constant, $r(t)$ is the rate random walk consisting of an integration of a white noise, b is the constant bias term, and τ is the time delay constant. The output is also quantised according to the sensor resolution and a range limit is applied. The Simulink model of the sensor is shown in Fig. 4.1.

Rate Sensor Model with Bias, Noise, Random Walk, Drift, Resolution, Range and Sample Time

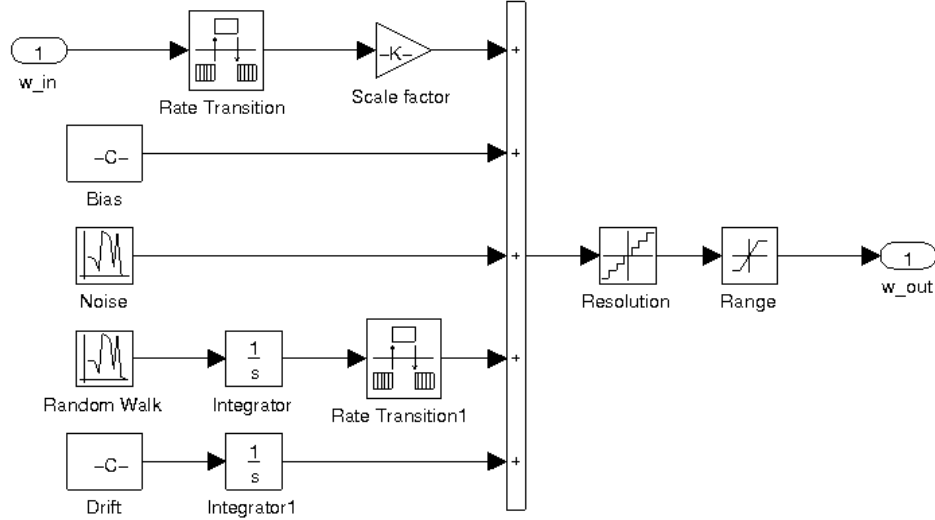


Figure 4.1: Simulink model of a rate sensor.

To evaluate the model, the rate sensor is loaded with the values of the SiREUS silicon rate sensor specified in Table 4.1 [14].

Sensor Characteristics	Value
Sensor Type	3-axis MEMS rate sensor
Rate Measurement Bandwidth	10 Hz (max)
Angular Rate Bias	10–20 °/h
Rate Bias Drift	5–10 °/h over 24h
Angular Random Walk	0.1–0.2 °/√h
Noise Equivalent Rate (defined as Flicker Rate)	< 1 °/h
Scale Factor Linearity	< 2000 ppm

Table 4.1: Specifications of the SiREUS rate sensor.

The standard deviation of the white noise is calculated through the ARW according to (4.1). The Noise Equivalent Rate (flicker rate) is a so called “pink noise”. It is generally difficult to model pink noise and in this case it is sufficient to model the

error as a random walk [20]. The random walk is an integration of white noise with standard deviation

$$\sigma = \frac{NER}{\sqrt{t}} \sqrt{BW} = NER \cdot BW \quad (4.4)$$

The SiREUS rate sensor is simulated with zero true rate signal for 3 different random walk sequences. The result is shown in Fig. 4.2.

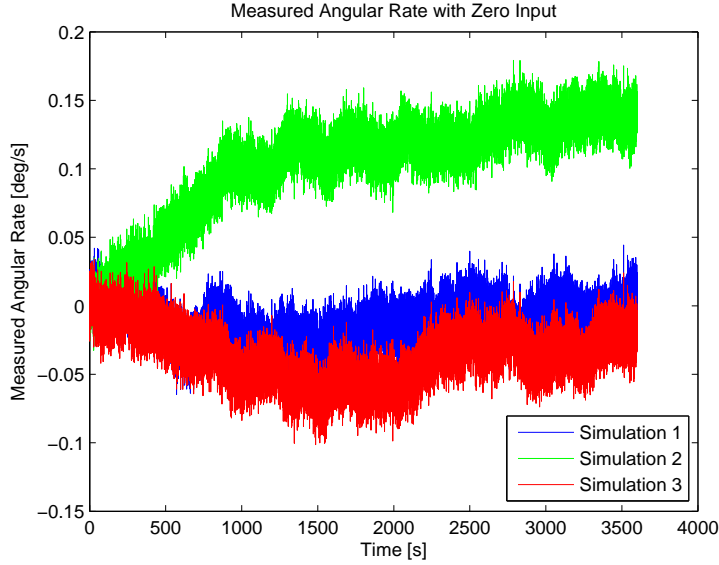


Figure 4.2: Simulation result of SiREUS noise model.

This result corresponds to similar simulations made in for example [13]. Models like these can for example be used in the design of Kalman filters for navigation and control.

4.1.2 Laser Range Finder

A Laser Range Finder (LRF) is a sensor that measures the distance to an object through time measurements of a reflected laser beam sent towards the target. With the use of at least three retro-reflectors attached to the outside of the target, the relative attitude between the chaser and target can be measured by geometrical relations. In this model, the sensor measures the relative position of a target, the Line of Sight (LOS) in the y- and z-direction, and the relative attitude of the target with respect to the chaser. It is assumed that the sensor is placed on the +x face of the satellite and is pointing in the positive x-direction.

The model of the LRF calculates the true position of a target relative to a chaser spacecraft, given the respective true positions in a reference frame. Bias and noise

errors which are dependent on the range are added to the output. The errors of a LRF are usually specified for long, medium, and short range. To calculate the bias error as a function of range, the bias is assumed to vary linearly between the values for long and medium range, between medium and short range, and stay constant from the short range to zero. A general graph of the bias as a function of range is shown in Fig. 4.3. The values are not specified in the graph, since it is just the shape of the curve that matters.

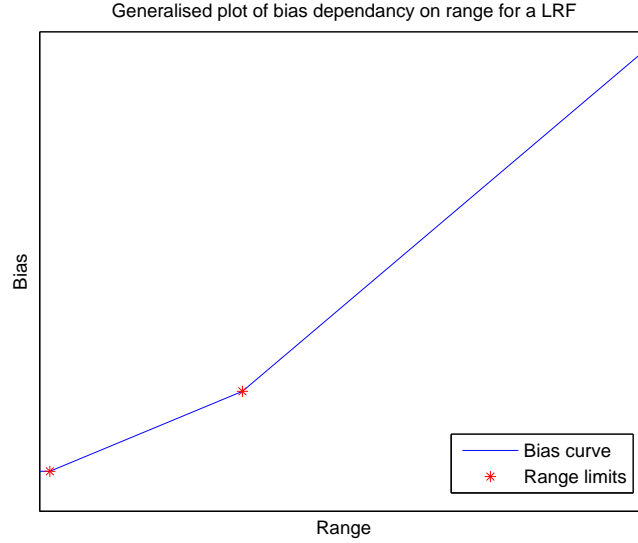


Figure 4.3: General model of the range dependent bias error.

The lines between the range limit points are modelled as linear function of the form $y = kx + m$. The error for a range greater than the long range limit is assumed to have a steeper inclination than the linear function for the shorter ranges. The k value of this curve is therefore modelled to have the same inclination proportionality to the line between long and medium range as this line to the line between medium and short range. The bias error is written in equations as

$$B = \begin{cases} B_s & \text{if } r < r_s \\ kr + m & \text{if } r > r_s \end{cases}$$

where

$$k = \begin{cases} k_1 = (B_m - B_s)/(r_m - r_s) & \text{if } r_s < r < r_m \\ k_2 = (B_l - B_m)/(r_l - r_m) & \text{if } r_m < r < r_l \\ k_3 = k_2^2/k_1 & \text{if } r > r_l \end{cases} \quad (4.5)$$

$$m = \begin{cases} B_m - k_1 r_m & \text{if } r_s < r < r_m \\ B_l - k_2 r_l & \text{if } r_m < r < r_l \\ B_l - k_3 r_l & \text{if } r > r_l \end{cases}$$

where r_s, r_m, r_l are the short, medium, and long range limits, and B_s, B_m, B_l are

the biases specified for the respective range. This maximum bias is then multiplied with a uniform random number varying between 1 and -1 to model the random bias error.

The noise error is modelled in the same way, only exchanging the bias for the specified noise values.

As an example, the Videometer Assembly made by EADS SODERN is considered [3]. This is the main sensor supporting the correct docking of the Automated Transfer Vehicle (ATV) to the International Space Station (ISS). The operating principle is the one of a LRF, e.g. illuminating the target with a laser beam and detecting the reflected light. The specifications of the Videometer are shown in Table 4.2.

Measurement	Error Characteristics	Value
Range	Bias	
	300 m	± 5.8 m
	100 m	± 0.9 m
	1.25 m	± 0.01 m
	Noise (3σ)	
	300 m	± 4 m
LOS	100 m	± 0.5 m
	1.25 m	± 0.0006 m
	Bias	
	from 10 to 300 m	$\pm 0.2^\circ$
Roll	below 10 m	$<\pm 0.5^\circ$
	Noise (3σ)	$<\pm 0.004^\circ$
	Bias	$\pm 0.8^\circ$
Pitch/Yaw	Noise (3σ)	$\pm 0.03^\circ$
	Bias	$\pm 0.9^\circ$
	Noise (3σ)	$\pm 1.2^\circ$

Table 4.2: Specifications of the EADS SODERN Videometer Assembly LRF.

The range measurements of the EADS Videometer as the chaser approaches a target from 500 m distance to docking at a constant speed is shown in Fig. 4.4.

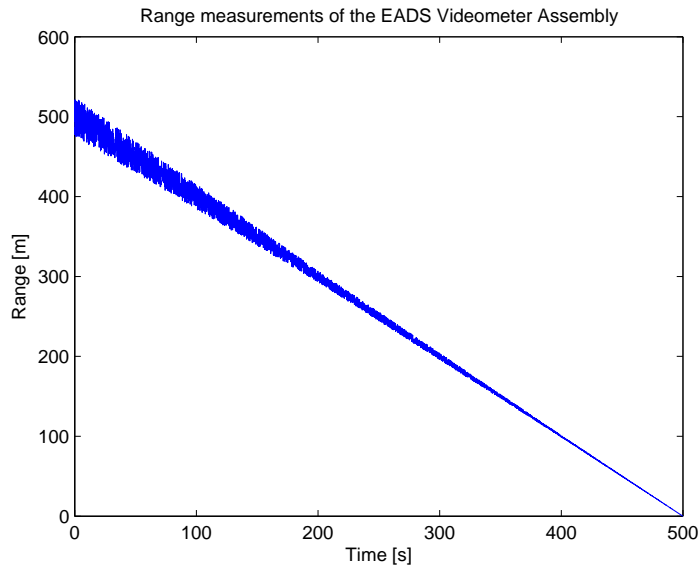


Figure 4.4: Simulation of range measurements of the EADS Videometer Assembly approaching a target at constant speed.

It is seen that the range measurements are more accurate as the chaser gets closer to the target.

The LOS and attitude errors are assumed to not vary with the range. The LOS and attitude measurements of the EADS Videometer using this model are shown in Fig. 4.5 and 4.6. The initial relative attitude is assumed to be zero.

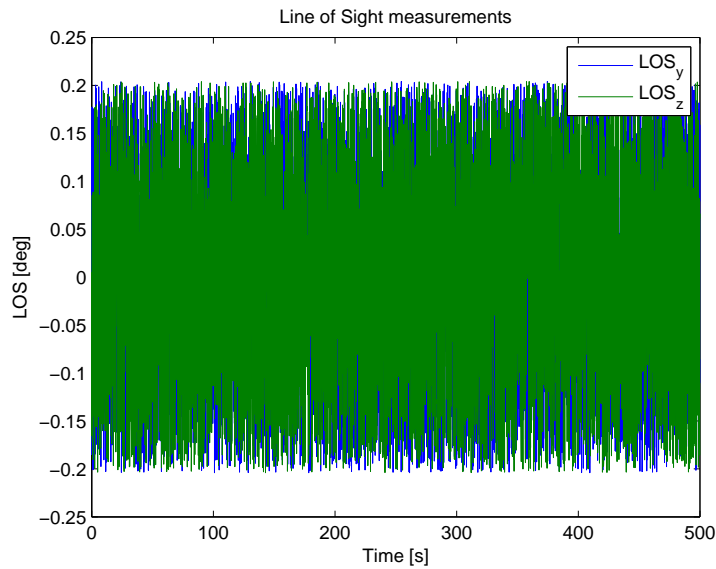


Figure 4.5: Simulation of LOS measurements of the EADS Videometer Assembly approaching a target at constant speed.

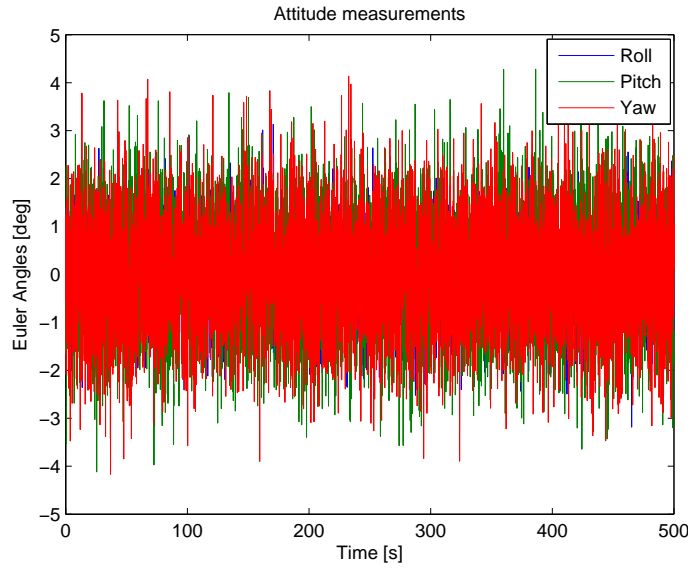


Figure 4.6: Simulation of attitude measurements of the EADS Videometer Assembly approaching a target at constant speed.

The LRF model can be used for simulations of Rendezvous and Docking missions.

4.1.3 Accelerometer

An accelerometer is a sensor which measures acceleration forces. The measured forces can be static, like the gravitational force, or dynamic forces caused by moving or vibrating the accelerometer.

In this simple model of an accelerometer, the errors are modelled by a bias and a white noise. The output of the accelerometer is written as

$$y(t) = a(t) + n(t) + b \quad (4.6)$$

where $a(t)$ is the true acceleration, $n(t)$ is the noise term, and b is the constant bias term. The output is also quantised according to the resolution of the sensor and a range model is added to limit the output. The Simulink model of the accelerometer is shown in Fig. 4.7.

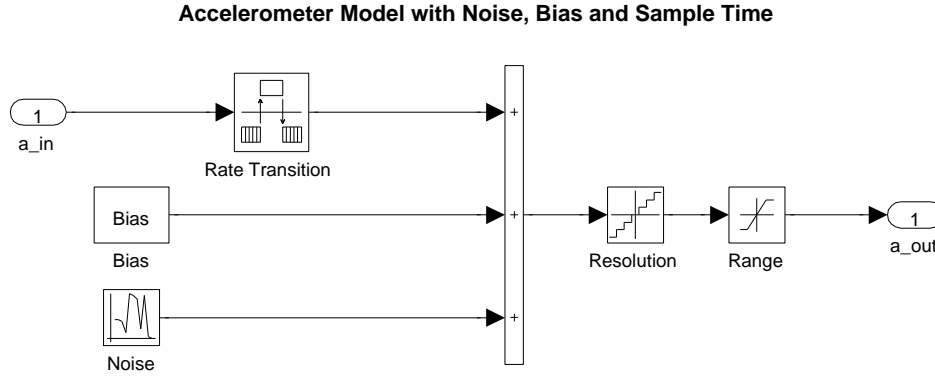


Figure 4.7: Simulink model of an accelerometer.

A comparison between the simulation of two different accelerometers using this model is made. The measured accelerations are integrated to show the calculated values of velocity and position when using only accelerometer measurements. The accelerometers used are the Honeywell Q-Flex QA-2000 [10] and the Litton LN200S [12]. The specifications of the sensors are listed in Table 4.3.

	Honeywell QA-2000 (Sensor 1)	Litton LN200S (Sensor 2)
Range	± 60 g	± 40 g
Bias	< 4 mg	0.3 mg
Noise (1σ)	< 1500 μg	35 μg
Resolution	< 1 μg	-

Table 4.3: Specifications of two accelerometers.

The resolution of the LN200S sensor is not specified, but the same resolution as for the QA-2000 sensor will be assumed. The acceleration is assumed to go from 60 g to 0 g in the x-direction. The simulation results together with the supposed true states are shown in Figs. 4.8, 4.9, and 4.10.

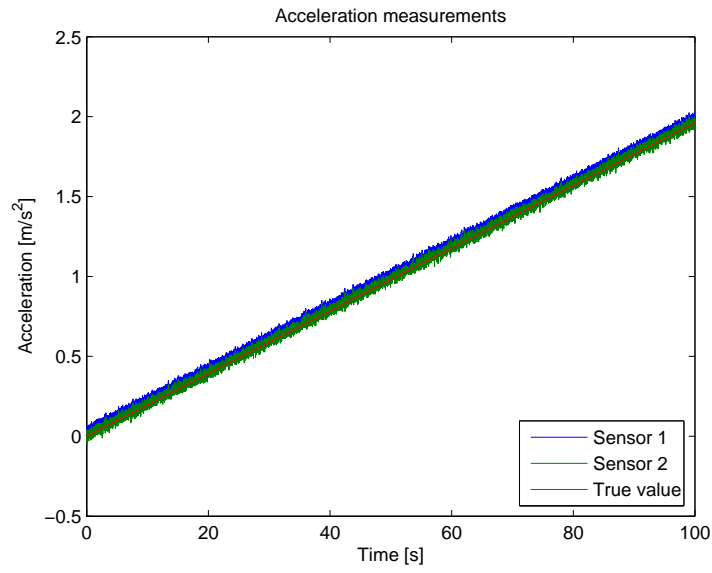


Figure 4.8: Acceleration measurements.

