Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Bjørn Andreas Kristiansen

Nonlinear Attitude Control of a Pointing and Slewing Spacecraft

Master's thesis in Cybernetics and Robotics, MTTK
Supervisor: Jan Tommy Gravdahl
June 2019

NTNU
Norwegian University of
Science and Technology

Bjørn Andreas Kristiansen

Nonlinear Attitude Control of a Pointing and Slewing Spacecraft

Master's thesis in Cybernetics and Robotics, MTTK
Supervisor: Jan Tommy Gravdahl
Advisor: Mariusz Eivind Grøtte
June 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

NTNU
Norges teknisk-naturvitenskapelige
universitet

Fakultet for informasjonsteknologi,
matematikk og elektroteknikk
Institutt for teknisk kybernetikk

MSc thesis assignment

Name of the candidate: Bjørn Andreas Kristiansen
Subject: Engineering Cybernetics
Title: Nonlinear Attitude Control of a Pointing and Slewing Spacecraft

Background

A nonlinear dynamics & control model shall be developed for a 6U CubeSat carrying a pushbroom hyperspectral imager as payload and spacecraft bus developed by NanoAvionics. Scientific observations with the hyperspectral imager require requirements for attitude knowledge, controller settling time and attitude controller accuracy. The spacecraft will have active actuation with Reaction Wheels and Magnetorquers, where controllers and estimators will be developed in Matlab/Simulink to present precise and robust slewing and pointing maneuvers. The following tasks will be executed:

Tasks:

- Present a mathematical model for the satellite. The model should be suitable for ADCS design. Include disturbances and uncertainties (solar pressure, gravity gradient, magnetic field, atmospheric drag…) in the model and environment. Also include a dynamic model of the reaction wheels as well as uncertainties in their assembly.
- Design 1) LQR based and 2) Sliding mode, ordinary and higher order, attitude controllers for both pointing and a slewing maneuver
- Compare the controllers with PD-control with respect to robustness and performance.
- Test software-in-the-loop controller(s) and state observer(s) by simulation in Matlab/Simulink. Characterize results for absolute angle error, steady-state error and settling time.
- Include time delays in the model used for simulation.
- Apply state observer(s)/estimator(s) to a dynamical model, where these can be variants of sliding mode observers.
- Investigate the stability properties for the cascaded systems

To be handed in by: 3/6-2019

Jan Tommy Gravdahl

Jan Tommy Gravdahl
Professor, supervisor

Abstract

Orbital mechanics and attitude dynamics are modeled for a satellite actuated by reaction wheels and magnetorquers. A nonlinear satellite model is implemented in Matlab/Simulink. The controls used are conventional sliding mode control, the generalized super-twisting algorithm, linear-quadratic regulator, and proportional-derivative controller for both target pointing and slewing using the reaction wheels, as well as a B-dot tumbling and momentum dumping controllers using the magnetorquers. Control simulations are presented with and without a sliding mode observer. The simulation results are based on HYPSO, a 6U CubeSat mission developed at NTNU that will host a hyperspectral imager to detect and monitor ocean color along the Norwegian coast.

Sammendrag

Dynamikken i en satellitts bane og attitude modelleres for en satellitt med reaksjonshjul og magnetorquere som aktuatorer. Den ulineære modellen er implementert i Matlab/Simulink. Regulatorene som blir brukt er konvensjonell sliding mode control, generalized super-twisting algorithm, linear-quadratic regulator og proporsjonal-derivasjon-regulator med reaksjonshjulene for peking og rotasjon. Magnetiske torquere brukes for å regulere tumblingen med B-dot algoritmen, samt en regulator for å regulere drivmomentet i reaksjonshjulene. Systemet blir simulert både med og uten en sliding mode observer. Resultatene fra simuleringen basererer seg på HYPSO-satellitten ved NTNU, en 6U CubeSat som vil bruke et hyperspektralt kamera for å overvåke havfarge langs norskekysten.

Preface

This thesis is submitted as part of earning a Master of Science degree in Cybernetics and Robotics at the Department of Engineering Cybernetics at Norwegian University of Science and Technology (NTNU). Professor Jan Tommy Gravdahl was supervisor and PhD research fellow Mariusz Eivind Grøtte was advisor during work on this thesis.

The work on the master's thesis has been a continuation of the specialization project from the fall of 2018 where some of the background theory was developed, as well as most of the simulator. Some of the chapters and sections in this thesis are based on the work in this project:

- Section 2.2, 2.3, and 2.4 are mostly based on the work done in the specialization project, with the wheel frame being added to section 2.2 and the definition of quaternion error in section 2.3 being the notable exceptions.
- Chapter 3 contains similar sections on orbital mechanics and perturbations. The derivation of the attitude kinetics has changed.
- The B-dot algorithm was used for detumbling in the specialization project as well. The results have been reproduced in section 8.1. The mechanical constraints of the magnetorquers have been added in this master's thesis.

The thesis work has been part of the HYPSO project at NTNU. A special thanks to Prof. Jan Tommy Gravdahl, PhD research fellow Mariusz Eivind Grøtte, as well as PhD research fellow Ida-Louise Garmann Borlaug for their advice and valuable input during my time working on this thesis. Additionally, I would like to thank my colleagues, particularly Arnt Erik Stene, Tord Sjeggestad Bjørnsen, Morten Lie, and Monica Gilde for good discussions.

Bjørn Andreas Kristiansen
Trondheim, June 3, 2019

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

This work is related to the HYPSO mission, part of the MASSIVE project at the NTNU SmallSat lab (see appendix A). The HYPSO mission consists of a CubeSat mounted with a camera for ocean observation. The attitude determination and control system, ADCS, is the subsystem that is responsible for aiming the hyperspectral imager at desired target areas. Control is to be developed to meet the demands to attitude control on the HYPSO mission, such as accuracy in target pointing and slewing.

1.2 Related work

Attitude control means to control the orientation of a reference frame, generally attached to a vehicle or some other controllable entity, relative to another frame. There exists a lot of research on attitude control, for vehicles such as ships [1], aircrafts [2], or satellites [3].

Sliding mode control is a form of nonlinear control that is robust to modeling errors, external disturbances and uncertainties [4]. The sliding mode control is widely used in control literature ([5], [6]), also for attitude control problems ([7], [8]).

The problem of attitude control for a satellite is well studied ([9], [10], [11]). Sliding mode control for attitude control on satellites actuated with reaction wheels also exists ([12], [13]). Research on the attitude control problem for satellites also exists from NTNU, both with [14] and without [15] observers, often with attitude control based on magnetorquers ([16], [17]), as the earlier satellite projects at NTNU were actuated by magnetorquers only, such as NUTS ([18], [19]).

1.3 Contributions

The contributions in this thesis are in the in the field of control theory.

(a) A PD controller with gyroscopic compensation is used on simulations on a 6U CubeSat using magnetorquers with a momentum dumping algorithm.

(b) Sliding mode control, in particular the second order sliding mode control generalized super-twisting algorithm, is applied to the attitude control problem of a pointing and slewing satellite actuated by reaction wheels.

(c) A comparison of the response of the controllers based on simulated results is presented.

1.4 Thesis outline

In chapter 2, the theory used in the rest of the thesis will be introduced. Relevant background theory concerning notation, coordinate frames and attitude representation, is introduced.

In chapter 3, the dynamics of the system is developed in a mathematical model. This includes the orbital mechanics, attitude dynamics, and perturbations that affect the system.

In chapter 4, the relevant controller and observer theory is included.

Chapter 5 contains the control laws designed for the attitude control problem. Four different controllers, namely PID, LQR, sliding mode control (SMC), and a second order

SMC, the generalized super-twisting algorithm, are included for both target pointing and the slewing maneuver. For the magnetorquers, the B-dot algorithm for detumbling and an algorithm for momentum dumping is shown.

In chapter 6 the observers are designed. Two complete observers will be presented: A sliding mode observer (SMO), and a higher order sliding mode observer (HOSMO). An extra SMO for angular velocity is also presented.

In chapter 7 the model created in Simulink for simulation is presented.

Chapter 8 includes results and the discussion of the results. Results for detumbling, pointing, and slewing is presented, both with and without a time delay present in the simulation, with and without observer. The chapter also includes results for the different observers, the B-dot detumbling algorithm, the momentum dumping algorithm, and how robust the controllers are to perturbations in the torque distribution matrix.

In chapter 9 the findings are summarized, and description of future work is included.

Chapter 2

Background theory

2.1 Notation

A vector is denoted as $\mathbf{x} \in \mathbb{R}^n$, where n is the dimension of the vector. Any lowercase, boldface symbol represents a vector, whereas a roman symbol represents a scalar. A matrix with m rows and n columns is represented with uppercase, boldface symbols such as $\mathrm{A} \in \mathbb{R}^{mxn}$. Vectors and matrices may be represented in coordinate frames using the notation $\mathbf{x}^a$, where $\mathbf{x}$ is represented in frame {a}. A similar notation is used for differentiation with respect to different frames. For example, differentiating $\mathbf{x}^a$ with respect to time in the {a} frame is shown using Leibniz's notation as $\frac{{}^a d}{dt}\mathbf{x}^a$. Angular velocity is defined between two coordinate frames, where $\boldsymbol{\omega}_{ab}^c$ is the angular velocity between frames {a} and {b} represented in frame {c}. The cross product between vectors in three dimensions such as $\mathbf{x} \times \mathbf{y}$ can be represented as $\mathrm{S}(\mathbf{x})\mathbf{y}$, where $\mathrm{S}(\mathbf{x})$ is a skew symmetric matrix formed by a vector $\mathbf{x}$, given by

$$\mathrm{S}(\mathbf{x}) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}. \tag{2.1}$$

2.2 Coordinate frames

2.2.1 Earth-centered inertial frame

The Earth-centered inertial frame, ECI, is denoted by {i}. Its origin is at the center of the Earth, with the first axis pointed towards the vernal equinox, the third axis points through the North Pole, and the second axis completes the right handed coordinate frame.

2.2.2 Earth-centered, Earth-fixed frame

The Earth-centered, Earth-fixed frame, ECEF, is denoted by {e}. Its origin is, like the ECI frame, at the center of the Earth, with the third axis pointed out of the North Pole. The first and second axes spans the equatorial plane, with the first axis pointing through the prime meridian. ECEF follows the Earth's rotation, having the angular velocity about the third axis $\omega_{ie} = 7.2921 \cdot 10^{-5}$ [20, p. 16].

2.2.3 Longitude and latitude

For a spherical body, a point on the surface can be represented by the angle from the pole, the angle from a position at the equator, and the distance from the center of the sphere. A three-dimensional Cartesian coordinate position vector can therefore be represented using a coordinate system like this. For the Earth, longitude, latitude, and distance is used to describe these three coordinates. Longitude is the angle that moves from east to west, from -180° in the east, 0° at the prime meridian, and 180° in the west. Latitude is 0° at the Equator, extending to 90° at either pole. The distance from the center of the Earth completes the coordinate frame.

2.2.4 Perifocal frame

The perifocal frame, or PQW, is an inertial frame centered in the Earth's center. The perifocal frame is useful because it defines a frame that is fixed to the orbit, not any of the moving parts of the modeled system, such as the Earth or the satellite. The first axis, p, is pointing toward the periapsis, which is the shortest radius in orbit. The second axis, q, has a true anomaly of 90° from p, making it orthogonal to the first axis. The third axis, w, is the cross product of the p and q axes.

By using the perifocal frames own set of parameters, called orbital parameters, one can uniquely desribe an orbit. The orbital parameters are defined as

- a - Semi-major axis
  - Half of the major axis of the ellipsis created by the orbit plane.
- e - eccentricity
  - A parameter that describes how unlike the orbit is a circle. When e is 0, the orbit is completely circular. Eccentricity between 0 and 1 gives an elliptical orbit.
- i, Ω, ω - inclination, longitude of the ascending node, and argument of periapsis
  - three angles defining the orbit plane's attitude in space relative to the equatorial plane.
- θ - true anomaly
  - Describes the position of the orbiting body in the orbit.

2.2.5 Orbit Frame

The orbit frame, denoted {o}, is the first of two frames centered in the satellite's center. Its purpose is to have a frame that follows the path of the satellite as it orbits the Earth, and therefore does not have any involvement when it comes to the satellite's attitude. The first axis is pointing in the direction of the velocity of the satellite, while the third axis is always pointing toward the center of the Earth. The second axis completes the right handed coordinate system. This is also known as Vehicle Velocity, Local Horizontal, or VVLH, frame.

$$\begin{aligned}
\hat{\mathbf{z}}^o &= -\frac{\mathbf{r}^i}{||\mathbf{r}^i||_2} \\
\hat{\mathbf{x}}^o &= \frac{\mathbf{v}^i}{||\mathbf{v}^i||_2} \\
\hat{\mathbf{y}}^o &= \frac{\hat{\mathbf{z}}^o \times \hat{\mathbf{x}}^o}{||\hat{\mathbf{z}}^o \times \hat{\mathbf{x}}^o||_2}
\end{aligned} \tag{2.2}$$

where $\mathbf{r}^i, \mathbf{v}^i$ are the position and velocity vectors, respectively, in the ECI frame.

2.2.6 Body frame

The body frame is, like the orbit frame, centered in the satellite's center. Denoted {b}, it follows the satellite as it rotates. It is therefore useful to describe attitude as a rotation between the body and orbit frame.

2.2.7 Wheel frame

The wheel frame is not a right handed reference frame. This is a frame has each axis fixed to the rotational axis of each reaction wheel. The wheel frame is useful as a tool to separate vectors that are represented in the body frame and the vectors that represents the behavior of all the reaction wheels. Due to the nature of reaction wheels, the wheel frame and the body frame will not rotate relative to each other, and the rotation between vectors in either of the frames can be described by a constant matrix A, called the torque distribution matrix. Vectors in the wheel frame have the same length as the number of reaction wheel in the satellite, and are denoted by a $\{w\}$.

2.2.8 Transport theorem

The differentiation of a vector u that is represented in the rotating reference frame b in frame a is given by the transport theorem [21, p. 243]

$$\frac{{}^a d}{dt}\mathbf{u} = \frac{{}^b d}{dt}\mathbf{u} + \omega_{ab}^b \times \mathbf{u} \tag{2.3}$$

2.3 Attitude representations

2.3.1 Euler angles

Euler angles are three angles used to describe an attitude, one about each axis. To specify a specific transformation, three rotations are required. Because only three parameters are used to define an attitude, there exists a singularity in the representation. The order of rotations for the Euler angles decides where the singularity in the representation appears. For some applications, choosing the correct order of rotations would be enough to avoid singularities. For a satellite, it is not realistic to assume that a given attitude will be avoided.

2.3.2 Quaternions

The quaternion between coordinate frame a and coordinate frame b is defined as [21, p. 233]

$$\mathbf{q}_{ab} = \begin{bmatrix} \eta \\ \boldsymbol{\epsilon} \end{bmatrix} = \begin{bmatrix} cos(\frac{\theta}{2}) \\ \mathbf{k}\ sin(\frac{\theta}{2}) \end{bmatrix} \tag{2.4}$$

where the angle θ and the vector $\mathbf{k}$ are the angle-axis parameters.

The version of quaternions used in this thesis are also referred to as unit quaternions, as they are being constrained by $\mathbf{q}^\intercal \mathbf{q} = 1 \Rightarrow \eta^2 + \boldsymbol{\epsilon}^\intercal \boldsymbol{\epsilon} = 1$.

The quaternion product is defined as

$$\mathbf{q} = \mathbf{q}_1 \otimes \mathbf{q}_2 = \begin{bmatrix} \eta_1 & -\boldsymbol{\epsilon}_1^\intercal \\ \boldsymbol{\epsilon}_1 & \eta_1 \mathbf{I}_{3x3} + \mathbf{S}(\boldsymbol{\epsilon}_1) \end{bmatrix} \mathbf{q}_2 \tag{2.5}$$

The inverse of a quaternion is defined as

$$\mathbf{q} = \begin{bmatrix} \eta \\ \boldsymbol{\epsilon} \end{bmatrix} \Rightarrow \mathbf{q}^{-1} = \begin{bmatrix} \eta \\ -\boldsymbol{\epsilon} \end{bmatrix} \tag{2.6}$$

The quaternion error is defined using the quaternion product from equation 2.5 as

$$\mathbf{q}_e = \mathbf{q}_d^{-1} \otimes \mathbf{q}_{val} \tag{2.7}$$

where $\mathbf{q}_d$ is the desired quaternion, while $\mathbf{q}_{val}$ is current value. The quaternion has reached its desired value when the quaternion error is

$$\mathbf{q}_e = \begin{bmatrix} \pm 1 \\ \mathbf{0}_{3x1} \end{bmatrix} \tag{2.8}$$

2.3.3 Rotations

A rotation is a coordinate transformation. A rotation matrix from {b} to {a} can be constructed by [21, p. 219-220]

$$\mathbf{R}_b^a = \{\vec{a} \cdot \vec{b}\} \tag{2.9}$$

where $\vec{a}$ and $\vec{b}$ are the orthogonal unit vectors that create the right handed coordinate frame {a} and {b}, respectively. Alternatively, it can be done by

$$\mathbf{R}_b^a = (\ \mathbf{b}_1^a\ \mathbf{b}_2^a\ \mathbf{b}_3^a\ ) \tag{2.10}$$

where $\mathbf{b}_i^a$ is the ith unit vector of the {b} frame represented in {a} frame.

A rotation matrix constructed this way is called a direction cosine matrix, DCM. It uses nine parameters to construct a rotation, is for that reason overparameterized and will not have any singularities.

Being a part of SO(3), the rotation matrix has the following properties[21, p. 219-221]

$$\text{SO(3)} = \{\mathbf{R} | \mathbf{R} \in R^{3x3}, \mathbf{R}^{\mathrm{T}}\mathbf{R} = \mathbf{I}_{3x3}, \det(\mathbf{R}) = 1\} \tag{2.11a}$$

$$\mathbf{R}_a^b = (\mathbf{R}_b^a)^{-1} = (\mathbf{R}_b^a)^T \tag{2.11b}$$

$$\mathbf{R}_a^d = \mathbf{R}_c^d \mathbf{R}_b^c \mathbf{R}_a^b \tag{2.11c}$$

Matrices to represent a full rotation can also be represented by Euler angles, section 2.3.1, and quaternions, section 2.3.2. For the Euler angles, a full rotation is represented by the product of three simple rotations. A simple rotation is a rotation by an angle about an axis. Using θ, ϕ and ψ as the angles of rotation about the 1st, 2nd and 3rd axis, respectively, gives

$$\mathbf{R}_b^a = \mathbf{R}_3(\psi)\mathbf{R}_2(\theta)\mathbf{R}_1(\phi) \tag{2.12}$$

Using quaternions, where η is the scalar part and $\boldsymbol{\epsilon}$ is the vector part, the rotation matrix can be constructed by using the formula [20, p. 28]

$$\mathbf{R}_{\eta,\boldsymbol{\epsilon}} = \mathbf{I}_{3x3} + 2\eta \mathrm{S}(\boldsymbol{\epsilon}) + 2\mathrm{S}^2(\boldsymbol{\epsilon}) \tag{2.13}$$

The time derivative of a rotation matrix is given by [21, p. 240]

$$\dot{\mathbf{R}}_b^a = \mathrm{S}(\boldsymbol{\omega}_{ab}^a)\mathbf{R}_b^a \tag{2.14a}$$

$$\dot{\mathbf{R}}_b^a = \mathbf{R}_b^a \mathrm{S}(\boldsymbol{\omega}_{ab}^b) \tag{2.14b}$$

2.4 Angular velocity

Angular velocity shows the rotational velocity relative to some point, i.e. the origin of some frame. Addition or subtraction can be applied to angular velocities in order to find different angular velocity properties

$$\boldsymbol{\omega}_{ac}^b = \boldsymbol{\omega}_{ab}^b + \boldsymbol{\omega}_{bd}^b - \boldsymbol{\omega}_{cd}^b \tag{2.15}$$

For this thesis, $\boldsymbol{\omega}_{ob}^b$ is an interesting property, as it shows the satellite's angular velocity relative to the orbit it is in. $\boldsymbol{\omega}_{ib}^b$ can be measured by an IMU, so the function giving the desired angular velocity is

$$\boldsymbol{\omega}_{ob}^b = \boldsymbol{\omega}_{ib}^b - \boldsymbol{\omega}_{io}^b = \boldsymbol{\omega}_{ib}^b - \mathbf{R}_o^b \boldsymbol{\omega}_{io}^o \tag{2.16}$$

where $\boldsymbol{\omega}_{io}^{o}$ is given by [22]

$$\boldsymbol{\omega}_{io}^{o} = \mathbf{R}_{i}^{o}\frac{\mathrm{S}(\mathbf{r}^{i})\mathbf{v}^{i}}{(\mathbf{r}^{i})^{\mathsf{T}}\mathbf{r}^{i}} \tag{2.17}$$

2.5 Relative degree

The definitions in this section are from [23]. The Lie derivative of $\mathbf{h}(\mathbf{x})$ along with respect to $\mathbf{f}(\mathbf{x})$ is defined as

$$L_{f}\mathbf{h}(\mathbf{x}) = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}) \tag{2.18}$$

Taking the Lie derivative several times is defined as

$$L_{g}L_{f}^{k}\mathbf{h}(\mathbf{x}) = \frac{\partial(L_{f}^{k}\mathbf{h})}{\partial \mathbf{x}}\mathbf{g}(\mathbf{x}) \tag{2.19}$$

For a system on the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} \tag{2.20a}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) \tag{2.20b}$$

the derivative is defined as

$$\dot{\mathbf{y}} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}(\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}) = L_{f}\mathbf{h}(\mathbf{x}) + L_{g}\mathbf{h}(\mathbf{x})\mathbf{u} \tag{2.21}$$

The relative degree ρ of a system is defined as the number of times the output has to be differentiated before the input any element of the input vector $\mathbf{u}$ appears. In other words, the ρth derivative of the output $\mathbf{y}$ is given by

$$\mathbf{y}^{(\rho)} = \frac{\partial(L_{f}^{\rho-1}\mathbf{h})}{\partial \mathbf{x}}(\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}) = L_{f}^{\rho}\mathbf{h}(\mathbf{x}) + L_{g}L_{f}^{\rho-1}\mathbf{h}(\mathbf{x})\mathbf{u} \tag{2.22}$$

where $L_{g}L_{f}^{i-1}\mathbf{h}(\mathbf{x}) = 0, i = 1, 2, \ldots, \rho - 1;\ L_{g}L_{f}^{\rho-1}\mathbf{h}(\mathbf{x}) \neq 0$

The vector relative degree is the relative degree when the chosen output is a vector.

$$\rho = \{\rho_1, \rho_2, \rho_3, \ldots\} \tag{2.23}$$

The total relative degree is defined as the sum of the elements in the vector relative degree,[24]

$$\rho_t = \Sigma \rho_i \tag{2.24}$$

Chapter 3

Mathematical modeling

3.1 Orbital Mechanics

Newton's law of universal gravitation is given by [25, p. 423]

$$\mathbf{f} = \frac{G m_1 m_2 \mathbf{r}}{||\mathbf{r}||^3} \tag{3.1}$$

where m_1 and m_2 are the masses of the object and G is the gravitational constant. In developing the orbital mechanics of the system, several assumptions will be made.

- This is a two-body problem, meaning that the objects (primarily) interact with each other.
- The objects can be modeled as point masses.
- The Earth is spherical.
- The mass of the satellite is a lot smaller than the Earth's mass.

Rewriting the equation by dividing both sides of the equation by the mass of the satellite gives

$$\mathbf{a} = \frac{\mathbf{f}}{m_{total}} = \frac{\mu \mathbf{r}}{||\mathbf{r}||^3} \tag{3.2}$$

where μ is the standard gravitational parameter for the Earth, given as the product of the gravitational constant G and the Earth's mass.

Using Newton's second law, $\frac{{}^i d}{dt}(m\mathbf{v}) = \mathbf{f}$ with $\dot{\mathbf{r}} = \mathbf{v}$, the equation describing the orbital motion of the satellite is found to be

$$\frac{{}^i d}{dt}\mathbf{v}^i = \mu \frac{\mathbf{r}^i}{||\mathbf{r}^i||^3} + \frac{\mathbf{f}^i_{perturbations}}{m_{total}} \tag{3.3}$$

The superscript i denotes a vector represented in the ECI frame, which is inertial. By representing the position and velocity vectors in an inertial reference frame, Newton's second law may be applied directly.

The force from the gravitational pull from the Earth is the largest force acting on the satellite and is for this reason included in equation 3.3. The remaining forces make up the $\mathbf{f}^i_{perturbations}$ vector. The perturbations are discussed and modeled in section 3.5.

3.2 Kinematic Equation

The kinematic equation, showing how the attitude changes over time, is given by [20]

$$\dot{\mathbf{q}}_{ob} = \begin{bmatrix} \dot{\eta}_{ob} \\ \dot{\boldsymbol{\epsilon}}_{ob} \end{bmatrix} = \mathrm{T}(\mathbf{q}_{ob})\boldsymbol{\omega}^b_{ob} = \frac{1}{2}\begin{bmatrix} -\boldsymbol{\epsilon}^{\mathsf{T}}_{ob} \\ \eta_{ob}\mathbf{I}_{3x3} + \mathrm{S}(\boldsymbol{\epsilon}_{ob}) \end{bmatrix}\boldsymbol{\omega}^b_{ob} = \begin{bmatrix} -\frac{1}{2}\boldsymbol{\epsilon}^{\mathsf{T}}_{ob} \\ \mathrm{U}(\mathbf{q}_{ob}) \end{bmatrix}\boldsymbol{\omega}^b_{ob} \tag{3.4}$$

Remark. *The quaternion in 3.4 is for the attitude between the orbit and body frame. The same kinematic equation would hold for different quaternions.*

Note that the determinant of $2\mathrm{U}(\boldsymbol{\epsilon}_{ob})$,

$$
\begin{aligned}
det(2\mathbf{U}(\mathbf{q})) = det(\eta \mathbf{I}_{3\text{x}3} + \mathbf{S}(\boldsymbol{\epsilon})) = det(\begin{bmatrix} \eta & -\epsilon_3 & \epsilon_2 \\ \epsilon_3 & \eta & -\epsilon_1 \\ -\epsilon_2 & \epsilon_1 & \eta \end{bmatrix}) \\
= \eta(\eta^2 - (\epsilon_1^2)) - (-\epsilon_3)(\eta\epsilon_3 - \epsilon_1\epsilon_2) + \epsilon_2(\epsilon_3\epsilon_1 - (-\eta\epsilon_2)) \\
= \eta^3 + \eta(\epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2)
\end{aligned}
\tag{3.5}
$$

is singular when η is zero.

Another property of note is [26]

$$
\mathbf{T}^\mathsf{T}(\mathbf{q})\mathbf{T}(\mathbf{q}) = \frac{1}{4}\mathbf{I}_{3x3} \tag{3.6}
$$

3.3 Kinetic Equation

The angular momentum of the satellite given in body frame is [27]

$$
\mathbf{H}_s^b = \mathbf{J}\boldsymbol{\omega}_{ib}^b + \mathbf{A}\mathbf{J}_w\boldsymbol{\omega}_{bw}^w \tag{3.7}
$$

where $\mathbf{J}$ is the inertia of the spacecraft including the reaction wheels, $\mathbf{J}_w$ is a matrix with the inertia of each reaction wheel on the diagonal entries. $\mathbf{J}_s$ denotes the inertia of spacecraft without the reaction wheels. It is defined as $\mathbf{J}_s = \mathbf{J} - \mathbf{A}\mathbf{J}_w\mathbf{A}^+$, where $\mathbf{A}$ is the torque distrubution matrix and $\mathbf{A}^+$ pseudoinverse of the is the $\mathbf{A}$ matrix, which may be defined as the Moore-Penrose inverse [28].

By Euler's second axiom, the kinetic differential equation is

$$
\frac{^i d}{dt}(\mathbf{H}_s^b) = \mathbf{J}\frac{^b d}{dt}\boldsymbol{\omega}_{ib}^b + \mathbf{A}\mathbf{J}_w\frac{^w d}{dt}\boldsymbol{\omega}_{bw}^w + \mathbf{S}(\boldsymbol{\omega}_{ib}^b)\mathbf{H}_s^b = \boldsymbol{\tau}_{mtq}^b + \boldsymbol{\tau}_e^b \tag{3.8}
$$

where the fact that there is no rotation between the body ({b}) and wheel ({w}) frames has been used, as well that the torque distribution matrix $\mathbf{A}$ is constant in both frames. $\boldsymbol{\tau}_{mtq}^b$ and $\boldsymbol{\tau}_e^b$ are the torques from the magnetorquers and the sum of the external perturbing torques, respectively.

3.4 Reaction wheel dynamics

The angular momentum of the reaction wheels given in the body frame, is [27]

$$\mathbf{H}_w^b = \mathbf{A}\mathbf{J}_w\mathbf{A}^+(\boldsymbol{\omega}_{ib}^b + \mathbf{A}\boldsymbol{\omega}_{bw}^w) = \mathbf{A}\mathbf{J}_w\boldsymbol{\omega}_{iw}^w = \mathbf{A}\mathbf{J}_w\mathbf{A}^+\boldsymbol{\omega}_{iw}^b \tag{3.9}$$

The differential equation for the reaction wheels is found by using Euler's second axiom

$$\begin{aligned}\frac{{}^i d}{dt}(\mathbf{H}_w^b) = \frac{{}^b d}{dt}(\mathbf{A}\mathbf{J}_w\mathbf{A}^+\boldsymbol{\omega}_{iw}^b) + \mathbf{S}(\boldsymbol{\omega}_{iw}^b)\mathbf{H}_w^b &= \mathbf{A}\mathbf{J}_w\mathbf{A}^+\frac{{}^b d}{dt}\boldsymbol{\omega}_{iw}^b + \mathbf{S}(\boldsymbol{\omega}_{iw}^b)(\mathbf{A}\mathbf{J}_w\mathbf{A}^+\boldsymbol{\omega}_{iw}^b)\\ &= \mathbf{A}\mathbf{J}_w\mathbf{A}^+\frac{{}^b d}{dt}\boldsymbol{\omega}_{iw}^b + \mathbf{S}(\mathbf{A}\boldsymbol{\omega}_{iw}^w)(\mathbf{A}\mathbf{J}_w\boldsymbol{\omega}_{iw}^w) = \mathbf{A}\mathbf{J}_w\mathbf{A}^+\frac{{}^b d}{dt}\boldsymbol{\omega}_{iw}^b\\ = \mathbf{A}\mathbf{J}_w\mathbf{A}^+(\frac{{}^b d}{dt}\boldsymbol{\omega}_{ib}^b + \frac{{}^b d}{dt}\boldsymbol{\omega}_{bw}^b) &= \mathbf{A}\mathbf{J}_w\mathbf{A}^+(\frac{{}^b d}{dt}\boldsymbol{\omega}_{ib}^b + \mathbf{A}\frac{{}^w d}{dt}\boldsymbol{\omega}_{bw}^w) = \boldsymbol{\tau}_u^b\end{aligned} \tag{3.10}$$

Several assumptions were taken in developing equation 3.10. Among those are that the body and the wheel frames do not rotate relative to each other. Due to this assumption, no gyroscopic term will appear when differentiating a vector represented in the body frame in the wheel frame, or the other way around. All that is needed is a rotation matrix between the frames, a role the torque distribution matrix **A** has here. **A** and $\mathbf{J_w}$ are given in the wheel frame. Since the wheel frame and the body frame has no relative rotation the same matrices are constant in the body frame as well.

On the second line of equation 3.10, the gyroscopic term $\mathbf{S}(\mathbf{A}\boldsymbol{\omega}_{iw}^w)(\mathbf{A}\mathbf{J}_w\boldsymbol{\omega}_{iw}^w)$ has been cancelled. This is based on an assumption about the physical characteristics of the reaction wheels. $\mathbf{J_w}$ is an $\mathbb{R}^{nxn}$ matrix with the inertia of each reaction wheel about the rotating axis on the diagonal [27]. If the inertia of every reaction wheel is identical, the cross-product term disappears since

$$\begin{aligned}\mathbf{S}(\mathbf{A}\boldsymbol{\omega}_{iw}^w)(\mathbf{A}\mathbf{J}_w\boldsymbol{\omega}_{iw}^w) &= \mathbf{S}(\mathbf{A}\boldsymbol{\omega}_{iw}^w)(\mathbf{A}\sigma\mathbf{I}_{nxn}\boldsymbol{\omega}_{iw}^w) = \mathbf{S}(\mathbf{A}\boldsymbol{\omega}_{iw}^w)(\sigma\mathbf{A}\boldsymbol{\omega}_{iw}^w)\\ &= \sigma\mathbf{S}(\mathbf{A}\boldsymbol{\omega}_{iw}^w)(\mathbf{A}\boldsymbol{\omega}_{iw}^w) = \sigma\mathbf{S}(\boldsymbol{\omega}_{iw}^b)(\boldsymbol{\omega}_{iw}^b) = \mathbf{0}\end{aligned} \tag{3.11}$$

where the property $\mathrm{S}(\mathbf{x})\mathbf{x} = 0$ has been used. In this thesis the inertia of the reaction wheels are assumed to be identical, and therefore the cross-product term will not be included in the reaction wheel dynamics.

$\boldsymbol{\tau}_u^b$ is the torque input for the reaction wheels given in the body frame. The torque input in the wheel frame is $\boldsymbol{\tau}_u^b$ distributed by the torque distribution matrix, $\mathbf{A}$, or $\boldsymbol{\tau}_u^w = \mathbf{A}^+\boldsymbol{\tau}_u^b$.

3.5 Perturbations

Perturbations are any force or torque that affects the satellite other than the gravitational force from the Earth, defined in section 3.1. These perturbations sum up to the torques denoted as $\boldsymbol{\tau}_e^b$ in equation 3.8 and the forces denoted as $\mathbf{f}^i_{perturbations}$ in equation 3.3. It should be noted that not all external forces and torques are included, such as lunar gravitation, tidal waves, and the geodesy of the Earth. With respect to perturbations, the perturbations will be modeled as worst-case scenarios. If the controller is able to handle the worst-case scenario, it would be able to handle less severe scenarios as well.

3.5.1 Gravity gradient

Because the satellite is not a point mass, the gravity from the Earth will apply a torque to the satellite. The torque from the gravity gradient is given by [29]

$$\boldsymbol{\tau}_g^b = 3(\frac{\mu}{||\mathbf{r}^i||^3})\mathrm{S}(\mathbf{c}_3)\mathbf{J}\mathbf{c}_3 \tag{3.12}$$

where $\mathbf{c}_3$ is the third column of the rotation matrix from the orbit frame to the body frame.

3.5.2 Solar radiation pressure

The sun emits radiation, which exerts pressure upon bodies moving in the Solar System. The force is given by [17]

$$\mathbf{f}^b_{SRP} = \frac{G_{SRP}}{c}\frac{A}{2}(1+\eta)cos(\alpha)\begin{bmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\\ 0\end{bmatrix} \tag{3.13}$$

where

- G_{SRP} is the solar constant, 1367 W/m^2.
- c is the speed of light in vacuum, $3 \cdot 10^8 m/s$.
- A is the area the solar radiation is hitting.
- η is the emittance, set to 0.2.
- α defines the angle of the sunlight.

The force from the solar radiation pressure is dependent on the area the radiation is hitting. In the simulation used in this thesis, a worst-case version of the solar radiation pressure is used. For this reason, the angle α is set to 0° and the area A is set to area of the face of the satellite with the largest surface area. The vector at end denotes the direction of the force. Together with the vector specifying the direction of the force, this maximizes the force from the solar radiation pressure. There is also radiation due to Earth albedo and black body radiation. These effects have not been explicitly modeled in this thesis.

Remark. *The direction of the solar radiation pressure is constant. This is a simplification that removes the need for a model of the Sun's position. However, it does introduce an unphysical element to the model: When the satellite rotates, the source of the solar radiation pressure, the Sun, rotates with it.*

The torque is given by

$$\boldsymbol{\tau}^b_{SRP} = \mathbf{r} \times \mathbf{f}_{SRP} = (\mathbf{x}_{CP} - \mathbf{x}_{CG}) \times \mathbf{f}_{SRP} \tag{3.14}$$

where $\mathbf{x}_{CP}$ and $\mathbf{x}_{CG}$ are the position vectors to the center of pressure and the center of gravity, respectively. For this purpose, the centers are taken to be at each of the sides of the largest face of the satellite. It should be noted that this is not a reasonable to assume that the center of gravity would be in any of those points, as a placement that far outside of the geometric center violates the CubeSat standard. From the maximization of the force in equation 3.13, it can be seen that the center of pressure actually at the geometric center of the largest face of the satellite. However, the given choice of $\mathbf{x}_{CP}$ and $\mathbf{x}_{CG}$ will produce a higher torque, which is better for a worst-case scenario.

3.5.3 Atmospheric drag

Atmospheric drag comes from the resistance in the fluid the body moves through, here the atmosphere. The force works in the opposite direction of the motion of the body. The force acting on the satellite is given by

$$\mathbf{f}_{atm}^{b} = -\frac{1}{2}\rho A \|\mathbf{v}^{b}\|^{2} C_D \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} \tag{3.15}$$

where ρ, A and C_D are constants denoting the fluid density, the area of the satellite and the drag coefficient, respectively. The area A is set to the surface area of the largest face of the satellite in order to maximize the force.

The torque is given by

$$\boldsymbol{\tau}_{atm}^{b} = \mathbf{r} \times \mathbf{f}_{atm} = (\mathbf{x}_{CP} - \mathbf{x}_{CG}) \times \mathbf{f}_{atm} \tag{3.16}$$

where $\mathbf{x}_{CP}$ and $\mathbf{x}_{CG}$ are chosen in the same way as for the solar radiation pressure in section 3.5.2 to maximize the effect for a worst-case perturbation.

3.5.4 Magnetic disturbance

A satellite with electronic equipment will be affected by a torque due to the Earth's magnetic field. When the electronics in the satellite are in use, a dipole is set up. This residual magnetic dipole induces a torque that can be represented as [16]

$$\boldsymbol{\tau}_m^b = D\mathbf{B}^b \tag{3.17}$$

where D is a constant. $\mathbf{B}^b$ is the magnetic field acting on the satellite, represented in the body frame. For the model used to simulate the Earth's magnetic field, see section 7.2.