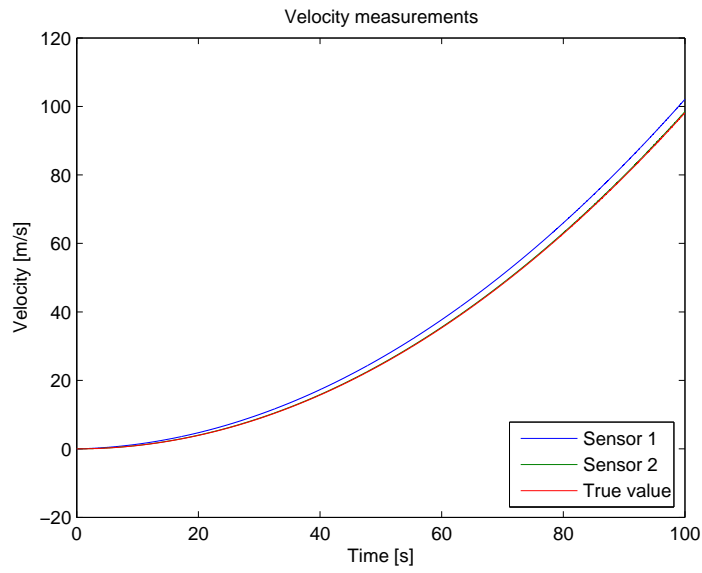


Figure 4.9: Velocity measurements.

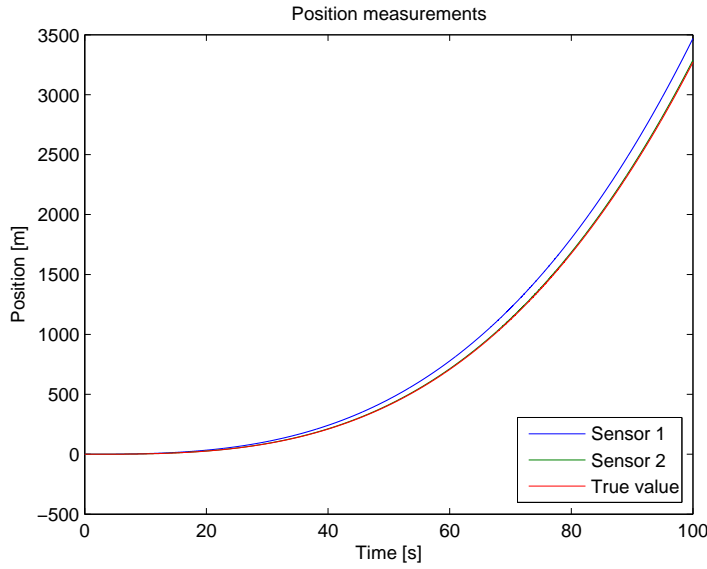


Figure 4.10: Position measurements.

The results show that Sensor 2 measures the true states with higher accuracy than Sensor 1. The velocity measurement error after 100 s is 3.96 m/s for Sensor 1 and 0.35 m/s for Sensor 2. The position measurement error after 100 s is 198.11 m for Sensor 1 and 17.25 m for Sensor 2.

This accelerometer model is very simple, but sufficient to account for significant errors in acceleration measurements and their propagation to velocity and position calculations.

4.2 Actuators

There are several options to control a satellite in a way which ensures stability. Depending on the mission requirements different techniques of passive or active control can be used. Active control systems generally provide greater control flexibility and larger control torques can be implemented. This is however a trade-off to greater subsystem mass and complexity.

Actuators are also subject to errors that have to be modelled in order to make realistic simulations. The commanded torque is modified according to range limits, deadzone, noise, time delay, command quantisation, and more, to give the actual actuator output.

The **range limits** are specified by the maximum and minimum force or torque that an actuator can produce.

A **deadzone** is a nonlinearity which is likely to exist in many mechanical actuators.

A deadzone is a region where no action occurs and a commanded force or torque within this region is output as zero. If the deadzone is small it has a negligible effect, however, if it is large it can lead to a degraded transient response [19].

A white **noise** is added to the command in the same way as for the sensors to model for example bearings noise.

As the output of the actuator is not changed instantly with a change in the input command, a **time delay** is added to the model.

The command can also be **quantised** in order to model actuators that only have a number of discrete output possibilities, such as a thruster that can only be on or off.

In the following sections, some actuators modelled for the GAST toolbox are described.

4.2.1 Reaction Wheels

A reaction wheel provides control torques through changes in the rotational momentum of the wheel. It operates with the principle of conservation of momentum. A symmetrical rotating body, which may have an initial momentum, produces angular torques when accelerated about its axis of rotation. Since the overall momentum of the system is not changed, the momentum change of the reaction wheel is negatively transferred to the satellite. A 3-axis control system producing a required angular torque can be achieved through a minimum of 3 reaction wheels mounted orthogonally in the satellite.

In this model of a 3-axis reaction wheel system, a bearings noise is added to account for the noise in the rotation mechanism of the reaction wheels. The noise is a white noise with a variation σ^2 specified for a certain reaction wheel. The commanded torque is quantised to a given resolution and a transport delay is added. Lastly, the torque is saturated to account for the torque range limits. The realised torques are integrated to give the current wheel momentum of each reaction wheel. The Simulink model of the reaction wheel is shown in Fig. 4.11.

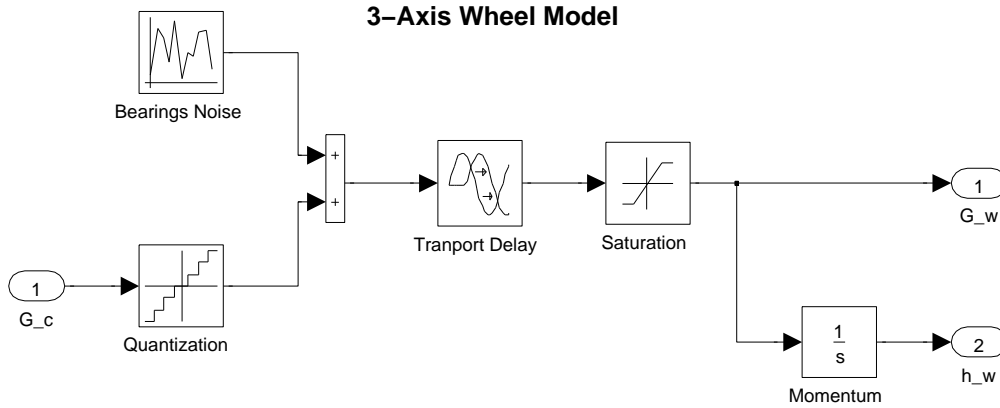


Figure 4.11: Simulink model of a reaction wheel control system.

Fig. 4.12 shows an example, using this model, of the commanded torque compared to the realised torque around the x-axis of the satellite. The commanded torque is set to be periodical with $T_c = 10 \sin(t)$ Nm. The command quantisation is 0.1 Nm, the variance of the bearings noise is $\sigma^2 = 0.1$ Nm, the transport delay is 1 s and the range limit is ± 5 Nm.

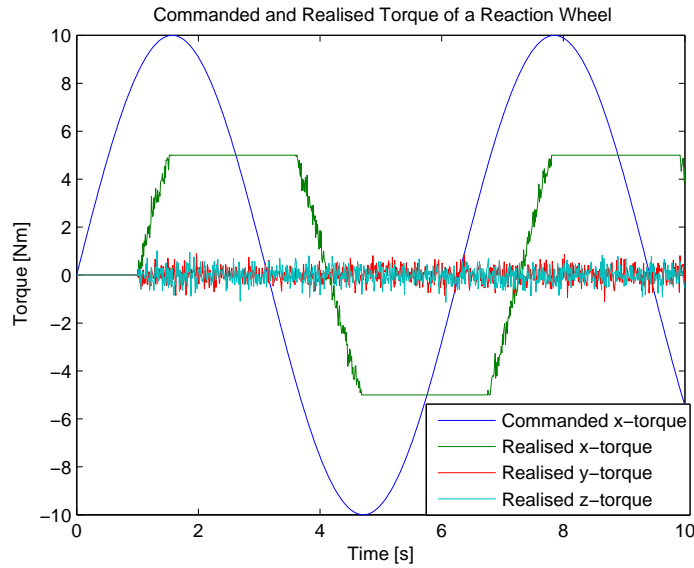


Figure 4.12: Commanded torque compared to the output torque of a reaction wheel system.

It is seen that the realised torques around the y- and z-axes are not completely zero due to the bearings noise in the model. The reaction wheel model has to be included in a complete simulation to determine if the reaction wheel torque is able to sufficiently realise the attitude requirements for a mission.

4.2.2 Control Moment Gyroscope

A Control Moment Gyroscope (CMG) can also be used for spacecraft attitude control. A CMG produces a torque by tilting the angular momentum of a flywheel around a gimbal axis. As the rotating flywheel tilts, the changing angular momentum causes a gyroscopic torque that rotates the spacecraft. This section describes the models of a single gimbaled CMG and a 3-axis control CMG system.

1-axis Single Gimbaled CMG

In this model, the input is the commanded torque and the output is the realised torque of a single gimbaled CMG.

The CMG has a reference frame F_G with unit axes $(\mathbf{e}_s, \mathbf{e}_t, \mathbf{e}_g)$, which are denoted spin, transverse, and gimbal axis respectively. The unit axes are described in the spacecraft body frame, F_B . The flywheel rotates with a constant angular momentum, h_n , around the spin axis, which is orthogonal to the gimbal axis. The gimbal axis is fixed in F_B . For simplicity, the origin of F_G is defined to be located at the center of mass of the spacecraft. However, F_G does not have to coincide with F_B . With the gimbal angle denoted as δ , the unit axes can be written in F_B as

$$\mathbf{e}_s(t) = \cos(\delta)\mathbf{e}_{s0} + \sin(\delta)\mathbf{e}_{t0} \quad (4.7)$$

$$\mathbf{e}_t(t) = -\sin(\delta)\mathbf{e}_{s0} + \cos(\delta)\mathbf{e}_{t0} \quad (4.8)$$

$$\mathbf{e}_g(t) = \mathbf{e}_{g0} \quad (4.9)$$

The angular velocity of F_G with respect to F_B is $\dot{\delta}\mathbf{e}_g$. The output torque of the CMG in the body frame is the rate of change of angular momentum in the body frame and can be written as

$$\mathbf{T} = \dot{\mathbf{h}}_{CMG} = \dot{\delta} \times \mathbf{h} = |\dot{\delta}||\mathbf{h}|\mathbf{e}_t(t) = h_n\dot{\delta}\mathbf{e}_t(t) \quad (4.10)$$

From this, the required gimbal rate is then given as

$$\dot{\delta} = \frac{\mathbf{T}_c \cdot \mathbf{e}_t}{h_n} \quad (4.11)$$

The required gimbal rate is then limited to the maximum rate specified by the user. The gimbal rate is integrated over time to give the gimbal angle, which is also limited to a specified range. The time derivative of the realised gimbal angle gives the realised gimbal rate. The realised output torque is then given as

$$\mathbf{T}_{out} = h_n\dot{\delta}(-\sin(\delta)\mathbf{e}_{s0} + \cos(\delta)\mathbf{e}_{t0}) \quad (4.12)$$

4.2. ACTUATORS

Figure 4.13: 1-axis Single Gimbaled CMG model in Simulink.

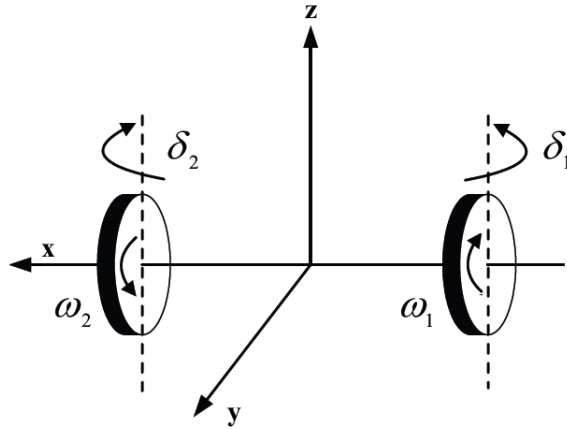


Figure 4.14: Two Single Gimbaled CMGs mounted on the same axis.

If $\delta_1 = -\delta_2$ and $\omega_1 = -\omega_2$, the output torque will be only in the y-direction, since the torques in the x-direction cancel each other out. If two such CMGs are placed on each body axis of a satellite, the satellite can be 3-axis controlled with a quite simple dynamic model. Another advantage of this configuration is that the wheel speeds can be increased or decreased without resulting in a torque as long as $\omega_1 = -\omega_2$.

The torque resulting from one wheel is written as

$$\mathbf{T} = \dot{\boldsymbol{\delta}} \times \mathbf{h} \quad (4.14)$$

The total torque from both wheels is then

$$\mathbf{T}_1 + \mathbf{T}_2 = 2\dot{\delta}h_n \cos(\delta)\hat{\mathbf{y}} \quad (4.15)$$

as the torques in the x-direction cancel out.

With two wheels mounted on each of the spacecraft body axes the total output torque becomes

$$\mathbf{T}_{out} = \begin{bmatrix} 2\dot{\delta}_z h_{nz} \cos(\delta_z) \\ 2\dot{\delta}_x h_{nx} \cos(\delta_x) \\ 2\dot{\delta}_y h_{ny} \cos(\delta_y) \end{bmatrix} \quad (4.16)$$

where δ_j and h_{nj} is the gimbal angle and the nominal angular momentum of one wheel mounted with the spin axis along the j -direction. The gimbal angular rates are

$$\dot{\delta} = \begin{bmatrix} T_y / (2h_{nx} \cos(\delta_x)) \\ T_z / (2h_{ny} \cos(\delta_y)) \\ T_x / (2h_{nz} \cos(\delta_z)) \end{bmatrix} \quad (4.17)$$

The gimbal rates are integrated to give the gimbal angles and both are limited to a user specified range. Thereafter, the output torque is calculated with the realised gimbal angles and rates as \mathbf{T}_{out} above.

The Simulink model of the 3-axis CMG system is seen in Fig. 4.15. The equations above are implemented to calculate the gimbal rate and gimbal angle for each axis. Saturation of the gimbal rate and gimbal angle are included to ensure that they do not exceed the maximum values of the CMG. The model contains the same error estimations as the previous model of a 1-axis CMG. The output is the realised torque, the realised angular momentum, and the gimbal angles of the CMGs. The torque is integrated to give the angular momentum, with the initial condition $\mathbf{h}_0 = 0$.

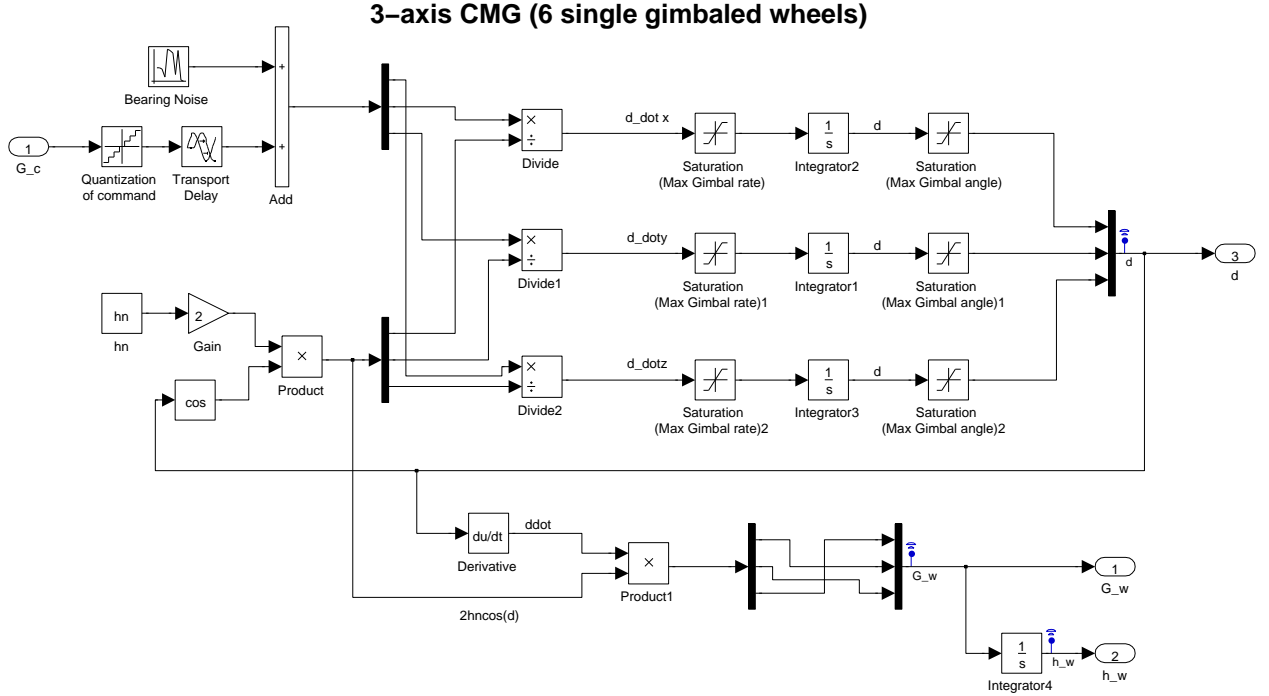


Figure 4.15: 3-axis Single Gimbaled CMG model in Simulink.

4.2.3 Thrusters

This section describes the model of a 3-axis thruster control system. Two thrusters mounted on each body axis, each one thrusting in the opposite direction of the other, builds up a 3-axis control system. The commanded thrust is first quantised to a number of discrete thrust levels. If the quantisation is equal to the maximum thrust level, the thrust model becomes an on/off model, where a thruster can only assume

the value of the maximum thrust or zero. The rate limiter models the rise and fall time of the thrust. If a step in thrust is commanded, the realised thrust is limited to rise with the rate F_{\max}/t_{rise} or fall with a rate $-F_{\max}/t_{\text{fall}}$. The transport delay models the response time of the thruster from command to execution. White noise is added to the model to account for thrust amplitude noise. Lastly, a deadzone is implemented in which the realised thrust is zero.

The Simulink model of a 3-axis thruster control system is shown in Fig. 4.16.

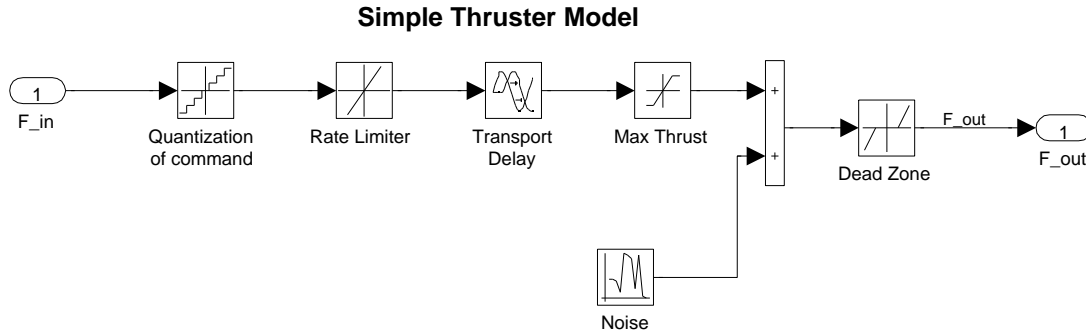


Figure 4.16: Simulink model of a thruster control system.

A simulation was made with the thruster requirements listed in a study made for the Lisa Pathfinder mission [6]. The specified requirements of the thrust control system are seen in Table 4.4.

Thruster Requirement Specification	Value
Thruster Type	FEED Thruster
Thrust Range	0.3–150 μN
Thrust Precision (Resolution)	<1 μN
Thrust Noise	<0.1 $\mu\text{N}/\sqrt{\text{Hz}}$
Response Time (Rise time for a thrust step of 30 μN)	<340 ms

Table 4.4: Thruster requirements of the Lisa Pathfinder thrusters.

Simulations using the GAST model was made for a thrust step of 1 μN and 10 μN . The result is shown in Figs. 4.17 and 4.18.

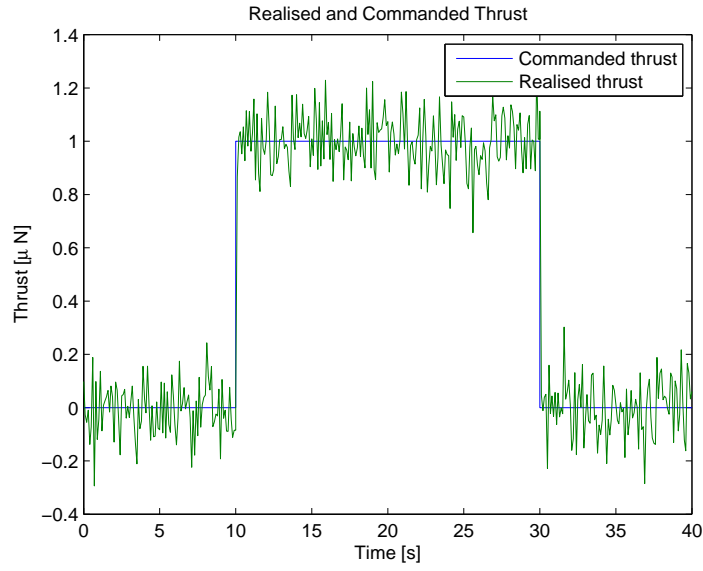


Figure 4.17: Thruster response to a step of $1 \mu\text{N}$.

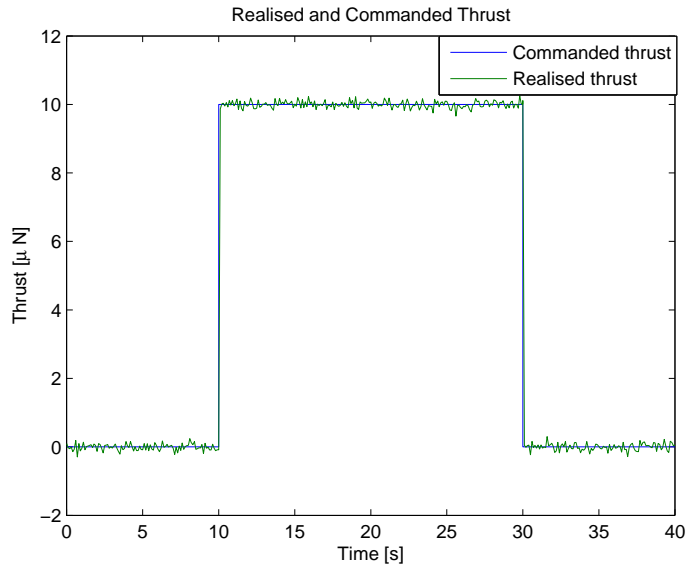


Figure 4.18: Thruster response to a step of $10 \mu\text{N}$.

The result is compared to the measurements made in [6], which are shown in Fig. 4.19.

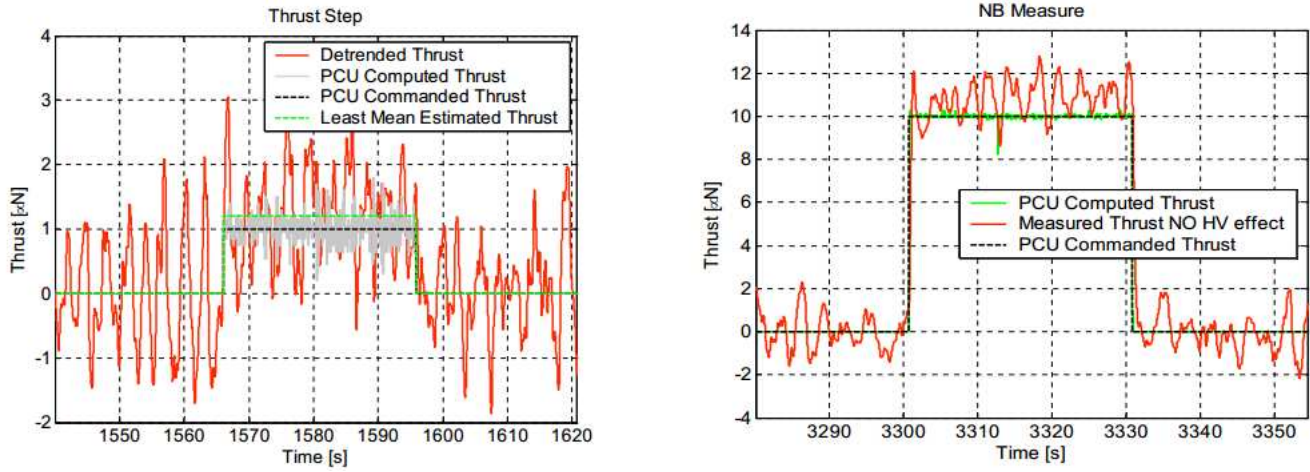


Figure 4.19: Simulation results from [6].

The GAST simulation results are similar to the PCU computed thrust from the measurements in Fig. 4.19. The noise is slightly larger in Fig. 4.19 than in Figs. 4.17 and 4.18 due to additional measurement noise, which is not present in the GAST model.

Chapter 5

Simulations using GAST

Several simulations were made to demonstrate the capabilities of the GAST toolbox. The simulations were added to the toolbox with the purpose to be used as templates and working examples for users who want to build quick simulations. The simulations of this chapter show, for example, how to set up a control system to follow a specified reference trajectory or attitude, how to compare simulations with different sensors and control laws, and how to validate analytical, linearised calculations with the nonlinear models of the toolbox.

5.1 De-spinning using Quaternion Control

A simulation was made using GAST to demonstrate the de-spinning of a satellite. Different reaction wheel parameters were compared in the simulation to determine the best actuator choice. The GAST simulation scheme is shown in Fig. 5.1.

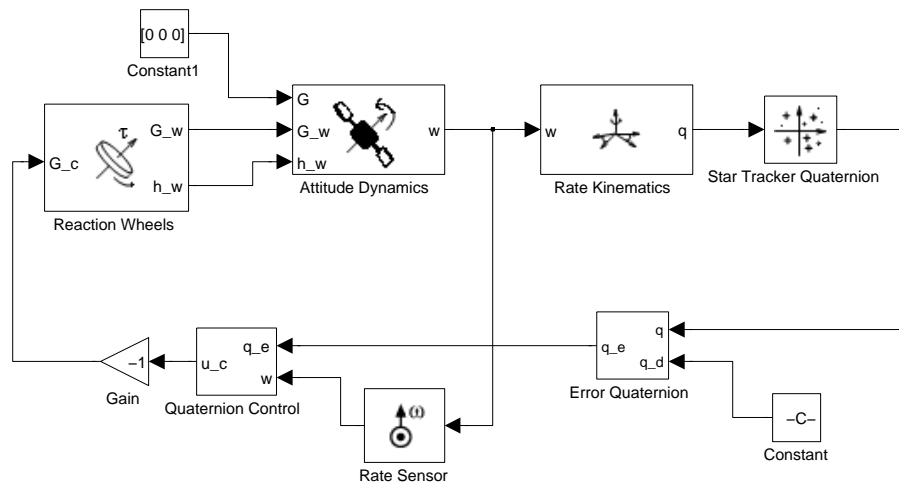


Figure 5.1: GAST Model of de-spinning using Quaternion Control.

The satellite has an inertia matrix with $I_x = 1500 \text{ kgm}^2$, and $I_y = I_z = 1000 \text{ kgm}^2$. The initial angular rates are $\omega_0 = [0.1, 0.1, 0.1]^\circ/\text{s}$ and the desired attitude is zero ($\bar{q} = [0, 0, 0, 1]$). The attitude is controlled with a reaction wheel using the model in section 4.3.1. The control law is a so called quaternion control where the feedback parameters are the angular rates and the vector part of the error quaternion. The control law is written as

$$\mathbf{u}_c = -\mathbf{K}_p \mathbf{q}_e - \mathbf{K}_d \boldsymbol{\omega} \quad (5.1)$$

where \mathbf{K}_p the 3×3 proportional quaternion gain matrix and \mathbf{K}_d the 3×3 proportional rate damping gain matrix. In this simulation \mathbf{K}_p and \mathbf{K}_d are chosen by experience to be \mathbf{I} and $10\mathbf{I}$ respectively, where \mathbf{I} is the identity matrix.

The sensors used to measure the quaternion and angular rates are a rate sensor (as modelled in Section 4.2.1) and a star tracker. The rate sensor is chosen as the Honeywell Miniature Inertial Measurement Unit sensor with specifications in [9] and the star tracker is chosen as the Active Pixel Sensor ASTRO APS with specifications in [11]. The reaction wheels parameters are chosen as in Table 5.1 for Reaction Wheel 1 and Reaction Wheel 2.

	Reaction Wheel 1	Reaction Wheel 2
Range	75 mNm	7.5 mNm
Resolution	0.02 mNm	0.02 mNm
Noise (1σ)	0.67 mNm	0.67 mNm
Time Delay	0.02 s	0.02 s

Table 5.1: Specifications of two reaction wheel configurations.

The results of the simulation with Reaction Wheel 1 are shown in Fig. 5.2.

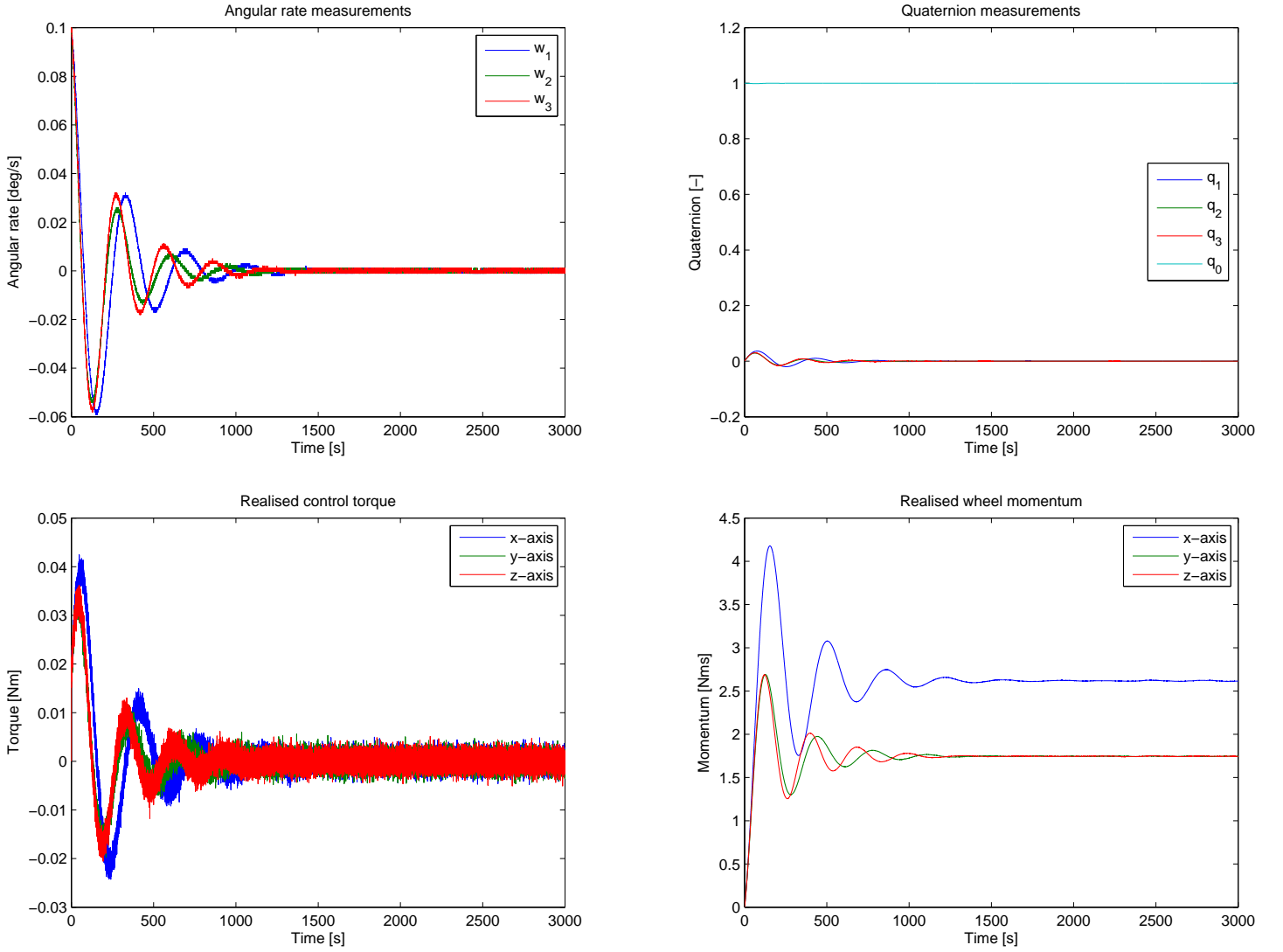


Figure 5.2: Results of GAST simulation of de-spinning of a satellite using Reaction Wheel 1.

The angular rates settle to zero after about 1500 s and the reaction wheel torque never reaches saturation. Thereafter, the same simulation is run the Reaction Wheel 2 configuration. The results are shown in Fig. 5.3

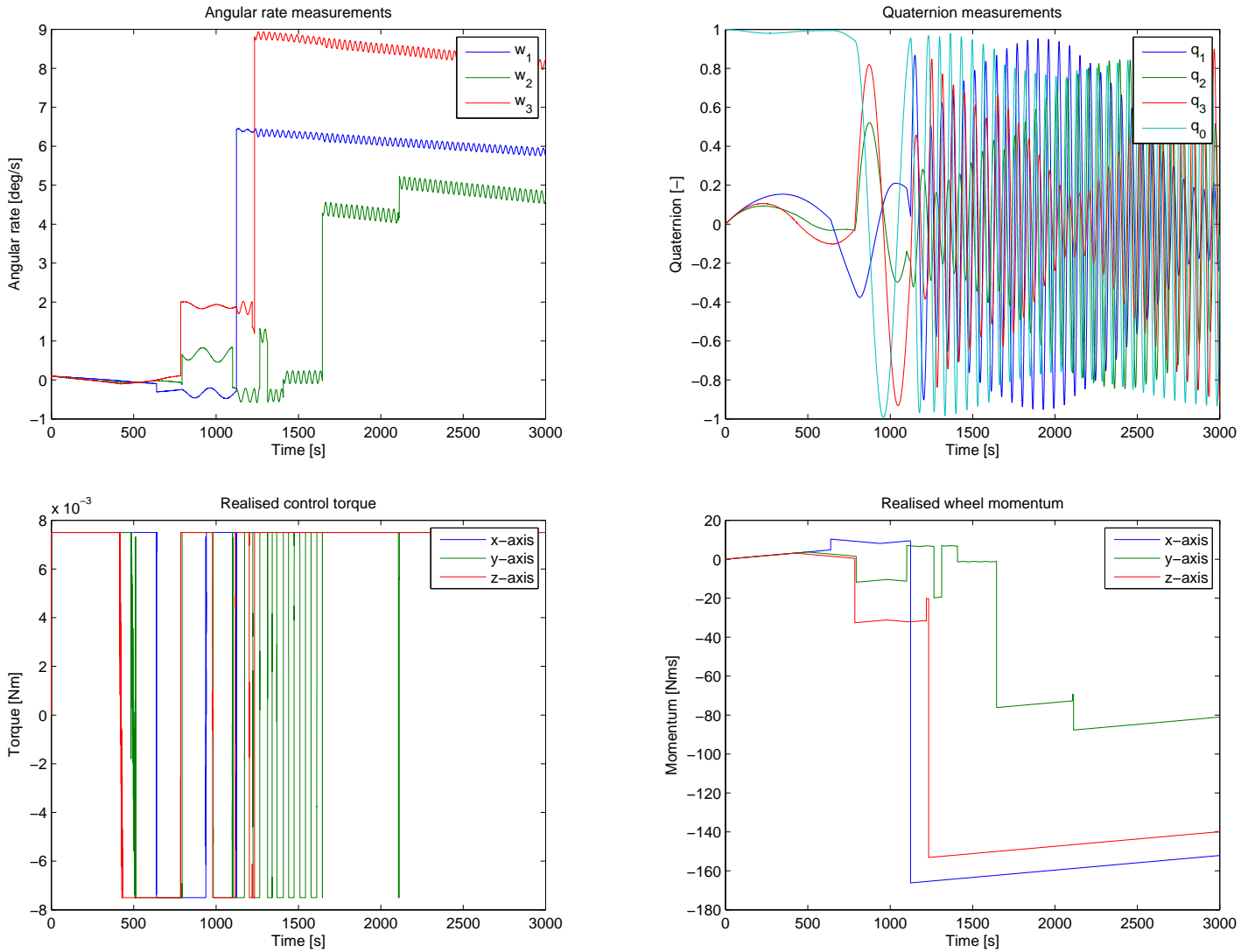


Figure 5.3: Results of GAST simulation of de-spinning of a satellite using Reaction Wheel 2.

In this simulation, it is seen that the requirements are not fulfilled and the angular rates do not settle to zero. The reason is the low saturation limit of Reaction Wheel 2, which means that this reaction wheel configuration does not provide sufficient torque to control and de-spin the satellite.