3.6 Dynamics

From equations 3.4, 3.8, and 3.10, the full set of equations for the attitude becomes

$$\mathbf{J}_w\mathbf{A}^+(\frac{{}^b d}{dt}\boldsymbol{\omega}_{ib}^b + \mathbf{A}\frac{{}^w d}{dt}\boldsymbol{\omega}_{bw}^w) = \boldsymbol{\tau}_u^w \tag{3.18a}$$

$$\dot{\mathbf{q}}_{ob} = \frac{1}{2}\mathbf{T}(\mathbf{q}_{ob})\boldsymbol{\omega}_{ob}^b \tag{3.18b}$$

$$\mathbf{J}\frac{{}^b d}{dt}\boldsymbol{\omega}_{ib}^b + \mathbf{A}\mathbf{J}_w\frac{{}^w d}{dt}\boldsymbol{\omega}_{bw}^w + \mathbf{S}(\boldsymbol{\omega}_{ib}^b)\mathbf{H}_s^b = \boldsymbol{\tau}_{mtq}^b + \boldsymbol{\tau}_e^b \tag{3.18c}$$

Taking advantage of the fact that $\mathbf{J}_s = \mathbf{J} - \mathbf{A}\mathbf{J}_w\mathbf{A}^+$, these equations become

$$\mathbf{J}_w\frac{{}^w d}{dt}\boldsymbol{\omega}_{bw}^w = \boldsymbol{\tau}_u^w - \mathbf{J}_w\mathbf{A}^+\frac{{}^b d}{dt}\boldsymbol{\omega}_{ib}^b \tag{3.19a}$$

$$\dot{\mathbf{q}}_{ob} = \frac{1}{2}\mathbf{T}(\mathbf{q}_{ob})\boldsymbol{\omega}_{ob}^b \tag{3.19b}$$

$$\mathbf{J}_s\frac{{}^b d}{dt}\boldsymbol{\omega}_{ib}^b = -\mathbf{S}(\boldsymbol{\omega}_{ib}^b)\mathbf{H}_s^b - \mathbf{A}\boldsymbol{\tau}_u^w + \boldsymbol{\tau}_{mtq}^b + \boldsymbol{\tau}_e^b \tag{3.19c}$$

With $(\mathbf{x}_1^\intercal, \mathbf{x}_2^\intercal, x_3, \mathbf{x}_4^\intercal)^\intercal = ((\boldsymbol{\omega}_{bw}^w)^\intercal, (\boldsymbol{\omega}_{ib}^b)^\intercal, \eta, \boldsymbol{\epsilon}^\intercal)^\intercal$ and $(\mathbf{u}_1^\intercal, \mathbf{u}_2^\intercal, \mathbf{u}_3^\intercal)^\intercal = ((\boldsymbol{\tau}_u^w)^\intercal, (\boldsymbol{\tau}_{mtq}^b)^\intercal, (\boldsymbol{\tau}_e^b)^\intercal)^\intercal$, the model used for linearization is

$$\mathbf{J}_w \frac{{}^w d}{dt}\mathbf{x}_1 = \mathbf{u}_1 - \mathbf{J}_w\mathbf{A}^+ \frac{{}^b d}{dt}\mathbf{x}_2 \tag{3.20a}$$

$$\mathbf{J}_s \frac{{}^b d}{dt}\mathbf{x}_2 = -\mathbf{S}(\mathbf{x}_2)\mathbf{H}_s^b - \mathbf{A}\mathbf{u}_1 + \mathbf{u}_2 + \mathbf{u}_3 \tag{3.20b}$$

$$\dot{x}_3 = -\frac{1}{2}\mathbf{x}_4^\top(\mathbf{x}_2 - \boldsymbol{\omega}_{io}^b) \tag{3.20c}$$

$$\dot{\mathbf{x}}_4 = \frac{1}{2}(x_3\mathbf{I}_{3\text{x}3} + \mathbf{S}(\mathbf{x}_4))(\mathbf{x}_2 - \boldsymbol{\omega}_{io}^b) \tag{3.20d}$$

3.7 Error dynamics

Since $\boldsymbol{\omega}_{ob}^b$ and $\mathbf{q}_{ob}$ are the variables of interest, the error variables are defined as

$$\boldsymbol{\omega}_e^b = \boldsymbol{\omega}_{ob}^b - \boldsymbol{\omega}_d^b \tag{3.21}$$

and

$$\mathbf{q}_e = \mathbf{q}_d^{-1} \otimes \mathbf{q}_{ob} \tag{3.22}$$

as defined in section 2.3.2.

To find the error dynamics, the error variables needs to be differentiated. From equation 2.16, it is clear that $\boldsymbol{\omega}_e^b$ is given by

$$\boldsymbol{\omega}_e^b = \boldsymbol{\omega}_{ob}^b - \boldsymbol{\omega}_d^b = \boldsymbol{\omega}_{ib}^b - \mathbf{R}_o^b\boldsymbol{\omega}_{io}^o - \boldsymbol{\omega}_d^b \tag{3.23}$$

Differentiating this equation with respect to time in the body frame yields the error dynamics of the angular velocity. The time derivative of $\boldsymbol{\omega}_d^b$ depends on the desired input signal, while the time derivative of $\boldsymbol{\omega}_{ib}^b$ can be found in 3.19. Differentiating the final term

$$\frac{{}^b d}{dt}\mathbf{R}_o^b\boldsymbol{\omega}_{io}^o = \frac{{}^b d}{dt}(\mathbf{R}_o^b)\boldsymbol{\omega}_{io}^o + \mathbf{R}_o^b\frac{{}^b d}{dt}(\boldsymbol{\omega}_{io}^o) = \mathbf{S}(\boldsymbol{\omega}_{ob}^b)\mathbf{R}_o^b\boldsymbol{\omega}_{io}^o + \mathbf{R}_o^b(\frac{{}^o d}{dt}\boldsymbol{\omega}_{io}^o - \mathbf{S}(\mathbf{R}_o^b\boldsymbol{\omega}_{ob}^o)\boldsymbol{\omega}_{io}^o) \tag{3.24}$$

Noting that the angular velocity of the orbit given in the orbit frame, $\boldsymbol{\omega}_{io}^o$, will be close to constant over time, its time derivative in orbit frame is approximately zero, $\frac{{}^o d}{dt}\boldsymbol{\omega}_{io}^o \approx \mathbf{0}$. Equation 3.24 becomes

$$\frac{{}^b d}{dt}\mathbf{R}_o^b\boldsymbol{\omega}_{io}^o = \mathbf{S}(\boldsymbol{\omega}_{ob}^b)\mathbf{R}_o^b\boldsymbol{\omega}_{io}^o - \mathbf{R}_o^b\mathbf{S}(\mathbf{R}_o^b\boldsymbol{\omega}_{ob}^o)\boldsymbol{\omega}_{io}^o \tag{3.25}$$

Putting the results together, the error dynamics become

$$\dot{\mathbf{q}}_e = \mathbf{T}(\mathbf{q_e})\boldsymbol{\omega}_e^b \tag{3.26a}$$

$$\begin{aligned} \mathbf{J}_s\frac{{}^b d}{dt}\boldsymbol{\omega}_e^b = & -\mathbf{S}(\boldsymbol{\omega}_{ib}^b)\mathbf{H}_s^b - \mathbf{A}\boldsymbol{\tau}_u^w + \boldsymbol{\tau}_{mtq}^b + \boldsymbol{\tau}_e^b - \mathbf{J}_s\mathbf{S}(\boldsymbol{\omega}_{ob}^b)\mathbf{R}_o^b\boldsymbol{\omega}_{io}^o \\ & + \mathbf{J}_s\mathbf{R}_o^b(\mathbf{S}(\mathbf{R}_b^o\boldsymbol{\omega}_{ob}^b)\boldsymbol{\omega}_{io}^o) - \mathbf{J}_s\frac{{}^b d}{dt}\boldsymbol{\omega}_d^b \end{aligned} \tag{3.26b}$$

Chapter 4

Control and observer theory

4.1 Control theory

The basic concept behind the controllers used for the reaction wheels are presented in this section. The controllers used with the magnetorquers, the detumbling control objective and momentum dumping, are only presented in the control design chapter (sections 5.1 and 5.2).

4.1.1 PID control

Proportional-integral-derivative control is on the form $\boldsymbol{\tau} = \mathbf{K}_p\mathbf{e} + \mathbf{K}_i \int \mathbf{e} dt + \mathbf{K}_d \frac{d}{dt}\mathbf{e}$, where $\mathbf{e}$ is the error value. The PID controller has three parts, where the P term gives a proportional contribution based on the error, the I term eliminates steady-state errors in the response, and the D term is a dampening term.

4.1.2 Linear-quadratic regulator

The linear-quadratic regulator, or LQR, is a form of optimal control. The LQR minimizes a cost function on the form $J = \mathbf{x}^\top\mathbf{Q}\mathbf{x} + \mathbf{u}^\top\mathbf{R}\mathbf{u}$, where $\mathbf{Q} = \mathbf{Q}^\top > \mathbf{0}, \mathbf{R} = \mathbf{R}^\top > \mathbf{0}$ are the matrices used to penalize the state and input vectors, respectively. The result of the minimization is a matrix **K**, which is the gain matrix that is to be applied in the state-feedback control, $\mathbf{u} = -\mathbf{K}\mathbf{x}$. As the equations describing the attitude of the satellite are nonlinear, it is necessary to linearize the equations before the LQR can be applied.

4.1.2.1 Linearization

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \tag{4.1a}$$

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{4.1b}$$

By linearizing the system, a system given on the form in equation 4.1a takes the form of the dynamic system in equation 4.1b. **A** and **B** are found by taking the Jacobian of $f(\mathbf{x}, \mathbf{u})$.

$$\mathbf{A} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} \cdots \frac{\partial f_1}{\partial x_n} \\ \vdots \ddots \vdots \\ \frac{\partial f_m}{\partial x_1} \cdots \frac{\partial f_m}{\partial x_n} \end{bmatrix}_{(\mathbf{x},\mathbf{u})=(\mathbf{x}_0,\mathbf{u}_0)}, \mathbf{B} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} \cdots \frac{\partial f_1}{\partial u_n} \\ \vdots \ddots \vdots \\ \frac{\partial f_m}{\partial u_1} \cdots \frac{\partial f_m}{\partial u_n} \end{bmatrix}_{(\mathbf{x},\mathbf{u})=(\mathbf{x}_0,\mathbf{u}_0)} \tag{4.2}$$

The point $(\mathbf{x}, \mathbf{u}) = (\mathbf{x}_0, \mathbf{u}_0)$ is referred to as the point of linearization.

4.1.3 Sliding mode control

Sliding mode control (SMC) is a form of nonlinear control that is robust to disturbances and unmodeled dynamics [30]. To apply the control, a sliding surface $\boldsymbol{\sigma}$ has to be constructed. The control has two phases: The reaching phase, where the state variables reach the sliding manifold, and the sliding phase, where the state slides along the sliding surface with $\boldsymbol{\sigma} = \mathbf{0}$ until the origin is reached.

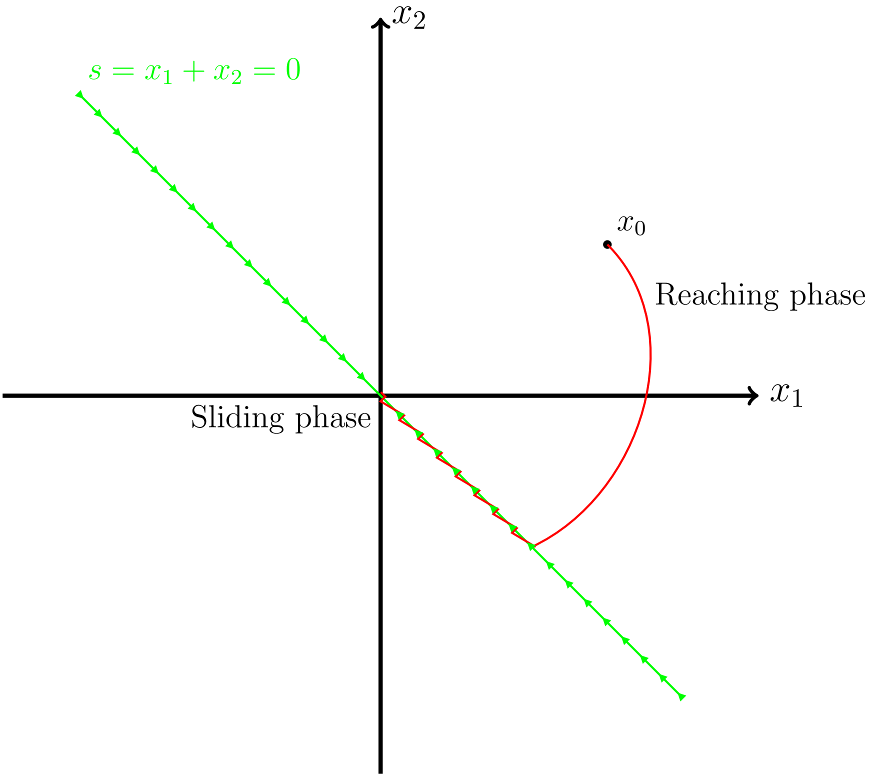


Figure 4.1: Figure illustrating the reaching and sliding phase. Credit: Arnt Erik Stene.

The choice of the sliding surface is for that reason important. Constructing a sliding surface based on the error variables would lead to a desirable result, as the sliding mode control ensures finite-time convergence to zero for the sliding variables [30].

4.1.3.1 Conventional sliding mode control

Equivalent control is the control signal that equates the dynamics of the system in the derivative of the sliding surface

$$\dot{\boldsymbol{\sigma}} = \mathbf{f}(\mathbf{x}, \mathbf{t}) + \mathbf{u} \Rightarrow \mathbf{u}_{eq} = -\mathbf{f}(\mathbf{x}, \mathbf{t}) \tag{4.3}$$

Note that this takes advantage of that the relative degree of the sliding surface is no more than one along any channel.
The conventional sliding mode control input is on the form

$$\mathbf{u}_{SMC} = \boldsymbol{\rho} sign(\boldsymbol{\sigma}) \tag{4.4}$$

where $\boldsymbol{\sigma}$ is the sliding suface, $\boldsymbol{\rho}$ is the control gain given by the bound on the noise on the system, and $sign(\cdot)$ is a function defined as

$$sign(\mathbf{x}) = \begin{cases} 1, & \text{if } x_i \geq 0. \\ -1, & \text{if } x_i < 0 \end{cases} \tag{4.5}$$

Combining the conventional sliding mode control in equation 4.4 with the equivalent control in equation 4.3, making the total control signal $\mathbf{u} = \mathbf{u}_{eq} + \mathbf{u}_{SMC}$. However, the equivalent control may be expensive to compute or difficult to find all together. The SMC can compensate for this by setting the control input to $\mathbf{u} = \mathbf{u}_{SMC}$ if the gain $\boldsymbol{\rho}$ is set to be $\boldsymbol{\rho} = \|\mathbf{u}_{eq}\|_{max} + \boldsymbol{\rho}_{old}$, as shown in [31].

4.1.3.2 Generalized super-twisting algorithm

A first order sliding mode control, such as the one discussed in 4.1.3.1 has a problem known as chattering. This effect appears due to the discontinuous switching in the control signal when the sliding surface is close being equal to zero, and thus the control signal switching between positive and negative. Second order sliding mode controls hold an advantage over the first order sliding mode controls in that their discontinuities are hidden behind an integrator, thus attenuating the chattering. In this thesis, a second order sliding mode controller known as the generalized super-twisting algorithm is used. The generalized super-twisting algorithm is given in the form [32]

$$\mathbf{u} = -\mathbf{k}_1 \boldsymbol{\phi}_1(\mathbf{x}) + \mathbf{z} \tag{4.6a}$$

$$\dot{\mathbf{z}} = -\mathbf{k}_2 \boldsymbol{\phi}_2(\mathbf{x}) \tag{4.6b}$$

where $\boldsymbol{\phi}_1(\mathbf{x})$ and $\boldsymbol{\phi}_2(\mathbf{x})$ are defined as

$$\boldsymbol{\phi}_1(\mathbf{x}) = \lceil \mathbf{x} \rfloor^{\frac{1}{2}} + \boldsymbol{\beta}\mathbf{x} \tag{4.7a}$$

$$\boldsymbol{\phi}_2(\mathbf{x}) = \frac{1}{2}\lceil \mathbf{x} \rfloor^{0} + \frac{3}{2}\boldsymbol{\beta}\lceil \mathbf{x} \rfloor^{\frac{1}{2}} + \boldsymbol{\beta}^2\mathbf{x} \tag{4.7b}$$

where $\lceil a \rfloor^b = |a|^b sign(a)$. The benefit of the generalized super twisting algorithm over the super twisting algorithm, another second order sliding mode control, is that it is robust in the presence of unmatched disturbances, as well as time and state varying perturbations. This is a necessity, as the perturbations present in this problem, defined in section 3.5, are both state and time dependent.

4.2 Observer theory

An observer exists to estimate the states of the system based on the inputs and outputs of the system. The idea a sliding mode observer builds on is to introduce an observer injection term that gets designed so that the estimated values go to the values they estimate. A sliding mode observer can make the output estimation error go to zero in finite time and the observer states converge to the system states asymptotically if the magnitude of the disturbances is bounded [30].

Chapter 5

Control design

The control design reflects the control objectives in section A.3. In this chapter, several controllers will be developed for both pointing and slewing. For detumbling, the results shown in the specialization project using the B-dot algorithm will be presented. As discussed in chapter A, the reaction wheels will be used for pointing and slewing. The B-dot algorithm is based on the magnetorquers for detumbling, and is one of two control schemes shown for the magnetorquers in this thesis. The other is an algorithm used for momentum dumping of the reaction wheels.

5.1 Detumbling controller design

The B-dot algorithm is an algorithm that controls against a magnetic field. It uses the derivative of the magnetic field, B-dot, in order to construct a control signal.

The control torque is given by

$$\boldsymbol{\tau}^b_{MTQ} = \mathbf{m}^b \times \mathbf{B}^b \tag{5.1}$$

where $\mathbf{B}^b$ is the representation of the Earth's magnetic field vector in the body frame. The B-dot algorithm itself comes from the form of $\mathbf{m}^b$,

$$\mathbf{m}^b = -k\dot{\mathbf{B}}^b, k > 0 \tag{5.2}$$

The $\mathbf{B}^b$ vector can be found from the IGRF model, see section 7.2. However, the derivative of the model cannot easily be read out of the model. Due to a simplification made by assuming that the time derivative of the magnetic field in the inertial frame is approximately zero [16], the derivative can be found as

$$\frac{{}^b d}{dt}\dot{\mathbf{B}}^b = -\boldsymbol{\omega}_{ib}^b \times \mathbf{B}^b \tag{5.3}$$

The vector $\mathbf{m}^b$ is the magnetic moment applied by the magnetorquers. $\mathbf{m}^b$ is given, in all three components, by the product of the current passing through the coils, the number of windings, and the surface area in the given direction.

$$\mathbf{m}^b = \begin{bmatrix} i_x N_x A_x \\ i_y N_y A_y \\ i_z N_z A_z \end{bmatrix} \tag{5.4}$$

In the simulation, only the B-dot algorithm in equation 5.2 is used to produce the $\mathbf{m}^b$ vector for the control torque. Equation 5.4 is used to produce the saturation limits for the magnetorquer control signals in the simulation, using the physical properties of the magnetorquers and the maximum values for current for the HYPSO mission as constraints.

5.2 Momentum dumping

When the reaction wheels are used to achieve a control objective, the reaction wheels will spin up in order to produce the correct motion. The reaction wheels work best when they are spinning above a certain thresholded angular velocity due to their mechanics. At a certain point the reaction wheels would saturate, so keeping the angular velocity of the reaction wheels away from the saturation limit is preferable. If the angular velocity from the reaction wheels along each axis in the body frame is zero, with the reaction wheels still spinning, the satellite is easier to control. In conclusion, what should be done is

- Spinning up the reaction wheels.
- Whenever the angular velocity of the reaction wheels change from the reference value, apply a torque to either increase or decrease their angular velocity.

This momentum dumping torque is given by [33]

$$\boldsymbol{\tau}_{MD} = \left(\frac{k_m}{||\mathbf{B}^b||}\left({\mathbf{h_e}}^b \times \frac{\mathbf{B}^b}{||\mathbf{B}^b||}\right)\right) \times \mathbf{B}^b, k_m > 0 \tag{5.5}$$

where $\mathbf{B}^b$ is the magnetic field vector, and ${\mathbf{h_e}}^b$ is the error in angular momentum for the reaction wheels, given by

$${\mathbf{h_e}}^b = \mathbf{A}\mathbf{J}_w(\boldsymbol{\omega}^w_{bw} - \boldsymbol{\omega}^w_{bw,ref}) \tag{5.6}$$

Remark. *The excess momentum in the reaction wheels that this torque dumps can only be dumped with external torque, such as it is done here with magnetorquers.*

5.3 Pointing control design

Pointing is one of the control objectives. It involves tracking the trajectory of the quaternion. The goal is to control the attitude quaternion, $\mathbf{q}_{ob}$, to a given attitude. When the satellite is pointing in some direction, both the angular velocity $\boldsymbol{\omega}^b_{ob}$ and the quaternion rate $\dot{\mathbf{q}}_{ob}$ will change when the attitude changes. For example, when the satellite has reached a setpoint, it will stay there if the angular velocity, or the quaternion rate, is zero. For this reason, either of them can be used when designing controllers for achieving the pointing objective.

5.3.1 PD control

The PD control used for pointing is in the form

$$\boldsymbol{\tau}^b = sign(\eta_e)\mathbf{K}_p\boldsymbol{\epsilon}_e + \mathbf{K}_d\boldsymbol{\omega}^b_{ob} + \mathbf{S}(\boldsymbol{\omega}^b_{ob})(\mathbf{A}\mathbf{J}_w\boldsymbol{\omega}^w_{bw}) \tag{5.7}$$

where $\mathbf{K}_p$ and $\mathbf{K}_d$ are positive definite matrices. $\mathbf{S}(\boldsymbol{\omega}^b_{ob})(\mathbf{A}\mathbf{J}_w\boldsymbol{\omega}^w_{bw})$ is a gyroscopic compensation term. As shown in [34], the PD controller without the gyroscopic compensation term controls the error quaternion to $[1\ \mathbf{0}_{3x1}]^\intercal$ or $[-1\ \mathbf{0}_{3x1}]^\intercal$ depending on the sign in front of the $\boldsymbol{\epsilon}_e$-term.

5.3.2 LQR

Using the definition of the Jacobians given in 4.2 with the representation of the attitude dynamics defined in equation 3.20, the linearized model becomes

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}^+\mathbf{J}_s^{-1}\mathbf{S}(\mathbf{x}_2)\mathbf{A}\mathbf{J}_w & \mathbf{A}^+\mathbf{J}_s^{-1}(\mathbf{S}(\mathbf{x}_2)\mathbf{J}-\mathbf{S}(\mathbf{J}\mathbf{x}_2)-\mathbf{S}(\mathbf{A}\mathbf{J}_w\mathbf{x}_1)) & \mathbf{0}_{nx1} & \mathbf{0}_{nx3} \\ -\mathbf{J}_s^{-1}\mathbf{S}(\mathbf{x}_2)\mathbf{A}\mathbf{J}_w & -\mathbf{J}_s^{-1}(\mathbf{S}(\mathbf{x}_2)\mathbf{J}-\mathbf{S}(\mathbf{J}\mathbf{x}_2)-\mathbf{S}(\mathbf{A}\mathbf{J}_w\mathbf{x}_1)) & \mathbf{0}_{3x1} & \mathbf{0}_{3x3} \\ \mathbf{0}_{1xn} & -\frac{1}{2}\mathbf{x}_4^\intercal & \mathbf{0}_{1x1} & -\frac{1}{2}(\mathbf{x}_2-\boldsymbol{\omega}^b_{io})^\intercal \\ \mathbf{0}_{3xn} & \frac{1}{2}(\mathbf{x}_3\mathbf{I}_{3x3}+\mathbf{S}(\mathbf{x}_4)) & \frac{1}{2}(\mathbf{x}_2-\boldsymbol{\omega}^b_{io}) & -\frac{1}{2}\mathbf{S}(\mathbf{x}_2-\boldsymbol{\omega}^b_{io}) \end{bmatrix}_{(\mathbf{x},\mathbf{u})=(\mathbf{x}_0,\mathbf{u}_0)} \tag{5.8a}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{J}_w + \mathbf{A}^+\mathbf{J}_s^{-1}\mathbf{A} & -\mathbf{A}^+\mathbf{J}_s^{-1} & -\mathbf{A}^+\mathbf{J}_s^{-1} \\ -\mathbf{J}_s^{-1}\mathbf{A} & \mathbf{J}_s^{-1} & \mathbf{J}_s^{-1} \\ \mathbf{0}_{1xn} & \mathbf{0}_{1x3} & \mathbf{0}_{1x3} \\ \mathbf{0}_{3xn} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \end{bmatrix}_{(\mathbf{x},\mathbf{u})=(\mathbf{x}_0,\mathbf{u}_0)} \tag{5.8b}$$

where the first, second, and third columns of the $\mathbf{B}$ matrix are related to the torque from the reaction wheels, the torque from the magnetorquers, and the perturbing torques, respectively. When using this model to calculate the LQR control gain, a reduced version is used.

$$\mathbf{A} = \begin{bmatrix} -\mathbf{J}_s^{-1}(\mathbf{S}(\mathbf{x}_2)\mathbf{J} - \mathbf{S}(\mathbf{J}\mathbf{x}_2) - \mathbf{S}(\mathbf{A}\mathbf{J}_w\mathbf{x}_1)) & \mathbf{0}_{3x3} \\ \frac{1}{2}(\mathbf{x}_3\mathbf{I}_{3x3} + \mathbf{S}(\mathbf{x}_4)) & -\frac{1}{2}\mathbf{S}(\mathbf{x}_2 - \boldsymbol{\omega}^b_{io}) \end{bmatrix}_{(\mathbf{x},\mathbf{u})=(\mathbf{x}_0,\mathbf{u}_0)} \tag{5.9a}$$

$$\mathbf{B} = \begin{bmatrix} -\mathbf{J}_s^{-1} \\ \mathbf{0}_{3x3} \end{bmatrix}_{(\mathbf{x},\mathbf{u})=(\mathbf{x}_0,\mathbf{u}_0)} \tag{5.9b}$$

Note that this model is rewritten to use the input torque to the reaction wheels in the body frame, not in the wheel frame.

The model is linearized about a chosen point. The points of interest are the target points. As Simulink does not allow for solving of the algebraic Riccati equation at each time step, the point has to be chosen offline. For more on this, see appendix C.

The point chosen is taken as the initial values for the reaction wheels for $\mathbf{x}_1$, the zero vector for $\mathbf{x}_2$ and $\boldsymbol{\omega}_{ob}^b$ due to the vectors representing small numbers. The quaternion is set to $[1; 0; 0; 0]$, as it is a valid quaternion.

Using these initial conditions the optimal gain can be calculated in Matlab and applied in Simulink as

$$\boldsymbol{\tau}_u^b = -\mathbf{K}\mathbf{x} = -\mathbf{K} \begin{bmatrix} \boldsymbol{\omega}_{ob,e}^b \\ \boldsymbol{\epsilon}_{ob,e} \end{bmatrix} \tag{5.10}$$

5.3.3 Sliding mode control

For the sliding mode controls, there are two parts that needs to be designed: the sliding surface and the control input.

5.3.3.1 Sliding surface

In choosing a sliding surface, it is desirable to choose one that has good stability properties. Two surfaces will be used for the pointing objective.

$$\boldsymbol{\sigma} = \dot{\boldsymbol{\epsilon}}_e + \mathbf{K}\boldsymbol{\epsilon}_e, \mathbf{K} = \mathbf{K}^\mathsf{T} > 0 \tag{5.11}$$

Setting $\boldsymbol{\sigma} = \mathbf{0}$, this becomes

$$\dot{\boldsymbol{\epsilon}}_e + \mathbf{K}\boldsymbol{\epsilon}_e = \mathbf{0} \Rightarrow \dot{\boldsymbol{\epsilon}}_e = -\mathbf{K}\boldsymbol{\epsilon}_e \tag{5.12}$$

which is asymptotically stable since $\mathbf{K}$ is a positive definite matrix. $\boldsymbol{\epsilon}_e \longrightarrow \mathbf{0}$, and from the identity that $\eta_e^2 + \boldsymbol{\epsilon}_e^\mathsf{T}\boldsymbol{\epsilon}_e = 1$, $\eta_e = \pm 1$. The other surface used is from [35]

$$\boldsymbol{\sigma} = \boldsymbol{\omega}_e + \mathbf{K}\boldsymbol{\epsilon}_e, \mathbf{K} = \mathbf{K}^\mathsf{T} > 0 \tag{5.13}$$

Again, setting $\boldsymbol{\sigma} = \mathbf{0}$, this becomes

$$\boldsymbol{\omega}_e + \mathbf{K}\boldsymbol{\epsilon}_e = \mathbf{0} \Rightarrow \boldsymbol{\omega}_e = -\mathbf{K}\boldsymbol{\epsilon}_e \tag{5.14}$$

Using the Lyapunov function candidate like in [35],

$$V = 1 - \eta_e \tag{5.15}$$

which positive definite as η_e lies between -1 and 1. The time derivative becomes

$$\dot{V} = -\dot{\eta}_e = \frac{1}{2}\boldsymbol{\epsilon}_e^\mathsf{T}\boldsymbol{\omega}_e = -\frac{1}{2}\boldsymbol{\epsilon}_e^\mathsf{T}\mathbf{K}\boldsymbol{\epsilon}_e \tag{5.16}$$

which makes the system asymptotically stable. $\boldsymbol{\epsilon}_e \longrightarrow \mathbf{0}$, and from the identity that $\eta_e^2 + \boldsymbol{\epsilon}_e^\mathsf{T}\boldsymbol{\epsilon}_e = 1$, $\eta_e = \pm 1$. When on the sliding surface, $\boldsymbol{\omega}_e = -\mathbf{K}\boldsymbol{\epsilon}_e$, and therefore $\boldsymbol{\omega}_e$ asymptotically converges to zero too.

Note that both these sliding surfaces have vector relative degree $\rho = \{1, 1, 1\}$, as the input can be found in the output of their first derivatives.

5.3.3.2 First order SMC

The first order sliding mode that will be used is the conventional one. As noted in section 4.1.3.1, setting the gain large enough would enable the control without equivalent control would work as well as calculating the equivalent control [31].

$$\boldsymbol{\tau}_u^b = \boldsymbol{\rho} sign(\boldsymbol{\sigma}) \tag{5.17}$$

The surface from equation 5.13 will be used by the first order SMC for pointing.

5.3.3.3 GSTA

The full GSTA algorithm is defined in equations 4.6 and 4.7, where the parameter $\mathbf{x}$ is the sliding surface. The surface from equation 5.11 will be used by the GSTA for pointing.

5.4 Slewing control design

With slewing, the objective is for the satellite to move at a given angular velocity. From this angular velocity, the rotation given by the angle-axis representation is given by the relationship [36]

$$\boldsymbol{\theta}(t) = \int \boldsymbol{\omega}(t)dt \tag{5.18}$$

When the satellite is set to slew at a constant angular velocity, $\boldsymbol{\theta}(t)$ would become $\boldsymbol{\omega}t$. The angle and axis parameters can be separated by

$$\theta(t) = ||\boldsymbol{\theta}(t)|| \tag{5.19a}$$

$$\mathbf{e}(t) = \frac{\boldsymbol{\theta}(t)}{||\boldsymbol{\theta}(t)||} \tag{5.19b}$$

In order to generate the reference trajectory used in the slewing objective, the angular velocity setpoint is used to calculate the quaternion reference by the relation in equation 5.19. The relation between the angle-axis parameters and quaternions is given by the definition of the quaternion in equation 2.4.

$$\mathbf{q}_d(t) = \begin{bmatrix} \eta(t) \\ \epsilon(t) \end{bmatrix} = \begin{bmatrix} cos(\frac{\theta(t)}{2}) \\ \mathbf{e}(t)\ sin(\frac{\theta(t)}{2}) \end{bmatrix} \tag{5.20}$$

5.4.1 PD control

For slewing, the PD control is defined as

$$\boldsymbol{\tau}^b = sign(\eta_e)\mathbf{K}_p\boldsymbol{\epsilon}_e + \mathbf{K}_d\boldsymbol{\omega}_e^b + \mathbf{S}(\boldsymbol{\omega}_{ob}^b)(\mathbf{A}\mathbf{J}_w\boldsymbol{\omega}_{bw}^w) \tag{5.21}$$

where $\mathbf{K}_p$ and $\mathbf{K}_d$ are positive definite matrices. $sign(\cdot)$ is the signum function defined in equation 4.5, $(\eta_e, \boldsymbol{\epsilon}_e)$ are the attitude error coordinates, and $\boldsymbol{\omega}_e^b$ is the $\boldsymbol{\omega}_{ob}^b$ angular velocity error. $\mathbf{S}(\boldsymbol{\omega}_{ob}^b)(\mathbf{A}\mathbf{J}_w\boldsymbol{\omega}_{bw}^w)$ is a gyroscopic compensation term. Note that this is the same PD controller as used for pointing, assuming that the angular velocity reference used in equation 5.7 is the zero vector.

5.4.2 LQR

The LQR control gain will be calculated the same way for slewing as it was for pointing in section 5.3.2.

5.4.3 Sliding mode control

For the sliding mode control, the control signals themselves will be the same for the slewing objective as they were in section 5.3.3 for the pointing objective. However, another surface is suggested, given by [26]

$$\boldsymbol{\sigma} = \boldsymbol{\omega}_e - 2c\frac{\partial W}{\partial \eta_e}\boldsymbol{\epsilon}_e \tag{5.22}$$

Using $W(\eta_e) = 1 - |\eta_e|$ as the Lyapunov function candidate, [26] shows that both $\eta_e = 1$ and $\eta_e = -1$ are asymptotically stable equilibrium points. Note that in equation 5.22, $\frac{\partial W}{\partial \eta_e} = -sign(\eta_e)$. This surface is used for both the first order SMC and for the GSTA when slewing.

Chapter 6

Observer design

6.1 Sliding mode observer

The first order sliding mode observer for the satellite attitude problem used here is from [31]

$$\dot{\hat{\boldsymbol{\epsilon}}}_{ob} = -L_1 sign(\hat{\boldsymbol{\epsilon}}_{ob} - \boldsymbol{\epsilon}_{ob}) \tag{6.1a}$$

$$\tau\dot{\mathbf{z}} = -\mathbf{z} - L_1 sign(\hat{\boldsymbol{\epsilon}} - \boldsymbol{\epsilon}) \tag{6.1b}$$

$$\dot{\hat{\boldsymbol{\omega}}}^b_{ob} = -L_2 sign(\mathrm{U}(\boldsymbol{\epsilon}_{ob})\hat{\boldsymbol{\omega}}^b_{ob} - z) \tag{6.1c}$$

where τ is a constant selected in order for equation 6.1b to work as a low pass filter. Note that the matrix $\mathrm{U}(\boldsymbol{\epsilon}_{ob})$ is defined in equation 3.4.

An IMU is available for HYPSO, so the angular velocity measurements will be available. An alternative sliding mode observer for the angular velocity would be

$$\dot{\hat{\boldsymbol{\omega}}}^b_{ib} = \mathbf{J}_s^{-1}(-\mathrm{S}(\hat{\boldsymbol{\omega}}^b_{ib})(\mathbf{J}\hat{\boldsymbol{\omega}}^b_{ib} + \mathbf{A}\mathbf{J}_w\hat{\boldsymbol{\omega}}^w_{bw}) - \boldsymbol{\tau}^b_u + \boldsymbol{\tau}^b_{MTQ}) + L sign(\boldsymbol{\omega}^b_{ib} - \hat{\boldsymbol{\omega}}^b_{ib}) \tag{6.2}$$

where the first terms are the dynamics, excluding the perturbing torques. From integrating this equation, $\boldsymbol{\omega}_{ob}^{b}$ is by equation 2.16 found by subtracting $\boldsymbol{\omega}_{io}^{b}$, which is assumed to be known, from the estimate of $\boldsymbol{\omega}_{ib}^{b}$.