5.2 Yaw Steering in Sun-Synchronous Orbit

This simulation example demonstrates a yaw steering control of a satellite in sun synchronous orbit. The satellite has a nadir-pointing requirement, meaning that the z-axis of the satellite body frame has to point towards the Earth throughout

the mission. The satellite is powered by solar panels and it is desired to direct the solar panels towards the sun to get as much power as possible. However, the nadir-pointing requirement limits the control to the yaw angle (rotation around the z -axis) and the rotation of the solar panels around the y -axis of the satellite body frame, where the panels are mounted. The principles of yaw steering is shown in Fig. 5.4 modified from [4], where a nadir-pointing satellite with solar panels on the y -axis is seen at different orbital positions. The rotation around the yaw axis and the rotation of the solar panels is seen at each position.

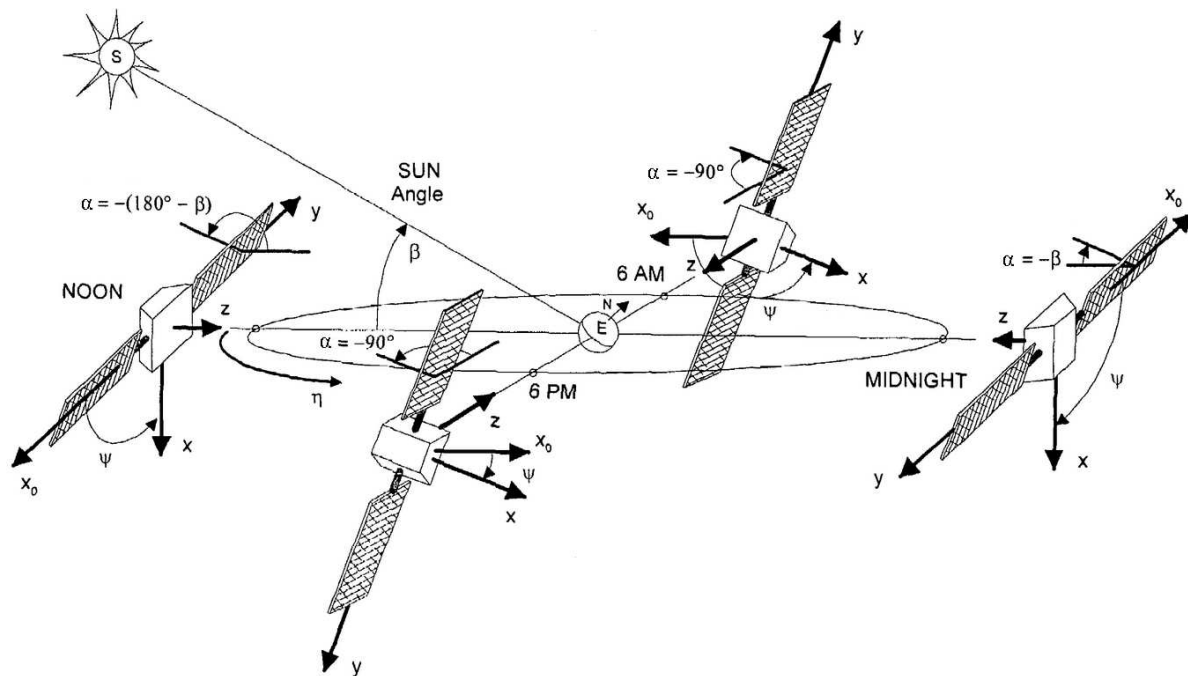


Figure 5.4: Principles of yaw steering.

In Fig. 5.4, \mathbf{x}_0 is the x -direction in the LVLH frame, $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ are the satellite body axes, and α denotes the solar panel rotation angle.

5.2.1 Orbit Definition

A sun-synchronous orbit is defined as an orbit with an altitude and inclination such that the regression of the nodes of approximately 1 degree per day, which allows the satellite orbit to follow the motion of the sun over the year. The satellite will have the same lighting conditions when passing over a certain point of the Earth and if placed in a near polar orbit, the satellite has coverage of the whole Earth. For this reason, sun synchronous orbits are useful for Earth observation and communication purposes. In this example orbit has an inclination of 98° , an altitude of 800 km, and a corresponding orbital angular velocity of 0.001 rad/s.

The right ascension of the ascending node, Ω , can be varied to analyse yaw steering with different illumination conditions.

5.2.2 Solar Panel Coordinate Frame

The solar panels are mounted so that they can rotate around the satellite y-axis, with the solar panel normal initially in the body z-direction. The solar panels are associated with a coordinate frame $(\hat{s}_x, \hat{s}_y, \hat{s}_z)$ which rotates with the panels around the y-axis with an angle α . The rotation of the solar panel coordinate frame (S) with respect to the LVLH frame is described by the yaw rotation of the satellite body, ψ , followed by the rotation of the solar panels around the y-axis, α , as

$$\begin{aligned} \mathbf{C}_{S,LVLH} &= \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} c(\alpha)c(\psi) & c(\alpha)s(\psi) & -s(\alpha) \\ -s(\psi) & c(\psi) & 0 \\ s(\alpha)c(\psi) & s(\alpha)s(\psi) & c(\alpha) \end{bmatrix} \end{aligned} \quad (5.2)$$

The solar panel coordinate frame is shown in Fig. 5.5 after a yaw rotation ψ and a solar panel rotation α , where \mathbf{s}_{z0} is the solar panel normal before the rotation.

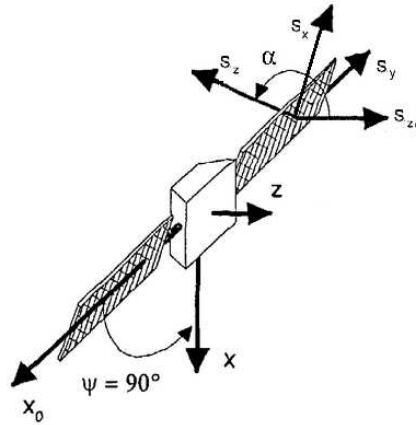


Figure 5.5: Solar panel reference frame.

5.2.3 Sun Direction Vector

The sun direction vector \mathbf{R}_s as seen by the satellite varies in time. However, for a satellite in sun-synchronous orbit, where the orbit follows the motion of the sun, the sun direction only varies due to the tilt of Earth's rotation with respect to the orbital plane. To simplify the problem, the sun is assumed to be located at the

Vernal Equinox, where the orbital and rotation plane of the Earth coincide. The sun direction in the ECI frame is therefore constant.

$$\mathbf{R}_s^{(ECI)} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (5.3)$$

The sun direction described in the LVLH reference frame is plotted for three different values of the right ascension of the ascending node $\Omega = 0^\circ, 45^\circ, 90^\circ$ in Fig. 5.6.

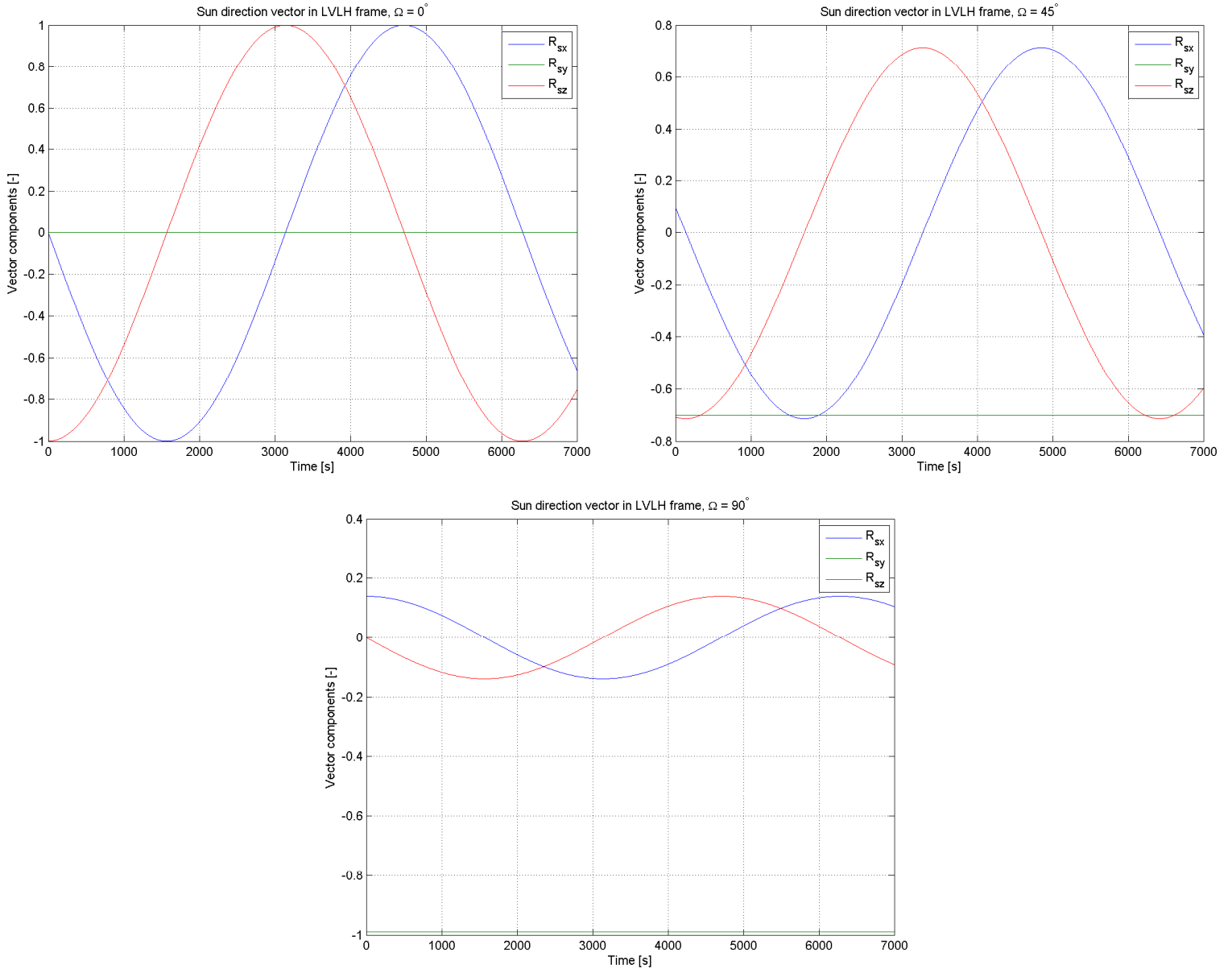


Figure 5.6: Sun direction vector in LVLH reference frame with $\Omega = 0^\circ, 45^\circ, 90^\circ$ respectively.

5.2.4 Optimal Yaw Angle

To achieve optimal sun illumination, it is desired that the satellite is turned around the yaw-axis so that the sun direction vector described in the body frame is perpendicular to the body y-axis, $\hat{\mathbf{y}}_b$. Thereafter, the solar panels can be rotated around the y-axis to achieve sun direction. The condition that must be satisfied is

$$\mathbf{R}_s^{(B)} \cdot \hat{\mathbf{y}}_b = 0 \quad (5.4)$$

Using this relation the optimal yaw angle can be calculated as

$$\begin{aligned} & \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{R}_s^{(LVLH)} \cdot \hat{\mathbf{y}}_b = 0 \Rightarrow \\ & -R_{sx} \sin(\psi) + R_{sy} \cos(\psi) = 0 \Rightarrow \\ & \tan(\psi) = \frac{R_{sy}}{R_{sx}} \Rightarrow \\ & \psi = \tan^{-1}\left(\frac{R_{sy}}{R_{sx}}\right) \end{aligned} \quad (5.5)$$

In Fig. 5.7, the optimal yaw angle is plotted for Ω ranging from 10° to 90° .

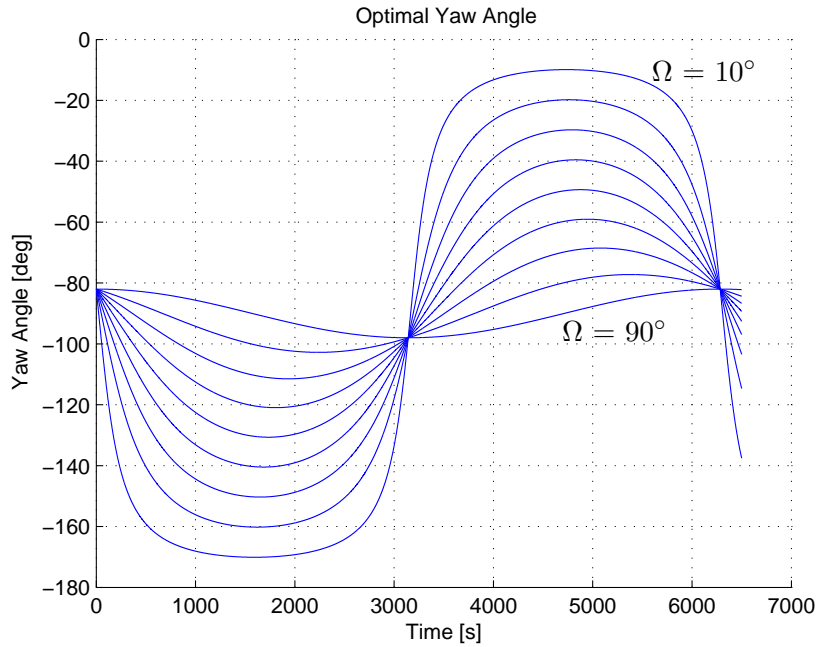


Figure 5.7: Optimal Yaw Angle for Ω between 10° and 90° .

It is seen that, as the right ascension of the orbit goes to 0° , the satellite has to

make a sharp turn of almost 180 degrees.

5.2.5 Optimal Solar Array Angle

For a maximum sun illumination angle, it holds that the sun direction and the solar array normal should satisfy

$$\mathbf{R}_s^{(LVLH)} \cdot \hat{\mathbf{s}}_z^{(LVLH)} = 1 \quad (5.6)$$

However, this is equivalent to that the unit vector in the z-direction of the solar array frame satisfies

$$\mathbf{R}_s^{(LVLH)} \cdot \hat{\mathbf{s}}_x^{(LVLH)} = 0 \quad (5.7)$$

The unit vector in the x-direction is described in the LVLH frame using the transpose of the rotation matrix derived in Section 5.2.2 as

$$\hat{\mathbf{s}}_x^{(LVLH)} = \mathbf{C}_{S,LVLH}^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\alpha) \cos(\psi) \\ \cos(\alpha) \sin(\psi) \\ -\sin(\alpha) \end{bmatrix} \quad (5.8)$$

Then the condition becomes

$$\left. \begin{aligned} R_{sx} \cos(\alpha) \cos(\psi) + R_{sy} \cos(\alpha) \sin(\psi) - R_{sz} \sin(\alpha) &= 0 \\ \cos(\alpha) [R_{sx} \cos(\psi) + R_{sy} \sin(\psi)] - R_{sz} \sin(\alpha) &= 0 \end{aligned} \right\} \Rightarrow$$

$$\tan(\alpha) = \frac{R_{sx} \cos(\psi) + R_{sy} \sin(\psi)}{R_{sz}} \Rightarrow \quad (5.9)$$

$$\alpha = \tan^{-1} \left(\frac{R_{sx} \cos(\psi) + R_{sy} \sin(\psi)}{R_{sz}} \right)$$

In Fig. 5.8, the optimal solar array angle is plotted for Ω ranging from 10° to 90° .

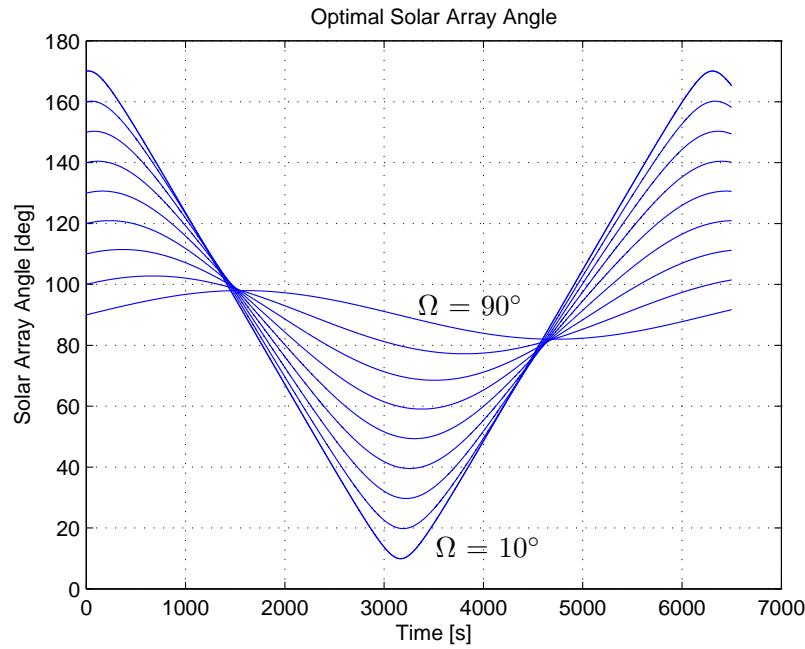


Figure 5.8: Optimal Solar Array Angle for Ω between 10° and 90° .

It is seen that as the right ascension approaches 0° , the optimal solar array angle varies as a triangle wave.

5.2.6 GAST Model

A Simulink model is made to simulate the results of a satellite with optimal yaw control to maximize the sun illumination of the solar panels. The model block diagram is shown in Fig. 5.9. It includes the dynamics and kinematics of a satellite with an inertia matrix with diagonal elements $I_{\text{diag}} = [1000, 1000, 1500] \text{ kgm}^2$. It also includes a star tracker model and a rate sensor model to feedback the measured quaternion and angular rate to the system. Quaternion control is implemented with proportional rate damping and proportional quaternion gain. The optimal yaw angle calculated as in the previous sections is given as a reference guidance attitude, with pitch and roll zero.

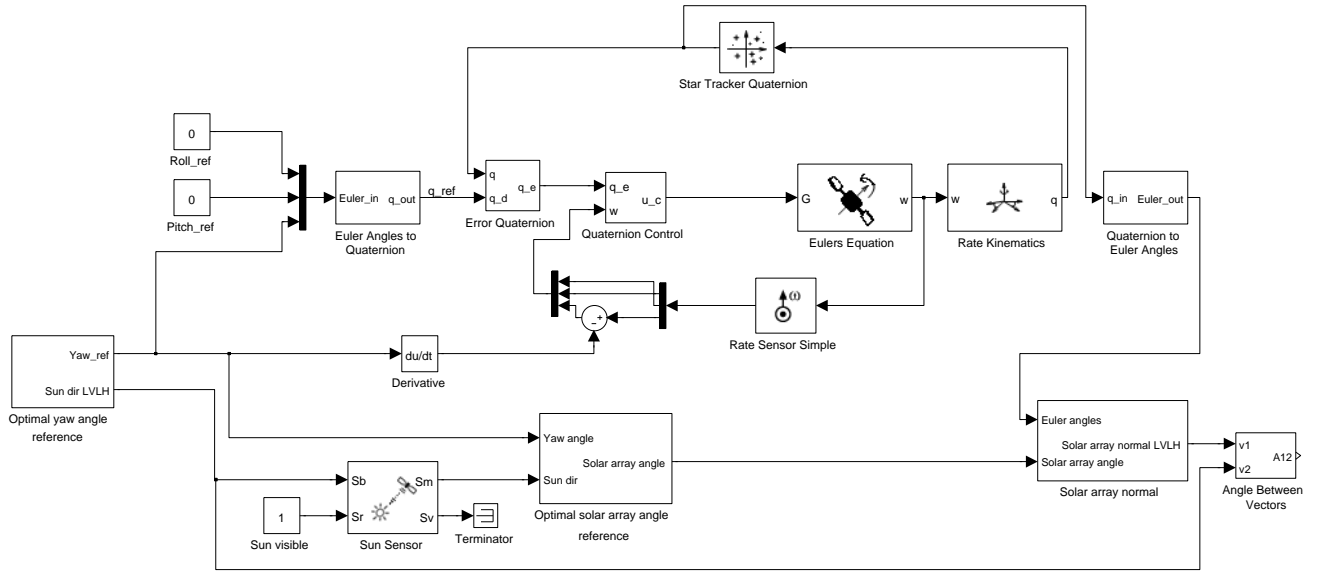


Figure 5.9: Simulink Model of Optimal Yaw Steering Control.

5.2.7 Results

The simulation is run with $\Omega = 45^\circ$. The proportional rate damping gain and the proportional quaternion gain are chosen by experience as $2\mathbf{I}$ and \mathbf{I} respectively. Fig. 5.10 shows the results of the simulation with no limitations on the control torque. The reference and realised yaw angle, the control torque and the error angle between the sun direction unit vector and the solar panel normal is plotted.

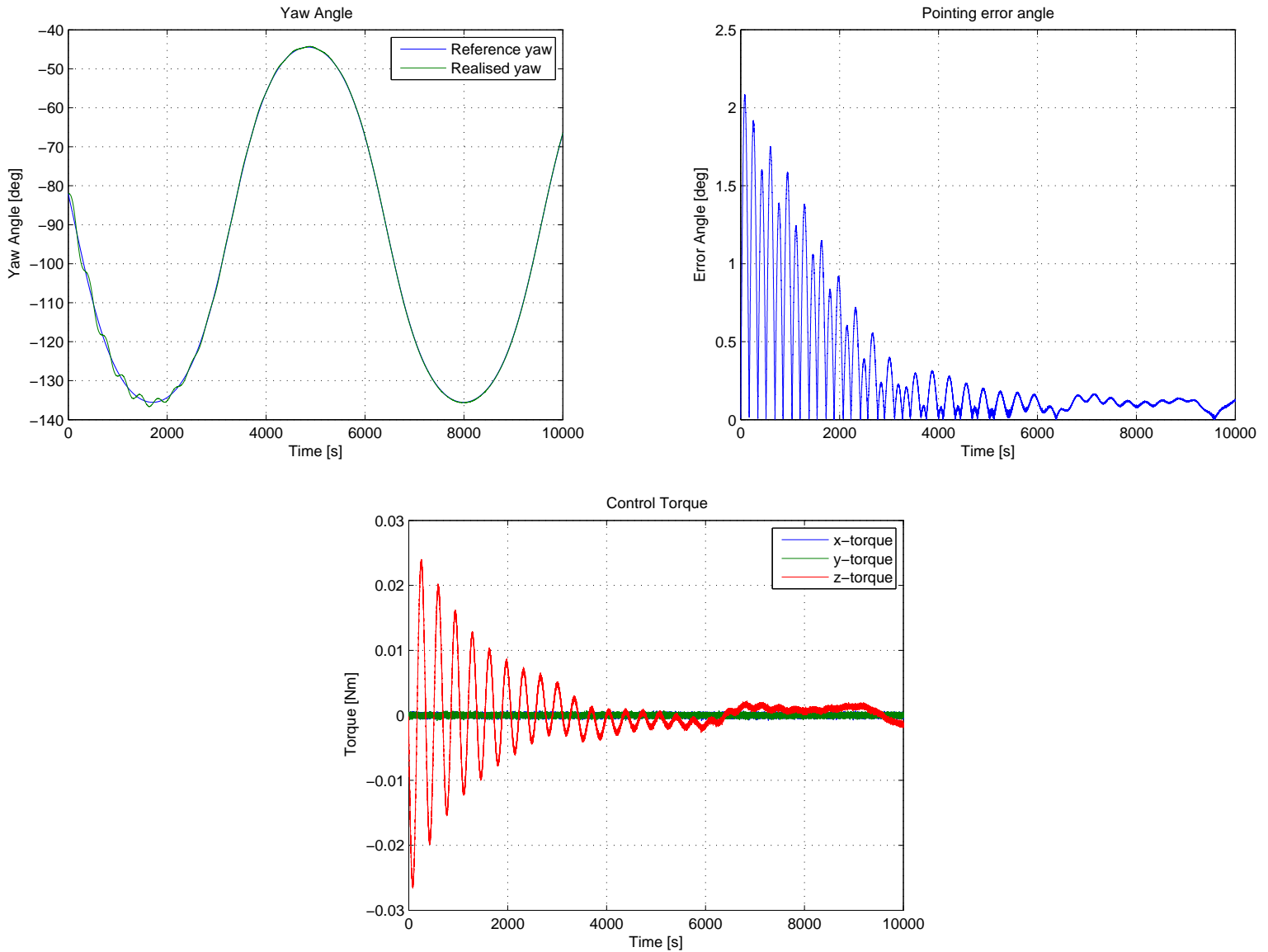


Figure 5.10: Results of yaw steering simulation with no limit on control torque.

The control is slow and the settling time is around 6000 s after which the pointing error angle stays below 0.15° . It is seen that the measurement noise is translated to the control torque, making it oscillate in an undesirable way.

The same simulation is performed with a saturation and quantisation of the control torque of 0.02 Nm, forcing the control torque to only assume the values 0.02, -0.02 , and 0 Nm. The result is shown in Fig. 5.11.

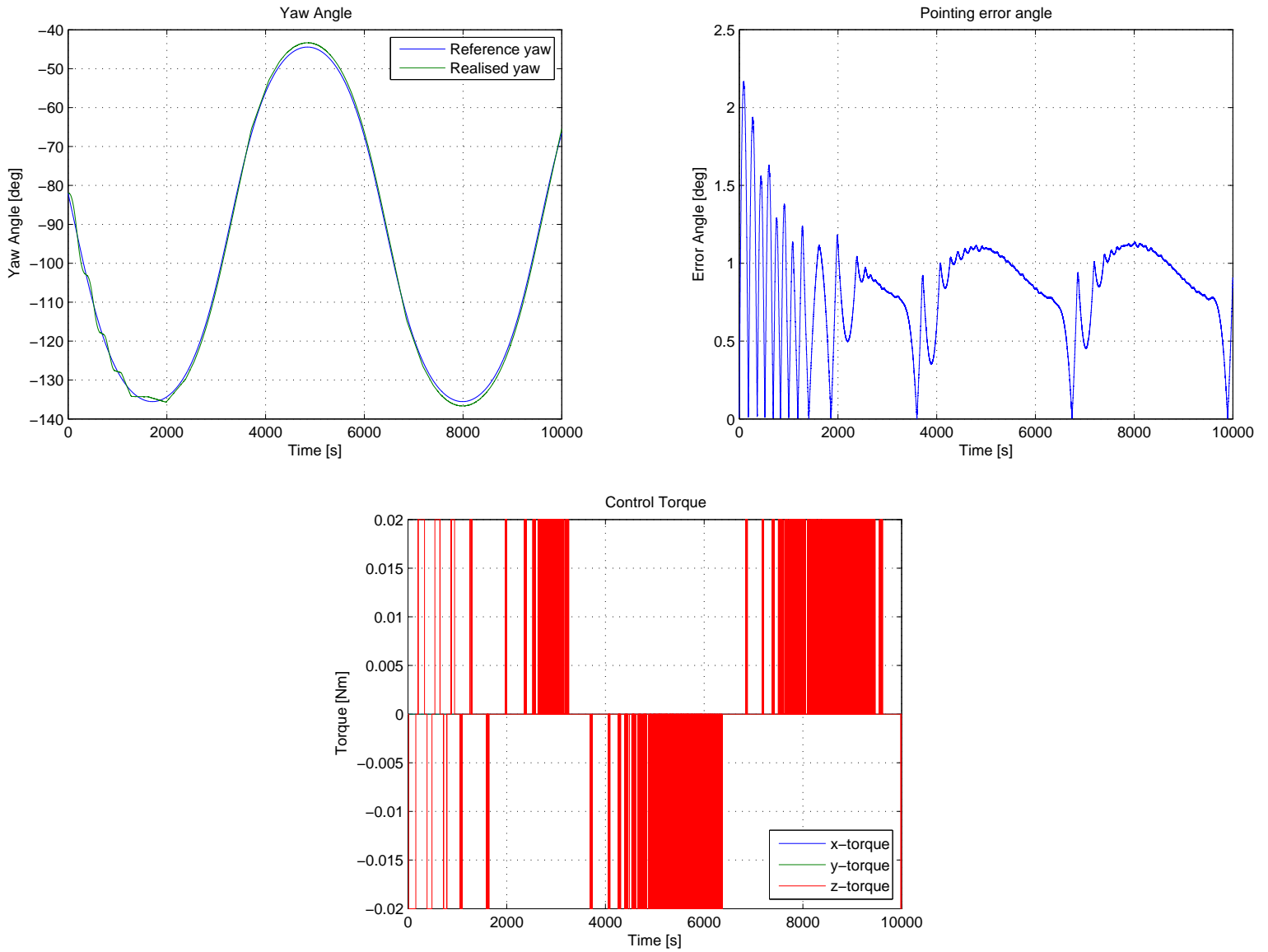


Figure 5.11: Results of yaw steering simulation with no limit on control torque.

In this case, the pointing error angle settles to a repeated pattern after about 3000 s. However, the maximum steady state error is more than 1° . Depending on the mission requirements, a better control law has to be developed to damp the oscillations around the reference angle and reduce the steady state error.

5.3 Control System Design for a Geostationary Satellite

For satellite missions demanding high life expectancies, control which ensures stability is of high priority. The control design must deal with disturbances such as aerodynamic torques, magnetic torques, gravity gradient torques, collisions (with other satellites and micrometeorites), solar radiation torques, and internal torques. This example simulation shows how the GAST Toolbox can be used to evaluate the stability of a control system design for a geostationary satellite.

A flight control system design for a geostationary communication satellite is designed. The control system is chosen to consist of an active, three-axis control, double-gimbaled, bias momentum control system. An Actuator Control System (ACS) can be used for desaturation, but is not considered in this example. The satellite is assumed to be symmetric with inertia $I_x = I_z = 3000 \text{ kgm}^2$ and $I_y = 660 \text{ kgm}^2$. The angular momentum bias is $h_n = 200 \text{ Nms}$ in the direction of the negative y-axis of the satellite.

The Euler equations used for this control system design are [1]

$$\begin{aligned} \mathbf{T} &= \dot{\mathbf{h}}^{(I)} = \dot{\mathbf{h}}^{(B)} + \boldsymbol{\omega} \times \mathbf{h} \\ \mathbf{h} &= \mathbf{h}_v + \mathbf{h}_w \\ \mathbf{h}_v &= I_x \omega_x \hat{\mathbf{x}}_b + I_y \omega_y \hat{\mathbf{y}}_b + I_z \omega_z \hat{\mathbf{z}}_b \end{aligned} \tag{5.10}$$

where $\dot{\mathbf{h}}^{(I)}$ is the derivative of the angular momentum in the Inertial frame and $\dot{\mathbf{h}}^{(B)}$ is the derivative in the body frame. The total angular momentum is the sum of the spacecraft angular momentum, \mathbf{h}_v , and the wheel angular momentum, \mathbf{h}_w . The wheel momentum components are

$$\begin{aligned} h_{wx} &= h_n \cos \delta \sin \gamma \\ h_{wy} &= -h_n \cos \delta \cos \gamma \\ h_{wz} &= -h_n \sin \delta \end{aligned} \tag{5.11}$$

where δ and γ are the gimbal deflections in the x- and z-directions respectively. It is assumed that the gimbal deflections are small, the deviation from nominal wheel angular momentum is small, and that the nominal wheel angular momentum is much greater than $\max[I_x \omega_0, I_y \omega_0, I_z \omega_0]$. With these assumptions, the dynamics above can be linearised to become three equations in yaw ψ , pitch θ , and roll ϕ .

$$\begin{aligned} T_x &= I_x \ddot{\phi} + \omega_0 h_n \phi + h_n \dot{\psi} + \dot{h}_x - \omega_0 h_z \\ T_y &= I_y \ddot{\theta} + \dot{h}_y \\ T_z &= I_z \ddot{\psi} + \omega_0 h_n \psi - h_n \dot{\phi} + \dot{h}_z + \omega_0 h_x \end{aligned} \tag{5.12}$$

where T_x , T_y , and T_z are the disturbance torques acting on the spacecraft. At the altitude of a geostationary communication satellite, the dominant disturbance torque is determined to be the solar radiation torque. The solar radiation torques are assumed in [1] to be periodic functions varying as

$$\begin{aligned} T_x &= 2 \cdot 10^{-5} (1 - 2 \sin(\omega_0 t)) \text{ Nm} \\ T_y &= 10^{-4} \cos(\omega_0 t) \text{ Nm} \\ T_z &= -5 \cdot 10^{-5} \cos(\omega_0 t) \text{ Nm} \end{aligned} \quad (5.13)$$

where ω_0 is the geostationary orbital rate of $7.28 \cdot 10^{-5}$ rad/s.

5.3.1 Pitch Control Design

The pitch equation is uncoupled from the x- and z-axes, which makes it easier to control. \dot{h}_y is the pitch control function, so to provide the required pitch damping, a linear PD pitch control law is selected as

$$\dot{h}_y = K_p(\tau_p \dot{\theta} + \theta) = u_y \quad (5.14)$$

where K_p is the proportional gain constant and τ_p is the time constant in seconds. The derivative gain is $K_p \tau_p$.

The closed loop of the pitch control system is written in terms of transfer functions as

$$\theta(s) = \frac{G(s)}{1 + G(s)U(s)} T(s) + \frac{G(s)U(s)}{1 + G(s)U(s)} R(s) \quad (5.15)$$

Where $G(s)$ is the plant transfer function, $U(s)$ is the controller transfer function, $T(s)$ is the disturbance transfer function and $R(s)$ is the reference.

With $M(s) = K_p(\tau_p s + 1)$, the closed loop becomes

$$\theta(s) = \frac{1}{I_y s^2 + K_p \tau_p s + K_p} T(s) + \frac{K_p \tau_p s + K_p}{I_y s^2 + K_p \tau_p s + K_p} R(s) \quad (5.16)$$

The stability of the system is investigated by looking at the denominator of the transfer function from disturbance to output. To ensure stability of the system, the Routh-Hurwitz stability criterion is a necessary and sufficient criterion to prove stability of an LTI system. The Routh-Hurwitz criterion states that for a characteristic equation of second order, the system has no right half plane poles if all coefficients of the polynomial are greater than zero. In this case, it implies a positive K_p and τ_p to ensure system stability.

The gain selection is based on the steady-state error and the response time of the system. The maximum steady-state error is estimated using the final value theorem, assuming that τ_p is much smaller than the orbital period and that the critical damping of the system is 1.

$$\begin{aligned}\lim_{t \rightarrow \infty} \theta(t) &= \lim_{s \rightarrow 0} s\theta(s) \\ \theta_{SS} &= \frac{T_y}{K_p} = \frac{10^{-4}}{K_p} < 0.000873 \\ K_p &> 0.1145 \text{ Nm/rad}\end{aligned}\tag{5.17}$$

The last relation is given from a pitch accuracy requirement of 0.05° .

The control parameters are chosen as $K_p = 0.4125 \text{ Nm/rad}$ and $\tau_p = 80 \text{ s}$.

5.3.2 Roll/Yaw Control Design

The roll and yaw dynamic equations are coupled, meaning that a control torque in roll will also affect the yaw angle. Using an Earth sensor it is not possible to make measurements of the yaw angle, therefore it is proposed to use a controller for both roll and yaw with only roll measurements. This can be done because of the gyroscopic coupling of the roll and yaw errors. The errors in yaw are translated to roll errors after a quarter of an orbit and are then controlled with the roll feedback law. For an active control without yaw measurements, the control laws are suggested to be [1]

$$\begin{aligned}u_{xc} &= \dot{h}_x - \omega_0 h_z = K(\tau \dot{\phi} + \phi) \\ u_{zc} &= \dot{h}_z + \omega_0 h_x = -kK(\tau \dot{\phi} + \phi)\end{aligned}\tag{5.18}$$

where K is the proportional gain constant, τ is the time constant, and k is the yaw-to-roll gain ratio of the feedback controller.