6.2 Higher order sliding mode observer

The higher order sliding mode observer, or HOSMO, is based on the observer in [37]. The system in [37] is based on a system on the form

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \tag{6.3}$$

Using the fact that $\mathbf{T}^{\mathsf{T}}(\mathbf{q})\mathbf{T}(\mathbf{q}) = \frac{1}{4}\mathbf{I}_{3x3}$ from equation 3.6, the attitude dynamics can be represented using this form by using the identity [26]

$$\dot{\mathbf{q}}_{ob} = \mathbf{T}(\mathbf{q}_{ob})\boldsymbol{\omega}_{ob}^{b} \Rightarrow \boldsymbol{\omega}_{ob}^{b} = 4\mathbf{T}^{\mathsf{T}}(\mathbf{q}_{ob})\dot{\mathbf{q}}_{ob} \tag{6.4}$$

$$\dot{\boldsymbol{\omega}}_{ob}^{b} = 4(\mathbf{T}^{\mathsf{T}}(\dot{\mathbf{q}}_{ob})\dot{\mathbf{q}}_{ob} + \mathbf{T}^{\mathsf{T}}(\mathbf{q}_{ob})\ddot{\mathbf{q}}_{ob}) = 4\mathbf{T}^{\mathsf{T}}(\mathbf{q}_{ob})\ddot{\mathbf{q}}_{ob} \tag{6.5}$$

From these identities, [35], [27] show that the system can be represented on the form in equation 6.3 with $\mathbf{M}(\mathbf{q})$ given on the form

$$\mathbf{M}(\mathbf{q}) = \frac{1}{2}\mathbf{T}(\mathbf{q}_{ob})\mathbf{J}_{s}\mathbf{T}^{\mathsf{T}}(\mathbf{q}_{ob}) \tag{6.6}$$

However, $\mathbf{M}(\mathbf{q})$ can be singular [27], which makes it unsuitable as a starting point. It is possible to find the error dynamics directly. As in [37], defining

$$\tilde{\mathbf{x}}_{1} = \boldsymbol{\epsilon}_{e} \tag{6.7a}$$

$$\tilde{\mathbf{x}}_{2} = \dot{\boldsymbol{\epsilon}}_{e} = \mathbf{U}(\mathbf{q})_{e}\boldsymbol{\omega}_{e} \tag{6.7b}$$

Differentiating the equations with respect to time, taking advantage of the equations in section 3.7

$$\dot{\tilde{\mathbf{x}}}_1 = \tilde{\mathbf{x}}_2 = \mathbf{U}(\mathbf{q}_e)\boldsymbol{\omega}_e = \frac{1}{2}(\eta_e \mathbf{I}_{3x3} + \mathbf{S}(\boldsymbol{\epsilon}_e))\boldsymbol{\omega}_e \tag{6.8a}$$

$$\begin{aligned}\dot{\tilde{\mathbf{x}}}_2 = \frac{d}{dt}(\mathbf{U}(\mathbf{q}_e)\boldsymbol{\omega}_e) &= \frac{d}{dt}(\mathbf{U}(\mathbf{q}_e))\boldsymbol{\omega}_e + \mathbf{U}(\mathbf{q}_e)\frac{d}{dt}(\boldsymbol{\omega}_e) = \frac{1}{2}(\dot{\eta}_e \mathbf{I}_{3x3} + \mathbf{S}(\dot{\boldsymbol{\epsilon}}_e))\boldsymbol{\omega}_e + \mathbf{U}(\mathbf{q}_e)\frac{d}{dt}(\boldsymbol{\omega}_e) \\ &= (-\frac{1}{4}\boldsymbol{\epsilon}_e^\top \boldsymbol{\omega}_e \mathbf{I}_{3x3} + \mathbf{S}(\frac{1}{4}(\eta_e \mathbf{I}_{3x3} + \mathbf{S}(\boldsymbol{\epsilon}_e))\boldsymbol{\omega}_e))\boldsymbol{\omega}_e + \mathbf{U}(\mathbf{q}_e)\mathbf{J}_s^{-1}(-\mathbf{S}(\boldsymbol{\omega}_{ib}^b)\mathbf{H}_s^b - \mathbf{A}\boldsymbol{\tau}_u^w + \boldsymbol{\tau}_{mtq}^b \\ &\quad + \boldsymbol{\tau}_e^b - \mathbf{J}_s\mathbf{S}(\boldsymbol{\omega}_{ob}^b)\mathbf{R}_o^b\boldsymbol{\omega}_{io}^o + \mathbf{J}_s\mathbf{R}_o^b(\mathbf{S}(\mathbf{R}_b^o\boldsymbol{\omega}_{ob}^b)\boldsymbol{\omega}_{io}^o) - \mathbf{J}_s\frac{{}^b d}{dt}\boldsymbol{\omega}_d^b)\end{aligned} \tag{6.8b}$$

Equivalent results can be found using the Lie derivative. Choosing the output as

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) = \begin{bmatrix} h_1(x) \\ h_2(x) \\ h_3(x) \end{bmatrix} = \boldsymbol{\epsilon}_{ob} \tag{6.9}$$

It can be seen from the definition of the relative degree in section 2.5 that the vector relative degree of this output is {2,2,2}, as input vectors appear in the derivative the first time after two differentiations. Using the equations in equation 3.20, rewriting so that $\mathbf{x}_1$ is represented in the body frame and the model is on the form used for Lie derivatives, shown in equation 2.20a.

$$\begin{aligned}\dot{\mathbf{x}} = \begin{bmatrix} \frac{{}^b d}{dt}\boldsymbol{\omega}_{bw}^b \\ \frac{{}^b d}{dt}\boldsymbol{\omega}_{ib}^b \\ \dot{\eta}_{ob} \\ \dot{\boldsymbol{\epsilon}}_{ob} \end{bmatrix} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} + \mathbf{d} &= \begin{bmatrix} \mathbf{J}_s^{-1}\mathbf{S}(\mathbf{x}_2)(\mathbf{J}\mathbf{x}_2 + \mathbf{A}\mathbf{J}_w\mathbf{A}^+\mathbf{x}_1) \\ -\mathbf{J}_s^{-1}\mathbf{S}(\mathbf{x}_2)(\mathbf{J}\mathbf{x}_2 + \mathbf{A}\mathbf{J}_w\mathbf{A}^+\mathbf{x}_1) \\ -\frac{1}{2}\mathbf{x}_4^\top(\mathbf{x}_2 - \boldsymbol{\omega}_{io}^b) \\ \frac{1}{2}(x_3\mathbf{I}_{3x3} + \mathbf{S}(\mathbf{x}_4))(\mathbf{x}_2 - \boldsymbol{\omega}_{io}^b) \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{A}\mathbf{J}_w\mathbf{A}^+ + \mathbf{J}_s^{-1} & \mathbf{J}_s^{-1} \\ -\mathbf{J}_s^{-1} & \mathbf{J}_s^{-1} \\ \mathbf{0}_{1x3} & \mathbf{0}_{1x3} \\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \end{bmatrix} \begin{bmatrix} \boldsymbol{\tau}_u^b \\ \boldsymbol{\tau}_{MTQ}^b \end{bmatrix} + \begin{bmatrix} \mathbf{J}_s^{-1} \\ \mathbf{J}_s^{-1} \\ \mathbf{0}_{1x3} \\ \mathbf{0}_{3x3} \end{bmatrix} \boldsymbol{\tau}_e^b\end{aligned} \tag{6.10}$$

As the vector relative degree is $\{2, 2, 2\}$, the solution will be on the form

$$\ddot{\mathbf{y}} = \begin{bmatrix} \ddot{y}_1 \\ \ddot{y}_2 \\ \ddot{y}_3 \end{bmatrix} = \boldsymbol{\phi}(\mathbf{x}) + \mathbf{E}(\mathbf{x})\mathbf{u}$$
$$\boldsymbol{\phi}(\mathbf{x}) = L_f^2\mathbf{h}(\mathbf{x}), \mathbf{E}(\mathbf{x}) = L_g L_f \mathbf{h}(\mathbf{x})\mathbf{u} \tag{6.11}$$

Note that these equations are for the dynamics of the system, not the error dynamics. Finding the Lie derivatives needed,

$$L_f\mathbf{h}(\mathbf{x}) = [\mathbf{0}_{3x3}, \mathbf{0}_{3x3}, \mathbf{0}_{3x1}, \mathbf{I}_{3x3}]\mathbf{f}(\mathbf{x}) = \frac{1}{2}(x_3\mathbf{I}_{3\mathrm{x}3} + \mathrm{S}(\mathbf{x}_4))(\mathbf{x}_2 - \boldsymbol{\omega}_{io}^b) \tag{6.12a}$$

$$\begin{aligned} L_f^2\mathbf{h}(\mathbf{x}) &= [\mathbf{0}_{3x3}, \frac{1}{2}(x_3\mathbf{I}_{3\mathrm{x}3} + \mathrm{S}(\mathbf{x}_4)), \frac{1}{2}(\mathbf{x}_2 - \boldsymbol{\omega}_{io}^b), -\frac{1}{2}\mathrm{S}(\mathbf{x}_2 - \boldsymbol{\omega}_{io}^b)]\mathbf{f}(\mathbf{x}) \\ &= -\frac{1}{2}(x_3\mathbf{I}_{3\mathrm{x}3} + \mathrm{S}(\mathbf{x}_4))\mathbf{J}_s^{-1}\mathrm{S}(\mathbf{x}_2)(\mathbf{J}\mathbf{x}_2 + \mathbf{A}\mathbf{J}_w\mathbf{A}^+\mathbf{x}_1) \\ -\frac{1}{4}(\mathbf{x}_2 - \boldsymbol{\omega}_{io}^b)\mathbf{x}_4^\mathsf{T}(\mathbf{x}_2 - \boldsymbol{\omega}_{io}^b) &- \frac{1}{4}\mathrm{S}(\mathbf{x}_2 - \boldsymbol{\omega}_{io}^b)(x_3\mathbf{I}_{3\mathrm{x}3} + \mathrm{S}(\mathbf{x}_4))(\mathbf{x}_2 - \boldsymbol{\omega}_{io}^b) \\ &= \begin{bmatrix} L_f^2 h_1(\mathbf{x}) \\ L_f^2 h_2(\mathbf{x}) \\ L_f^2 h_3(\mathbf{x}) \end{bmatrix} = \boldsymbol{\phi}(\mathbf{x}) \end{aligned} \tag{6.12b}$$

$$\begin{aligned} L_g L_f\mathbf{h}(\mathbf{x}) &= [\mathbf{0}_{3x3}, \frac{1}{2}(x_3\mathbf{I}_{3\mathrm{x}3} + \mathrm{S}(\mathbf{x}_4)), \frac{1}{2}(\mathbf{x}_2 - \boldsymbol{\omega}_{io}^b), -\frac{1}{2}\mathrm{S}(\mathbf{x}_2 - \boldsymbol{\omega}_{io}^b)]\mathbf{g}(\mathbf{x}) \\ &= \begin{bmatrix} -\frac{1}{2}(x_3\mathbf{I}_{3\mathrm{x}3} + \mathrm{S}(\mathbf{x}_4))\mathbf{J}_s^{-1} & \frac{1}{2}(x_3\mathbf{I}_{3\mathrm{x}3} + \mathrm{S}(\mathbf{x}_4))\mathbf{J}_s^{-1} \end{bmatrix} = \mathbf{E}(\mathbf{x}) \end{aligned} \tag{6.12c}$$

A coordinate transformation as described in [38] requires $\mathbf{E}(\mathbf{x})$ to be a square, nonsingular matrix. In [24] it is shown that the matrix can have more columns than rows and still satisfy the conditions, as long as the matrix has full rank. From equation 3.5 it can be seen that this does not hold when η is zero. For this reason, this approach is scrapped, and the one used in [37] is attempted. This gives an observer in the form

$$\dot{\hat{\mathbf{x}}}_1 = \hat{\mathbf{x}}_2 + \mathbf{z}_1 \tag{6.13a}$$

$$\dot{\hat{\mathbf{x}}}_2 = \hat{\mathbf{x}}_3 + \mathbf{z}_2 + \mathbf{U}(\cdot)\mathbf{J}_s^{-1}(-\boldsymbol{\tau}_u^b + \boldsymbol{\tau}_{MTQ}^b) \tag{6.13b}$$

$$\dot{\hat{\mathbf{x}}}_3 = \mathbf{z}_3 \tag{6.13c}$$

where

$$\mathbf{z}_1 = k_1|\mathbf{e_1}|^{2/3}sgn(\mathbf{e_1}) \tag{6.14a}$$

$$\mathbf{z}_2 = k_2|\mathbf{e_1}|^{1/2}sgn(\mathbf{e_1}) \tag{6.14b}$$

$$\mathbf{z}_3 = k_3 sgn(\mathbf{e_1}) \tag{6.14c}$$

where the error variables are defined as $\mathbf{e}_1 = \mathbf{x}_1 - \hat{\mathbf{x}}_1$, $\mathbf{e}_2 = \mathbf{x}_2 - \hat{\mathbf{x}}_2$, and $\mathbf{e}_3 = -\hat{\mathbf{x}}_3 + \mathbf{F}(\cdot)$. $\mathbf{F}(\cdot)$ is given by the remaining terms of the equation defining $\mathbf{x}_2$ once the terms with the control input are removed. This is a third order higher order sliding mode observer. The error dynamics of the observer becomes

$$\dot{\mathbf{e}}_1 = -k_1|\mathbf{e}_1|^{2/3}sgn(\mathbf{e}_1) + \mathbf{e}_2 \tag{6.15a}$$

$$\dot{\mathbf{e}}_2 = -k_2|\mathbf{e}_1|^{1/2}sgn(\mathbf{e}_1) + \mathbf{e}_3 \tag{6.15b}$$

$$\dot{\mathbf{e}}_3 = -k_3 sgn(\mathbf{e}_1) + \dot{\mathbf{F}}(\cdot) \tag{6.15c}$$

Using the sliding surface

$$\hat{\boldsymbol{\sigma}} = \hat{\hat{\mathbf{x}}}_1 + \hat{\hat{\mathbf{x}}}_2 \tag{6.16}$$

where $\hat{\hat{\mathbf{x}}} = \hat{\mathbf{x}} - \mathbf{x}_d$. As shown in section 5.3.3, the state variables would asymptotically converge to zero along $\hat{\boldsymbol{\sigma}} = 0$. Due to the stability of the HOSMO, $\hat{\boldsymbol{\sigma}}$ will go to $\boldsymbol{\sigma}$ in finite time.

In [37] the stability of the cascaded system is proven when the third order sliding mode observer is using Euler angles. The idea from this point would be to follow the same path in order to prove the stability for the cascaded system in this problem using an observer based on quaternions: design a control input such that $\dot{\hat{\boldsymbol{\sigma}}} = \mathbf{u}_{GSTA}$, so that the GSTA controller can drive the sliding surface to zero.

Differentiating the surface and solving for the control input gives

$$\begin{aligned}\dot{\hat{\boldsymbol{\sigma}}} &= \dot{\hat{\tilde{\mathbf{x}}}}_1 + \dot{\hat{\tilde{\mathbf{x}}}}_2 = \dot{\hat{\mathbf{x}}}_1 - \dot{\mathbf{x}}_{1,d} + \dot{\hat{\mathbf{x}}}_2 - \dot{\mathbf{x}}_{2,d} \\ &= \hat{\mathbf{x}}_2 + \mathbf{z}_1 - \dot{\mathbf{x}}_{1,d} + \hat{\mathbf{x}}_3 + \mathbf{z}_2 + \mathbf{U}(\cdot)\mathbf{J}_s^{-1}(-\boldsymbol{\tau}_u^b + \boldsymbol{\tau}_{MTQ}^b) - \dot{\mathbf{x}}_{2,d} \\ \boldsymbol{\tau}(\cdot) &= -\boldsymbol{\tau}_u^b + \boldsymbol{\tau}_{MTQ}^b = \mathbf{U}(\cdot)^{-1}\mathbf{J}_s(-\hat{\mathbf{x}}_2 - \mathbf{z}_1 + \dot{\mathbf{x}}_{1,d} - \hat{\mathbf{x}}_3 - \mathbf{z}_2 + \dot{\mathbf{x}}_{2,d} + \mathbf{u}_{GSTA})\end{aligned} \tag{6.17}$$

With this control torque, the full system would be

$$\dot{\tilde{\mathbf{x}}}_1 = -\tilde{\mathbf{x}}_1 + \hat{\boldsymbol{\sigma}} + \mathbf{e}_1 + \mathbf{e}_2 \tag{6.18a}$$

$$\dot{\hat{\boldsymbol{\sigma}}} = -\mathbf{k}_1\boldsymbol{\phi}_1(\hat{\boldsymbol{\sigma}}) + \mathbf{z} \tag{6.18b}$$

$$\dot{\mathbf{z}} = -\mathbf{k}_2\boldsymbol{\phi}_2(\hat{\boldsymbol{\sigma}}) \tag{6.18c}$$

$$\dot{\mathbf{e}}_1 = -k_1|\mathbf{e}_1|^{2/3}sgn(\mathbf{e}_1) + \mathbf{e}_2 \tag{6.19a}$$

$$\dot{\mathbf{e}}_2 = -k_2|\mathbf{e}_1|^{1/2}sgn(\mathbf{e}_1) + \mathbf{e}_3 \tag{6.19b}$$

$$\dot{\mathbf{e}}_3 = -k_3 sgn(\mathbf{e}_1) + \dot{\mathbf{F}}(\cdot) \tag{6.19c}$$

which is what is in [37], and the stability proof proving that the cascaded system is UGAS would be given by following the calculations in the paper. However, there are some problems that makes the proof impossible to follow directly. One is that the quaternions have two equilibrium points, in $[\pm 1, \mathbf{0}]$, making the stability results local. The other is that the matrix $\mathbf{U}(\cdot)$ becomes singular when η is zero, as it reduces to the skew-symmetric matrix with zeros on the diagonal that represents the cross product in three dimensions. Avoiding the hypersurface that is $\eta^2 = \boldsymbol{\epsilon}^\intercal\boldsymbol{\epsilon} = 0$ is not feasible for a satellite, particularly not a slewing satellite. Due to this, the control signals will not be multiplied with the inverse of the $\mathbf{U}(\cdot)$ matrix. Finding a way around this singularity in order to prove the stability of the cascaded system will be a task for future work.

Chapter 7

Simulation

The full model of the satellite's movement, including the attitude dynamics, equation 3.19, and the orbital mechanics, equation 3.3, is implemented in Simulink. The parameters and functions used in the simulations are defined in Matlab scripts and functions.

Remark. *The structure of the Simulink model has been kept from the specialization project. The internal structure of some of the subsystems has been updated to reflect the equations developed in the master's thesis. See the preface for a list of the changes in the model between the specialization project and the master's thesis.*

7.1 Reference model

A reference model is a model that outputs a smooth signal to the control system when given a reference signal as input. Here, a first-order low-pass filter is used in cascade with a mass-damper-system, as suggested in [20].

$$\begin{aligned} \dot{\boldsymbol{\eta}}_d &= \boldsymbol{\nu}_d \\ \dot{\boldsymbol{\nu}}_d + 2\zeta\omega_n\boldsymbol{\nu}_d + \omega_n^2\boldsymbol{\eta}_d &= \omega_n^2\mathbf{r} \end{aligned} \tag{7.1}$$

Here $\mathbf{r}$ is the reference value and $\boldsymbol{\eta}_d$ is the smooth output. For the simulation, ζ is set to 1 for critical damping. ω_n is set to 1.

To enable the reference model to output correct values from the beginning of the simulation, initial values are assigned to the integrator in the low pass filter as well as the final integrator. This is particularly important for the quaternion reference, as initializing the reference model for the quaternion reference as $[0; 0; 0; 0]$, the Simulink standard, would give a singularity in the model due to $[0; 0; 0; 0]$ not being a possible quaternion coordinate, as the norm is not equal 1. Initializing the reference models with the initial values of the simulations avoids this problem. It should be noted, however, that during slewing, the reference model for the quaternion reference is bypassed, and the desired quaternion is simply calculated based on the output of the angular velocity reference model.

7.2 International Geomagnetic Reference Field Model - IGRF

The International Geomagnetic Reference Field Model [39] is used to generate a model of the Earth's magnetic field. Given the satellite's position relative to the Earth and the satellite's attitude, the IGRF block returns the magnetic field vector represented in the body frame. This field vector is used to calculate the torque acting satellite by the magnetic disturbance, the B-dot algorithm, and the momentum dumping torque. In order to create the IGRF model block in Simulink, the code from [40] was used. This was done in the specialization project.

7.3 Noise

Noise is added to the system in three places: on the angular velocity on the reaction wheels, on the angular velocity of the body frame about the inertial frame, and on the quaternion. These are meant to represent jittering on the reaction wheels, measurement noise on the measurement from the inertial measurement unit (IMU), and quaternion measurement provided by the star tracker. All the noises are represented as white noise.

Name, variable	Variance
Jittering, $\boldsymbol{\omega}_{bw}^{w}$	$\sqrt{\frac{30}{\frac{2*\pi}{60}}}$
IMU noise, $\boldsymbol{\omega}_{ib}^{b}$	1e-7 $\frac{rad}{s}$
Quaternion noise, $\mathbf{q}_{ob}$	0.01°

The measurement noise on the $\boldsymbol{\omega}_{ib}^{b}$ measurements are not measured from a real IMU, but rather a value used here for simulation purposes. Note as well that this thesis does not deal with issues such as drift in the measurements.

7.4 Time delay

Signals from the control block is delayed before applied to the satellite dynamics system by using a transport delay block. This block is meant to represent the total delay in the system. Physically, this is meant to represent the delay from when a measurement is taken to when a control action can be applied to the actuators, due to factors such as pulse per second, real time clock, overhead in physical transfer of data in the cables, and the processing power of each sub-block.

7.5 Uncertainty in assembly of reaction wheels

Nominally, the reaction wheels are set in a formation such that there is one reaction wheel on each of axes of the body frame, as well as a fourth reaction wheel mounted such that it has equal components along the three axes. The nominal torque distribution matrix is given as

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & \sqrt{\frac{1}{3}} \\ 0 & 1 & 0 & \sqrt{\frac{1}{3}} \\ 0 & 0 & 1 & \sqrt{\frac{1}{3}} \end{bmatrix} \tag{7.2}$$

This torque distribution matrix is used when distributing the torque calculated in the control system block to the reaction wheels. To simulate an uncertainty in the assembly of the reaction wheels, the torque distribution matrix representing the true positions of the reaction wheels is perturbed by a small value on each column, before each column is normalized.

7.6 Performance

To compare the efficacy of the different controllers for each of the control objectives, three measures will be compared: settling time, accuracy, and steady-state error. Settling time is the time it takes for a system to reach the desired output value, within a given error band. The accuracy is taken as the RMSE value from the time the system has settled. The accuracy measurements is used as a measurement on the error in attitude when pointing, and the on the error in angular velocity when slewing. The steady-state error is taken as the absolute error in the final point of each simulation.

7.7 Setup

The setup of the Simulink model is shown in figure 7.1.

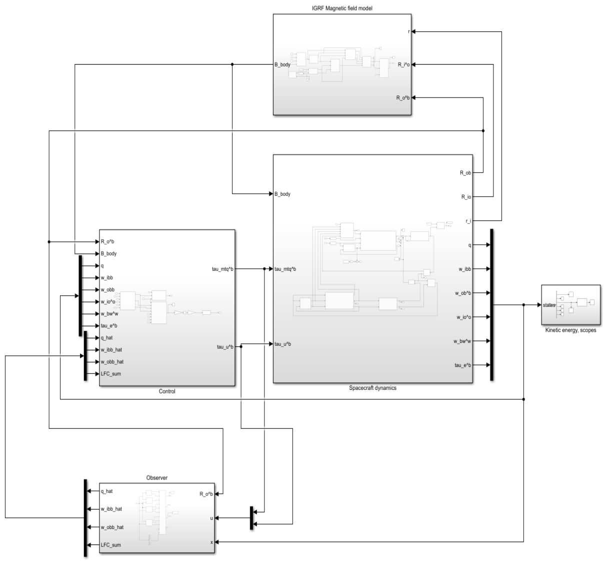


Figure 7.1: Top level of the Simulink model.

The control block produces a torque for both the magnetorquers and the reaction wheels, using the control laws designed in chapter 5. The observer block estimates the states from the satellite. In the satellite dynamics block, the dynamics of the satellite are simulated. This includes the attitude kinetics, attitude kinematics, as well as the orbital mechanics. The insides of the satellite dynamics block is shown in figure 7.2.

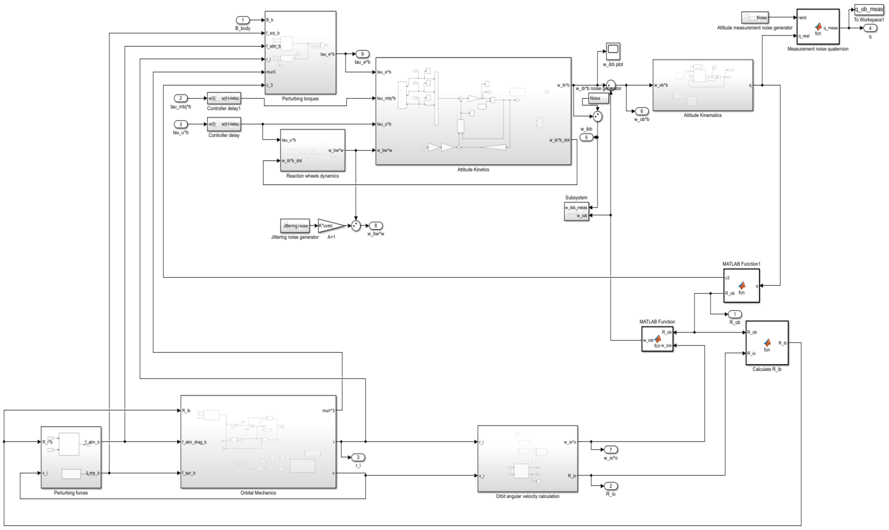


Figure 7.2: Top level of the Simulink model.

The sampling rate for the simulation is set to 4Hz, as this is the sampling rate that will be used on the HYPSO mission due to limitations due to the star tracker.

Chapter 8

Results and Discussion

The results from the simulations are presented with and without time delays in the system, with and without an observer present, and for both the target pointing and the slewing control objective. The results from each of the controllers designed in chapter 5 are examined. The B-dot algorithm, section 5.1, the algorithm used in the specialization project for detumbling, is shown in section 8.1. In section 8.2, a discussion of the available observers is performed. Section 8.6 shows the effect of the momentum dumping algorithm given in section 5.2. The effect of uncertainty in the assembly of the reaction wheels is shown in section 8.7. In the final section, section 8.8, target pointing and slewing are put together for the nominal operations that will be run with the HSI on the HYPSO satellite.

The controllers are compared based on the performance measures defined in section 7.6 in section 8.3 and 8.4. Both target pointing and slewing are analyzed. When pointing is analyzed, the initial attitude is set to be [20°;0;10°], and after a given time, the reference is set to [-20°;0;0]. For the slewing plots, the angular velocity reference starts at zero, then the second channel is set to the desired slew rate, which is 0.7238°/s. Note that the error measures are defined for errors in attitude for pointing, and for errors in angular velocity for slewing. The error bands used for settling time are defined to be 0.5° for pointing and 0.0004 rad/s for slewing throughout the chapter. Due to the number of results in sections 8.3 and 8.4, some of the discussion will be contained in the paragraphs named *Comparison*, along with a table showing the performance of the controllers. A discussion summarizing the results of these two sections is found in section 8.5.

The measurement noise is turned on for all simulations in section 8.3 and 8.4, and the momentum dumping algorithm on the magnetorquers is active. The time delay is, when in use, 0.25 seconds.

The group of plots that show the results come in two variants. The state variable plots show the $\boldsymbol{\epsilon}_{ob}$ and $\boldsymbol{\omega}_{ob}^{b}$ values together with the reference values, in addition to the quaternion error. The difference between the pointing and slewing variant of the plots is that the pointing variant has the $\boldsymbol{\epsilon}_{ob}$ plot on top, whereas the $\boldsymbol{\omega}_{ob}^{b}$ will be on top for slewing. The torque plot shows the angular velocity of the reaction wheels, the torque input *to* the reaction wheels and the torque provided *by* the magnetorquers. Any plot showing estimated values has the estimated values in dashed lines. For the observers, the $\boldsymbol{\epsilon}_{ob}$ plot is above the $\boldsymbol{\omega}_{ob}^{b}$ plot.

8.1 Detumbling

To test the detumbling of the satellite, the simulation is initialized with a significant value for $\boldsymbol{\omega}_{ib}^{b}$, set at [0.1;0.4;0.9] radians per second. The B-dot algorithm is used on the magnetorquers, while the control of the reaction wheels have been disabled, simulating the conditions of the detumbling phase of the satellite after launch.

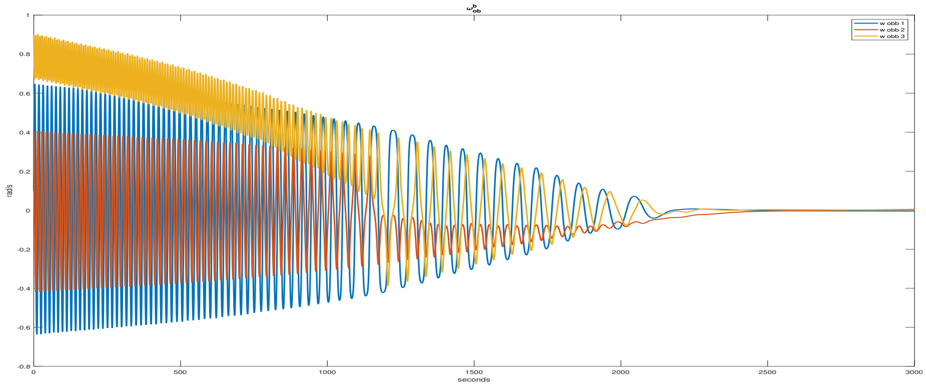


Figure 8.1: Detumbling of angular velocity

The B-dot algorithm manages to detumble the high initial angular velocity over around forty minutes. As is clear from the figure, the detumbling algorithm works well.

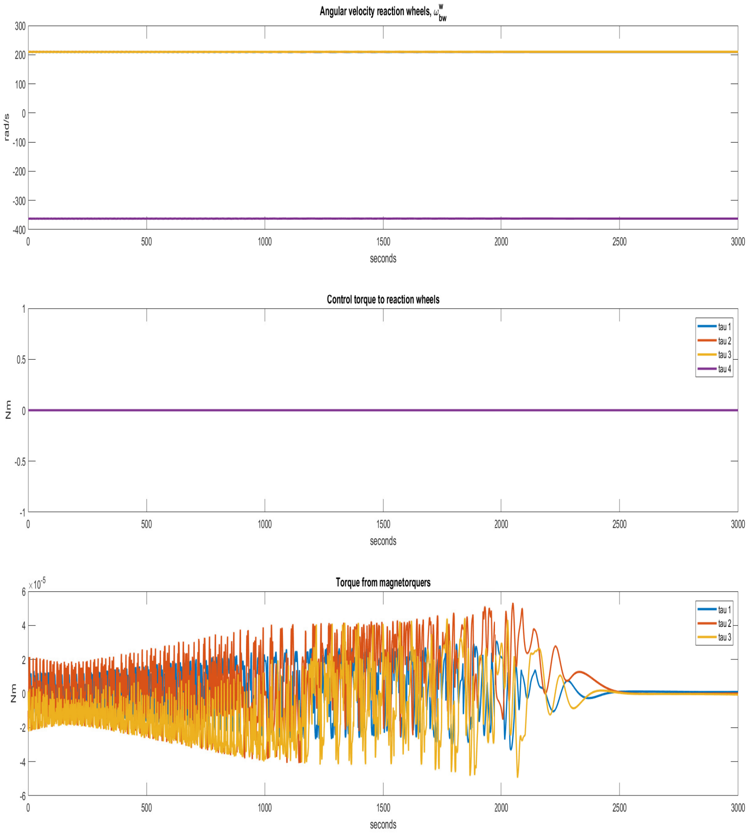


Figure 8.2: Torque from the magnetorquers during detumbling. The torque to the reaction wheels are included to show that the magnetorquers are the only actuations in use.

8.2 Observer results

The observers used in this thesis estimate the vector part of the quaternion, $\boldsymbol{\epsilon}$. From these three coordinates the final coordinate, the scalar part η, is to be reconstructed. η is calculated from the estimated $\boldsymbol{\epsilon}$ in the different observers using the identity $\eta^2 + \boldsymbol{\epsilon}^\intercal \boldsymbol{\epsilon} = 1$. When the simulator propagates the quaternion value using equation 3.4, the η value reaches both positive and negative values. In the function that reconstructs η from $\boldsymbol{\epsilon}$, the positive solution to $\eta = \sqrt{1 - \boldsymbol{\epsilon}^\intercal \boldsymbol{\epsilon}}$ is always chosen. This leads to a discontinuity when η reaches zero during slewing, as the wave representing η is rectified, as opposed to following a sinusoidal wave. The kinematic equation, equation 3.4, uses both η and $\boldsymbol{\epsilon}$. This equation is used by both the first order sliding mode observer and the higher order sliding mode observer in chapter 6, based on their estimates of the given attitude, to give an estimate of the angular velocity and epsilon rate, respectively. During a slewing maneuver, these estimates exhibit strange behavior around $\eta = 0$, where the signal becomes mirrored about the 0 line, as seen in figure 8.3, where the calculated angular velocity based on $\dot{\eta}_{ob}$ in the HOSMO is shown.

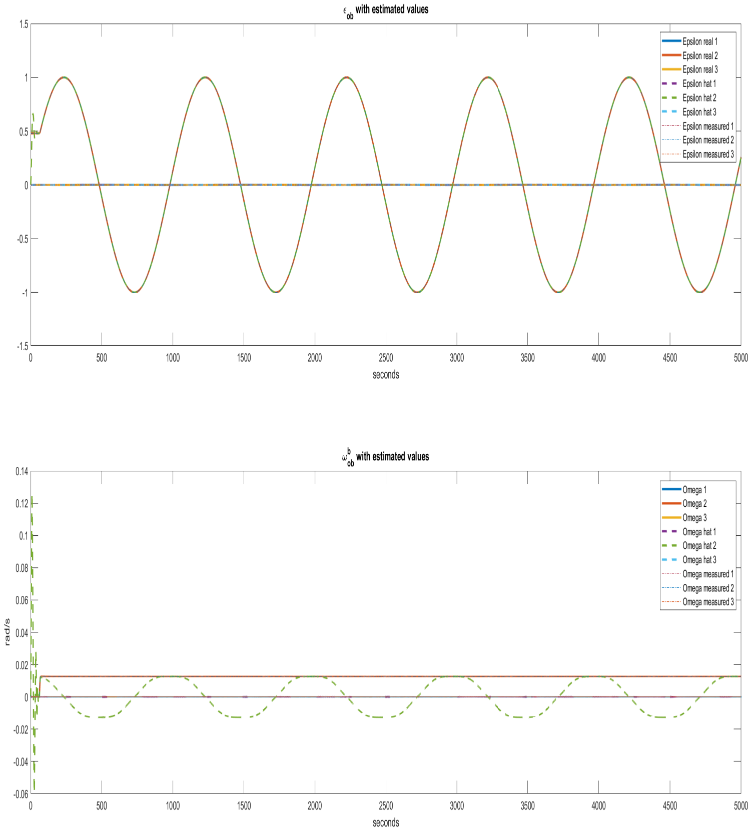


Figure 8.3: HOSMO during slewing.

Figuring out a way to fix it is a task for future work.

Due to the shortcomings of angular velocity estimates of the sliding mode observers that are based on the quaternion measurements, the sliding mode observer using the measurements from the IMU, from equation 6.2, will be used together with one of the other sliding mode observers for $\boldsymbol{\epsilon}$ estimation. The $\eta = 0$ problem does not appear in the estimates for ϵ for either the sliding mode observer or the higher order sliding mode observer.

The errors of the SMO for ϵ goes toward zero faster than those of the HOSMO, but the error ends up being smaller for the HOSMO. For this reason, the HOSMO estimates for the ϵ values are used together with the SMO based on the IMU measurements.

8.3 Results without time delays

8.3.1 Without observer

8.3.1.1 Pointing

PD control without time delay, without observer. Pointing.

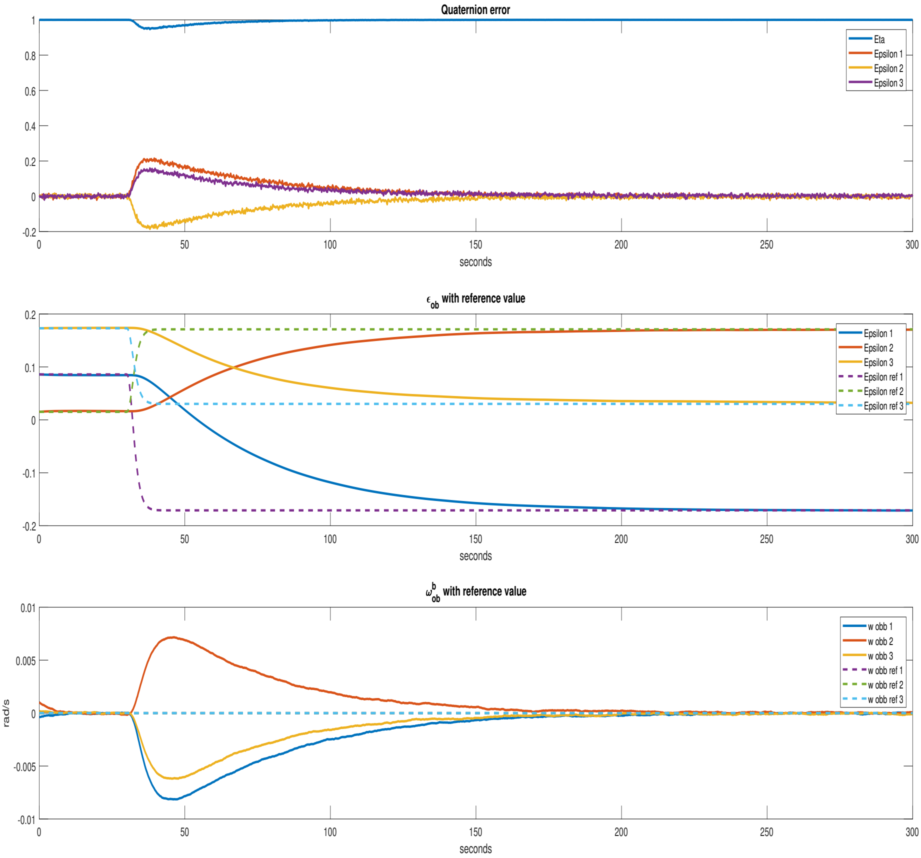
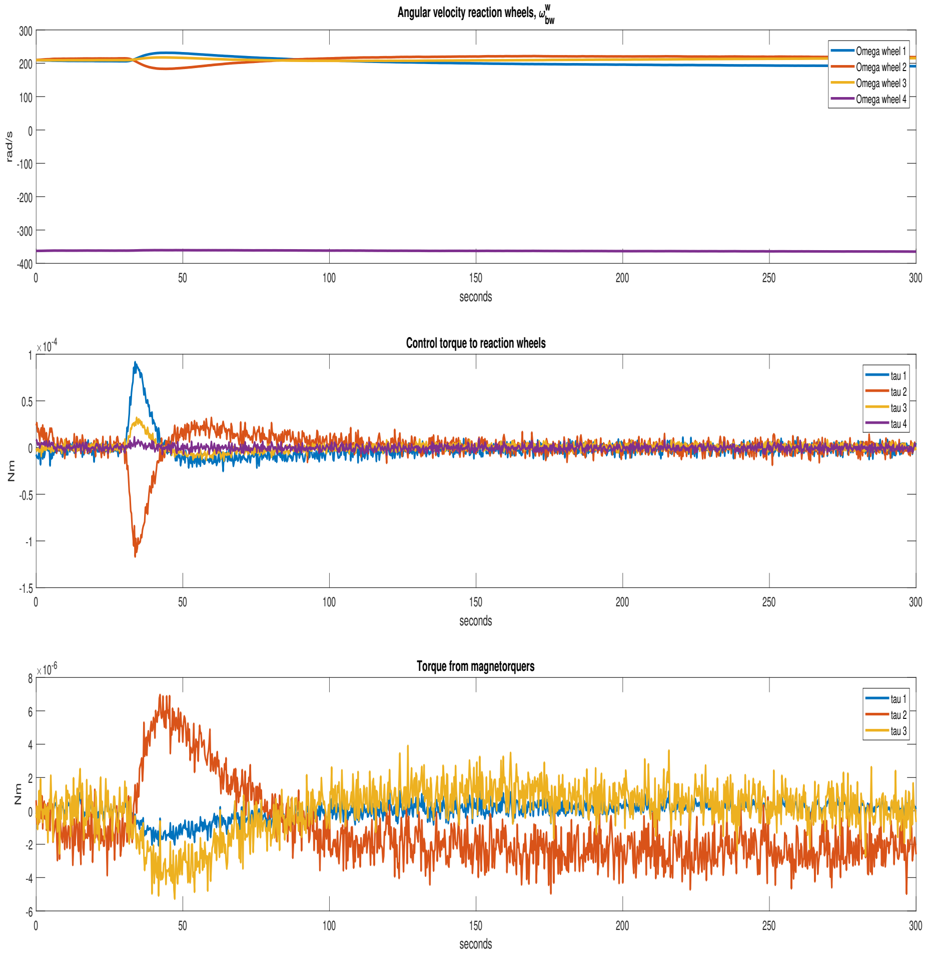


Figure 8.4: PD pointing without delay, without observer. State variables



Angular velocity reaction wheels, ω_{bw}^{w}

Control torque to reaction wheels

Torque from magnetorquers

Figure 8.5: PD pointing without delay, without observer. Torques

LQR without time delay, without observer. Pointing.