The system dynamics becomes

$$\begin{aligned}T_x - u_{xc} &= I_x \ddot{\phi} + \omega_0 h_n \dot{\phi} + h_n \dot{\psi} \\ T_z - u_{zc} &= I_z \ddot{\psi} + \omega_0 h_n \dot{\psi} - h_n \dot{\phi}\end{aligned}\tag{5.19}$$

In the Laplace domain, the system becomes

$$\begin{aligned}T_x &= I_x \phi(s)s^2 + \omega_0 h_n \phi(s) + h_n \psi(s)s + K\tau \phi(s)s + K\phi(s) \\ T_z &= I_z \psi(s)s^2 + \omega_0 h_n \psi(s) - h_n \phi(s)s - kK\tau \phi(s)s - kK\phi(s)\end{aligned}\tag{5.20}$$

or in matrix form

$$\begin{bmatrix} T_x \\ T_z \end{bmatrix} = \begin{bmatrix} I_x s^2 + K\tau s + \omega_0 h_n + K & h_n s \\ -(h_n + kK\tau)s - kK & I_z s^2 + \omega_0 h_n \end{bmatrix} \begin{bmatrix} \phi(s) \\ \psi(s) \end{bmatrix} \quad (5.21)$$

Solving for the roll and yaw angles and setting $I_x = I_z$

$$\begin{aligned} \begin{bmatrix} \phi(s) \\ \psi(s) \end{bmatrix} &= \begin{bmatrix} I_x s^2 + K\tau s + \omega_0 h_n + K & h_n s \\ -(h_n + kK\tau)s - kK & I_z s^2 + \omega_0 h_n \end{bmatrix}^{-1} \begin{bmatrix} T_x \\ T_z \end{bmatrix} \\ &= \frac{1}{\det(A)} \begin{bmatrix} I_z s^2 + \omega_0 h_n & -h_n s \\ (h_n + kK\tau)s + kK & I_x s^2 + K\tau s + \omega_0 h_n + K \end{bmatrix} \begin{bmatrix} T_x \\ T_z \end{bmatrix} \\ &= G(s) \begin{bmatrix} T_x \\ T_z \end{bmatrix} \\ \det(A) &= (I_x s^2 + \omega_0 h_n)^2 + (I_x s^2 + \omega_0 h_n)(K + \omega_0 h_n) + h_n^2 s^2 + h_n kKs(\tau s + 1) \end{aligned} \quad (5.22)$$

If the control parameters are chosen to be $K = 1.875$, $k = 0.0033$, and $\tau = 80$, the poles of the closed loop system can be calculated from the determinant $\det(A)$ and are

$$\begin{aligned} p_1 &= -0.02 + 0.07i \\ p_2 &= -0.02 - 0.07i \\ p_3 &= -3.44 \cdot 10^{-5} + 7.7 \cdot 10^{-4}i \\ p_4 &= -3.44 \cdot 10^{-5} - 7.7 \cdot 10^{-4}i \end{aligned}$$

Since all real parts of the poles are negative, the system is stable.

5.3.3 Simulation Results

The simulation results of the linearised system described in the previous sections is shown in Fig. 5.12.

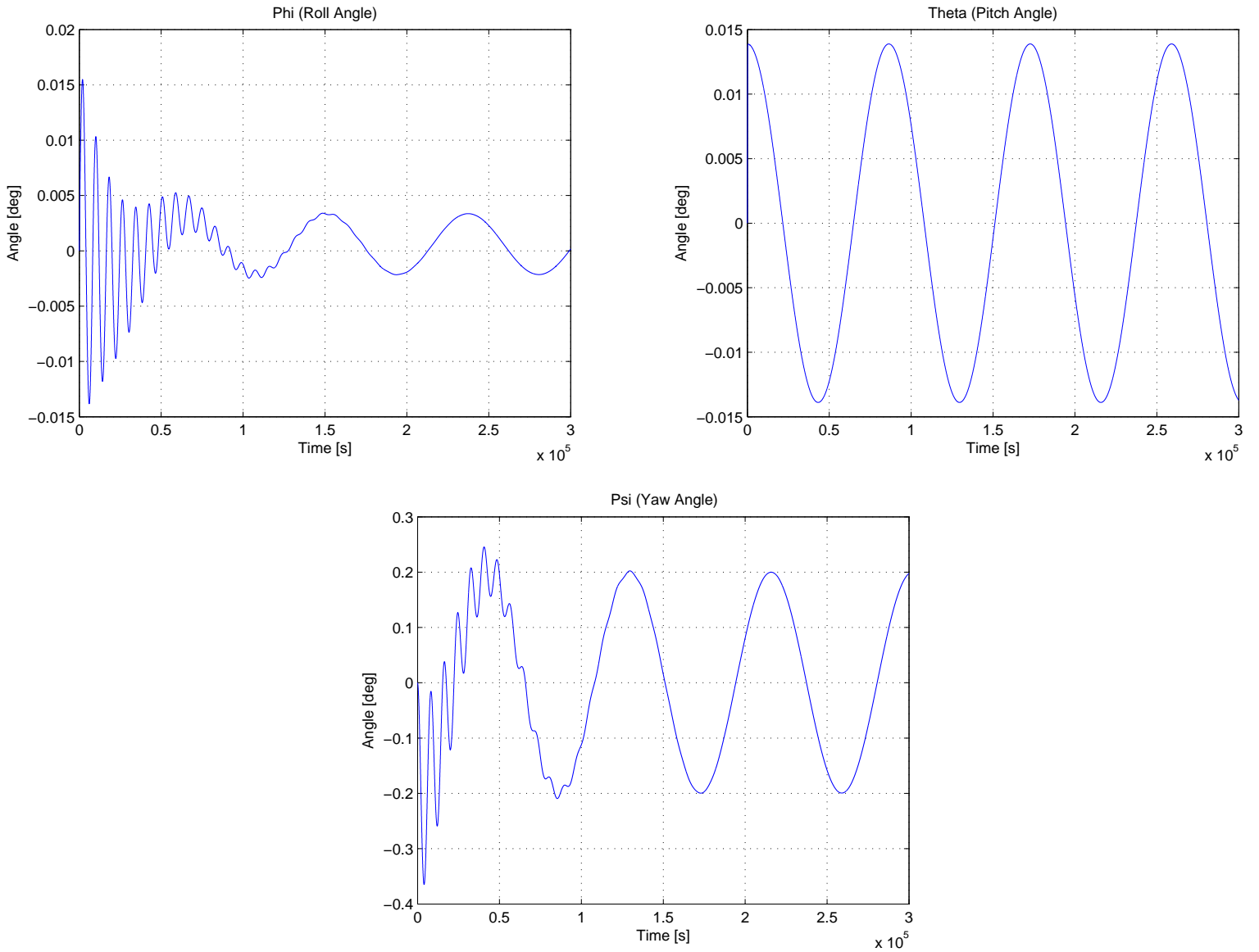


Figure 5.12: Results of linearised simulation model.

All angles settle to a stable oscillation with steady-state errors 0.003° , 0.014° , and 0.2° for the roll, pitch, and yaw angles respectively.

To compare the results, a simulation using GAST blocks with nonlinear satellite dynamics was made. The transient results for the roll, pitch, and yaw angles are shown in Fig. 5.13.

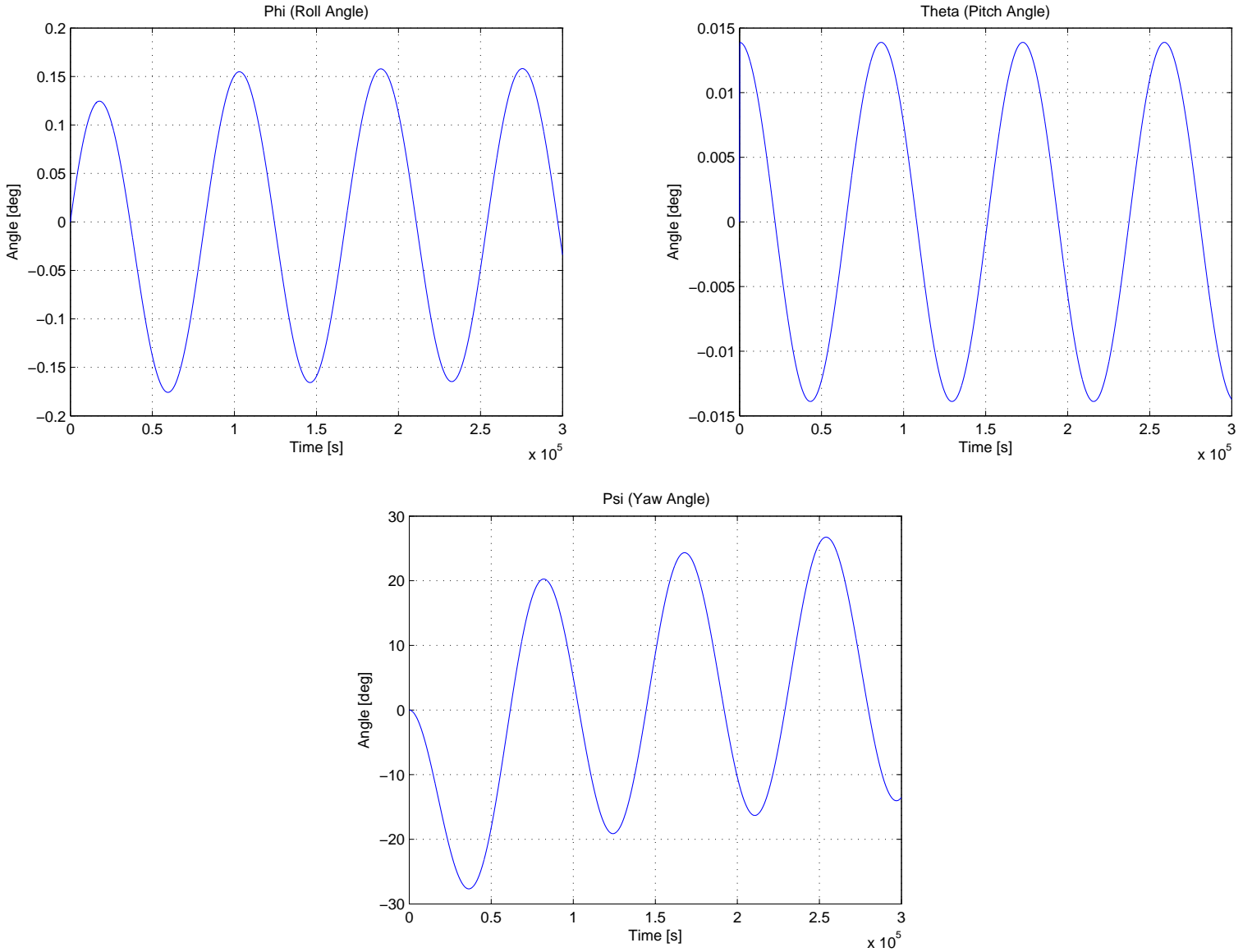


Figure 5.13: Results of the nonlinear simulation model using GAST.

The roll and pitch angles both settle to a periodic oscillation due to the presence of the disturbance torques. The maximum values are 0.16° and 0.014° respectively. The pitch response is the same as in the linear model, while the roll response has a significantly greater steady state error. The yaw angle is clearly diverging and reaching values of several tens of degrees. This result shows that, even though the linear control system design indicated stability, only roll measurements are not enough to stabilise the yaw angle in the presence of disturbance torques. If there are stricter requirements on the yaw angle, either it has to be controlled with yaw angle measurement feedback or a more appropriate control law has to be developed.

5.4 Mars Ascent Vehicle

Missions to Mars have recently become a hot topic in the space industry. A simulation model of a Mars Ascent Vehicle (MAV) was made and included in the GAST toolbox, as well as all the blocks created for the simulation. The simulation is based on the results from a study named Planetary Ascent Vehicle (PAV) made by ESA in collaboration with EADS Space Transportation and SciSys within the Aurora Programme [5]. In this case the recovery of a Mars soil sample collected by a robotic mission is considered. The simulation is built up according to the block diagram from [5] shown in Fig. 5.14.

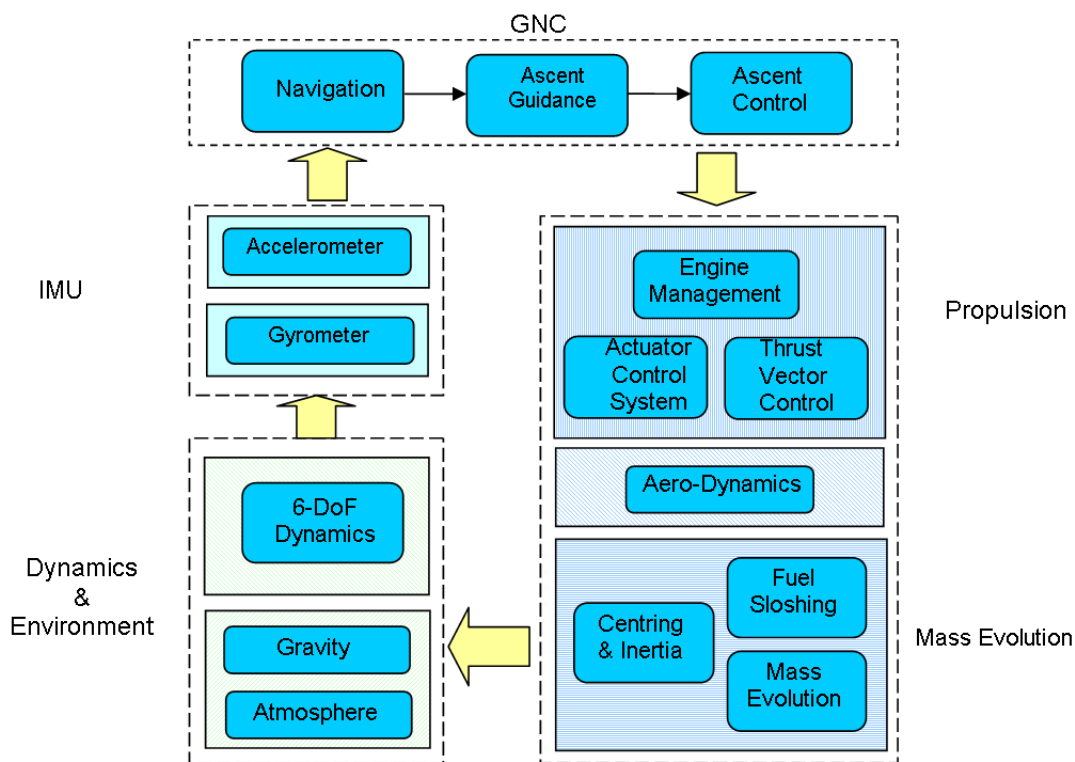


Figure 5.14: Simulation block diagram.

The MAV has a liquid propellant 2-stage configuration. Four engines are required for the first stage and one for the second stage. The simulation is designed so that the first stage separates when its fuel is depleted and thereafter the second stage ignites. The design of the GNC algorithms are out of the scope of this document. The following sections describe the Simulink block models created for the rest of the simulation. Users can then create custom GNC algorithms suited for the desired mission requirements. The GAST blocks created can easily also be used in similar simulations of ascent vehicles.

5.4.1 Dynamics

The simulation contains a 6 DoF dynamics model of a launcher, with 3 DoF translational motions and 3 DoF rotational motions. The forces and torques acting on the vehicle cause translational and rotational accelerations as

$$\begin{aligned}\mathbf{F} &= m\mathbf{a} \\ \mathbf{T} &= \mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega}\end{aligned}\tag{5.23}$$

The equations hold as long as the current mass and the current inertia matrix are not varying quickly with time. The equations are integrated to give the translational and rotational velocities as well as the position and attitude of the vehicle. The rate kinematic equations using quaternions are given by

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_0 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}\tag{5.24}$$

The initial conditions are given as the initial position, velocity, attitude, and rotational velocity in the PCI frame. The user can specify the initial position with longitude, latitude, altitude, launch azimuth and initial quaternion with respect to the launch pad and use the appropriate GAST coordinate transformation blocks to convert the values to PCI coordinates.

5.4.2 Environment

Mars Gravity Model

To model the gravitational accelerations and forces acting on the MAV, a gravity model is constructed in Simulink.

All planets depart from the symmetry of a perfect sphere and this effect has to be taken into account using spherical harmonics. To simplify the model, the planet is assumed to be symmetrical around the axis of rotation so that the potential only depends on the latitude. The model with this assumption is called zonal harmonics. Then the gravitational potential can be written as in [16]

$$\Phi(r, \theta) = -\frac{GM}{r} \left\{ 1 - \sum_{n=2}^{\infty} \left(\frac{R_e}{r} \right)^n J_n P_n(\cos \theta) \right\}\tag{5.25}$$

where J_n are the so called Jeffrey's constants or zonal gravity coefficients which are unique for a planet. P_n are the Legendre polynomials.

The potential is defined as approaching zero as r goes to infinity and decreasing when coming closer to the planet. The gravitational acceleration can be written as the negative gradient of the potential as

$$\mathbf{g} = -\nabla\Phi = -\frac{\partial\Phi}{\partial r}\mathbf{e}_r - \frac{1}{r}\frac{\partial\Phi}{\partial\theta}\mathbf{e}_\theta = g_r\mathbf{e}_r + g_\theta\mathbf{e}_\theta \quad (5.26)$$

For this simulation block the Legendre polynomials up to the fourth order are taken into consideration. Taking the derivative, the gravitational acceleration components become

$$g_r = -\frac{GM}{r^2} \left[1 - 3J_2\left(\frac{R_e}{r}\right)^2 P_2(\cos\theta) - 4J_3\left(\frac{R_e}{r}\right)^3 P_3(\cos\theta) - 5J_4\left(\frac{R_e}{r}\right)^4 P_4(\cos\theta) \right] \quad (5.27)$$

$$g_\theta = \frac{3GM}{r^2} \left(\frac{R_e}{r}\right)^2 \sin\theta \cos\theta \left[J_2 + \frac{1}{2}J_3\left(\frac{R_e}{r}\right) \sec\theta(5\cos^2\theta - 1) + \frac{5}{6}J_4\left(\frac{R_e}{r}\right)^2(7\cos^2\theta - 1) \right] \quad (5.28)$$

The coefficients J_n are shown in Table 5.2 for Earth and Mars [16].

	J_2	J_3	J_4
Earth	0.0010826269	-0.0000025323	-0.0000016204
Mars	0.001964	0.000036	—

Table 5.2: Zonal gravity coefficients for Earth and Mars.

Mars Atmosphere

The dynamics of the vehicle including loads depends on the position and velocity of the vehicle in the atmosphere, since the properties of the atmosphere is varying.

The data of the Mars atmosphere taken from the model in the PAV mission is placed in a table and for a given position and velocity, the data is interpolated to give values of the Mach number, speed of sound, density, scale height, and free stream temperature.

5.4.3 Aerodynamic Model

The aerodynamic forces and torques are calculated as

$$\begin{aligned}
F_x &= -C_A(M, \alpha) S_{ref} q_\infty \\
F_y &= -C_N(M, \beta_c) S_{ref} q_\infty \\
F_z &= -C_N(M, \alpha_c) S_{ref} q_\infty
\end{aligned} \tag{5.29}$$

$$\begin{aligned}
T_x &= 0 \\
T_y &= C_m(M, \alpha_c) S_{ref} l_{ref} q_\infty \\
T_z &= -C_m(M, \beta_c) S_{ref} l_{ref} q_\infty
\end{aligned} \tag{5.30}$$

where $T = T_n$ is the torque around the nose of the launch vehicle. It is assumed that the Center of Pressure is not changing with the angle of attack and sideslip angle. The torque around the Center of Gravity (CoG) is then calculated as

$$\mathbf{T}_G = \mathbf{r}_{CoG,CoP} \times \mathbf{F} = (\mathbf{r}_{CoG,n} - \mathbf{r}_{CoP,n}) \times \mathbf{F} = \mathbf{r}_{CoG,n} \times \mathbf{F} + \mathbf{T}_n \tag{5.31}$$

The aerodynamic coefficients, C , are predefined in tables, as defined in the PAV simulator, and calculated as a function of M and α , α_c , or β_c using interpolation of the tabulated data. The angles α , α_c , and β_c are defined as in Fig. 5.15 from [7], where α is the total angle of attack between the body x-axis and the relative wind, α_c is the angle between the stability x-axis (the projection of the relative wind vector onto the xz-plane) and the body x-axis, and β_c is the sideslip angle between the relative wind vector and the stability x-axis.

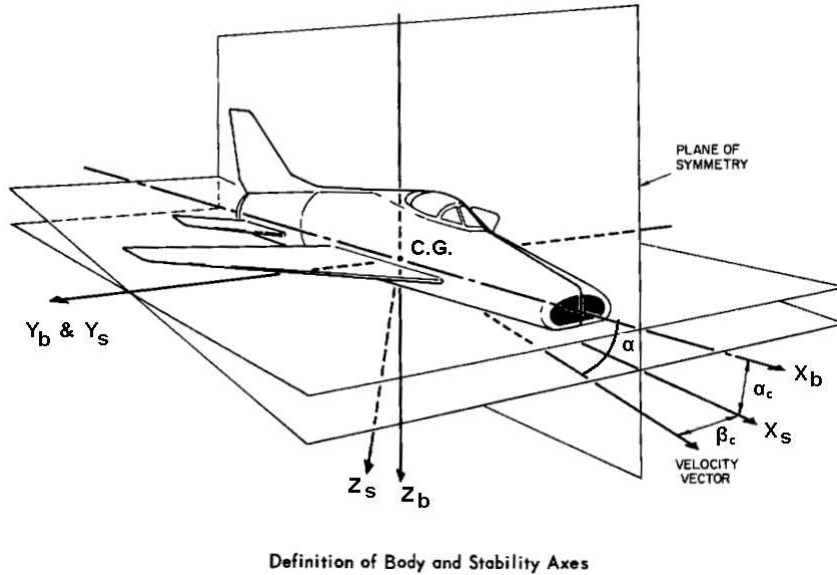


Figure 5.15: Definition of angle of attack and sideslip angle [7].

The angles are calculated as

$$\begin{aligned}
\alpha &= \cos^{-1}(\cos(\alpha_c) \cos(\beta_c)) \\
\alpha_c &= \tan^{-1}(v_{Bz}/v_{Bx}) \\
\beta_c &= \tan^{-1}(v_{By}/v_{Bx})
\end{aligned} \tag{5.32}$$

5.4.4 Propulsion

The propulsion system consists of 5 main engines which are throtttable and double gimballed, and an Actuator Control System for attitude control during coasting phases.

Engine Management

The Engine Management block is designed to calculate the thrust for each of the main engines. The engines are throtttable using differential thrust. The thrust is calculated for each engine as

$$F_c = \Delta F + qI_{sp}g_0 - p_aA_{exit} \tag{5.33}$$

where ΔF is the commanded differential thrust of one engine, q is the nominal propellant flow rate, I_{sp} is the specific impulse, g_0 is the standard Earth gravitation, p_a is the atmospheric pressure, and A_{exit} is the engine nozzle exit area. q , I_{sp} and the mixture ratio, MR , of the propellant are interpolated from a predefined table as a function of time. A separate table is also specified for the decay period after the engines are turned off, where the properties are functions of the time after the shut down. The total propellant flow rate, including the differential thrust, is calculated as

$$Q_p = \frac{\Delta F}{I_{sp}g_0} + q \tag{5.34}$$

Thrust Vector Control

The Thrust Vector Control block is calculating the force and torque vector acting on a spacecraft resulting from the engines (n engines in total). The engines are double gimballed, with a commanded rotation of the nominal thrust direction (body x-axis) first around the body z-axis with an angle β_z and sequentially around the body y-axis with and angle β_y . The resulting thrust direction becomes

$$\begin{aligned}
\mathbf{e}_F &= \begin{bmatrix} \cos \beta_y & 0 & \sin \beta_y \\ 0 & 1 & 0 \\ -\sin \beta_y & 0 & \cos \beta_y \end{bmatrix} \begin{bmatrix} \cos \beta_z & -\sin \beta_z & 0 \\ \sin \beta_z & \cos \beta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} \cos \beta_y \cos \beta_z \\ \sin \beta_z \\ -\sin \beta_y \cos \beta_z \end{bmatrix}
\end{aligned} \tag{5.35}$$

The realised angles β_y and β_z are derived from the commanded angles subject to limitations in the deflection angular rates and the deflection angles. Thereafter, the thrust direction is calculated as above and the force vector as the magnitude of the force times the thrust direction for each engine. The total force is the sum of all engine forces. The torque from each thruster around the CoG is calculated as

$$\mathbf{T}_i = (\mathbf{r}_i - \mathbf{CoG}) \times \mathbf{F}_i \tag{5.36}$$

where \mathbf{r}_i is the location of the thrust application of engine i and \mathbf{CoG} is the location of the CoG in the body frame.

Actuator Control System

The Actuator Control System (ACS) consists of 6 thrusters to give 3 DoF attitude control.

This block calculates the propellant consumption, total force, and total torque of an Actuator Control System (ACS) with n thrusters. For each thruster the thrust is

$$F = \begin{cases} F_{\text{thruster}} & \text{if } AO = 1 \\ 0 & \text{if } AO = 0 \end{cases} \tag{5.37}$$

where AO is the activation order.

The thrust direction is defined by two angles, α and β , where α represents a rotation of the nominal thrust direction vector (defined as the direction of the body x-axis) about the body y-axis. Thereafter the vector is rotated again around the body x-axis with angle β to give the final thrust direction as

$$\begin{aligned}
\mathbf{e}_F &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \beta & -\sin \beta \\ 0 & \sin \beta & \cos \beta \end{bmatrix} \begin{bmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} \cos \alpha \\ \sin \alpha \sin \beta \\ -\sin \alpha \cos \beta \end{bmatrix}
\end{aligned} \tag{5.38}$$

The torque from each thruster is calculated as in (5.36). The forces and torques for each thruster are summed up to give the total force and torque from the ACS.

The propellant consumption of the ACS is calculated as

$$M_{ACS} = \int \frac{F}{I_{sp}g_0} dt \quad (5.39)$$

5.4.5 Mass Evolution

The propellant consumption of one stage is calculated as

$$M_c = \int (\sum_{i=1}^n Q_{pi}) dt \quad (5.40)$$

where Q_{pi} are the propellant flow rates of each engine.

The total remaining fuel in the tanks is calculated as

$$M_{\text{fuel}} = M_{f,\text{init}} - \int (\sum \frac{Q_{pi}}{1 + MR}) dt \quad (5.41)$$

where MR is the mixture ratio, i.e. the oxidiser mass divided by the fuel mass.

In the same way the total remaining oxidiser is calculated as

$$M_{\text{oxidiser}} = M_{o,\text{init}} - \int (\sum MR \frac{Q_{pi}}{1 + MR}) dt \quad (5.42)$$

The current mass of the stage is calculated as the sum of the dry mass and the remaining propellant. If there is no more fuel or oxidiser in the tanks, the depletion flag is set to 1. If either the stage is depleted or the separation order is true, the stage does not consume any more propellant. The current mass is then calculated as the initial mass minus the consumed propellant and minus the mass releases due to stage separation.

The CoG and inertia matrix of the vehicle change as a function of the vehicle mass throughout the ascent. The CoG and inertia matrix evolutions are given in tables of pre-estimated values according to the PAV simulator. The fuel sloshing model is not considered in this example.

5.4.6 Inertial Measurement Unit

For navigation purposes, an Inertial Measurement Unit (IMU) is used, which consists of a gyroscope (rate sensor) to measure angular rates and an accelerometer to measure accelerations. These models already exist in the GAST toolbox and can be exported directly to the simulation.

5.5 Active Debris Removal

In the last decades, the increasing space activity has lead to a huge amount of space debris. Space debris include non-operational spacecraft, used upper stages, and fragments from collisions and explosions. As the debris population grows, the risk of collisions increase and it is therefore necessary to reduce the number of large and massive objects in space. Studies at NASA and ESA show that the space environment can be stabilized by actively removing 10 large object per year [17].

Envisat is an ESA satellite that was launched on 1 March 2002 and it is the largest Earth observation satellite ever built. However, communication with the Envisat satellite was suddenly lost on 8 April 2012, and after rigorous attempts to re-establish contact, the end of the mission was declared on 9 May 2012. To prevent Envisat from damaging other spacecraft an option that has been studied by ESA is to actively remove the satellite using a debris removal satellite. In this simulation example, the GAST toolbox is used to simulate the rendezvous with Envisat. Characteristics and orbital data of Envisat are summarized in Table 5.3.

Orbit type	Sun-synchronous, polar orbit
Inclination	98.55°
Mean altitude	800 km
Orbital angular velocity	0.001 rad/s
Repeat cycle	35 days
Mass	8211 kg

Table 5.3: Specifications of Envisat satellite.

5.5.1 Approach Strategy

The removal of uncooperative targets is a big challenge, especially when the target is subject to uncontrolled spinning. The strategy here is to approach the target from a distance below and behind using an impulsive Hohmann transfer to achieve the same orbit. When the chaser is 3000 m behind the target a “hop” is made by an impulsive thrust in the LVLH z-direction to achieve a distance of 500 m to the target, where the chaser stays station keeping while identifying the target. Thereafter, a forced motion along \bar{v} is made to get as close as 50 m to the

target. An out of plane inspection is made to analyse the target, where the attitude of the spacecraft is controlled. The last step is a fly-around in the same rotational speed so that the bodies are spinning around a mutual center point. From this state it is possible to either dock, grab, or throw the chosen capture mechanism at the target. The different phases are summarized in Table 5.4 and shown graphically in Fig. 5.16, where v-bar and r-bar denotes the x- and z-axis of the target body frame respectively.

Manoeuvre	Final relative position [m]
1) Far range Hohmann transfer	$[-3000, 0, 0]$
2) Impulsive hop on v-bar	$[-500, 0, 0]$
3) Station keeping on v-bar	$[-500, 0, 0]$
4) Forced motion along v-bar	$[-50, 0, 0]$
5) Out of plane inspection	$[-50, 0, 0]$
6) Fly-around matching rotation	—

Table 5.4: Phases of approach strategy.

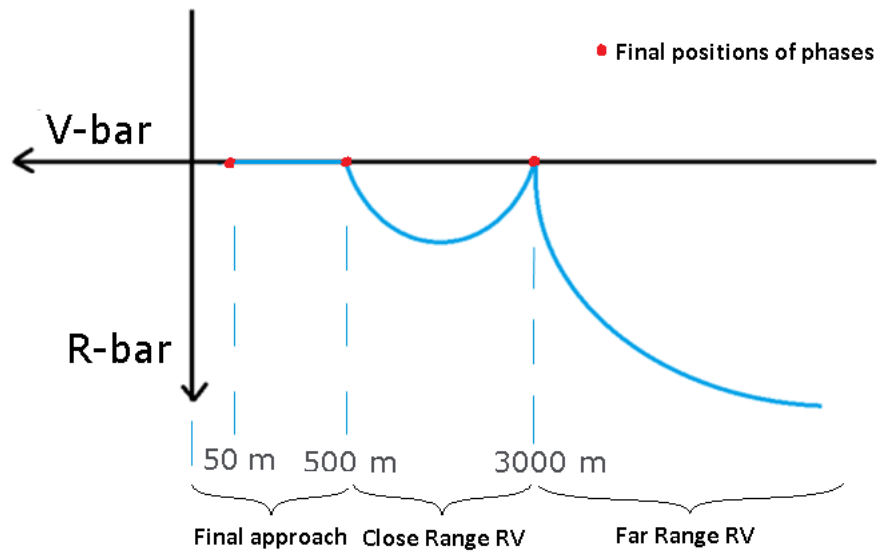


Figure 5.16: Phases of rendezvous approach. Relative position of the target (origin) and chaser (blue line).

In this simulation example, an analysis of the manoeuvres 2–5 are made. The Hohmann transfer and fly-around manoeuvres are not considered. The chaser is assumed to weigh 1500 kg. The main engine of the chaser is a 400 N thruster.

5.5.2 Relative Dynamics

For the relative dynamics of the two satellite bodies the linearised Hill Equations are used [8].

$$\begin{aligned} \ddot{x} - 2\omega\dot{z} &= \frac{1}{m_c}F_x \\ \ddot{y} + \omega^2y &= \frac{1}{m_c}F_y \\ \ddot{z} + 2\omega\dot{x} - 3\omega^2z &= \frac{1}{m_c}F_z \end{aligned} \quad (5.43)$$

where ω is the orbital rate of the target and m_c is the mass of the chaser satellite. The target is situated at the origin. The equations are assuming circular orbits and relatively close ranges in z-direction (because of the difference in orbital rates). The GAST block simulating the relative dynamics integrates the state space model of the above equations. The state space model is

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \end{aligned} \quad (5.44)$$

where

$$\begin{aligned} \mathbf{x} = \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \end{bmatrix}, \mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2\omega \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\omega^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -2\omega & 0 & 0 & 3\omega^2 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{m} \end{bmatrix} \\ \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{D} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{u} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \end{aligned} \quad (5.45)$$

5.5.3 Close Range Rendezvous

A reference trajectory for the close range rendezvous is made by assuming impulse forces, meaning instantaneous changes of radial velocity at start and end of the trajectory. The distance travelled along the x-axis when crossing it after half an orbital period is calculated as [8]

$$\Delta x = \frac{4}{\omega} \Delta v_z \quad (5.46)$$

Another impulse is fired when the desired position is reached to stop the motion. The Δv_z needed to travel 2500 m is calculated to be 0.625 m/s. The reference trajectory in the xz-plane is shown in Fig. 5.17.

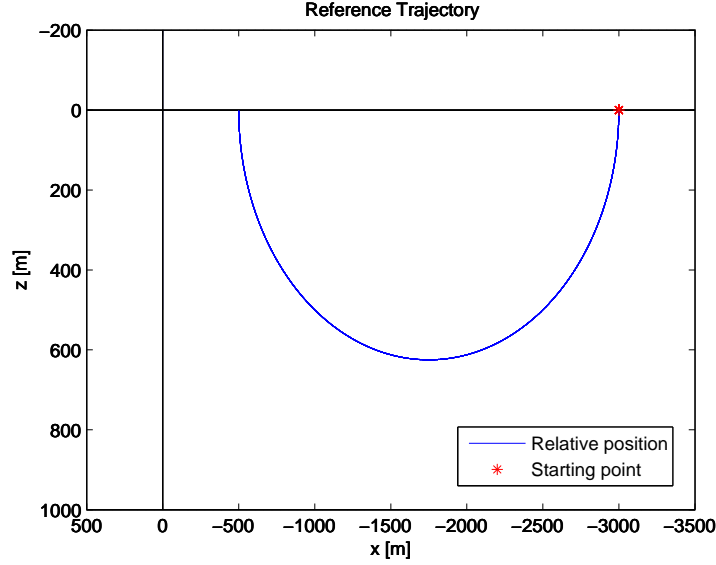


Figure 5.17: Reference trajectory with $\Delta v_z = 0.625$ m/s.

Since an impulse force cannot be achieved in practice, the time that the 400 N thruster has to fire to achieve the same Δv is calculated as

$$\Delta t = \frac{m \Delta v}{F} \quad (5.47)$$

In this case the thruster has to fire for 2.344 s at the start and end points of the trajectory. In this part of the simulation, the attitude of the chaser is not taken into account.

The control algorithm used for the station keeping is a simple PD-controller with $K_p = 150$ and $K_d = 1500$. The control force is limited to 10 N, having an ACS with 10 N thrusters in mind.

The fuel usage is calculated as

$$m_{fuel} = \int \frac{|F|}{I_{sp} g_0} dt \quad (5.48)$$

where the I_{sp} for the 400 N thruster is 318 s and for the 10 N thrusters 210 s.

The results of the simulation are shown in Fig. 5.18.