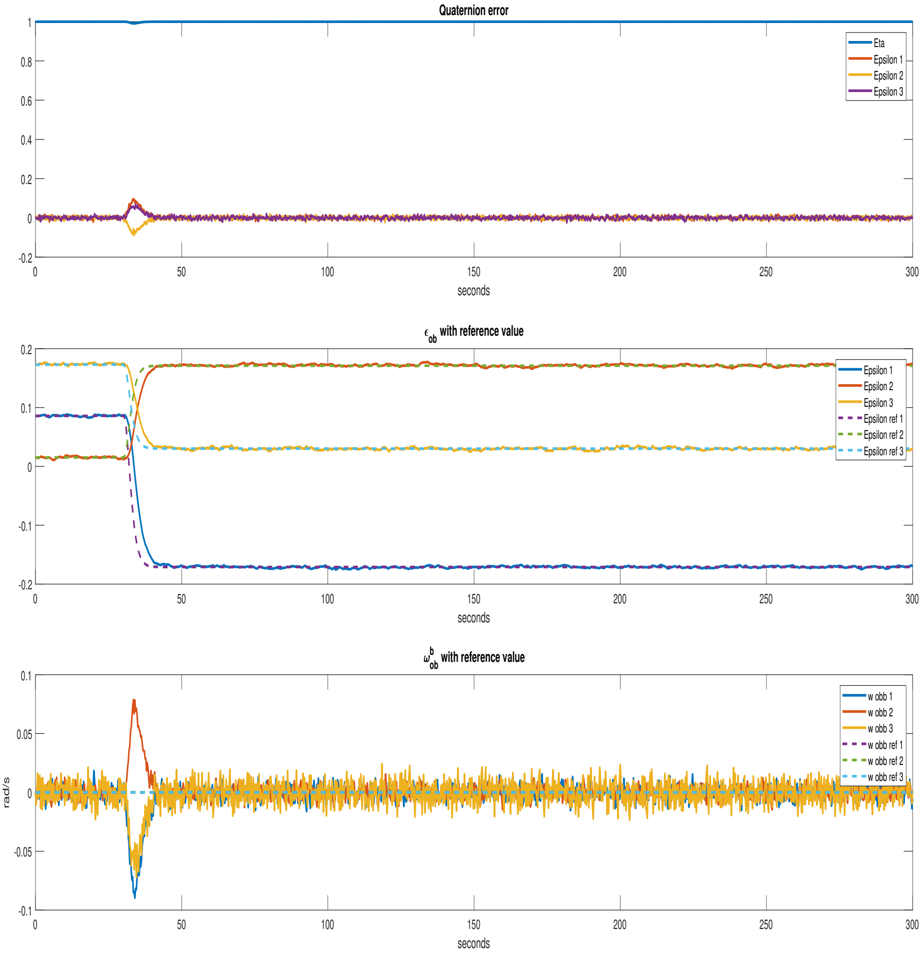


Figure 8.6: LQR pointing without delay, without observer. State variables

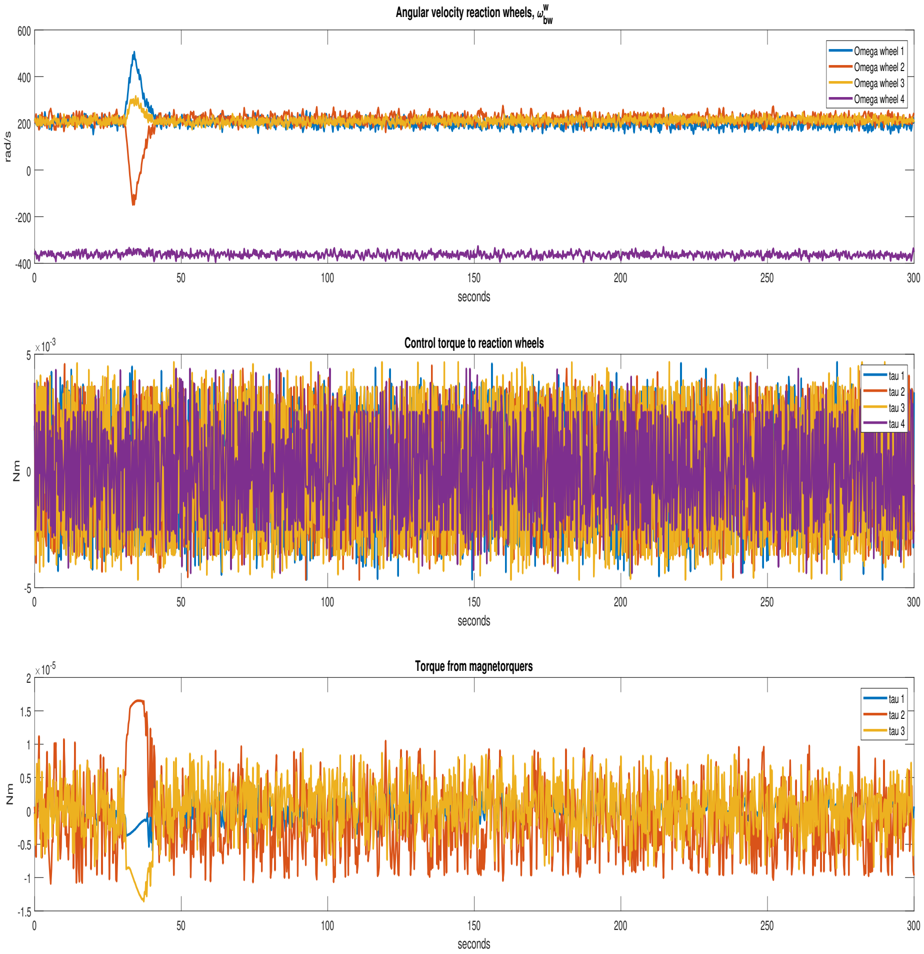


Figure 8.7: LQR pointing without delay, without observer. Torques

Sliding mode control without time delay, without observer. Pointing.

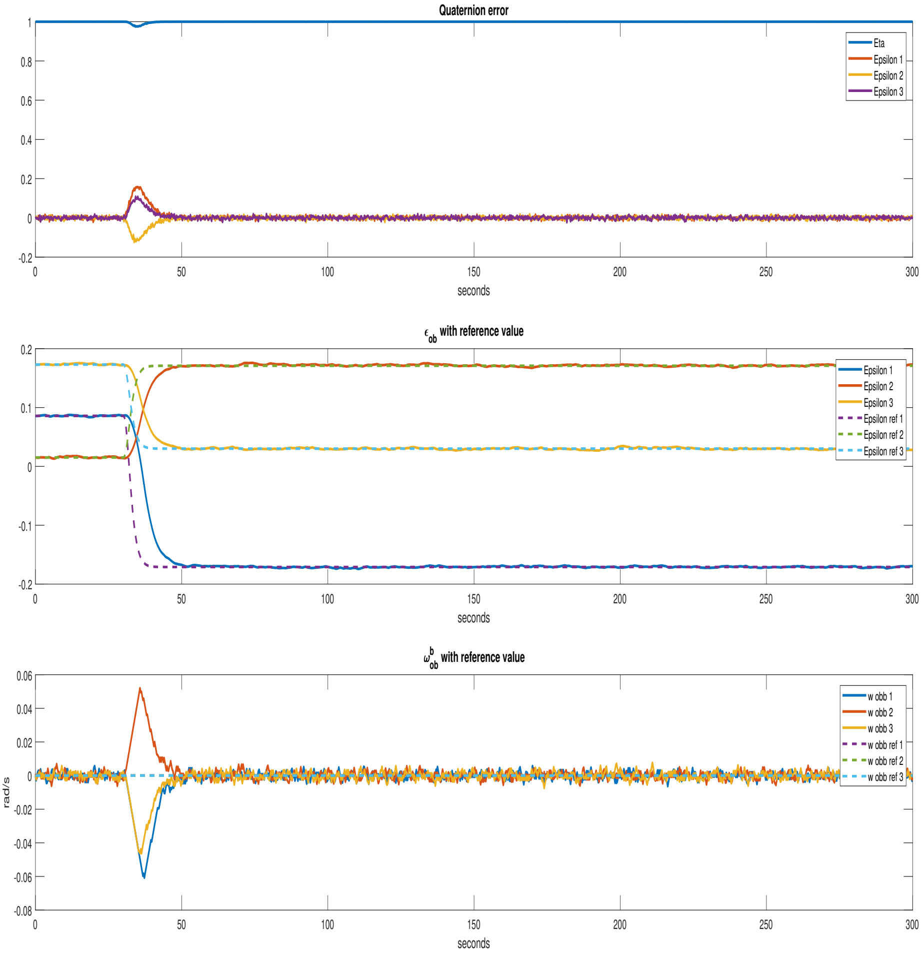


Figure 8.8: SMC pointing without delay, without observer. State variables

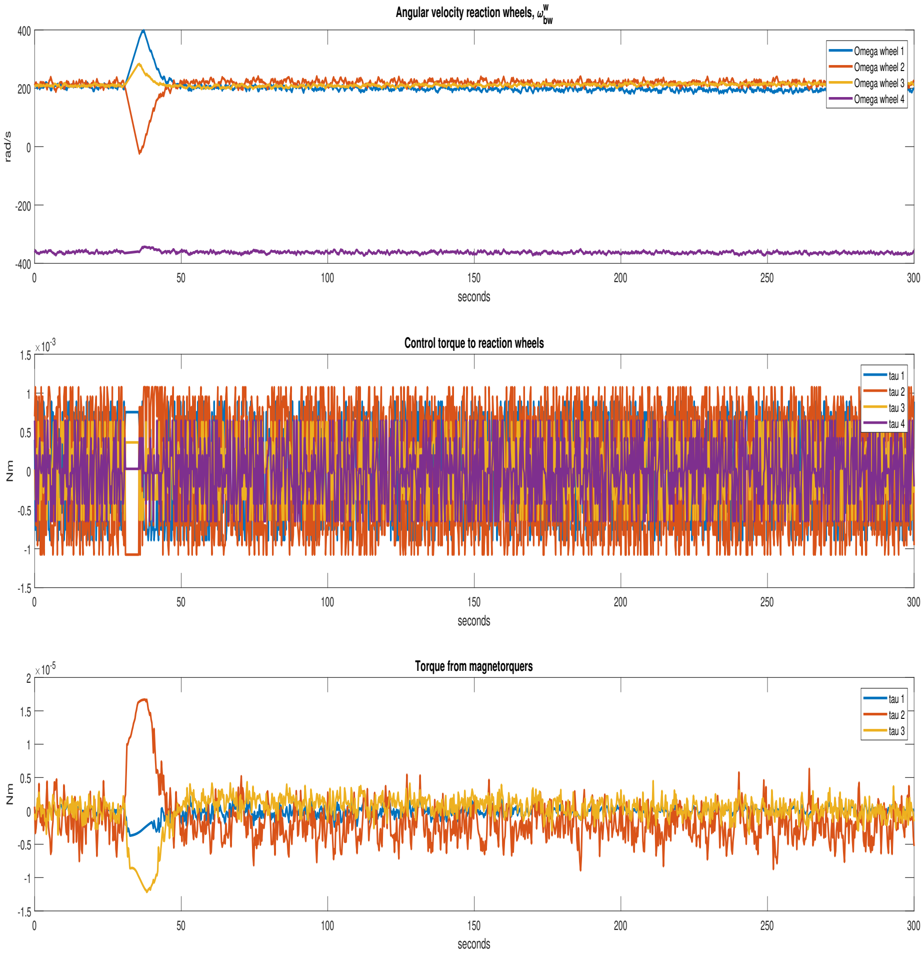


Figure 8.9: SMC pointing without delay, without observer. Torques

GSTA without time delay, without observer. Pointing.

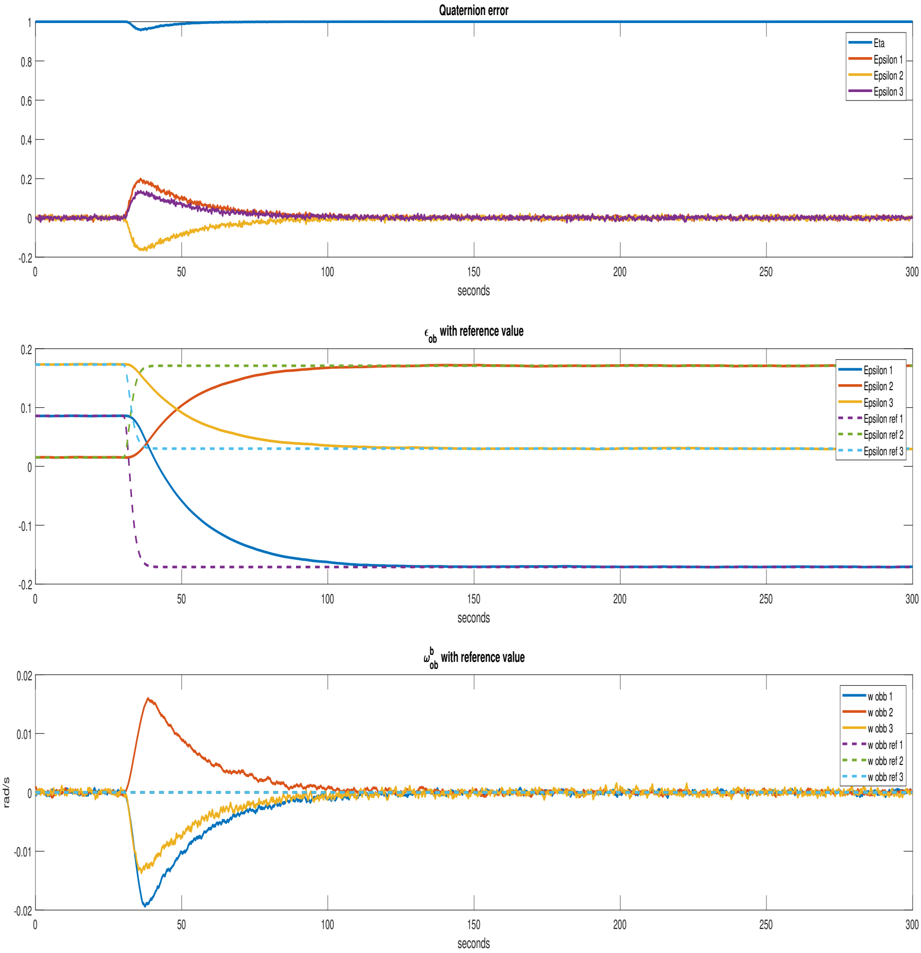


Figure 8.10: GSTA pointing without delay, without observer. State variables

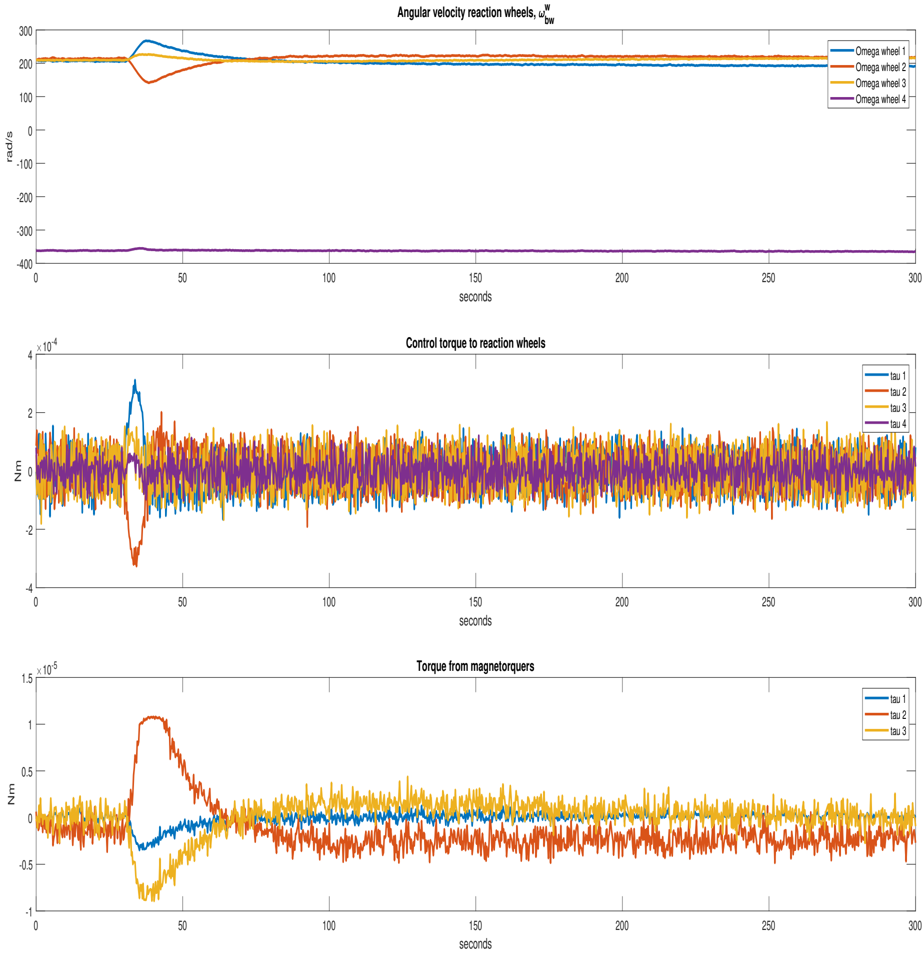


Figure 8.11: GSTA pointing without delay, without observer. Torques

Comparison of the different controllers based on settling time, accuracy, and steady-state error.

	Errors								
	Settling time[s]			RMSE [°]			Steady-state error[°]		
	ϵ_1	ϵ_2	ϵ_3	ϵ_1	ϵ_2	ϵ_3	ϵ_1	ϵ_2	ϵ_3
PD	164.25	148.75	202	0.1892	0.2116	0.3576	0.0244	0.0706	0.2255
LQR	15.5	208.5	267.5	0.1676	0.1872	0.2778	0.0814	0.1751	0.2401
SMC	19.25	45	171.25	0.1347	0.1658	0.1577	0.1206	0.2071	0.2364
GSTA	79	67	73.25	0.1144	0.1055	0.1216	0.0465	0.0265	0.0820

Table 8.1: Error comparison. Pointing, without delay, without observer

Notice that the LQR has the best settling time for the first output channel, the SMC has the best settling times for the second, while the GSTA is the best on the third. From the plots looks like the LQR and then the SMC are the two fastest controllers. However, the SMC and LQR in particular comes out with worse settling times as the control is unable to hold the reference value as well as the two others, and thus the line exits the error band at some point. The control input to the reaction wheels from the SMC and LQR are of higher magnitude than the input for PD and GSTA, also after the plots indicate they have settled. This may give rise to the behavior channel escapes the error band and settles again. Relaxing the size of the error band would have made the settling time numbers look better for both the LQR and the SMC.

The steady-state error can vary when the signal experience noise such as the SMC and LQR experience in some form. This is due to the measurement rather than the controller: The steady-state error is measured at one point. Choosing a different property for performance measuring could have been preferable. However: It would work well for a signal that are robust to noise, i.e. signals that would stay at their steady-state value when settled. The largest value for steady-state error is for the first output channel for PD control. Given the long settling time of the PD controller, it is possible this larger value is due to simulation being stopped while it was still converging.

When it comes to RMSE, the GSTA is the best one. The RMSE is taken from when the system has settled. Notice that there is not much difference between the SMC output channels that immediately settles and the one that jumps back inside the error band after 171.25 seconds.

8.3.1.2 Slewing

PD control without time delay, without observer. Slewing.

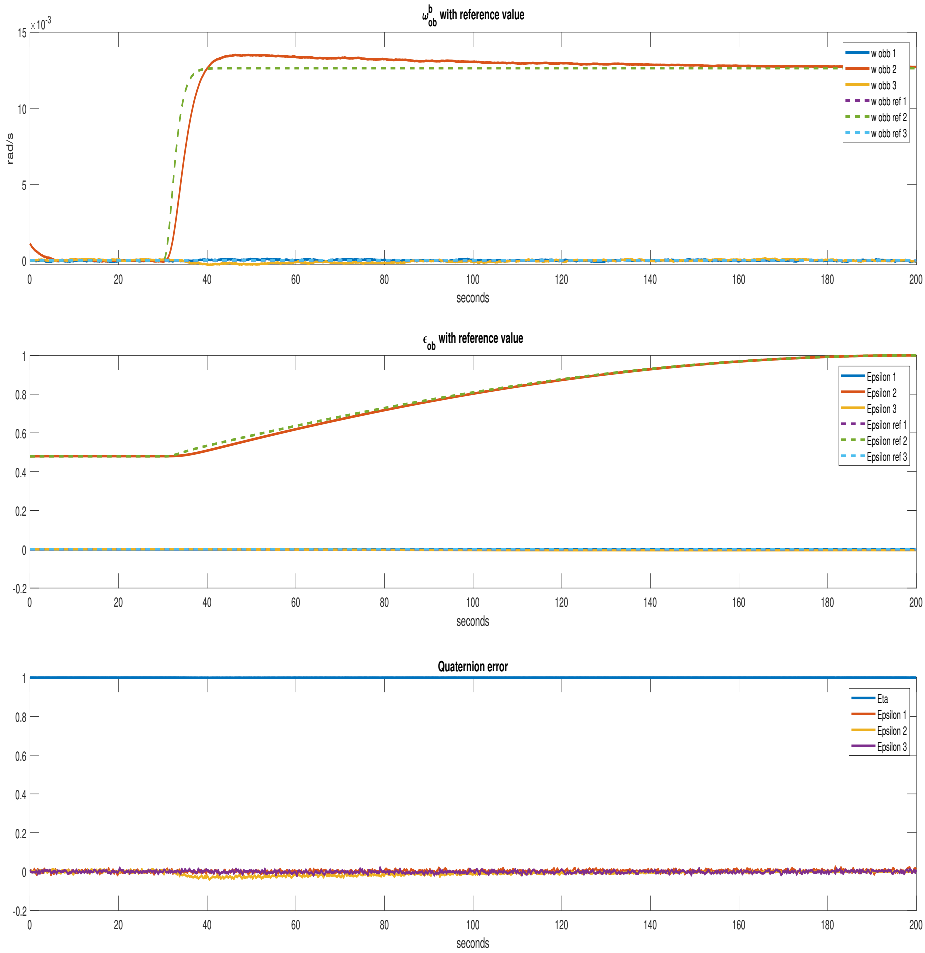


Figure 8.12: PD slewing without delay, without observer. State variables

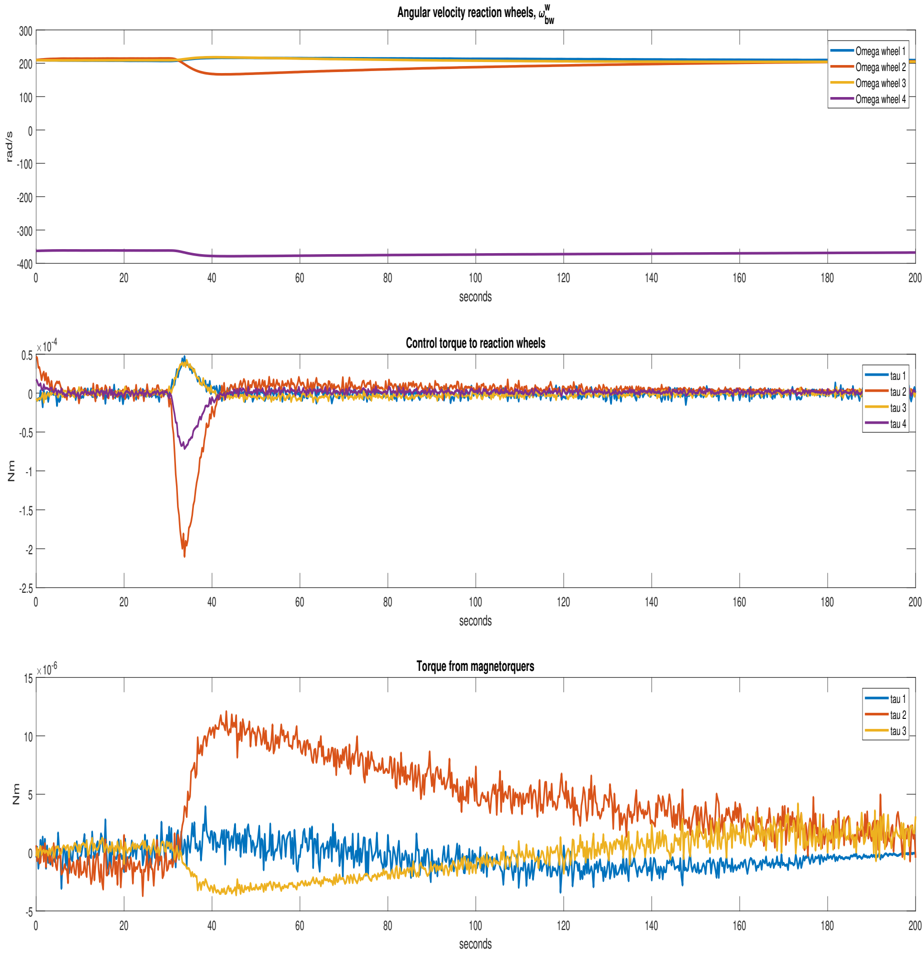


Figure 8.13: PD slewing without delay, without observer. Torques

LQR without time delay, without observer. Slewing.

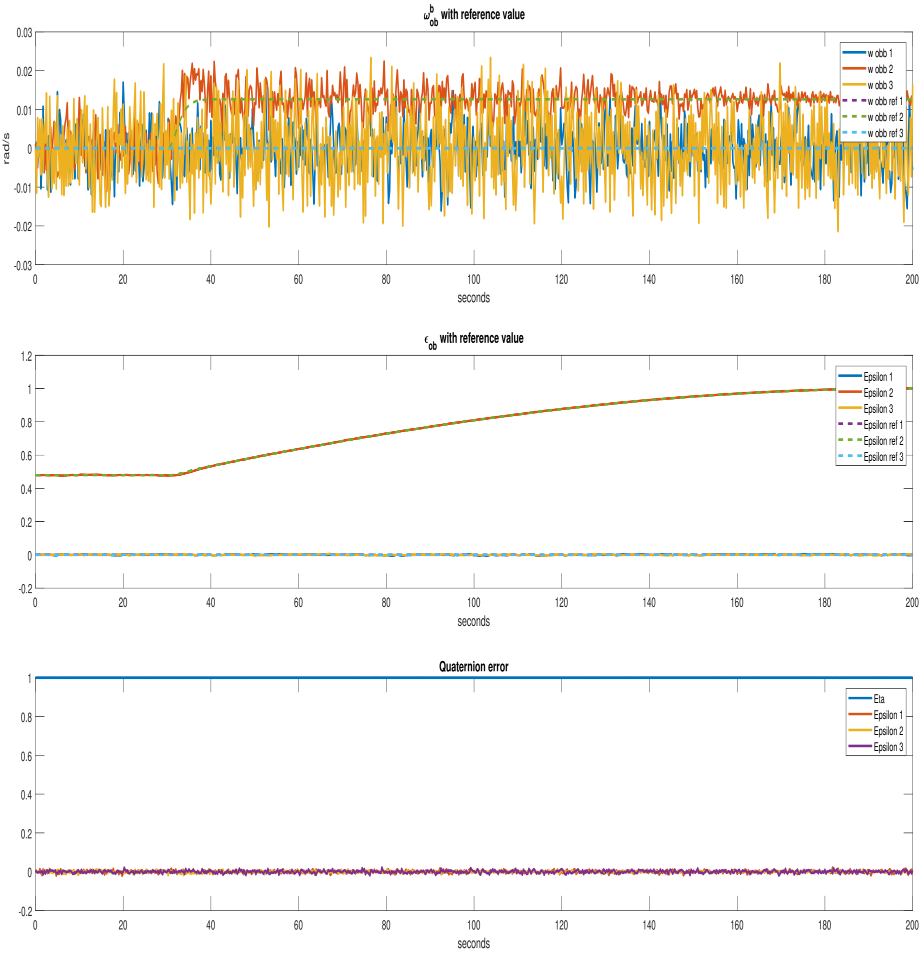


Figure 8.14: LQR slewing without delay, without observer. State variables

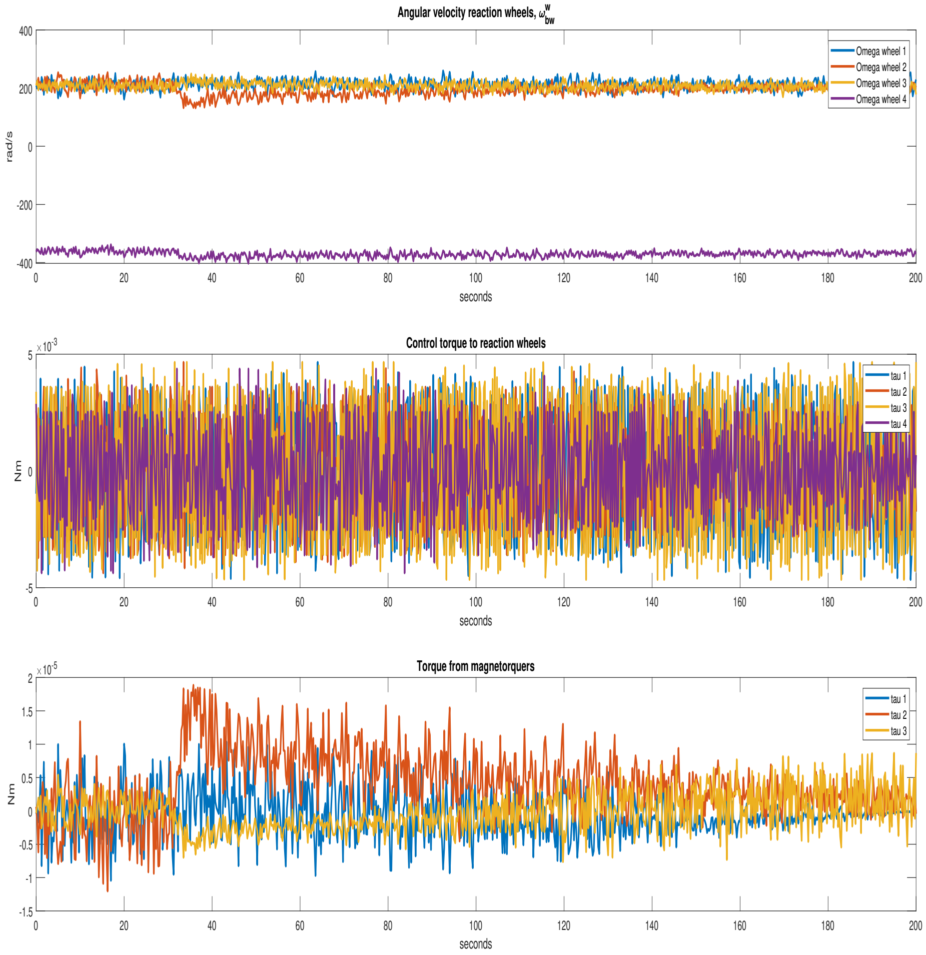


Figure 8.15: LQR slewing without delay, without observer. Torques

Sliding mode control without time delay, without observer. Slewing.

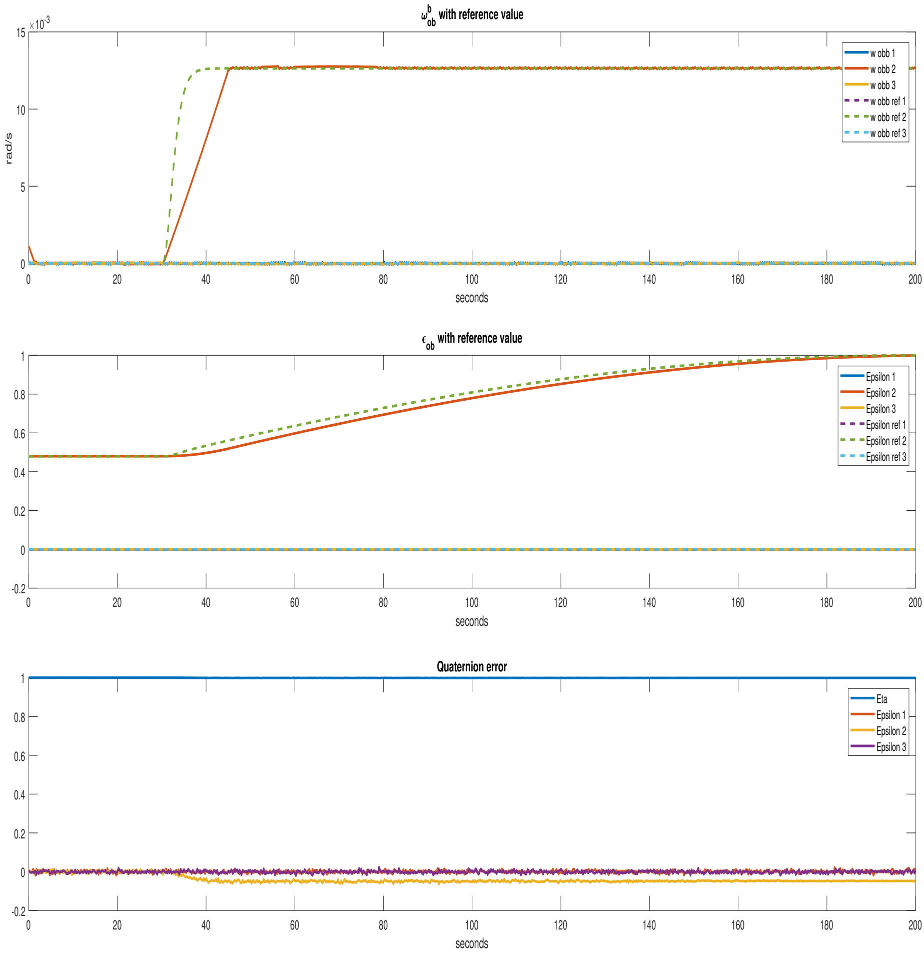


Figure 8.16: SMC slewing without delay, without observer. State variables

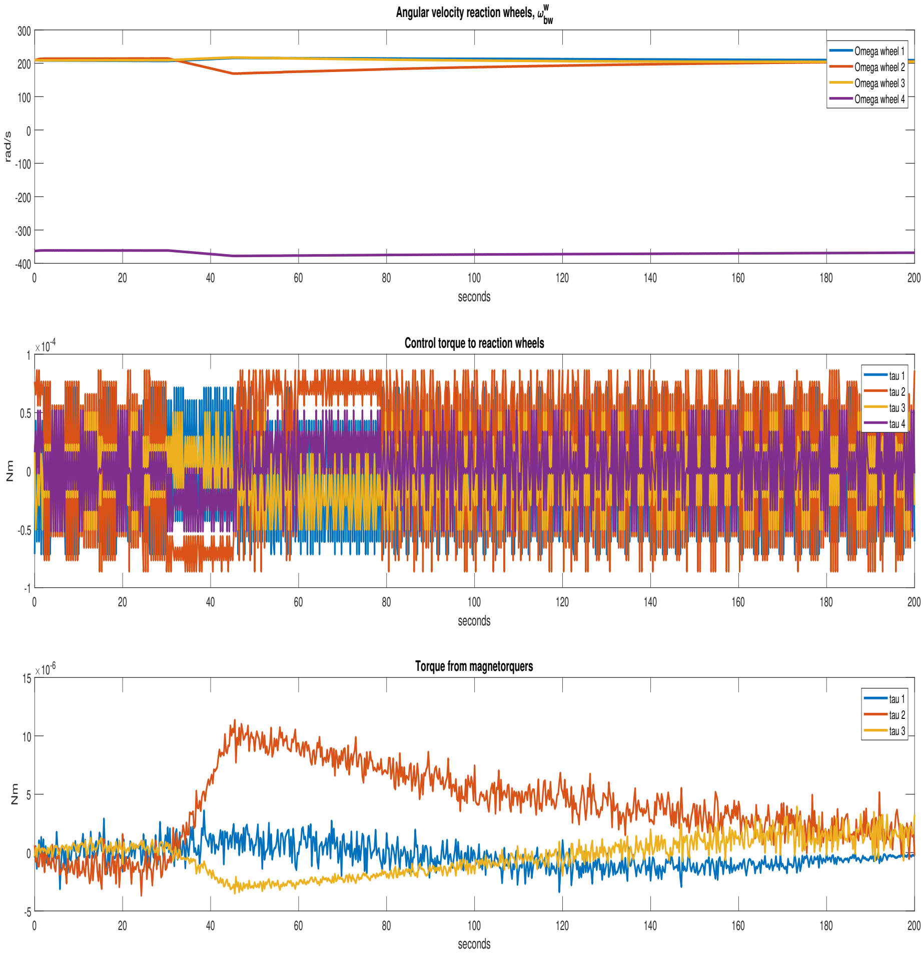


Figure 8.17: SMC slewing without delay, without observer. Torques

GSTA without time delay, without observer. Slewing.

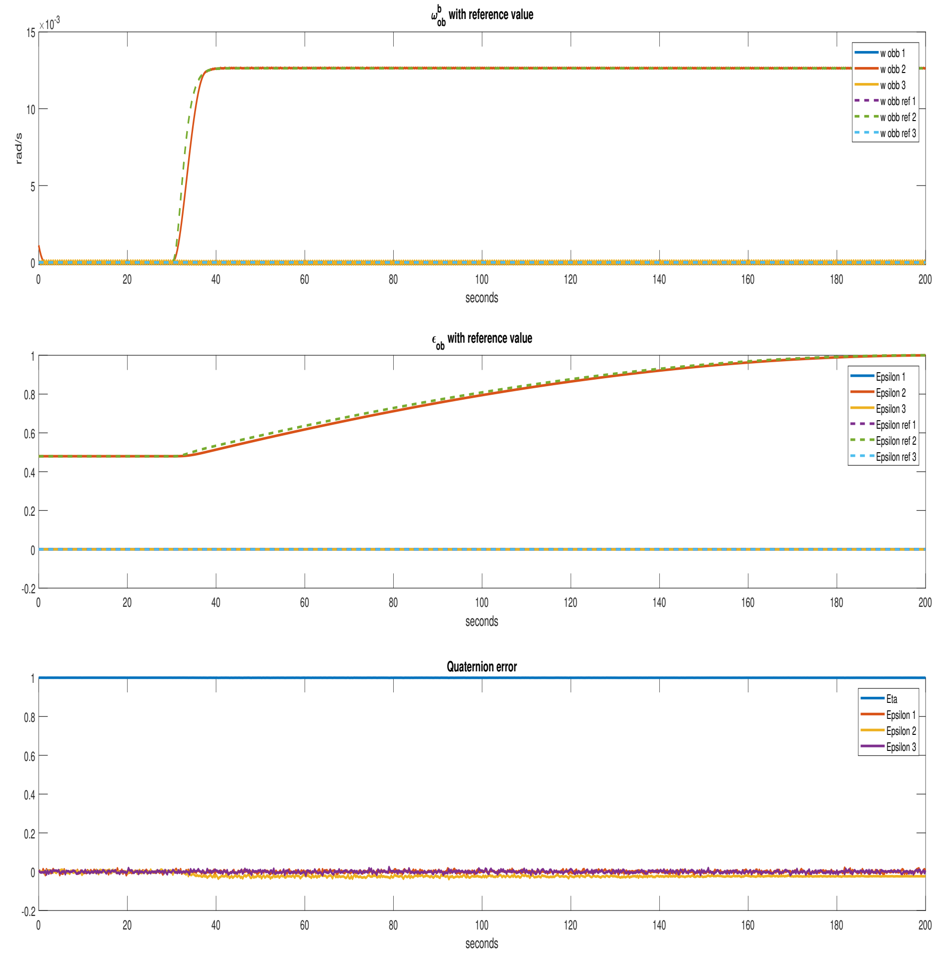


Figure 8.18: GSTA slewing delay, without observer. State variables

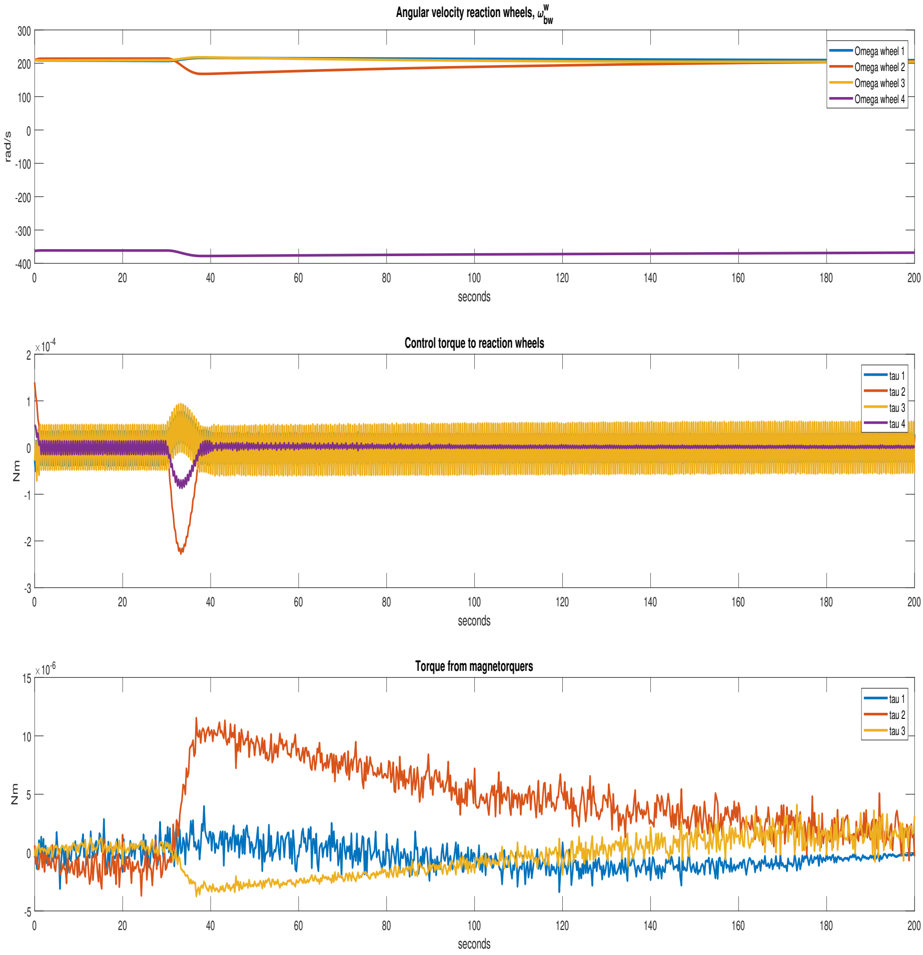


Figure 8.19: GSTA slewing without delay, without observer. Torques

Comparison of the different controllers based on settling time, accuracy, and steady-state error.

	Errors								
	Settling time[s]			RMSE $[\frac{rad}{s}]$			Steady-state error$[\frac{rad}{s}]$		
	$\boldsymbol{\omega}_1$	$\boldsymbol{\omega}_2$	$\boldsymbol{\omega}_3$	$\boldsymbol{\omega}_1$	$\boldsymbol{\omega}_2$	$\boldsymbol{\omega}_3$	$\boldsymbol{\omega}_1$	$\boldsymbol{\omega}_2$	$\boldsymbol{\omega}_3$
PD	-1	71.5	-1	0.388e-4	2.175e-4	1.008e-4	0.902e-4	0.762e-4	0.377e-4
LQR	-	160.5	-	-	1.1e-3	-	5.5e-3	3e-4	1.64e-2
SMC	-1	14.75	-1	0.3897e-4	0.5723e-4	0.4037e-4	0.373e-4	0.44e-6	0.477e-4
GSTA	-1	6.75	-1	0.309e-4	0.201e-4	1.245e-4	0.32e-4	0.224e-4	1.192e-4

Table 8.2: Error comparison. Slewing, without delay, without observer

Entries in the table has been set to - whenever the value does not exists. For example, the first and third channel for the LQR have - as settling time as they do not end up inside the error band during the simulation time. -1 denotes that the value never leaves the error band.

The LQR manages to control towards a slew rate, but it does not manage to hold the other two channels at zero. The signal, also the settled slew rate, is dominated by noise. The GSTA has the best settling time for slewing, followed by the SMC, then PD. The RMSE values are also better for the GSTA, apart from on the third channel, where it is the worst of the three that settled.

The GSTA and SMC goes towards the angular velocity reference, but there is a lag in the epsilon value, showing in the quaternion error plot. This is fine, as regulation towards the angular velocity reference is all that matters for the slewing objective. The PD control, however, has a small overshoot in the angular velocity in order to follow both references, and ends up tracking the reference in both angular velocity and epsilon.

8.3.2 With observer

8.3.2.1 Pointing

PD control without time delay, with observer. Pointing.

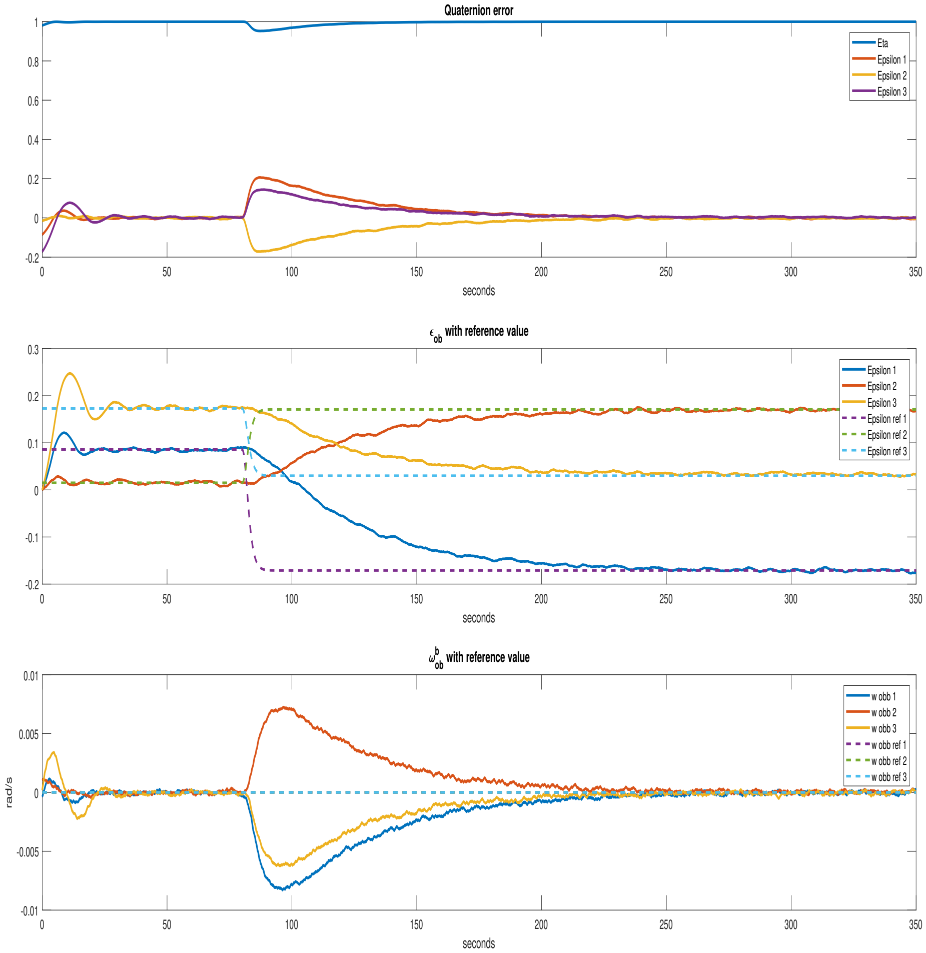


Figure 8.20: PD pointing without delay, with observer. State variables

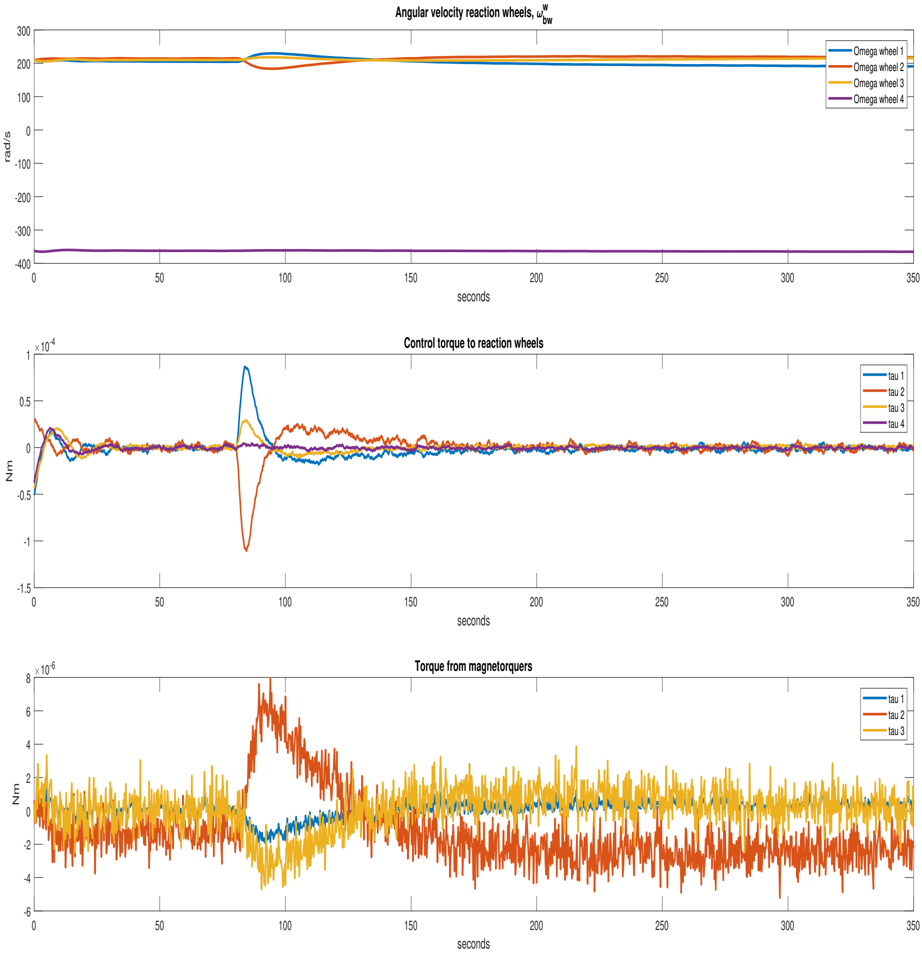


Figure 8.21: PD pointing without delay, with observer. Torques

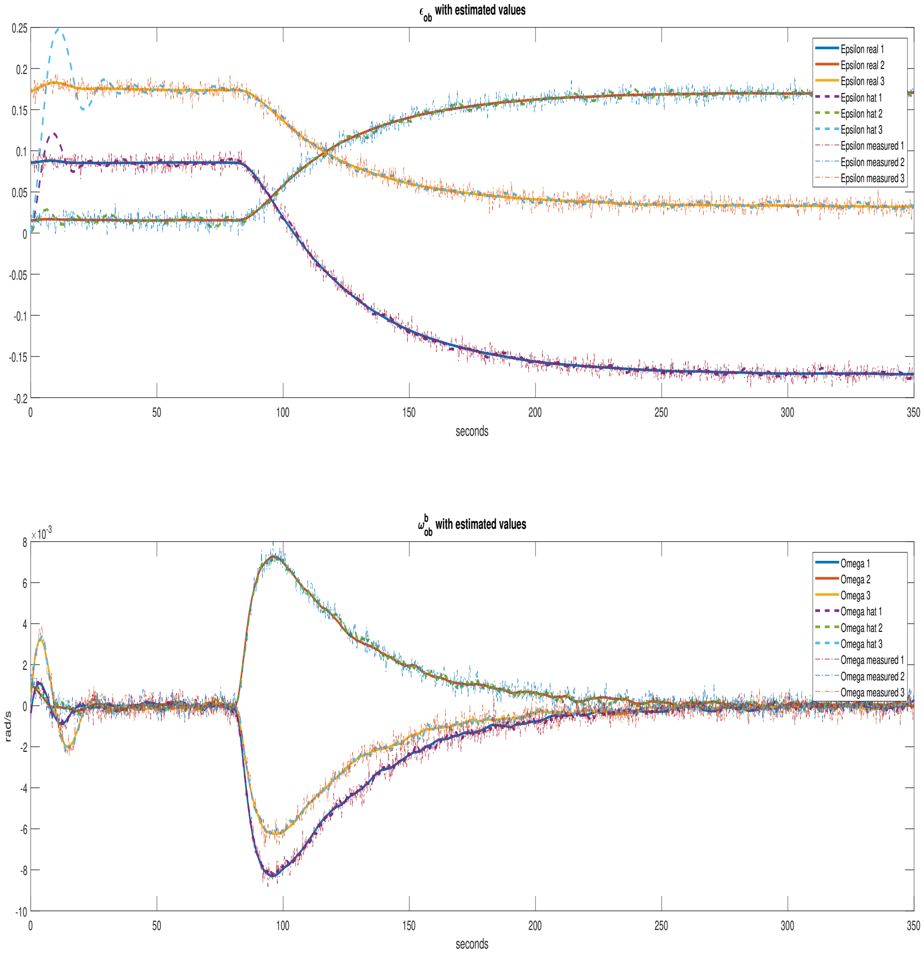


Figure 8.22: PD pointing without delay, with observer. Estimated values

LQR without time delay, with observer. Pointing.

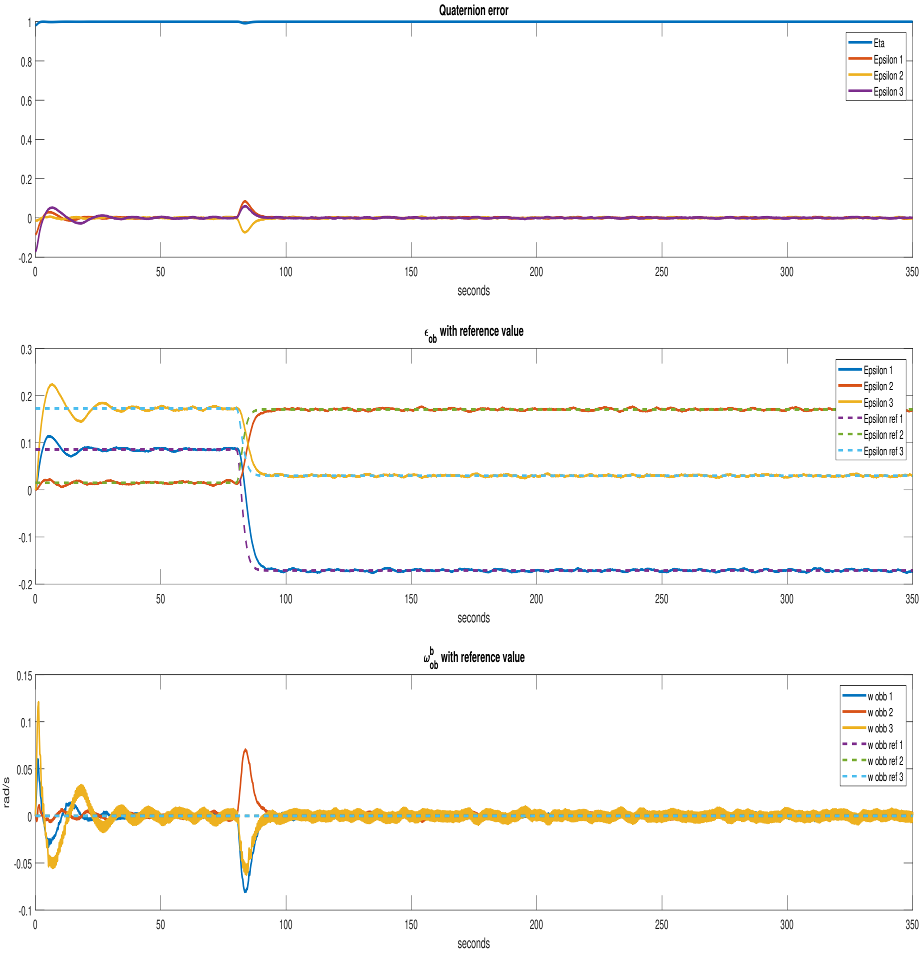


Figure 8.23: LQR pointing without delay, with observer. State variables

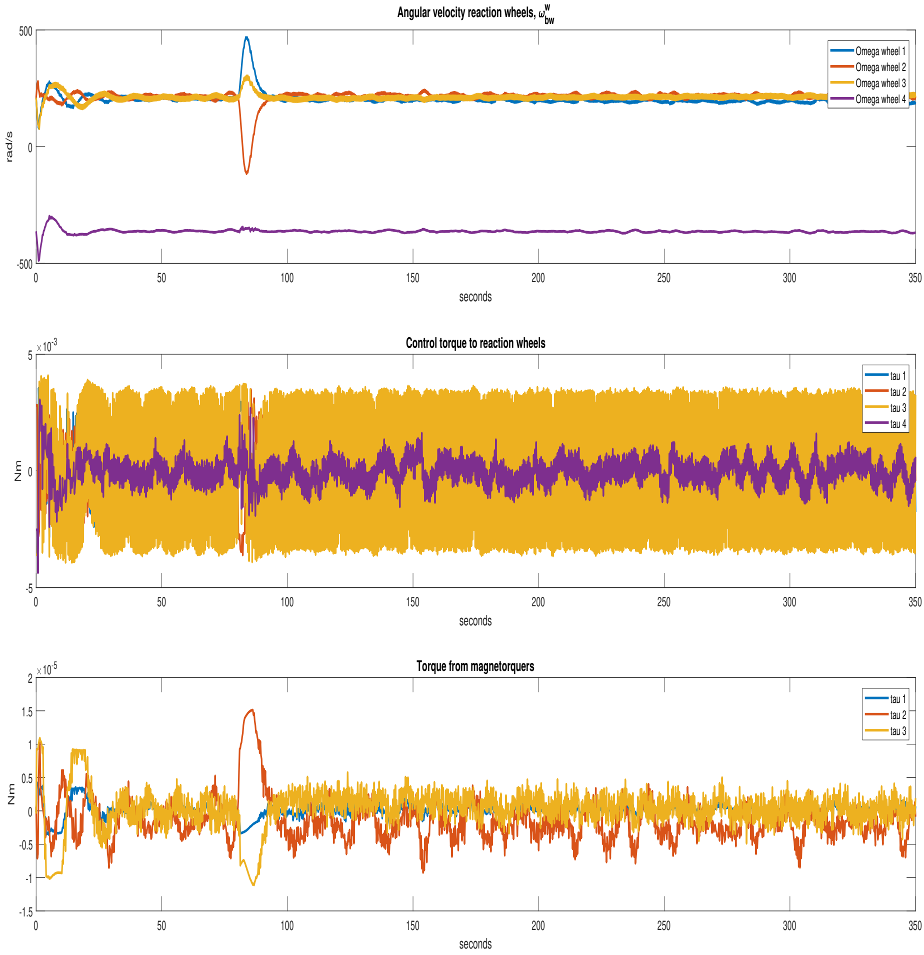


Figure 8.24: LQR pointing without delay, with observer. Torques

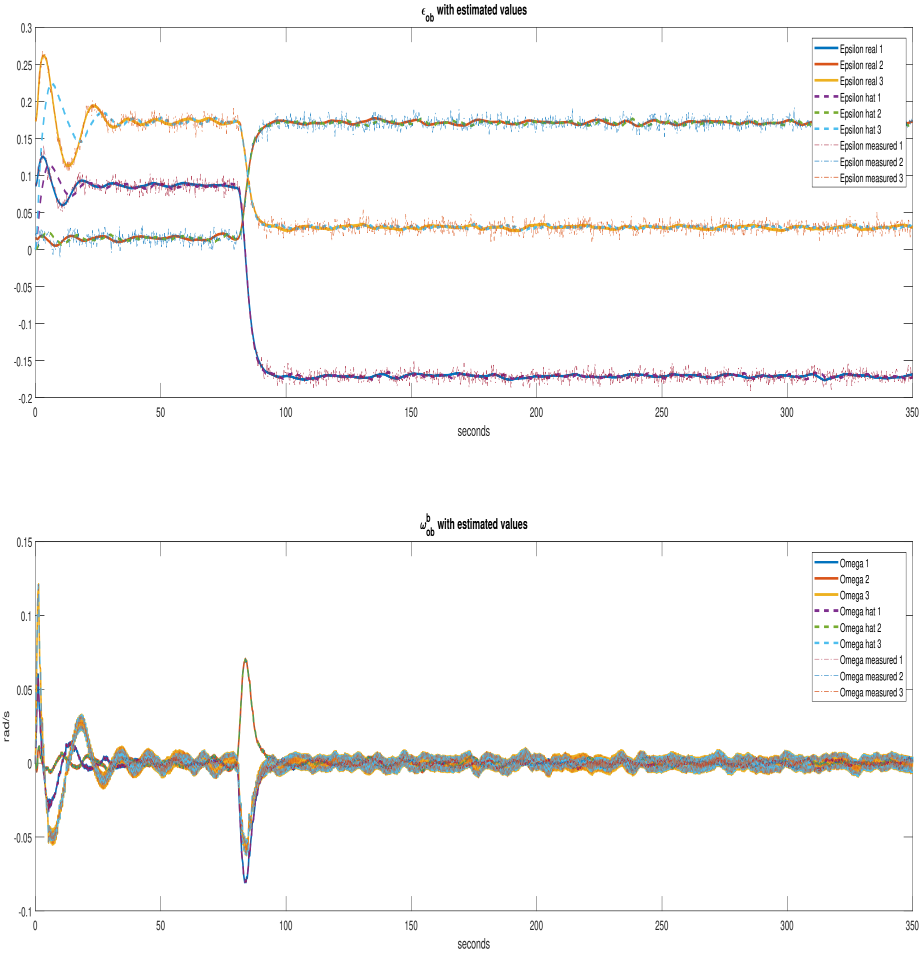


Figure 8.25: LQR pointing without delay, with observer. Estimated values

Sliding mode control without time delay, with observer. Pointing.

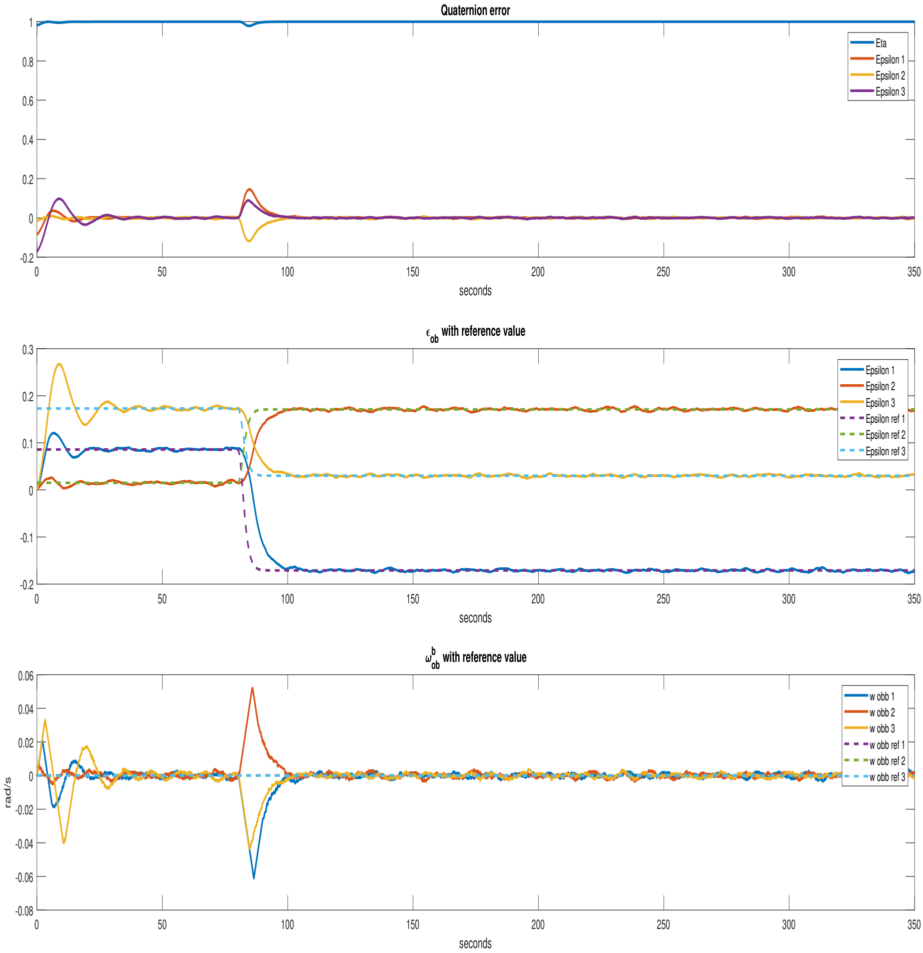


Figure 8.26: SMC pointing without delay, with observer. State variables

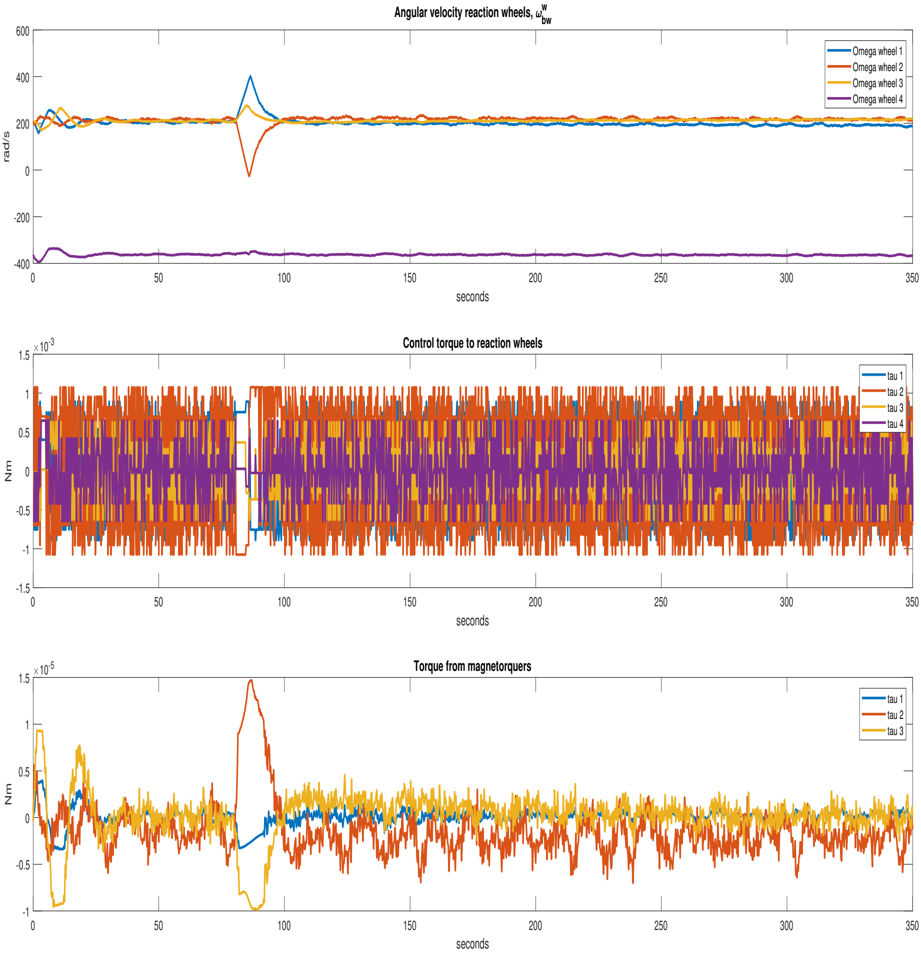


Figure 8.27: SMC pointing without delay, with observer. Torques

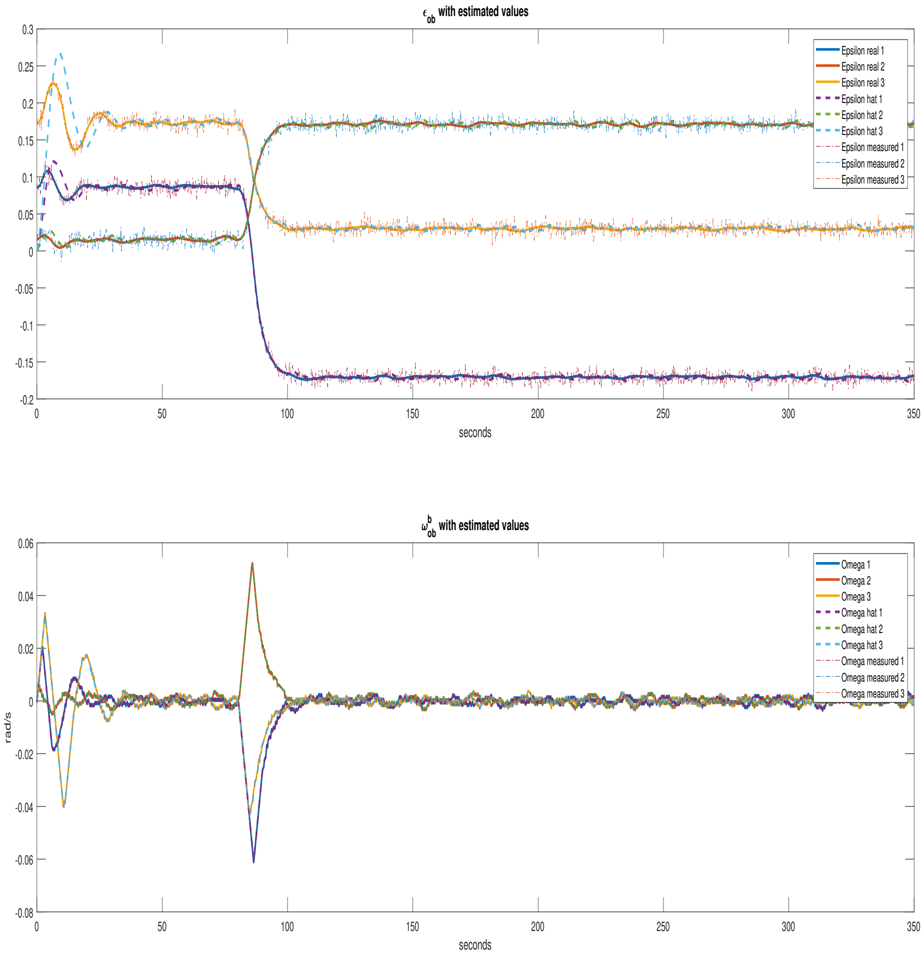


Figure 8.28: SMC pointing without delay, with observer. Estimated values

GSTA without time delay, with observer. Pointing.

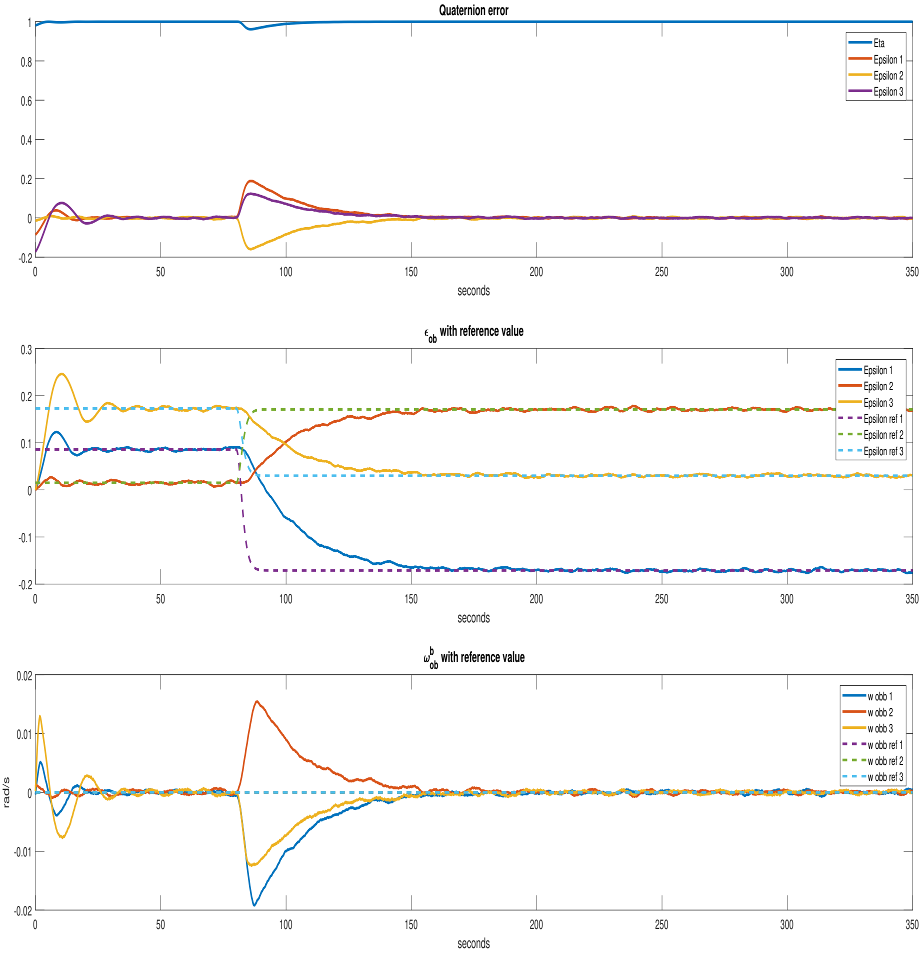


Figure 8.29: GSTA pointing without delay, with observer. State variables

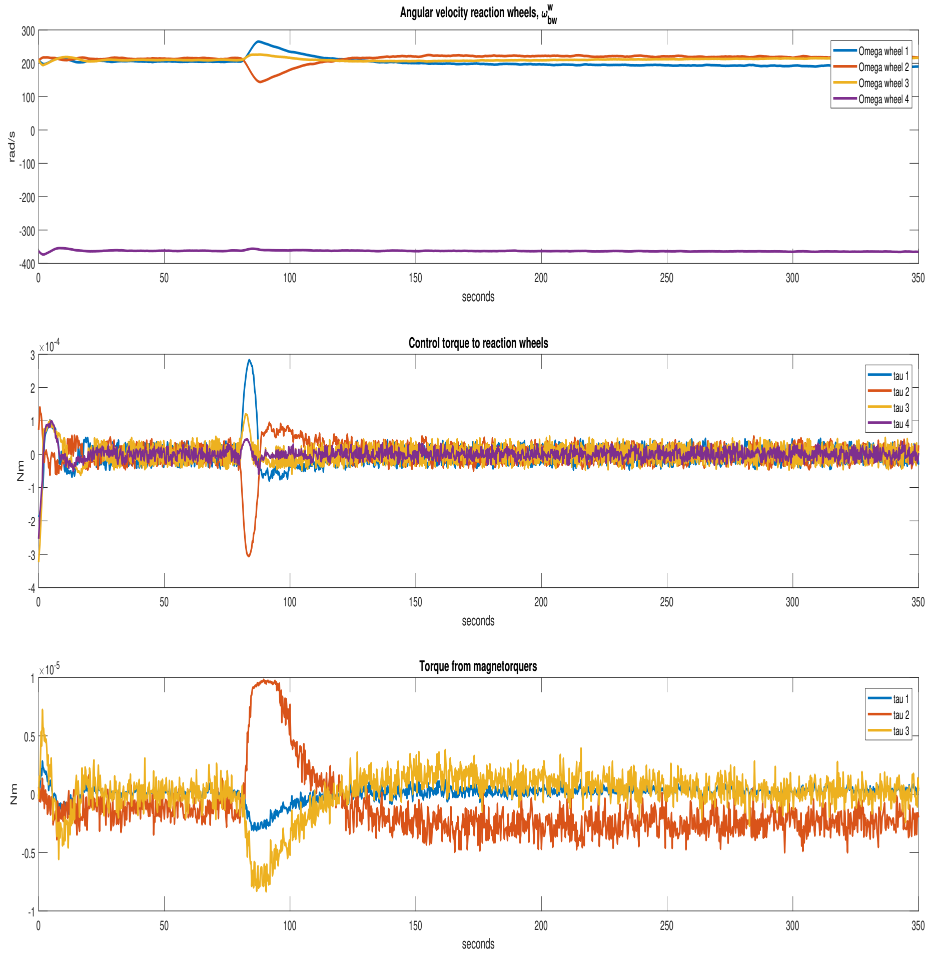


Figure 8.30: GSTA pointing without delay, with observer. Torques

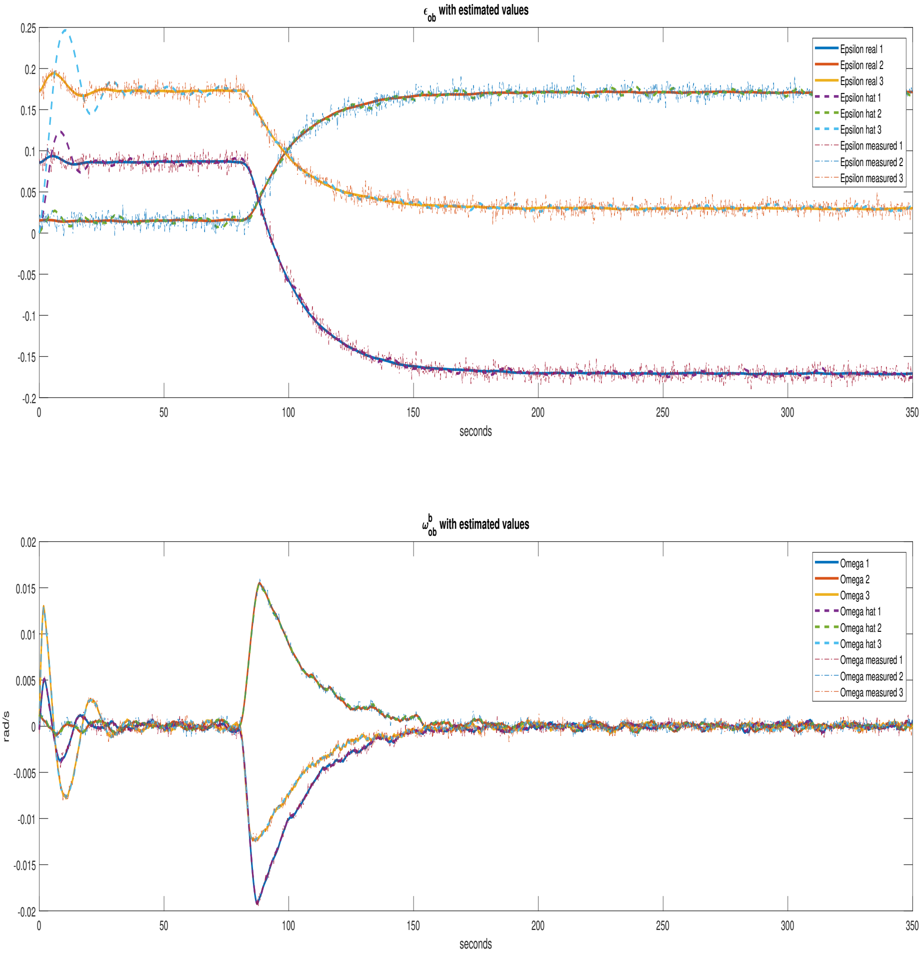


Figure 8.31: GSTA pointing without delay, with observer. Estimated values

Comparison of the different controllers based on settling time, accuracy, and steady-state error.

	Errors								
	Settling time[s]			RMSE [°]			Steady-state error[°]		
	ϵ_1	ϵ_2	ϵ_3	ϵ_1	ϵ_2	ϵ_3	ϵ_1	ϵ_2	ϵ_3
PD	166	147.75	169.75	0.1932	0.2099	0.3596	0.0551	0.0360	0.2285
LQR	236	58.25	218.25	0.2106	0.2144	0.2123	0.2631	0.2158	0.0991
SMC	21	57.75	17.25	0.1484	0.1601	0.1623	0.2748	0.1107	0.0741
GSTA	92	70.75	71.25	0.1315	0.1247	0.1304	0.0251	0.0578	0.0032

Table 8.3: Error comparison. Pointing, without delay, with observer

Compared to when the observer was not in use, the SMC does not jump out of the error band after the first settling. The same is the case for the second LQR channel. The third channel of the LQR settles faster, while the first jumps out of the error band and in again for the final time at 236 seconds. The PD control settles faster. As for the RMSE values, they are slightly worse for the PD controller, worse for all LQR channels apart from the third one, where it has improved, similar but slightly worse for the SMC, and worse for the GSTA.

8.3.2.2 Slewing

PD control without time delay, with observer. Slewing.