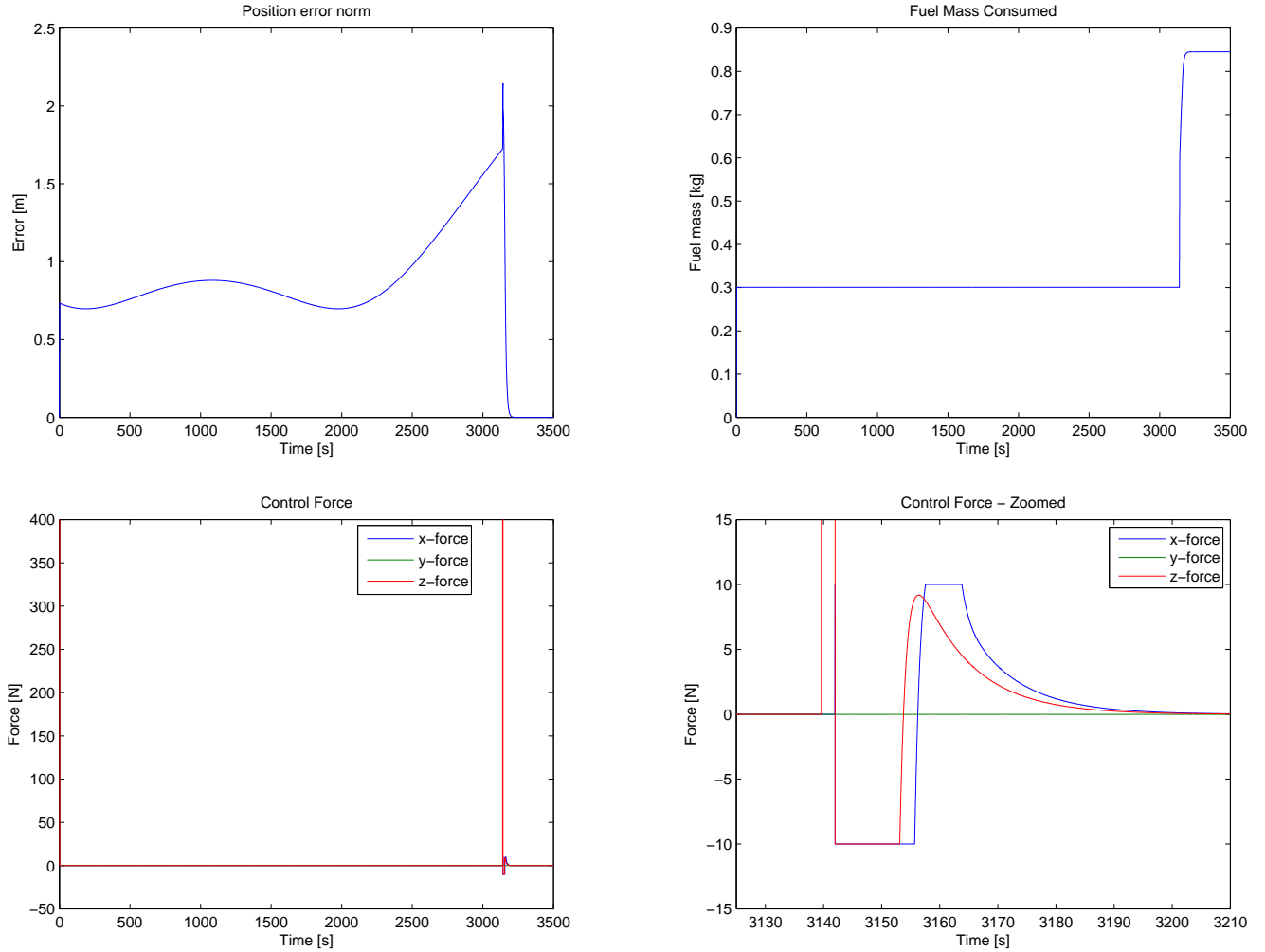


Figure 5.18: Results of GAST simulation of close range RV and station keeping.

The results show that the chaser is following the reference trajectory with less than 1.5 m error and settles to the desired target after about 3230 s with the use of the ACS. The final total fuel consumption is 0.75 kg.

5.5.4 Forced Motion

The forced motion is an approach from -500 to -50 m on v-bar. The reference trajectory is a straight line along v-bar with constant velocity $v = 0.5$ m/s. The control law used is a PD-controller with proportional gain 150 and derivative gain 1500. The gains are selected by experience while running the simulations. The resulting trajectory and the control force is shown in Fig. 5.19.

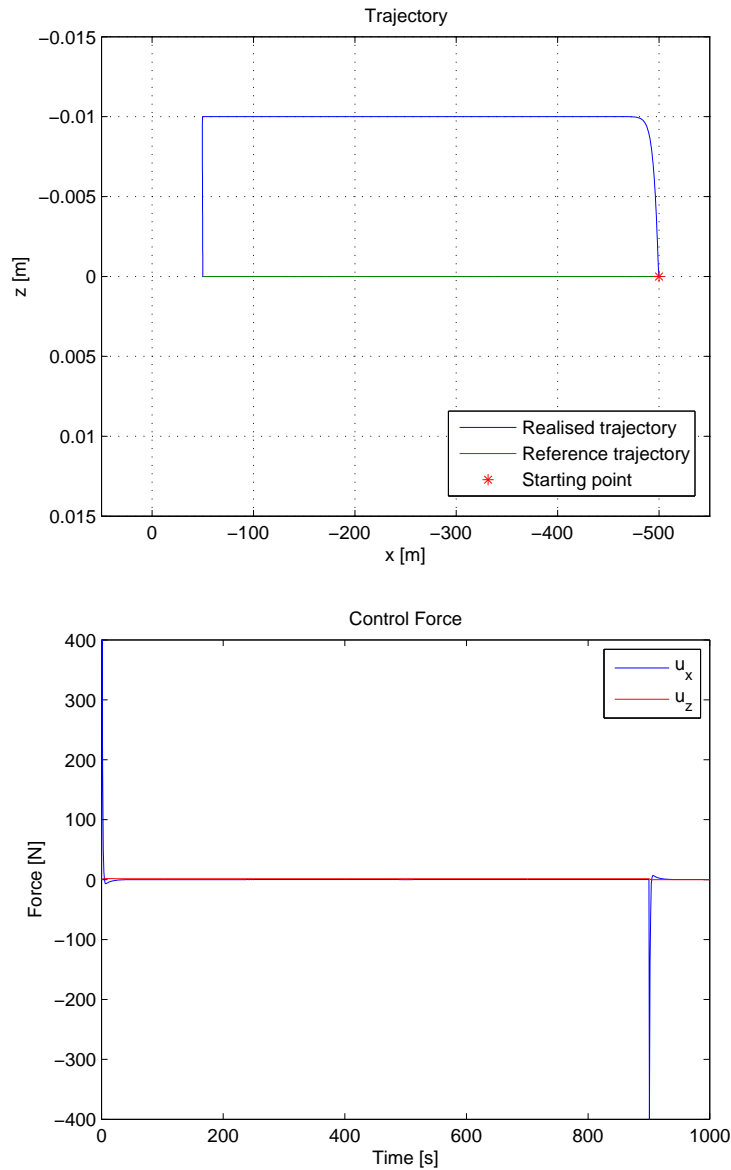


Figure 5.19: Results of GAST simulation of the forced motion approach.

The simulation shows that with this controller, the relative error during the motion is only 10 mm in z -direction. To achieve this trajectory, two boosts of the main engine are applied. The first one is 400 N in the positive x -direction at the beginning of the manoeuvre and the second is 400 N in the negative x -direction at 900 s. The control force u_z has a constant value of 1.5 N during the manoeuvre to keep the chaser from drifting in z -direction.

Trajectory safety must also be considered in rendezvous and docking missions. If the second boost cannot be applied it must be assured that the resulting trajectory does not provide a collision risk. If after 890 s, all forces are set to zero the trajectory for 5 orbits is as in Fig. 5.20.

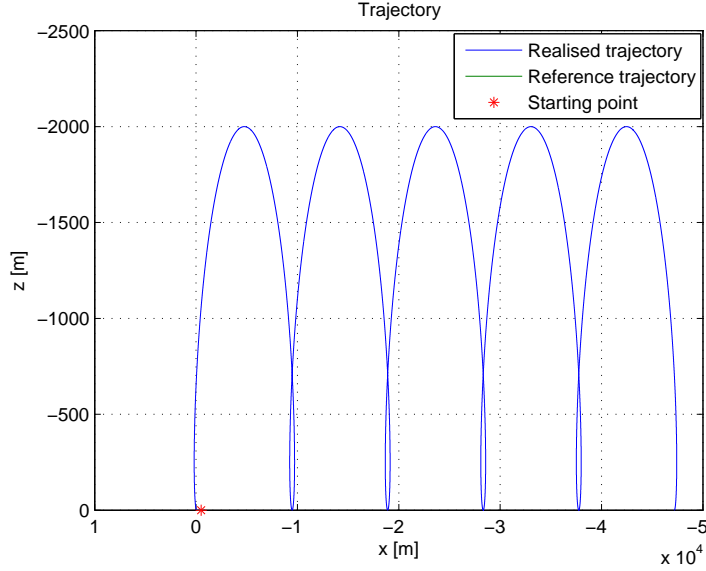


Figure 5.20: Trajectory resulting from loss of control force.

The chaser is moving further away from the target with every orbit, which is a safe path. However, the initial trajectory passes just 5 m above the target, which can cause a collision with the presence of disturbance forces. It is therefore advised to have an automatic collision avoidance manoeuvre which is applied in case of loss of control force or other unforeseen changes of the original plan.

5.5.5 Out-of-plane Inspection

A simulation of an out of plane (xy-plane) inspection with target attitude pointing is made. Assuming that the sensors of the chaser are on the positive x-side, the chaser should point the body x-axis towards the target. The desired attitude quaternion is calculated through a rotation of the angle between the initial body x-axis ($[1 \ 0 \ 0]$ in LVLH frame) and the negative relative position around the resulting vector of the cross product between the two vectors.

$$\begin{aligned}\phi &= \cos^{-1}([1, 0, 0] \cdot \frac{-\mathbf{x}}{|\mathbf{x}|}) \\ \mathbf{u} &= [1, 0, 0] \times \frac{-\mathbf{x}}{|\mathbf{x}|}\end{aligned}\tag{5.49}$$

The desired attitude quaternion can then be calculated as a rigid body rotation around a unit vector, as described in Chapter 2.

An impulsive velocity change of 0.266 m/s in the positive y-direction is assumed to generate the reference trajectory, which is a sinusoidal motion in the xy-plane. The

trajectory is then realised with a PD-controller ($K_p = 100$, $K_d = 1500$) and a limit of 10 N on the control force. The attitude is controlled using quaternion control with $K_p = K_d = \mathbf{I}$. The results from the simulation are presented in Fig. 5.21.

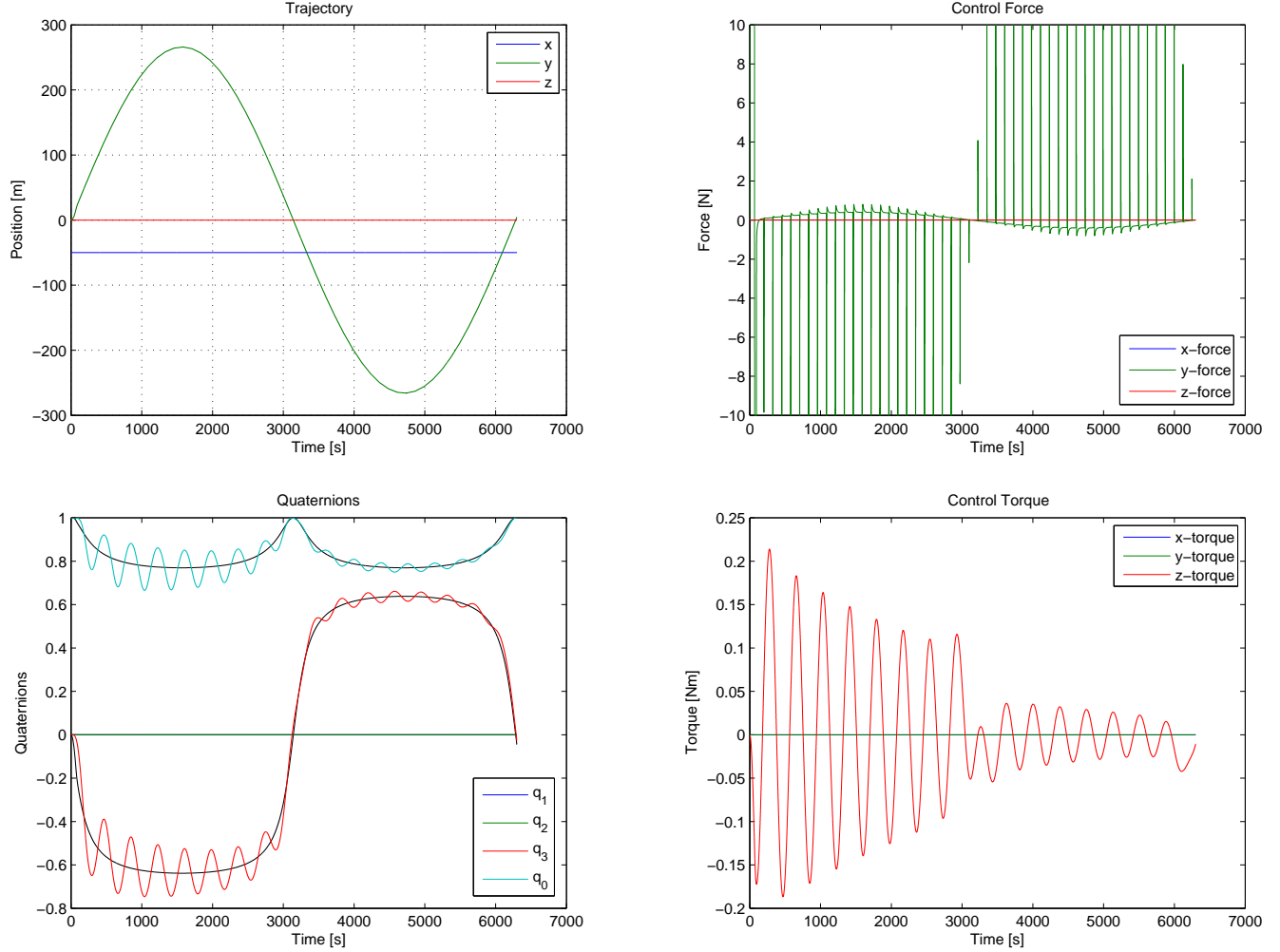


Figure 5.21: Results of GAST simulation of out-of-plane target inspection.

It is seen that the desired trajectory is followed, even with a control force limit of 10 N. The realised attitude quaternion follows the reference with some oscillations in the beginning, but settles after one orbit. Depending on the attitude pointing requirements, a more suitable quaternion control law has to be found.

Chapter 6

Conclusion

The aim of this project was to create a toolbox based in MATLAB/Simulink with the primary objective to facilitate the construction of rapid simulations for concept studies. The engineers of the Guidance, Navigation, and Control Systems Section of ESTEC will use the resulting GAST toolbox for simulations to support studies in the CDF and make preliminary designs of the AOCS subsystem and GNC algorithms. The simulation results can be used to motivate trade-offs, such as the level of autonomy or the number of sensors to be used in a particular mission.

The GAST toolbox contains about 100 element blocks, which can be used for customised simulations. The GAST simulation blocks are modular and can be assembled into different configurations to meet the needs of the user. The models were made simple and general to be able to use the same block to model different but similar components, such as sensors and actuators from different manufacturers. The sensors and actuators modelled in Chapter 4 show similar behaviours as could be expected from real sensor measurements and actuator outputs.

The simulations described in Chapter 5 were constructed using the GAST toolbox, predominantly using the blocks designed within the scope of this thesis. They show the toolbox's capability to produce fast and simple simulations, where the parameters can be changed easily to compare different design solutions. Simulations can be made arbitrarily complex using the existing GAST blocks and adding custom blocks and algorithms to simulate specific cases.

A continuation of the development of the GAST toolbox can be made by adding more simulation blocks according to the needs for CDF studies. The more blocks that are added, the more useful the toolbox will be in terms of the range and variety of simulations that are possible. The toolbox can be updated to support the GNC/AOCS systems development for future studies of innovative space missions.

Bibliography

- [1] BARRET, C. Spacecraft Flight Control System Design Selection Process for a Geostationary Communication Satellite. NASA Technical Paper 3289, Sept. 1992.
- [2] BERNER, R. Control moment gyro actuator for small satellite applications. Master's thesis, Department of Electrical & Electronic Engineering, University of Stellenbosch, South Africa, Apr. 2005.
- [3] EADS SODERN. VDM Videometer. EADS SODERN, 2004. Sensor Data Sheet.
- [4] EBERT, K., AND OESTERLIN, W. Dynamic yaw steering method for spacecrafts, Mar. 2009. Grant, US7510148 B2, <http://www.google.com/patents/US7510148>.
- [5] ESA, EADS SPACE TRANSPORTATION, SciSys. Aurora Programme - Planetary Ascent Vehicle - Executive Summary, 2005. ESA Contract Report.
- [6] EUROPEAN SPACE AGENCY, THALES ALENIA SPACE, GALILEO AVIONICA SPA, ALTA SPA. *Direct Thrust and Thrust Noise Measurements on the LISA Pathfinder Field Emission Thruster* (University of Michigan), 31st International Electric Propulsion Conference.
- [7] EXO CRUISER. Airplane's Body, Stability and Wind Axis, Sept. 2011. <http://dodlithr.blogspot.nl/2011/09/airplanes-stability-axis.html>.
- [8] FEHSE, W. *Automated Rendezvous and Docking of Spacecraft*. Cambridge University Press, 2003.
- [9] HONEYWELL INTERNATIONAL INC. MIMU - Miniature Inertial Measurement Unit. Honeywell International Inc., 2003. Sensor Data Sheet, <http://www51.honeywell.com/aero/common/documents/myaerospacecatalog-documents/MIMU.pdf>.
- [10] HONEYWELL INTERNATIONAL INC. Q-Flex QA-2000 Accelerometer. Honeywell International Inc., 2005. Sensor Data Sheet, <https://aero1.honeywell.com/inertsensor/docs/qa2000.pdf>.

- [11] JENA-OPTRONIK GMBH. Active Pixel Sensor ASTRO APS. Jena-Optronik GmbH. Sensor Data Sheet.
- [12] LITTON GUIDANCE AND CONTROL SYSTEMS. LN-200S Inertial Measurement Unit. Litton Guidance and Control Systems. Sensor Data Sheet.
- [13] PETKOV, P., AND SLAVOV, T. Stochastic modeling of mems inertial sensors. *Cybernetics and Information Technologies, Bulgarian Academy of Sciences* 10, 2 (2010), 31–40. http://www.cit.iit.bas.bg/CIT_2010/v10-2/31-40.pdf.
- [14] SELEX ES LTD. SiREUS The Silicon Rate Sensor. Selex ES Ltd., 2013. Sensor Data Sheet.
- [15] STOCKWELL, D. W. Angle Random Walk. Crossbow Technology, Inc. http://bullseye.xbow.com:81/Support/Support_pdf_files/AngleRandomWalkAppNote.pdf.
- [16] TEWARI, A. *Atmospheric and Space Flight Dynamics: Modeling and Simulation with MATLAB and Simulink*, 1st ed. No. ISBN 0817643737, 9780817643737. Birkhauser Basel, 2007.
- [17] THE EUROPEAN SPACE AGENCY. About Space Debris. http://www.esa.int/Our_Activities/Operations/Space_Debris/About_space_debris.
- [18] THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC. IEEE Standard for Inertial Sensor Terminology. The Institute of Electrical and Electronics Engineers, Inc., Nov. 2001. IEEE Std 528-2001.
- [19] TURNER, M. Actuator deadzone compensation: theoretical verification of an intuitive control strategy. *IEE Proc.-Control Theory Appl., IEE Proceedingsonline* 153, 1 (Jan. 2006), 59–68.
- [20] WOODMAN, O. J. An introduction to inertial navigation. Tech. Rep. UCAM-CL-TR-696, University of Cambridge, Computer Laboratory, Aug. 2007. <http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf>.