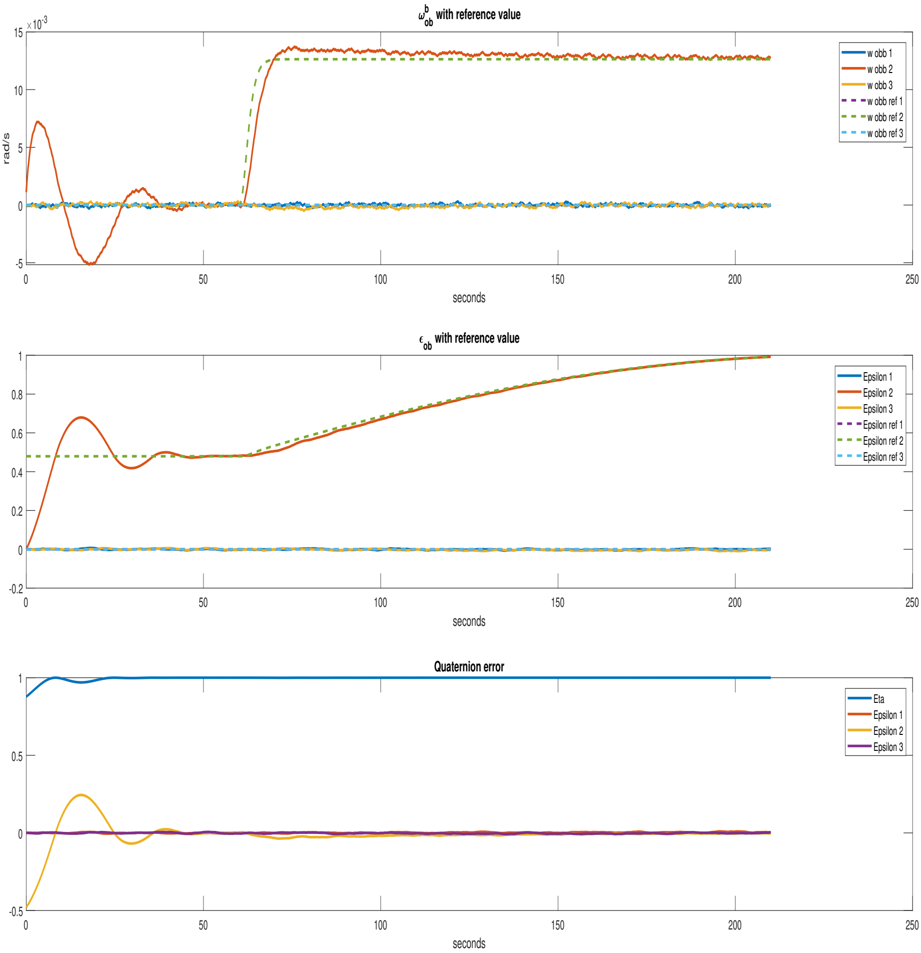


Figure 8.32: PD slewing without delay, with observer. State variables

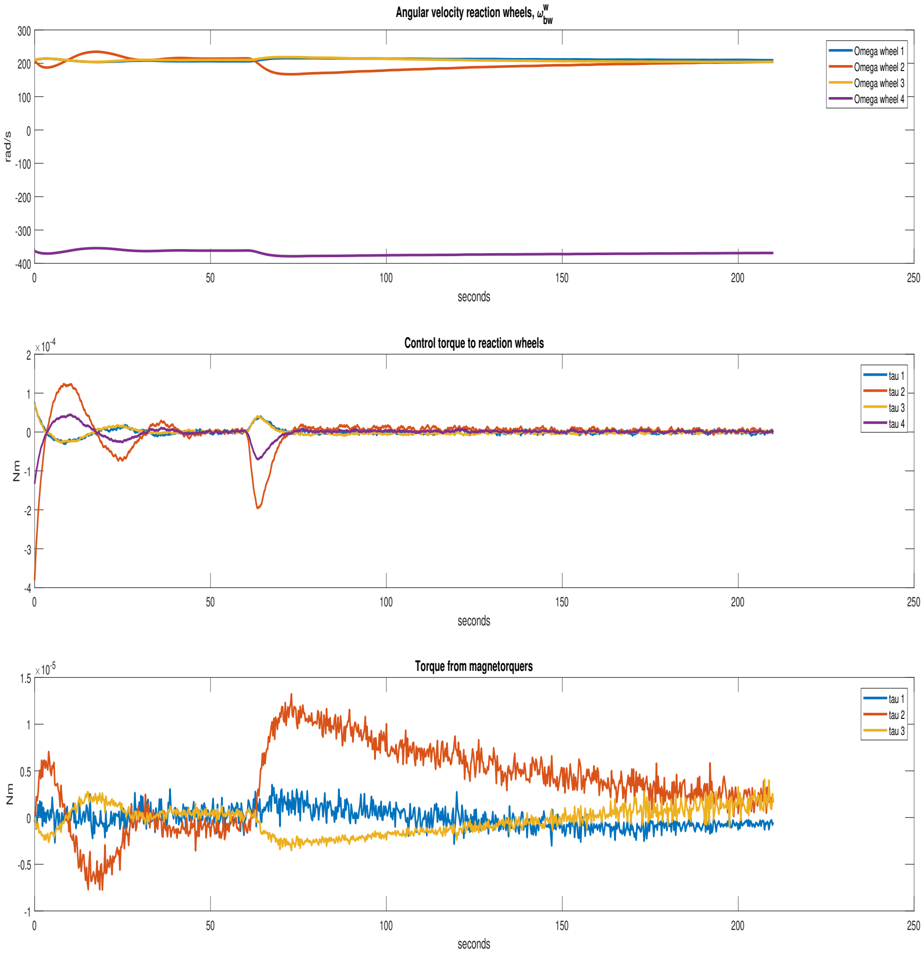


Figure 8.33: PD slewing without delay, with observer. Torques

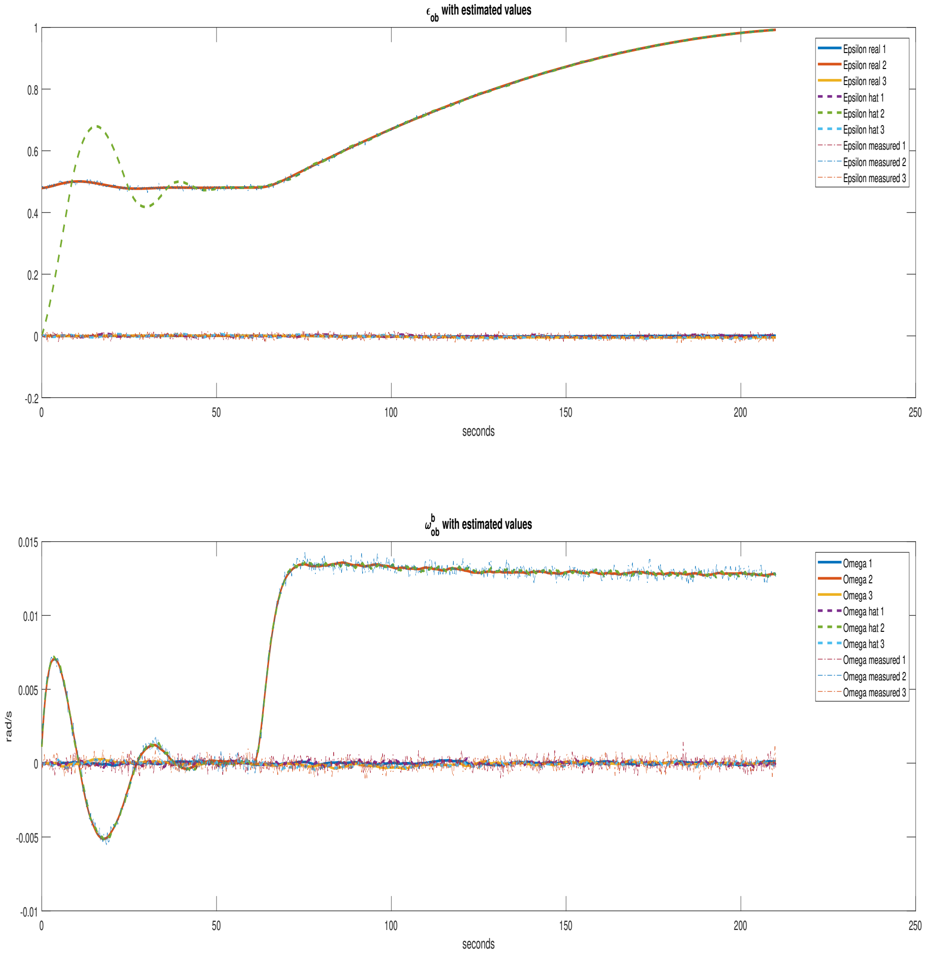


Figure 8.34: PD slewing without delay, with observer. Estimated values

LQR without time delay, with observer. Slewing.

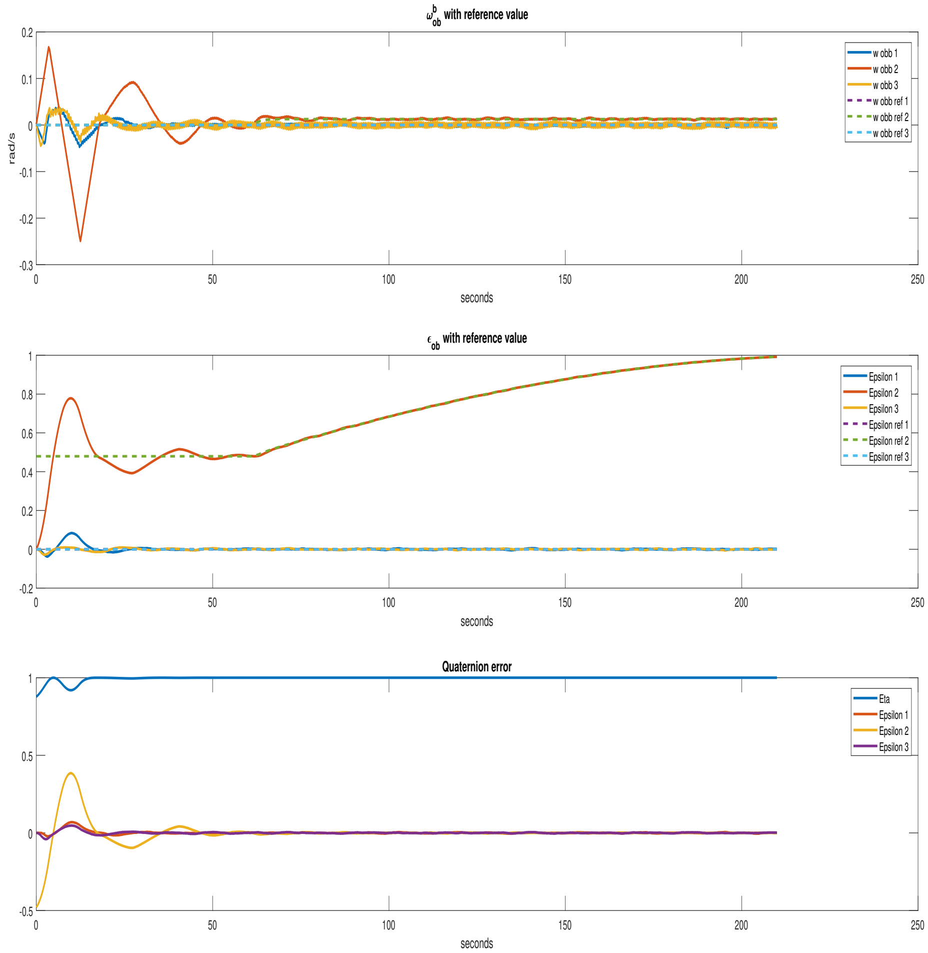


Figure 8.35: LQR slewing without delay, with observer. State variables

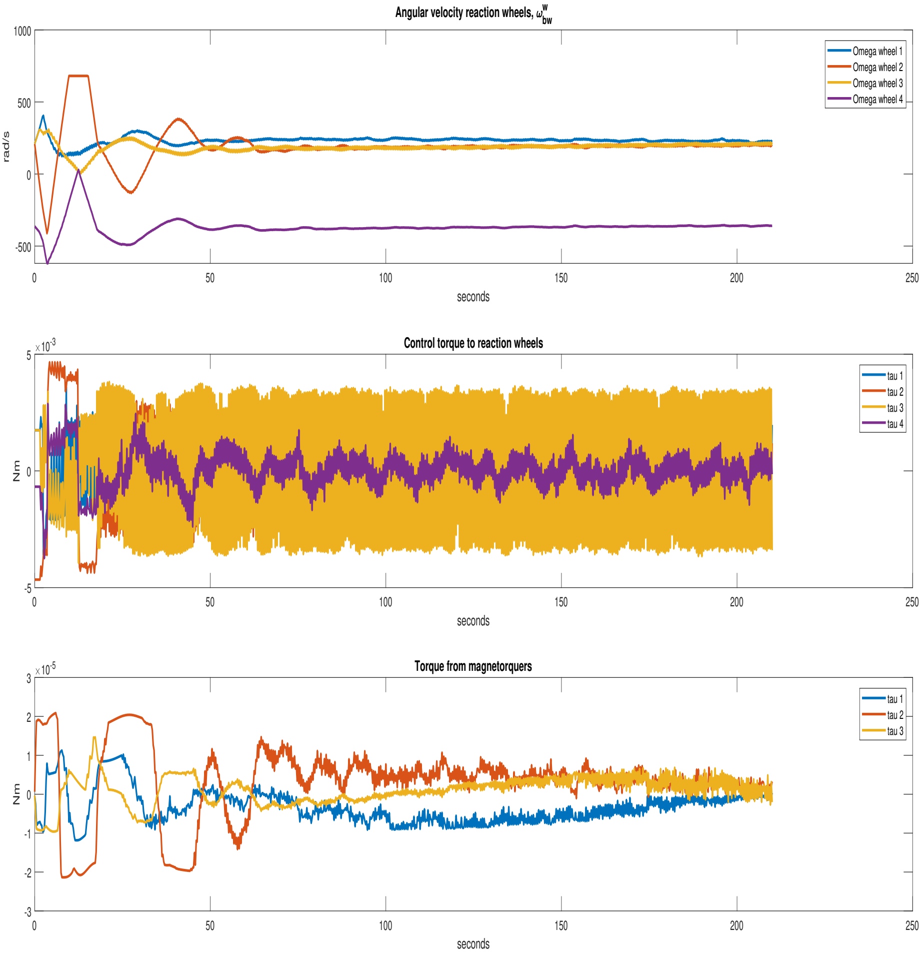


Figure 8.36: LQR slewing without delay, with observer. Torques

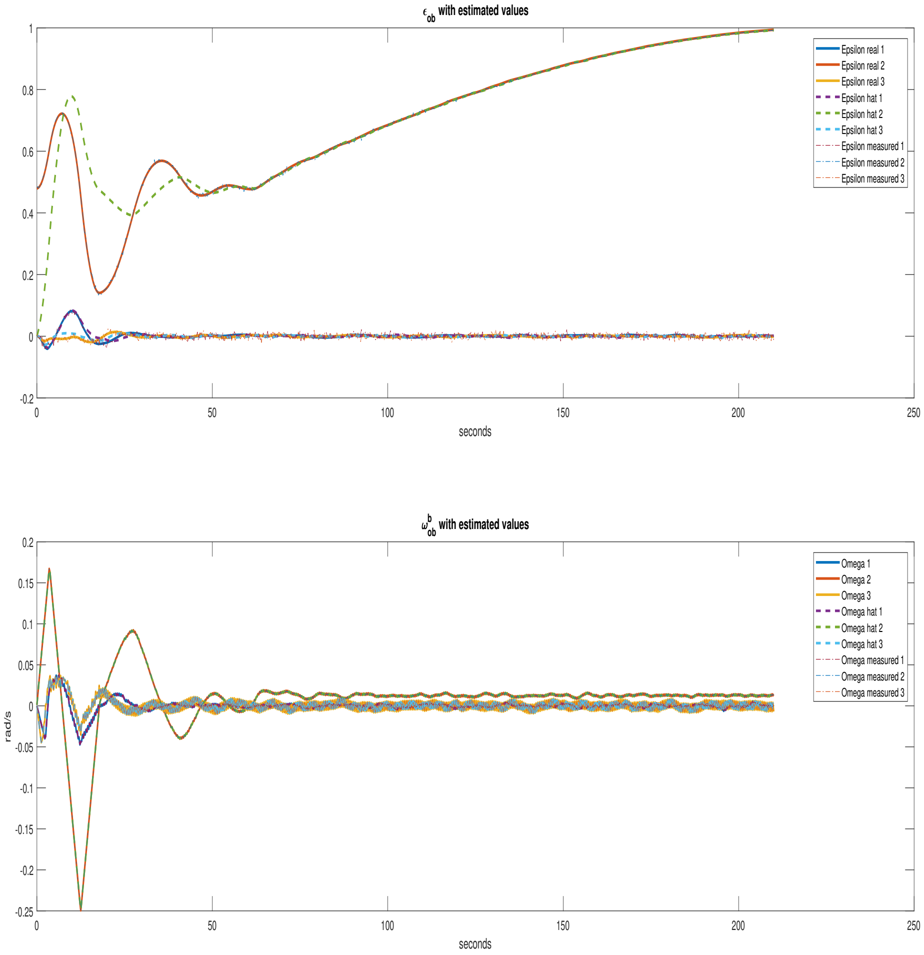


Figure 8.37: LQR slewing without delay, with observer. Estimated values

Sliding mode control without time delay, with observer. Slewing.

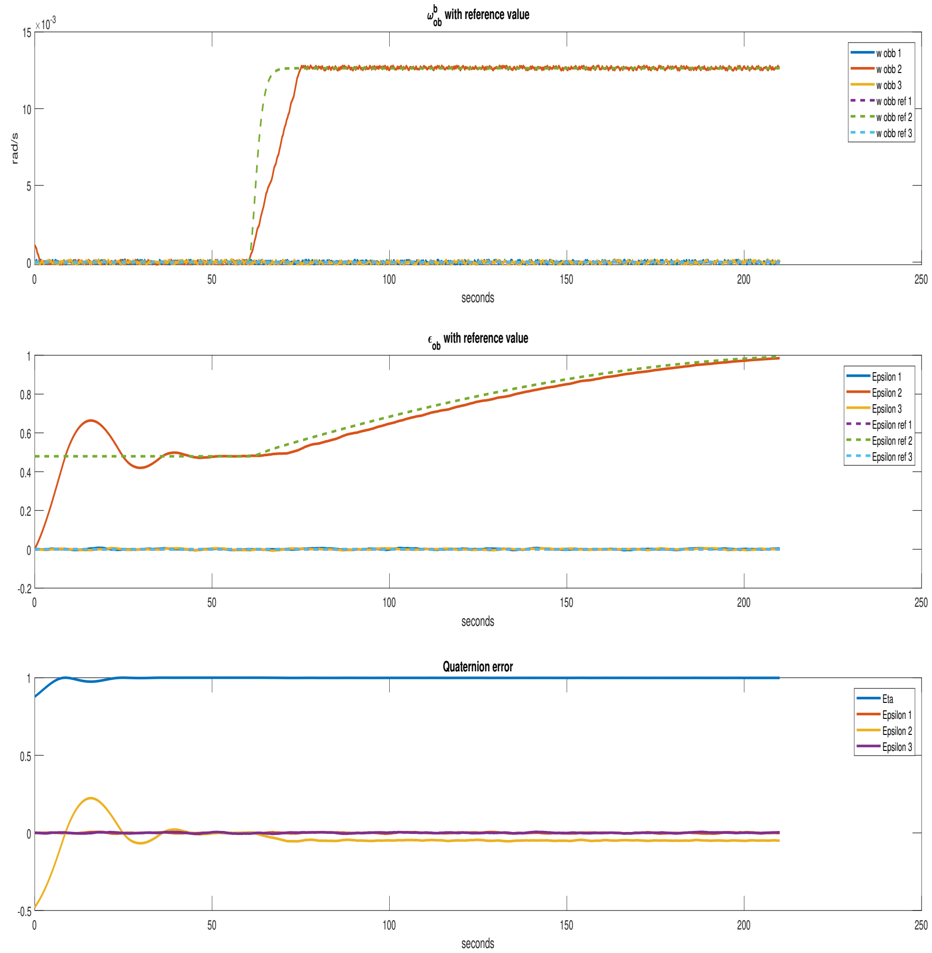


Figure 8.38: SMC slewing without delay, with observer. State variables

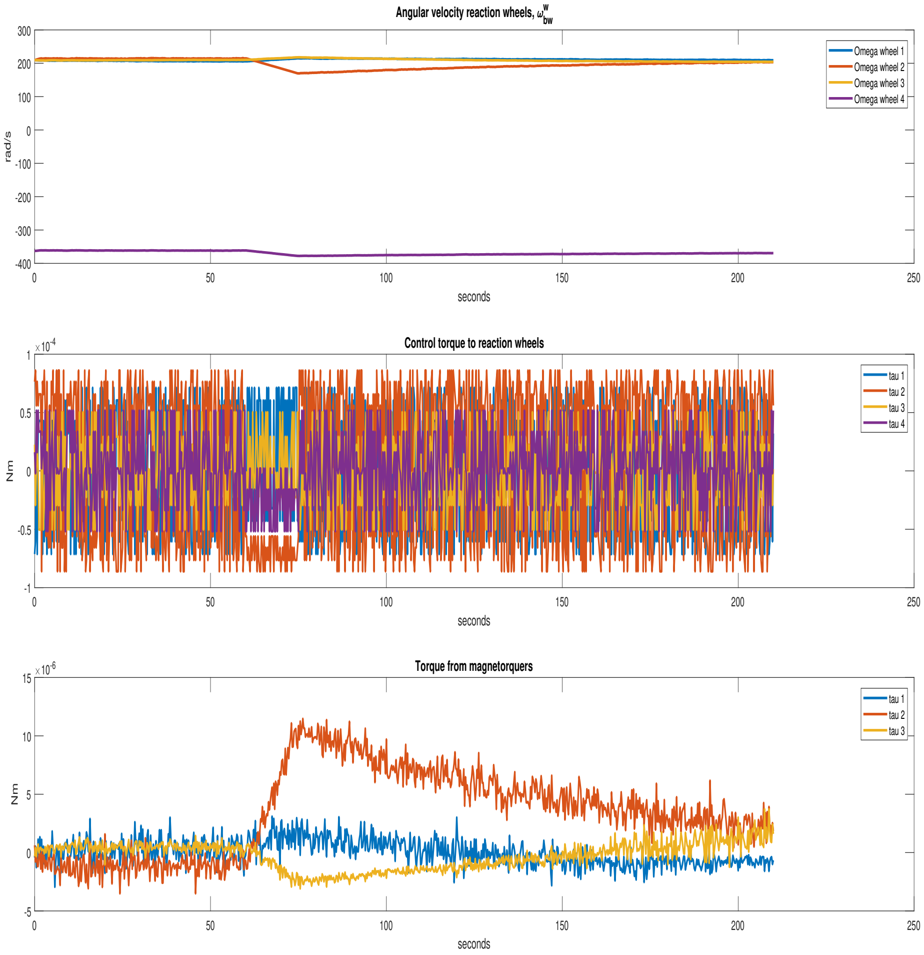


Figure 8.39: SMC slewing without delay, with observer. Torques

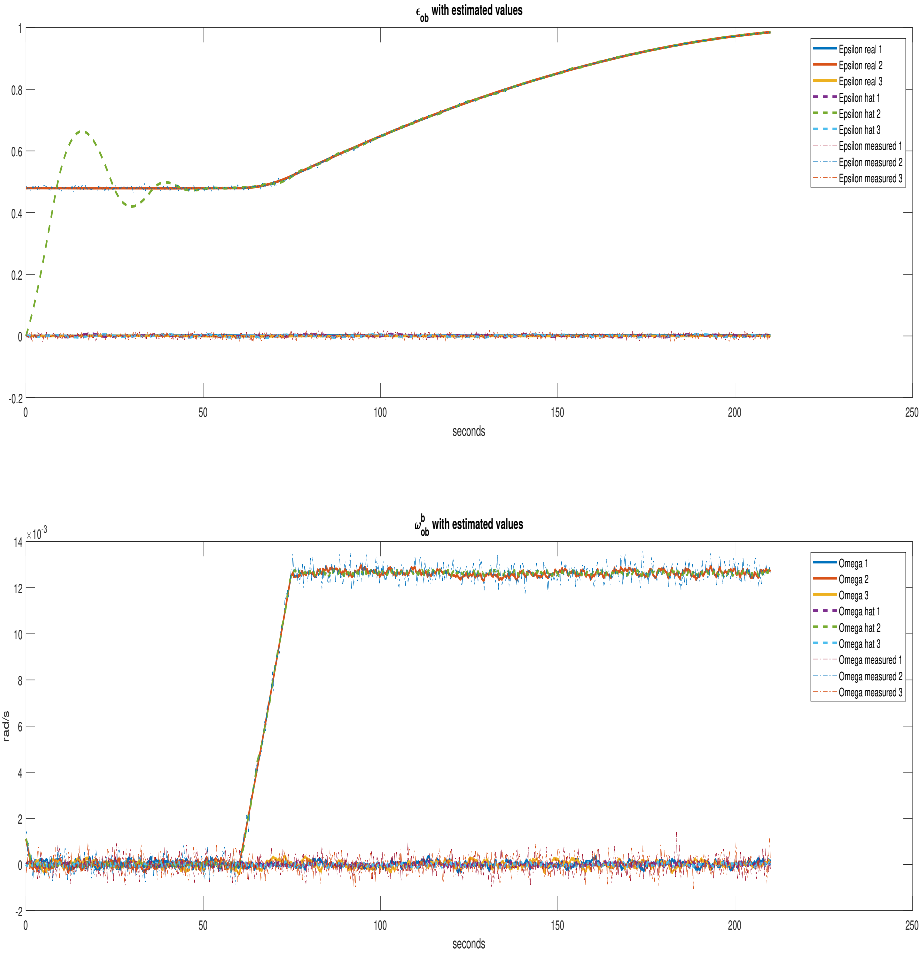


Figure 8.40: SMC slewing without delay, with observer. Estimated values

GSTA without time delay, with observer. Slewing.

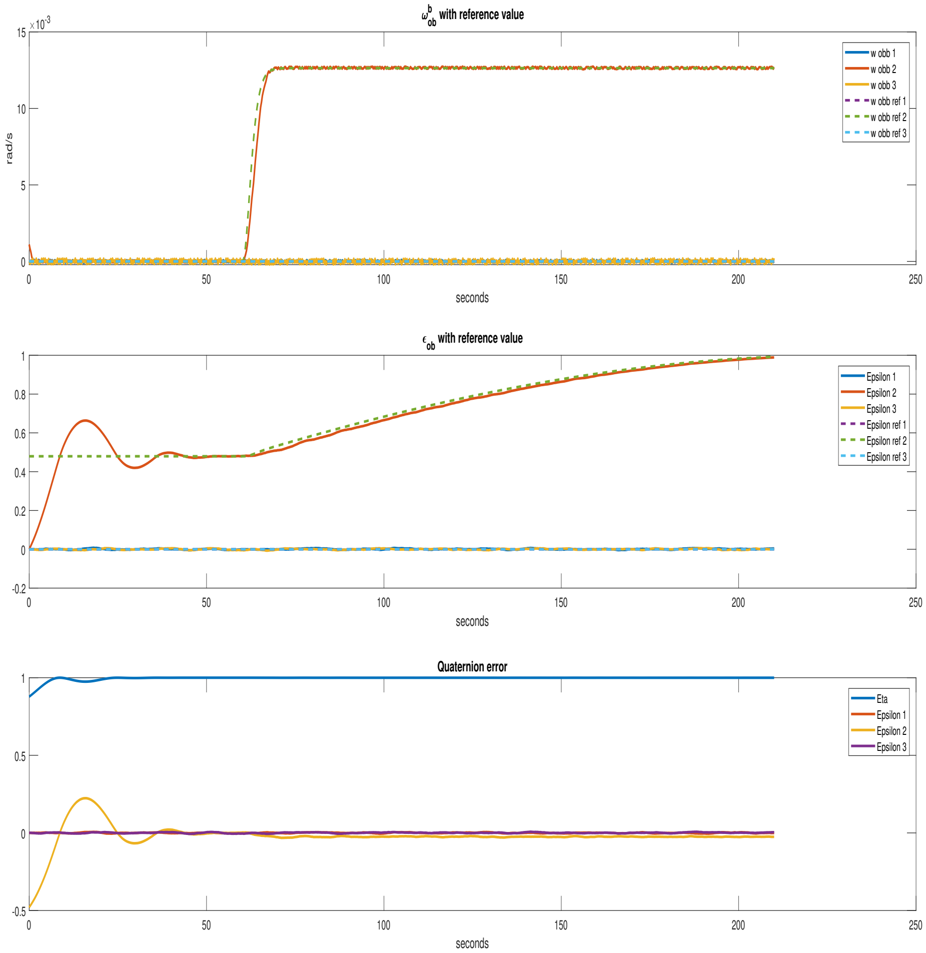


Figure 8.41: GSTA slewing without delay, with observer. State variables

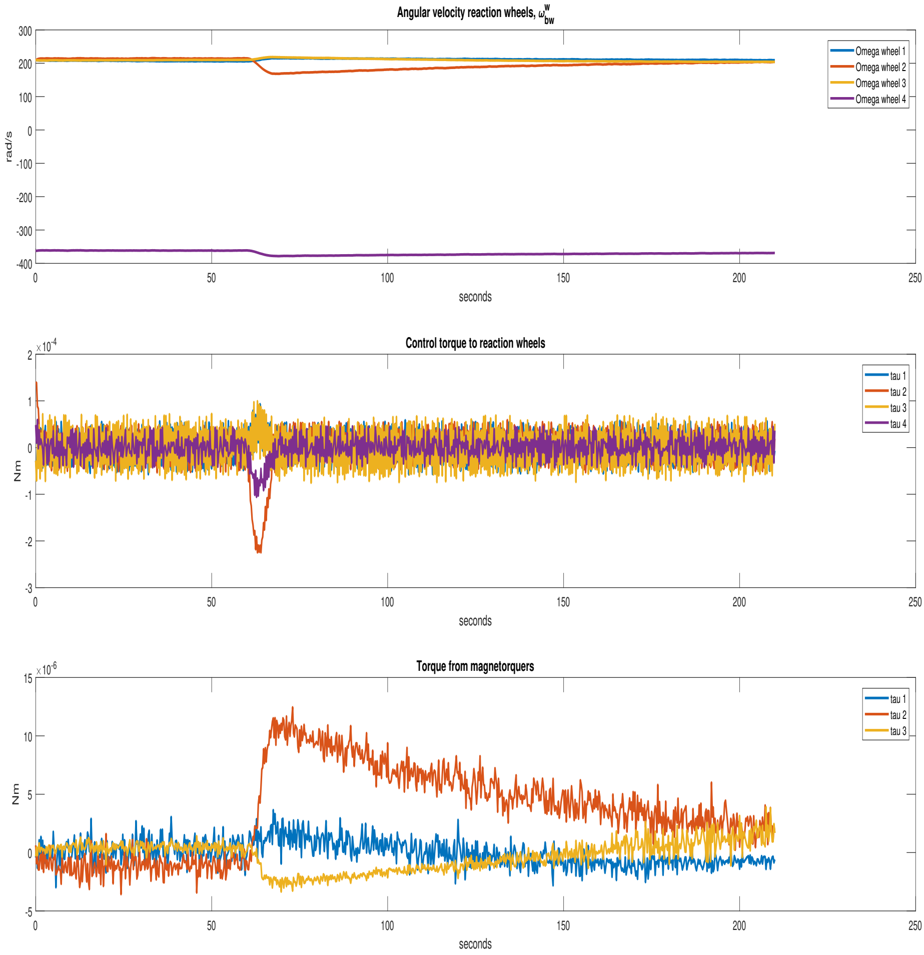


Figure 8.42: GSTA slewing without delay, with observer. Torques

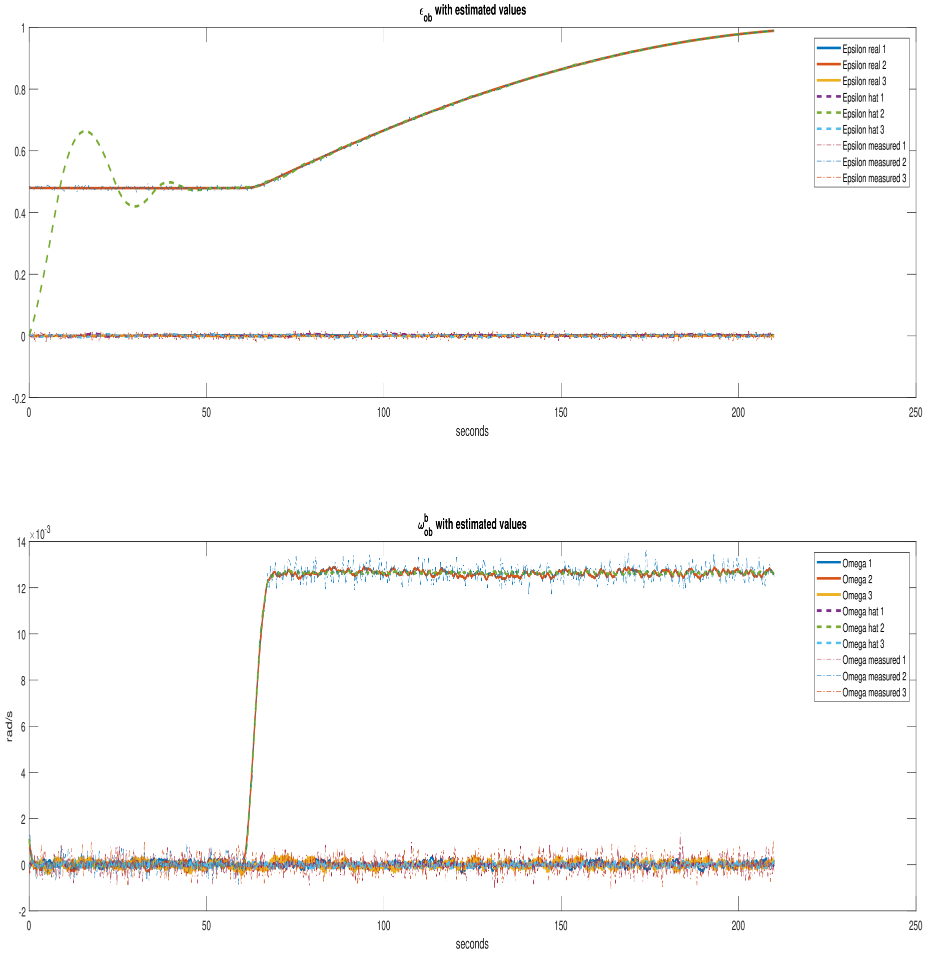


Figure 8.43: GSTA slewing without delay, with observer. Estimated values

Comparison of the different controllers based on settling time, accuracy, and steady-state error.

	Errors								
	Settling time[s]			RMSE $[\frac{rad}{s}]$			Steady-state error$[\frac{rad}{s}]$		
	ω_1	ω_2	ω_3	ω_1	ω_2	ω_3	ω_1	ω_2	ω_3
PD	-1	61.5	-1	0.694e-4	2.398e-4	1.244e-4	1.199e-4	1.567e-4	0.115e-4
LQR	-	-	-	-	-	-	1.1e-3	1.6e-3	6.9e-3
SMC	-1	14.5	0.25	1.183e-4	1.256e-4	1.335e-4	2.173e-4	0.656e-4	0.7e-4
GSTA	-1	6.75	13.25	1.19e-4	1.152e-4	1.627e-4	1.236e-4	0.178e-4	1.069e-4

Table 8.4: Error comparison. Slewing, without delay, with observer

The LQR does not manage to settle when the observer is in the loop. The settling times for SMC and GSTA are almost identical in the second channel, while the PD is a full 10 seconds faster. SMC and GSTA gets settling times for the third channel as well, meaning that it exits the error band during the first few seconds. The RMSE values are higher than without the observer, significantly higher for the SMC and the GSTA.

8.4 Results with time delays

8.4.1 Without observer

8.4.1.1 Pointing

PD control with time delay, without observer. Pointing.

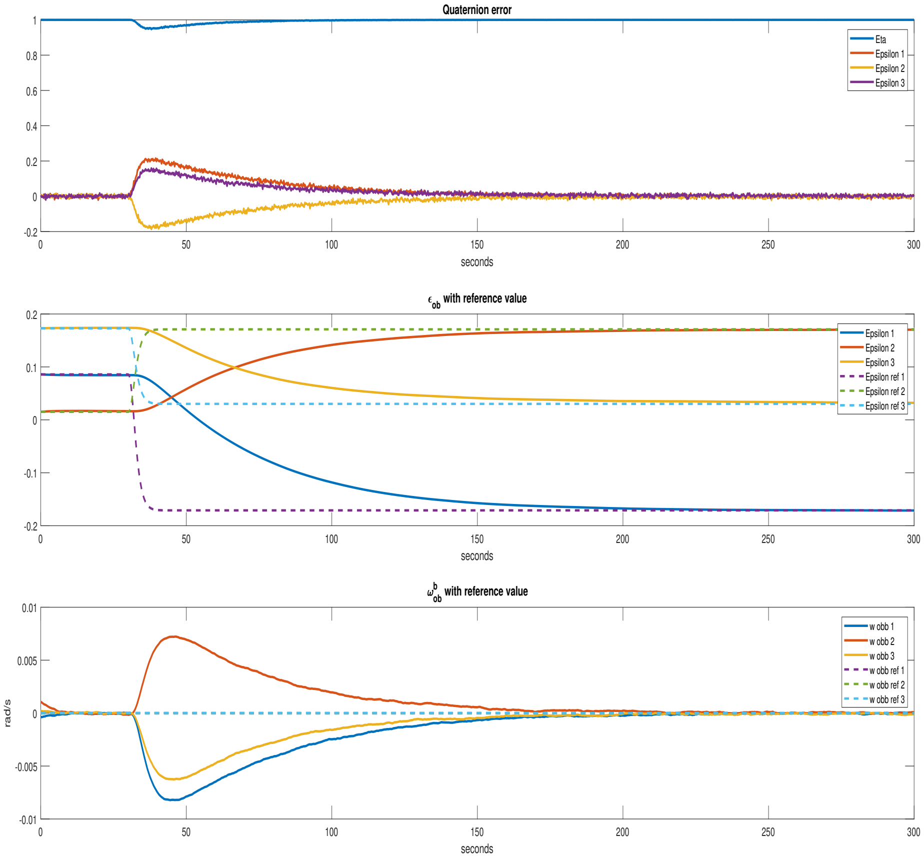
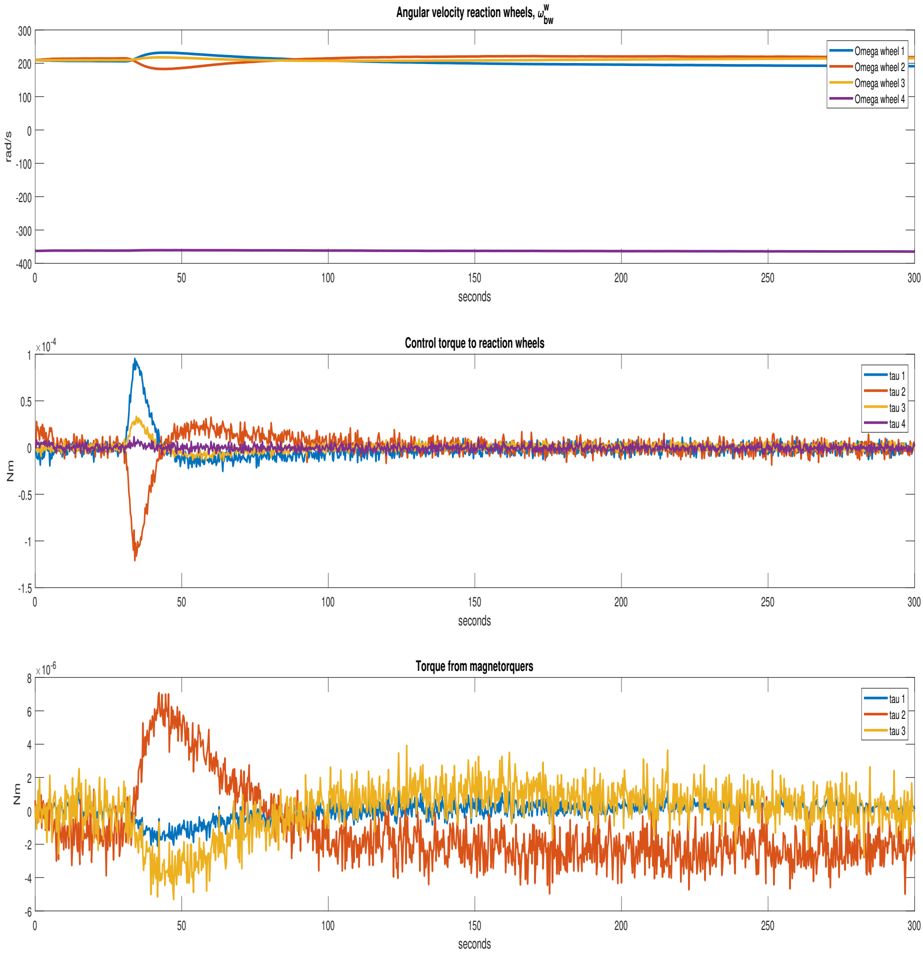


Figure 8.44: PD pointing with delay, without observer. State variables



Angular velocity reaction wheels, ω_{bw}^{w}

Control torque to reaction wheels

Torque from magnetorquers

Figure 8.45: PD pointing with delay, without observer. Torques

LQR with time delay, without observer. Pointing.

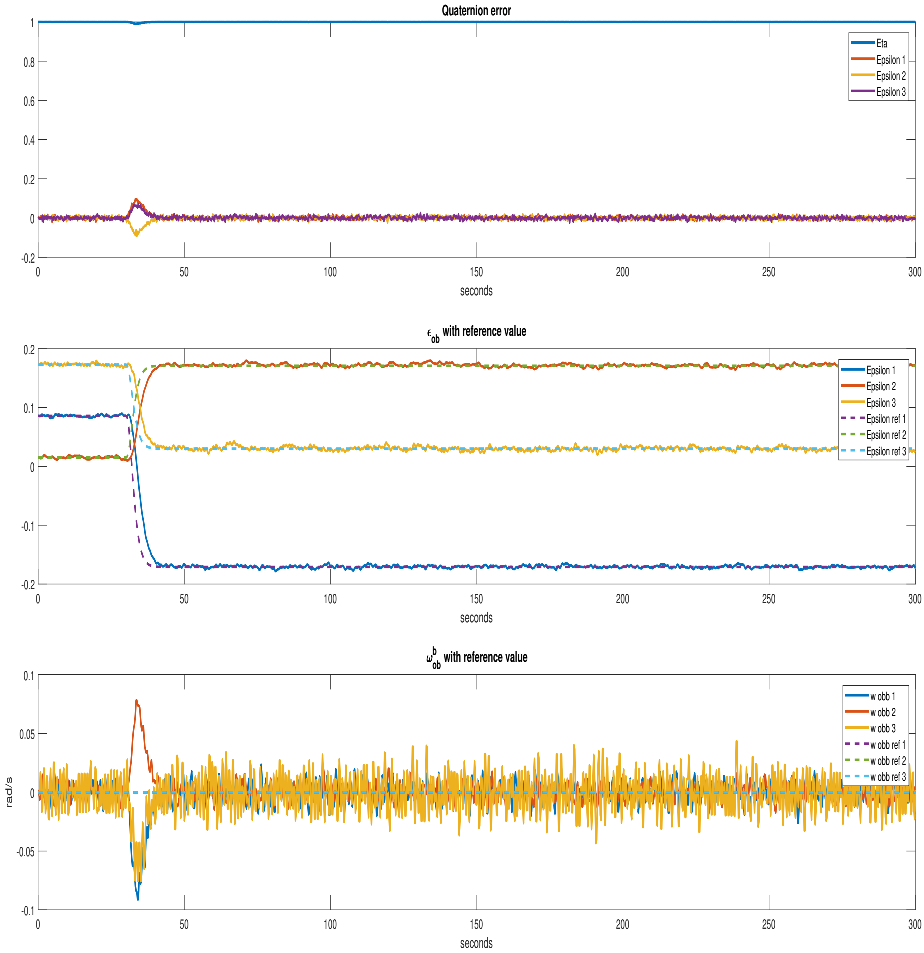


Figure 8.46: LQR pointing with delay, without observer. State variables

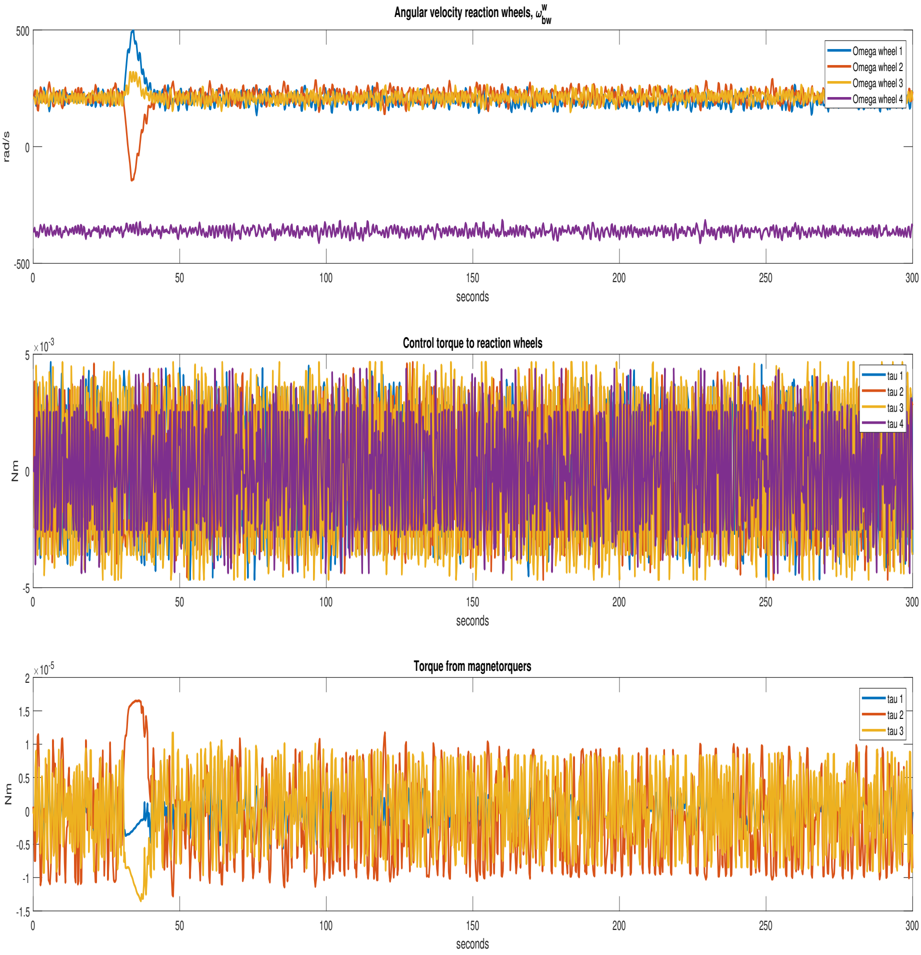


Figure 8.47: LQR pointing with delay, without observer. Torques

Sliding mode control with time delay, without observer. Pointing.

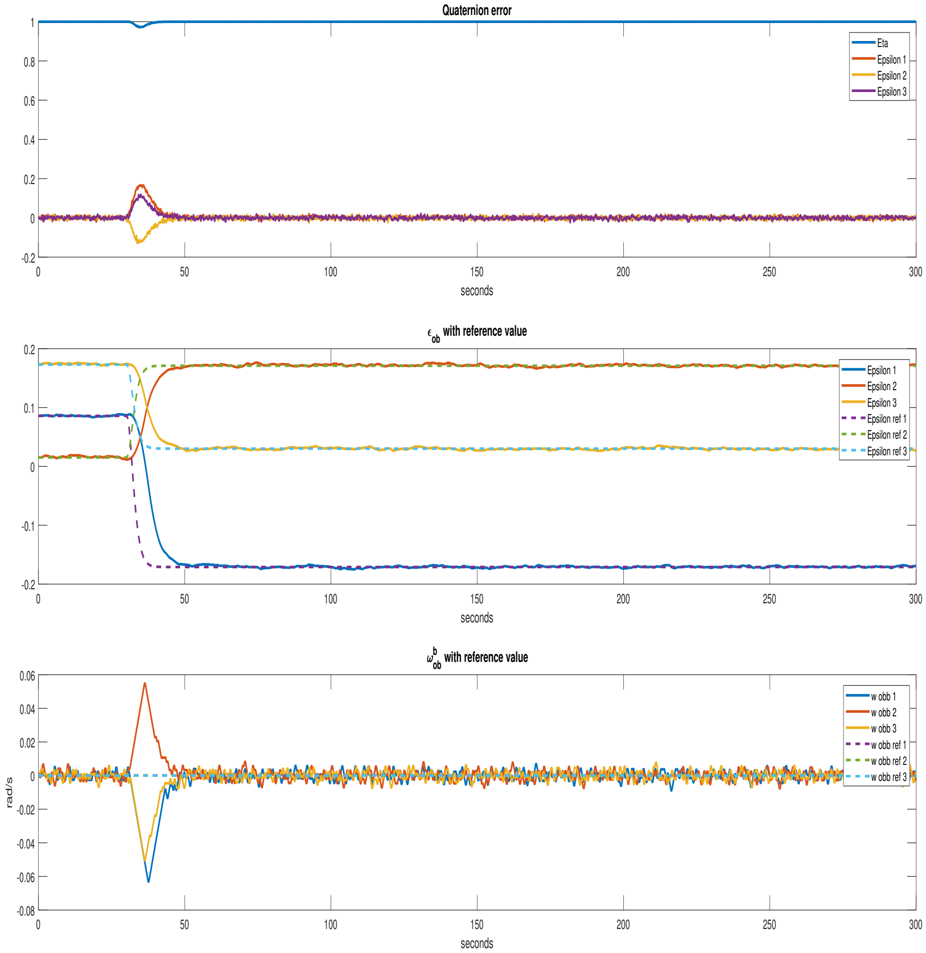


Figure 8.48: SMC pointing with delay, without observer. State variables

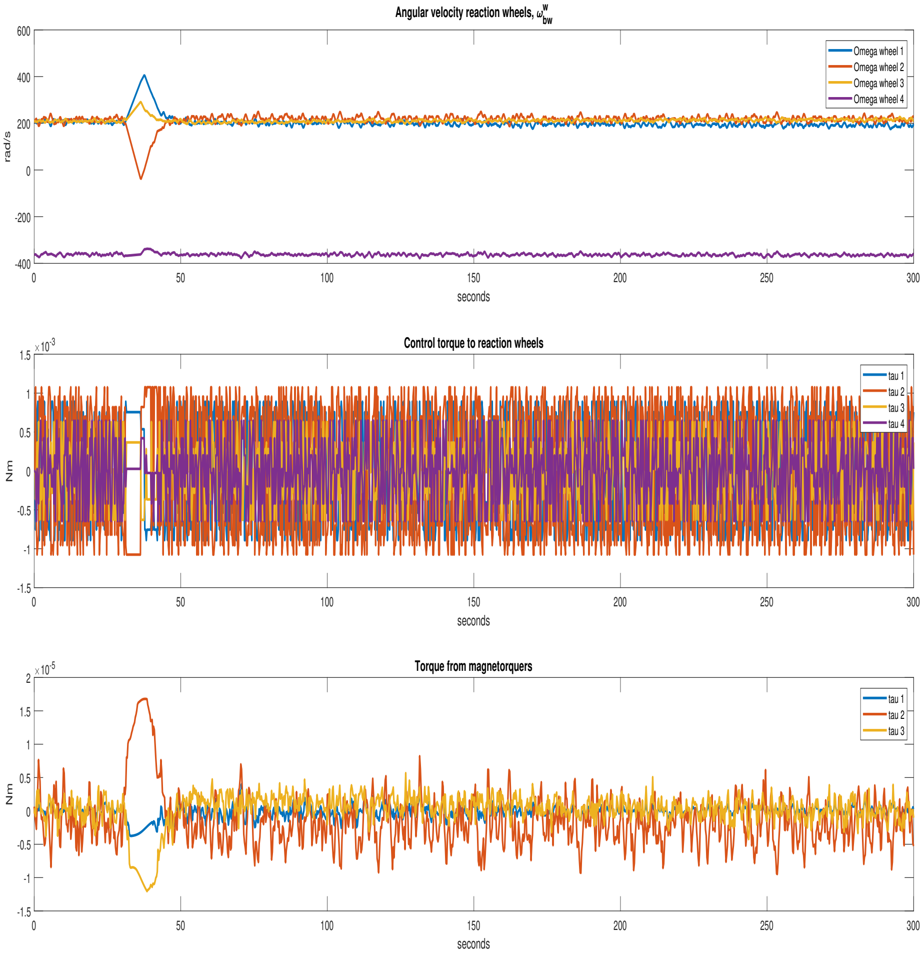


Figure 8.49: SMC pointing with delay, without observer. Torques

GSTA with time delay, without observer. Pointing.

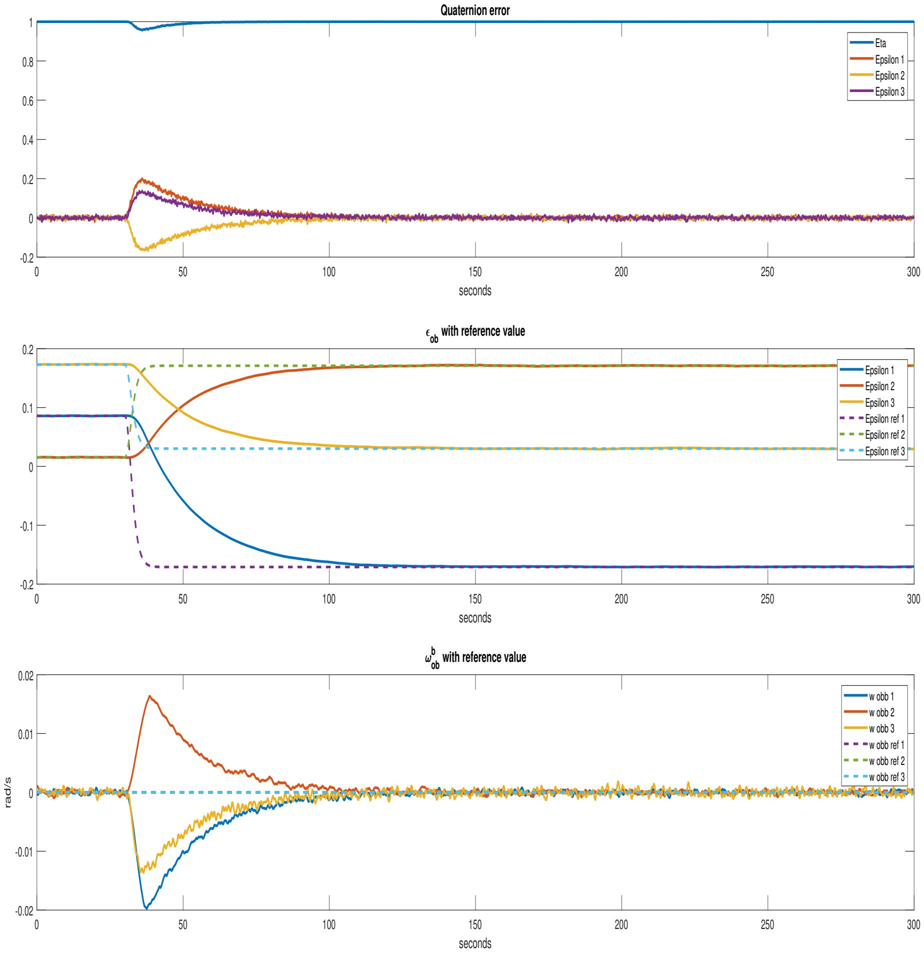


Figure 8.50: GSTA pointing with delay, without observer. State variables

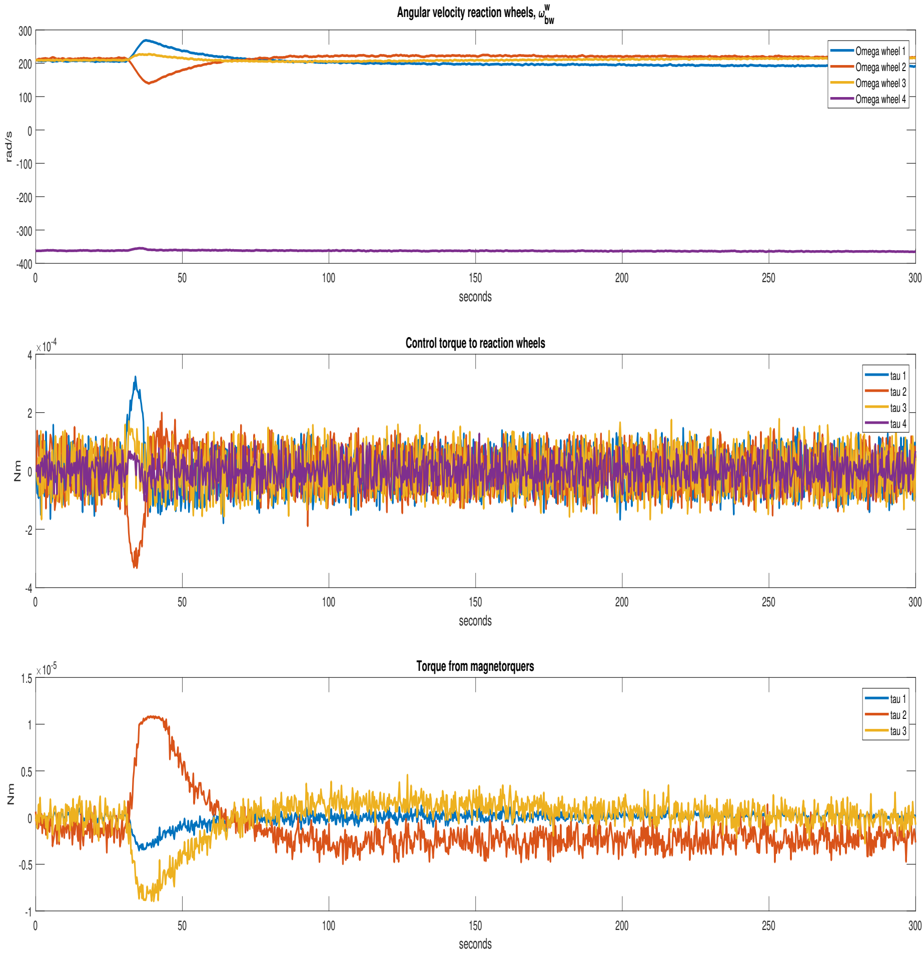


Figure 8.51: GSTA pointing with delay, without observer. Torques

Comparison of the different controllers based on settling time, accuracy, and steady-state error.

	Errors								
	Settling time[s]			RMSE [°]			Steady-state error[°]		
	ϵ_1	ϵ_2	ϵ_3	ϵ_1	ϵ_2	ϵ_3	ϵ_1	ϵ_2	ϵ_3
PD	164.25	149	202.25	0.1897	0.2111	0.3574	0.0241	0.0710	0.2251
LQR	261.5	239.5	-	0.1888	0.2129	-	0.1972	0.0314	0.7144
SMC	28.5	140.7	183.75	0.1620	0.1573	0.1699	0.1818	0.1749	0.3013
GSTA	79	66.75	73.75	0.1177	0.1054	0.1265	0.0753	0.0368	0.0737

Table 8.5: Error comparison. Pointing, with delay, without observer

Comparing these result to the equivalent without time delay, the LQR now performs worse far worse, as the settling time is a lot longer for the first channel and undefined for the third. PD and GSTA are close to unchanged from the idealized system without time delay. SMC has worse settling times on all channels, as well as worse RMSE values.

8.4.1.2 Slewing

PD control with time delay, without observer. Slewing.

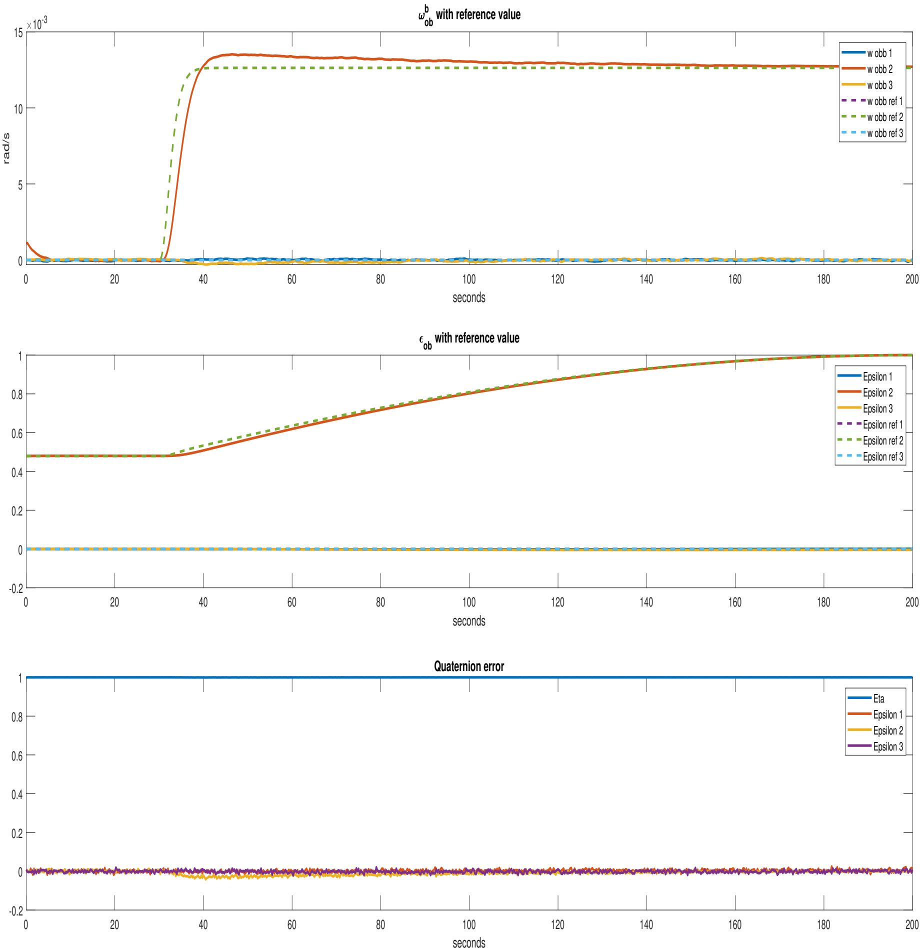


Figure 8.52: PD slewing with delay, without observer. State variables

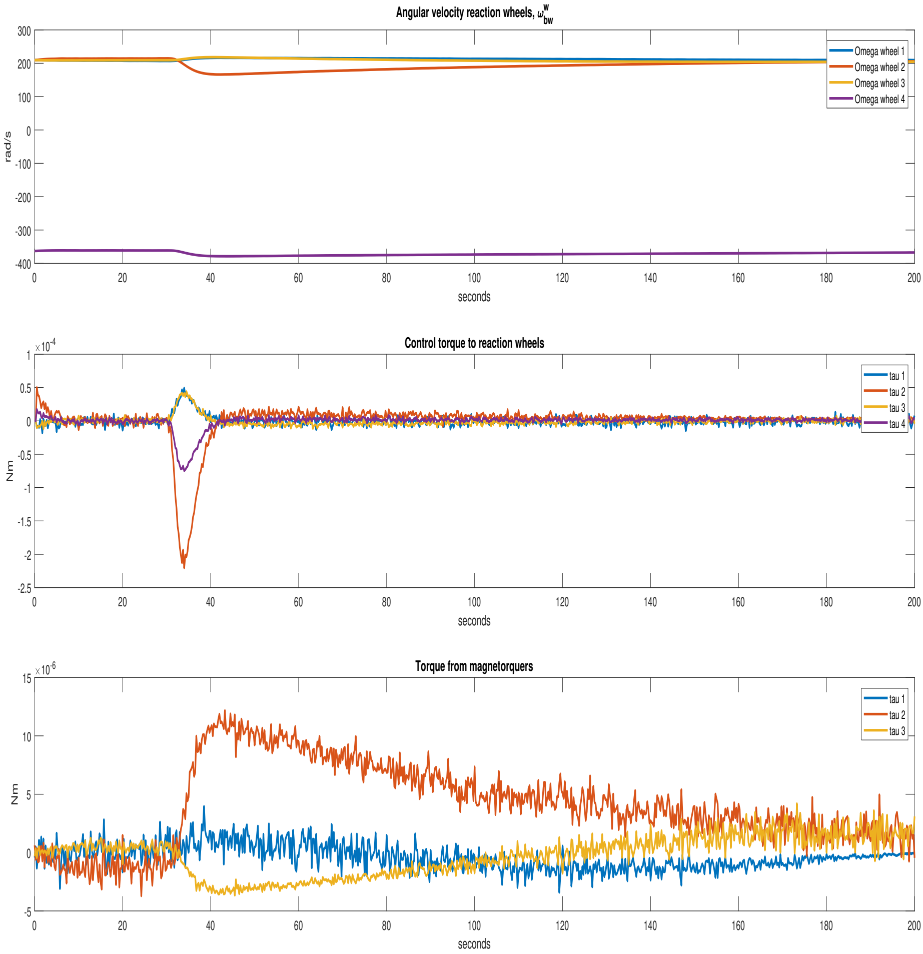


Figure 8.53: PD slewing with delay, without observer. Torques

LQR with time delay, without observer. Slewing.

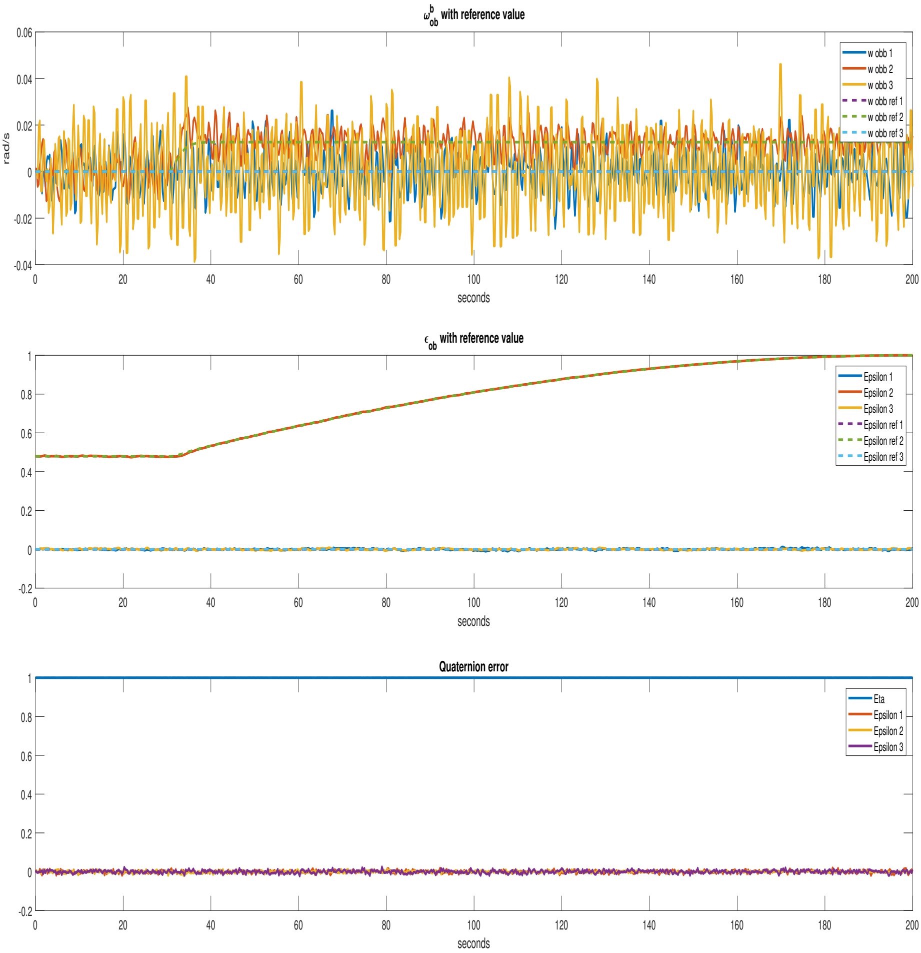


Figure 8.54: LQR slewing with delay, without observer. State variables

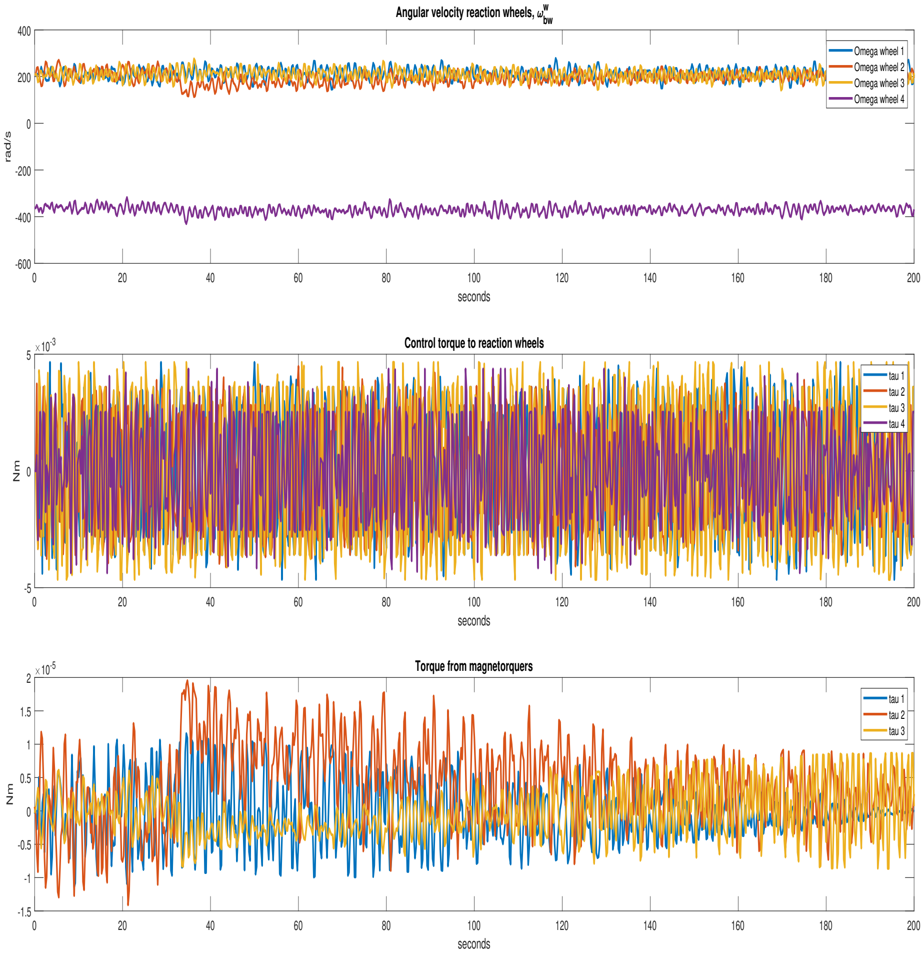


Figure 8.55: LQR slewing with delay, without observer. Torques

Sliding mode control with time delay, without observer. Slewing.

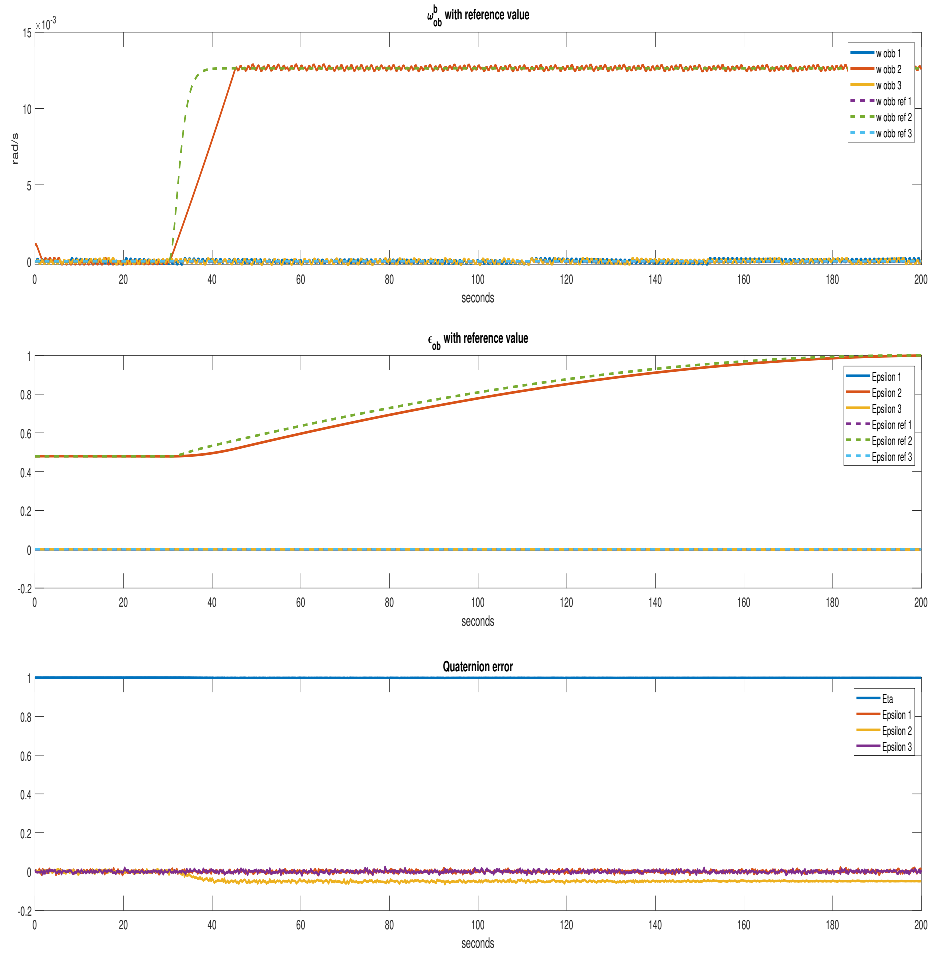


Figure 8.56: SMC slewing with delay, without observer. State variables

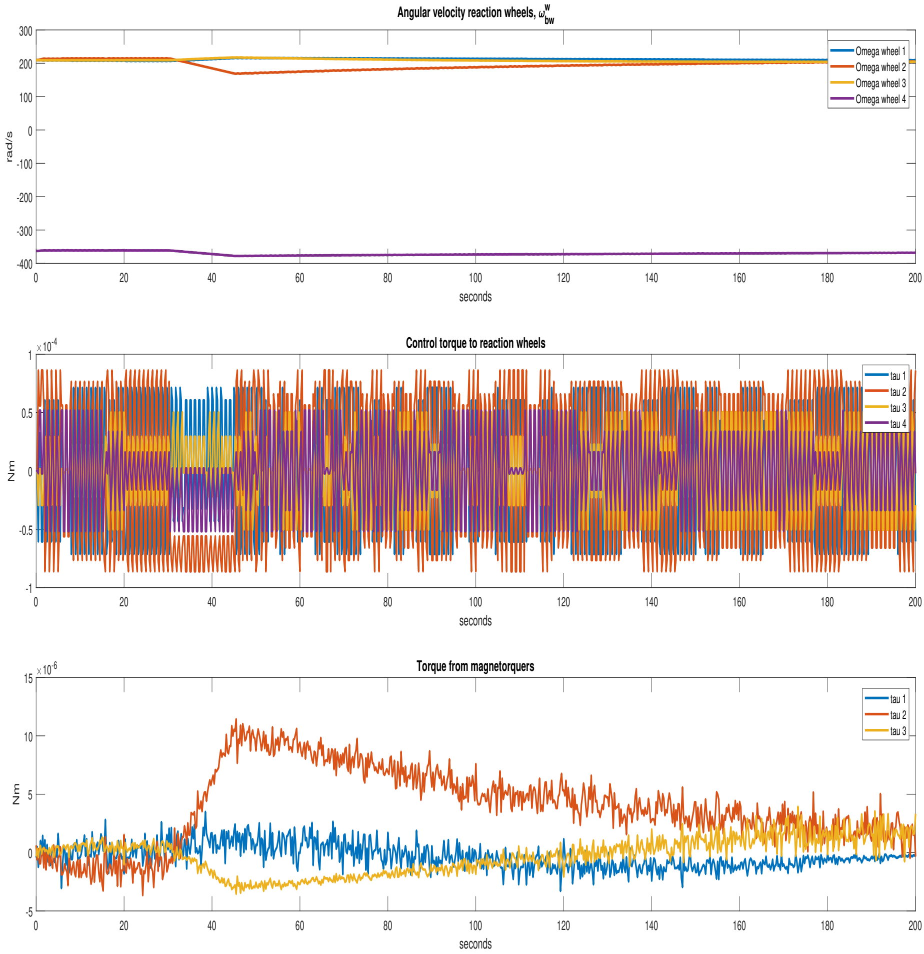
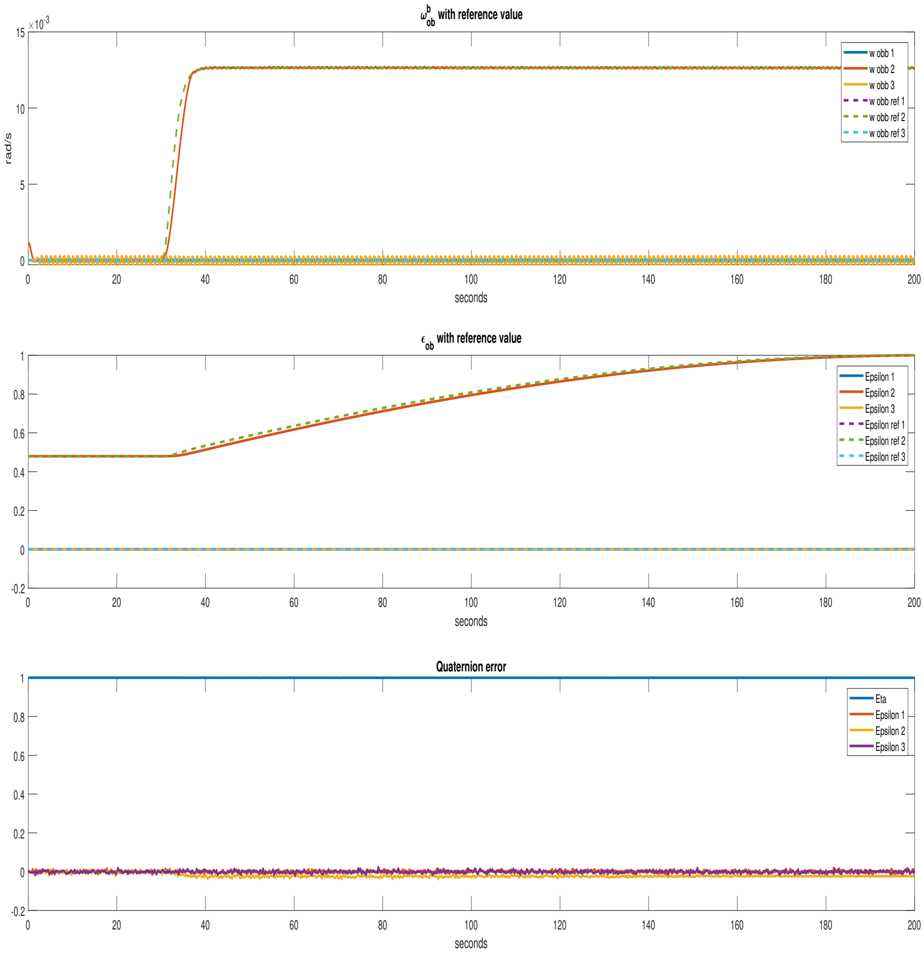


Figure 8.57: SMC slewing with delay, without observer. Torques

GSTA with time delay, without observer. Slewing.


ω^b_{ob} with reference value

ϵ_{ob} with reference value

Quaternion error

Figure 8.58: GSTA slewing with delay, without observer. State variables

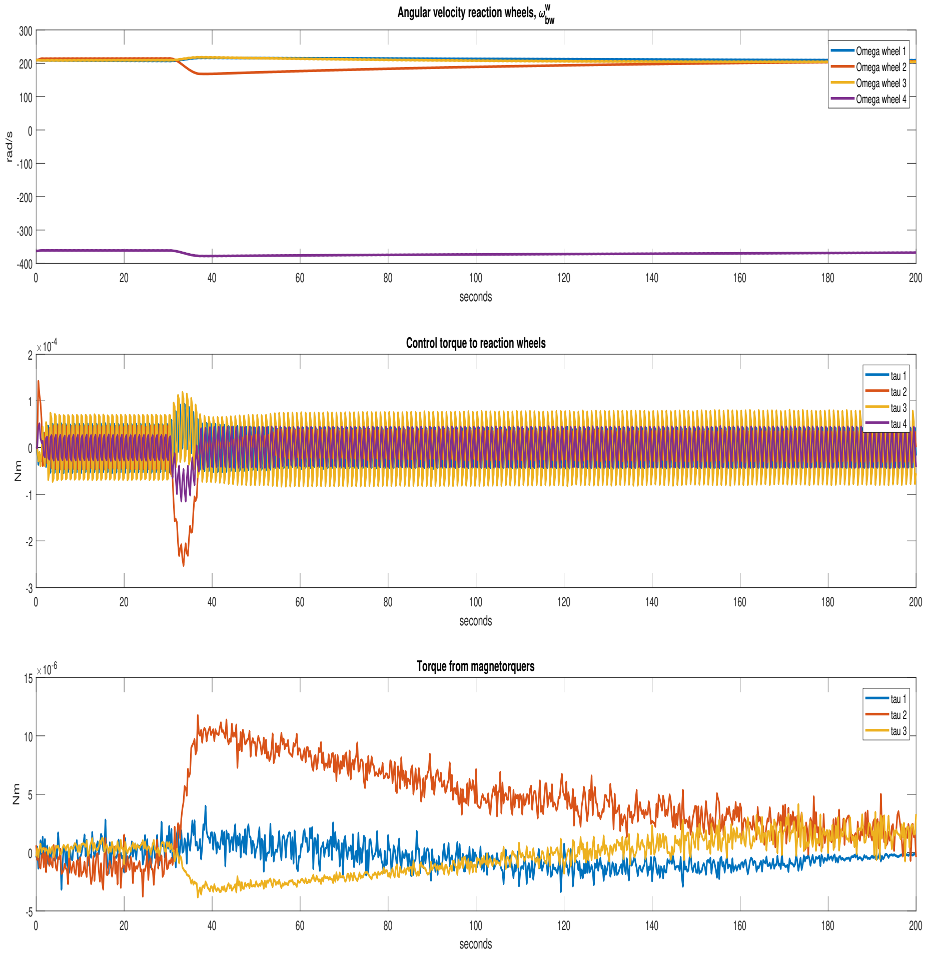


Figure 8.59: GSTA slewing with delay, without observer. Torques

Comparison of the different controllers based on settling time, accuracy, and steady-state error.

	Errors								
	Settling time[s]			RMSE $[\frac{rad}{s}]$			Steady-state error$[\frac{rad}{s}]$		
	ω_1	ω_2	ω_3	ω_1	ω_2	ω_3	ω_1	ω_2	ω_3
PD	-1	71.75	-1	0.397e-4	2.163e-4	1.020e-4	0.796e-4	0.7554e-4	0.5224e-4
LQR	24.75	-	-	9e-3	-	-	4e-4	2.8e-3	6.5e-3
SMC	-1	15	-1	1.123e-4	1.199e-4	1.168e-4	0.112e-4	0.7545e-4	0.7856e-4
GSTA	-1	6.5	-1	0.656e-4	0.346e-4	2.578e-4	0.635e-4	0.299e-4	2.468e-4

Table 8.6: Error comparison. Slewing, with delay, without observer

The settling time is unaffected by adding time delay for PD, SMC and GSTA. The LQR no longer settles the slew rate. The RMSE values go up slightly for the PD controller, while almost doubling for SMC and GSTA. From the plots, a chattering effect can be observed along the slew rate for the SMC.

8.4.2 With observer

8.4.2.1 Pointing

PD control with time delay, with observer. Pointing.

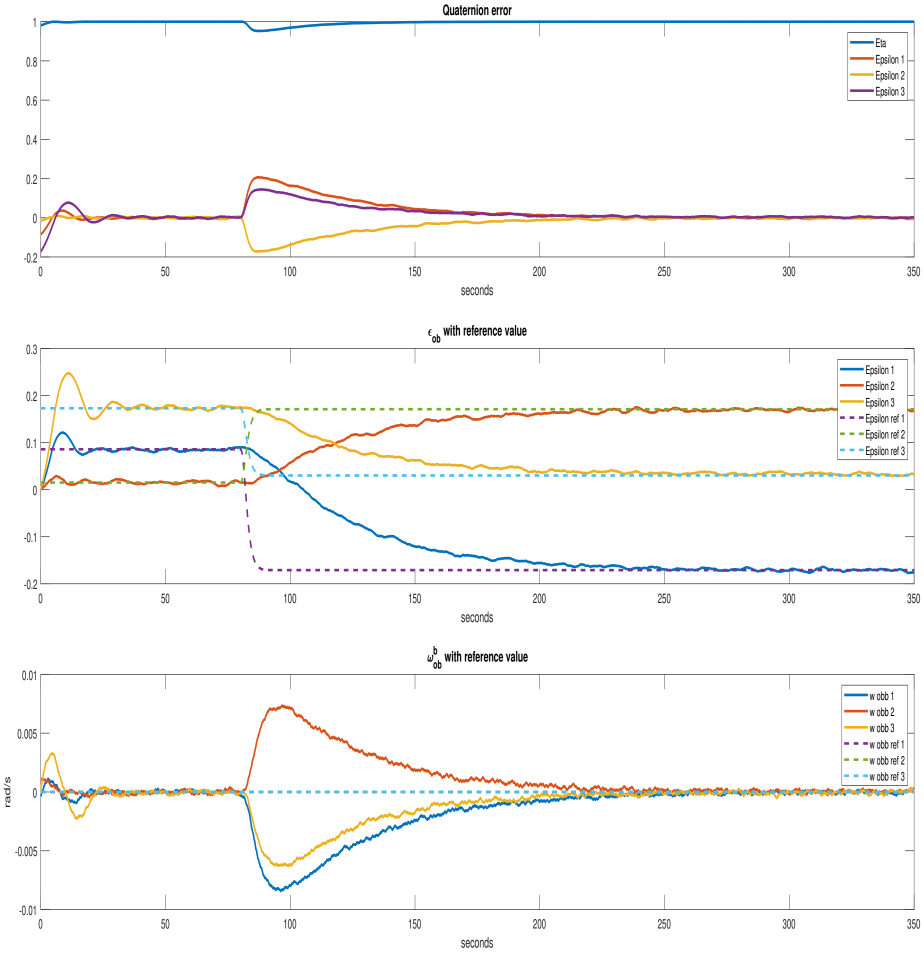


Figure 8.60: PD pointing with delay, with observer. State variables

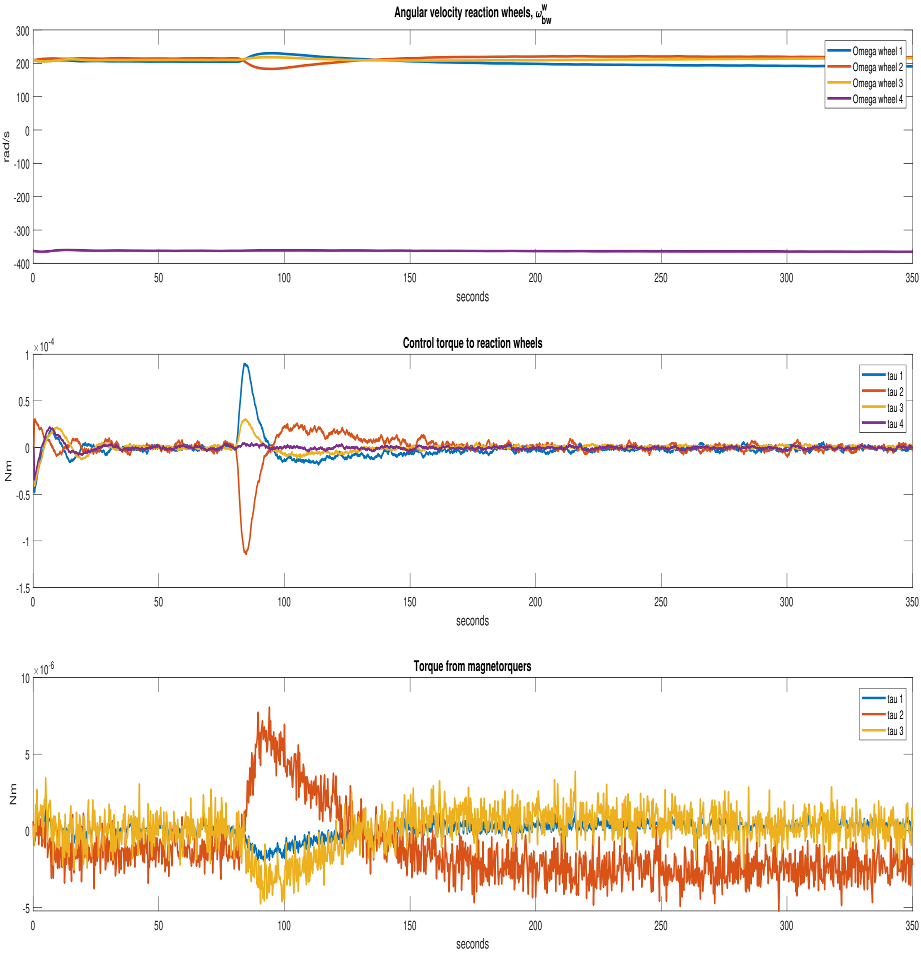


Figure 8.61: PD pointing with delay, with observer. Torques

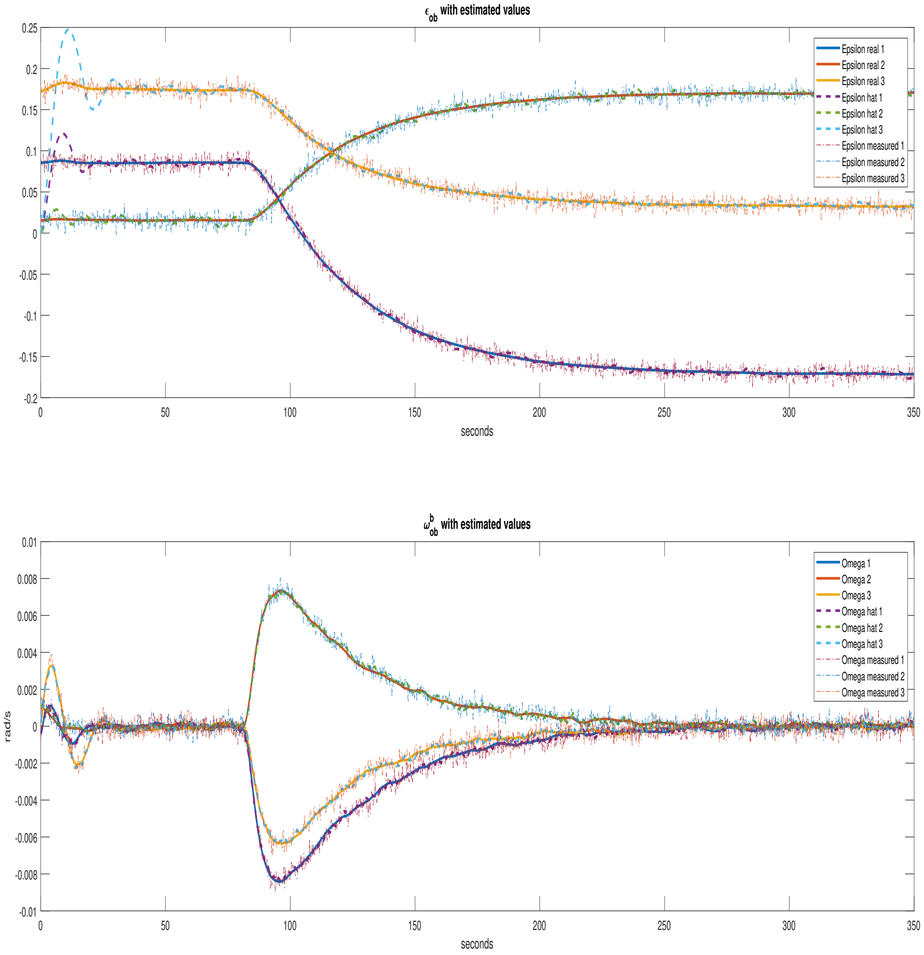


Figure 8.62: PD pointing with delay, with observer. Estimated values

LQR with time delay, with observer. Pointing.

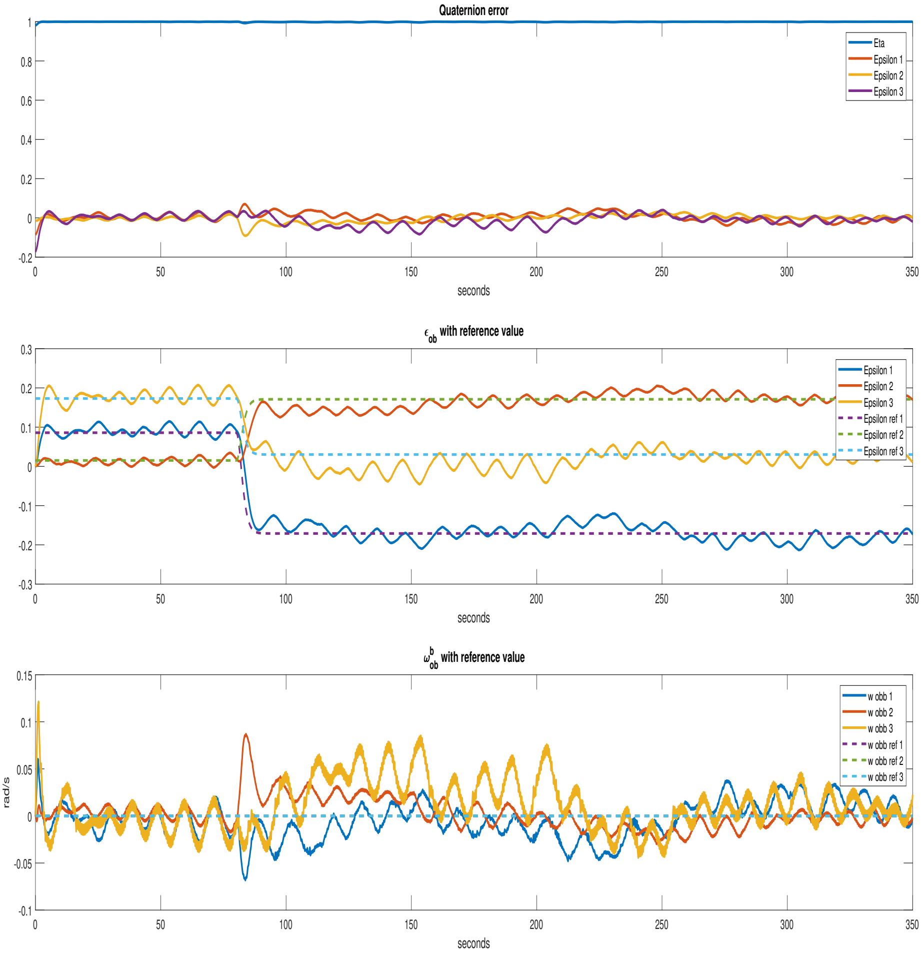


Figure 8.63: LQR pointing with delay, with observer. State variables

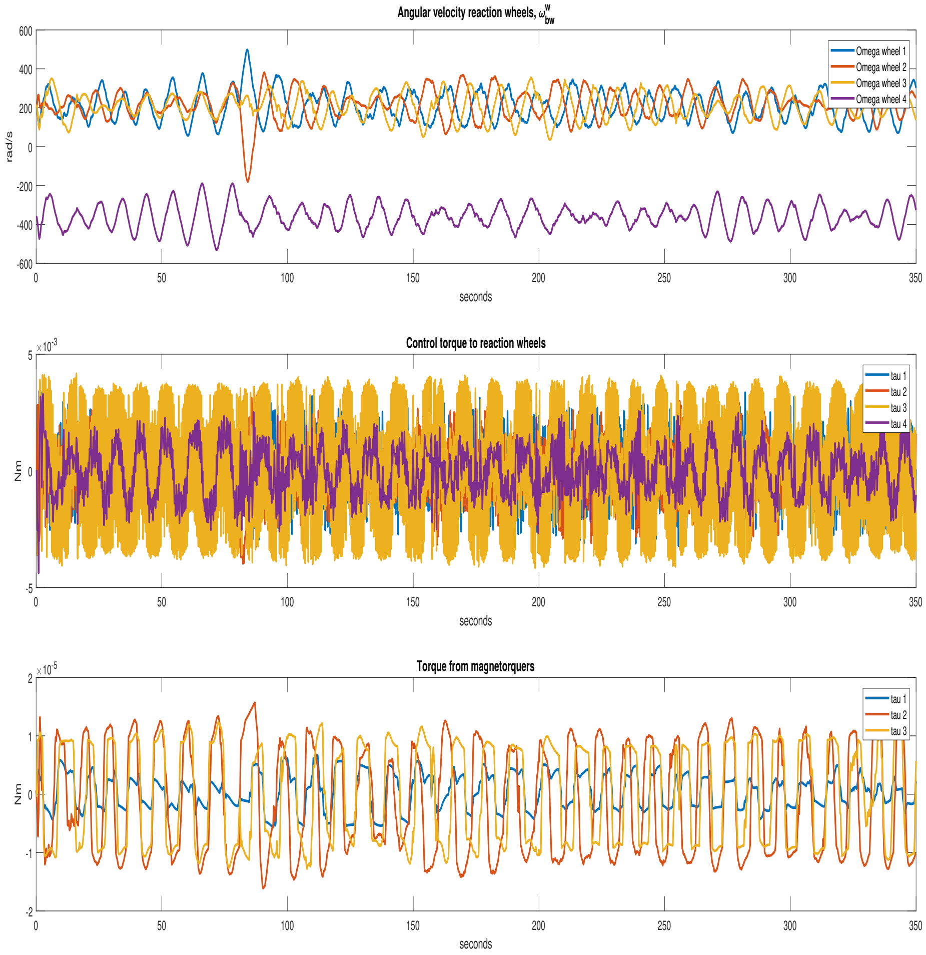


Figure 8.64: LQR pointing with delay, with observer. Torques

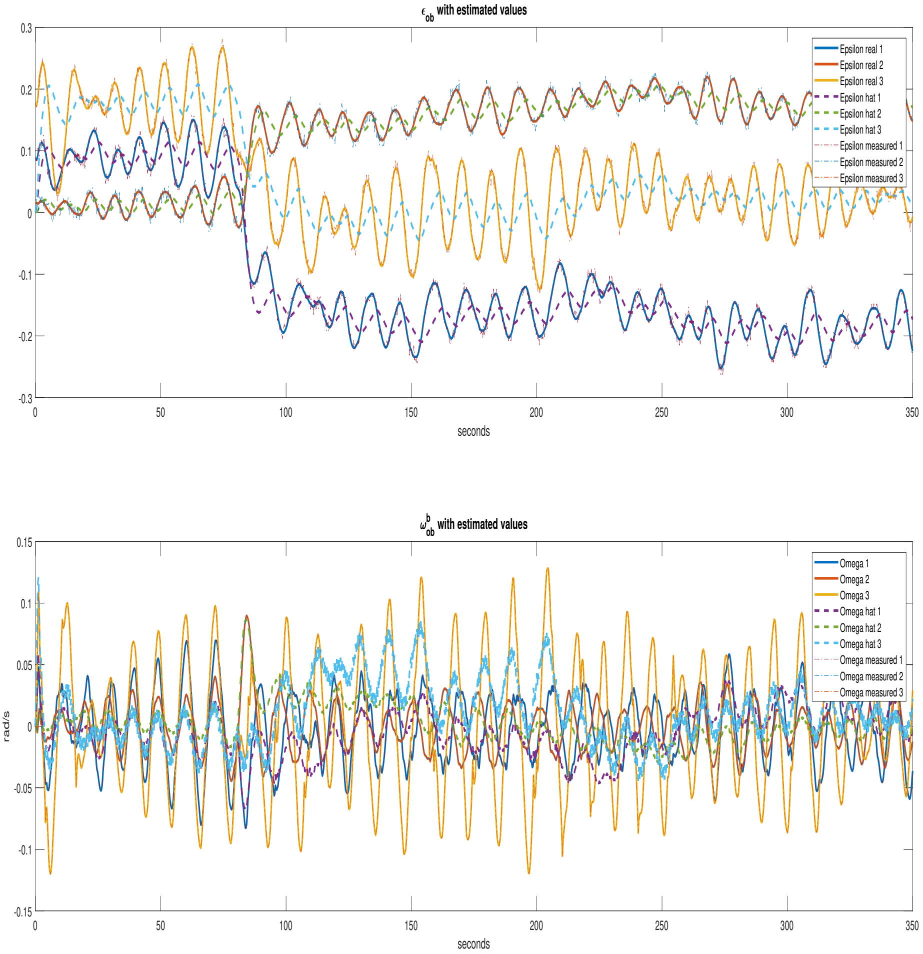


Figure 8.65: LQR pointing with delay, with observer. Estimated values

Sliding mode control with time delay, with observer. Pointing.

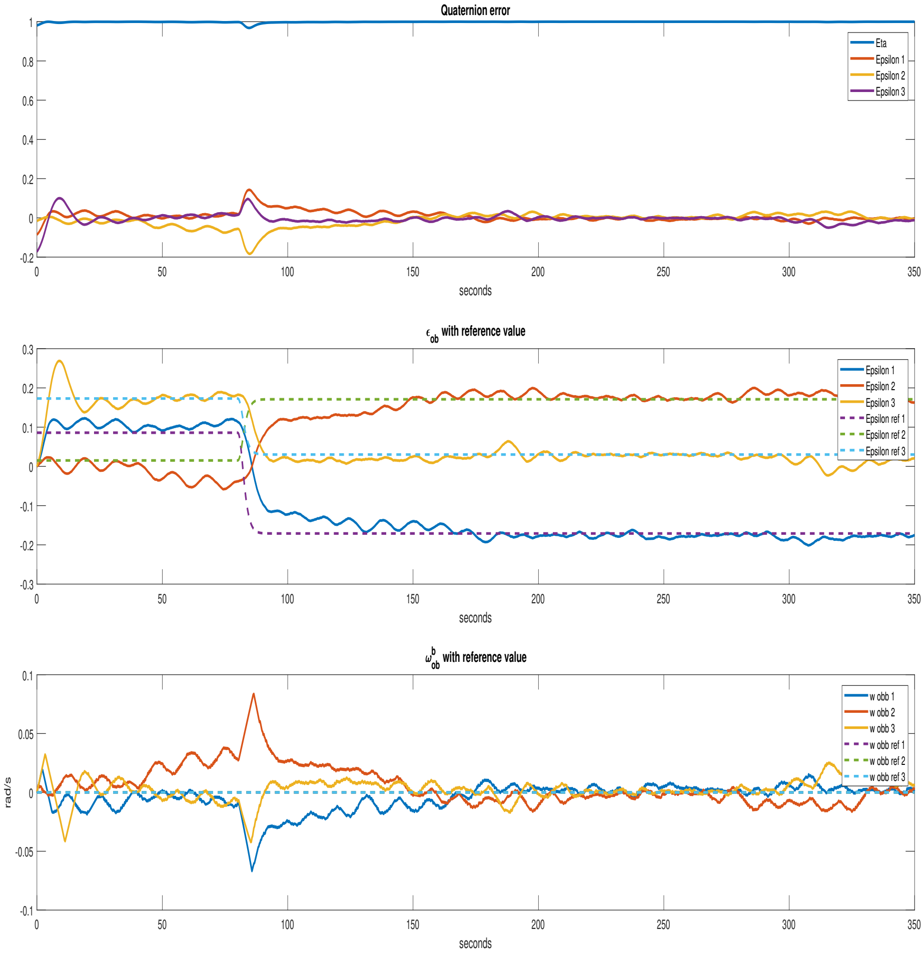


Figure 8.66: SMC pointing with delay, with observer. State variables

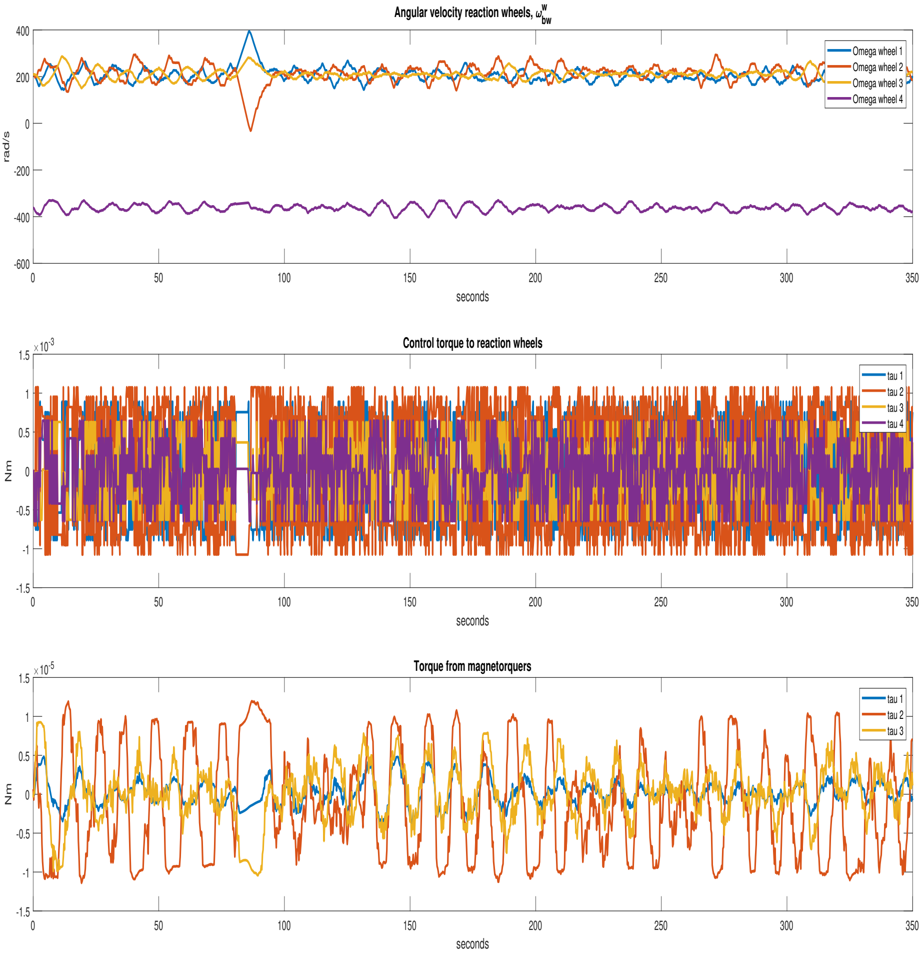


Figure 8.67: SMC pointing with delay, with observer. Torques

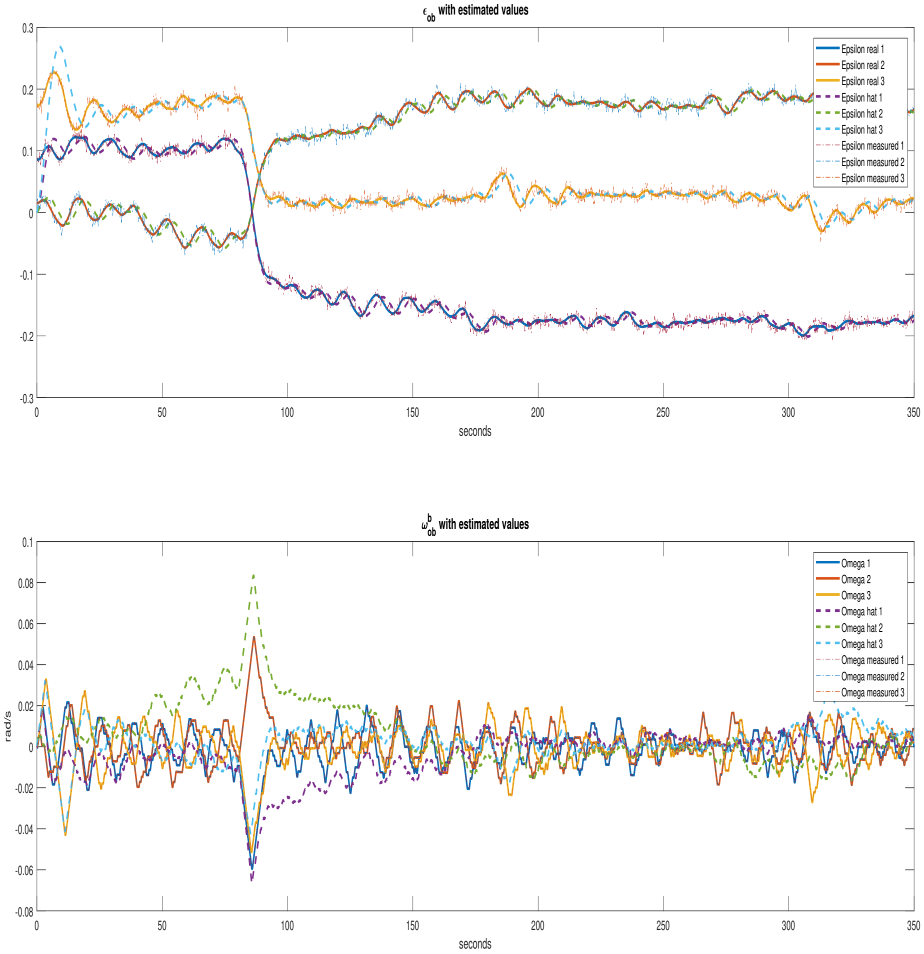


Figure 8.68: SMC pointing with delay, with observer. Estimated values

GSTA with time delay, with observer. Pointing.

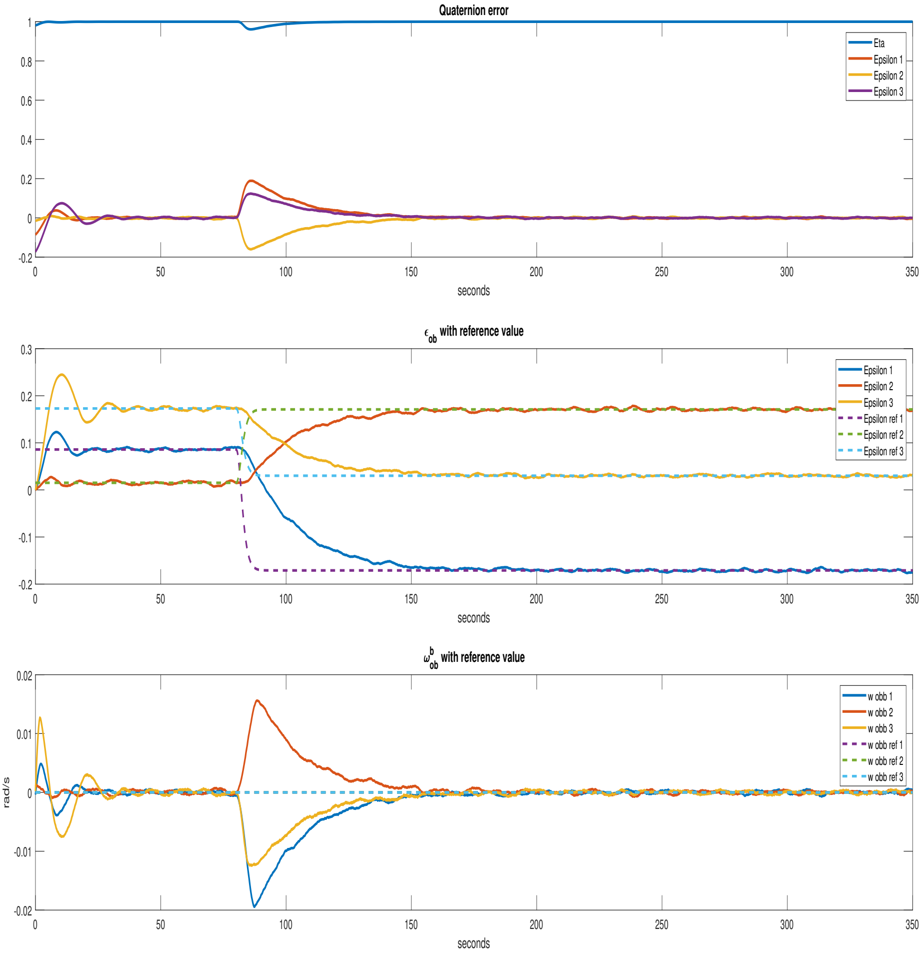


Figure 8.69: GSTA pointing with delay, with observer. State variables

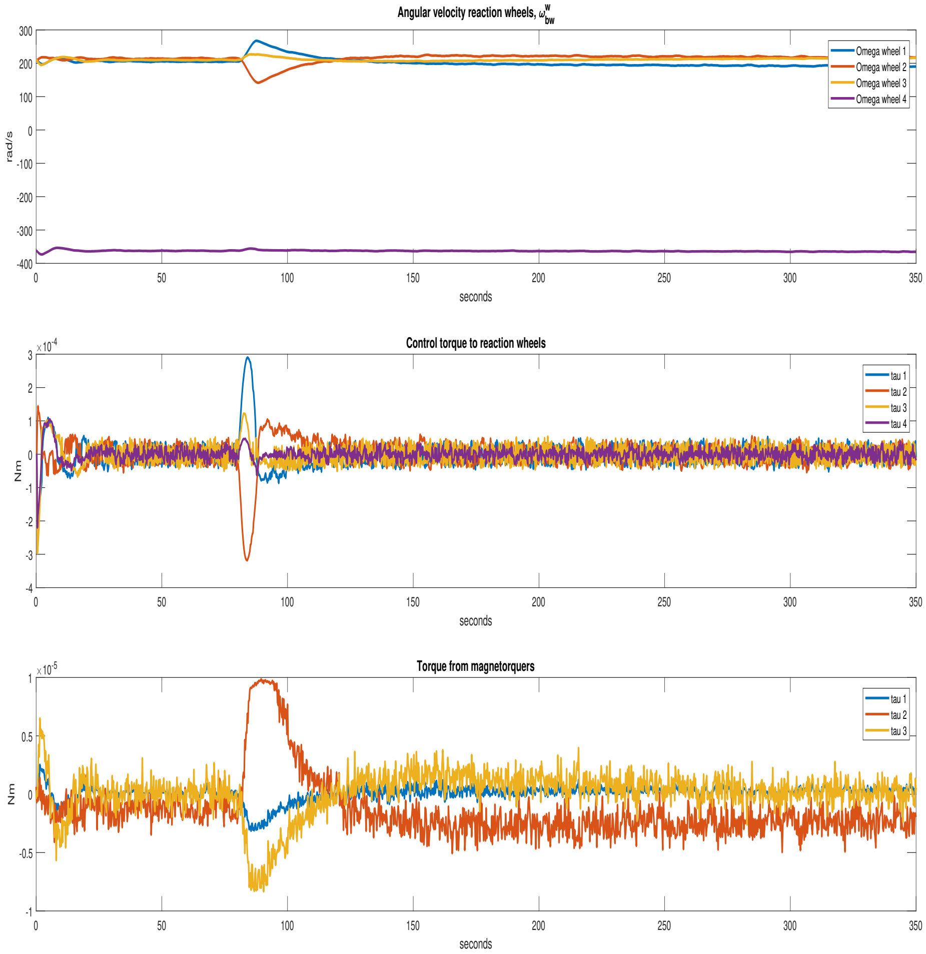


Figure 8.70: GSTA pointing with delay, with observer. Torques

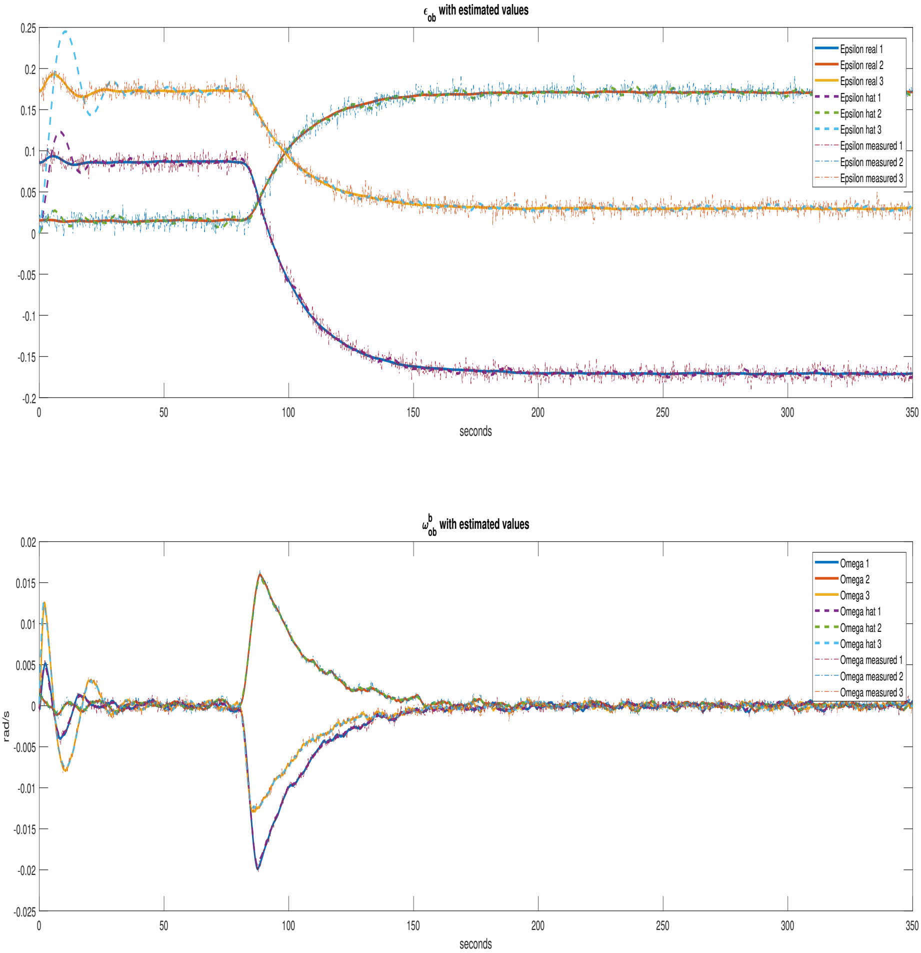


Figure 8.71: GSTA pointing with delay, with observer. Estimated values

Comparison of the different controllers based on settling time, accuracy, and steady-state error.

	Errors								
	Settling time[s]			RMSE [°]			Steady-state error[°]		
	ϵ_1	ϵ_2	ϵ_3	ϵ_1	ϵ_2	ϵ_3	ϵ_1	ϵ_2	ϵ_3
PD	166.25	147.5	171.25	0.1870	0.2199	0.3638	0.0516	0.0348	0.2481
LQR	-	-	-	-	-	-	6.1077	2.5892	4.2654
SMC	267.5	261.5	-	0.2594	0.5619	-	0.3681	0.4081	0.8239
GSTA	91.75	70.5	71.75	0.1273	0.1243	0.1236	0.0256	0.0602	0.0234

Table 8.7: Error comparison. Pointing, with delay, with observer

The LQR and SMC oscillate too rapidly for what the observer is able to follow. The controllers still manage to regulate towards the reference to a certain degree. The SMC does this better than the LQR, as two of the three channels end inside the error band. Comparing these results to the case where there was no time delay, the GSTA and PD keeps their responses almost identical, both in settling time and in RMSE values.

8.4.2.2 Slewing

PD control with time delay, with observer. Slewing.

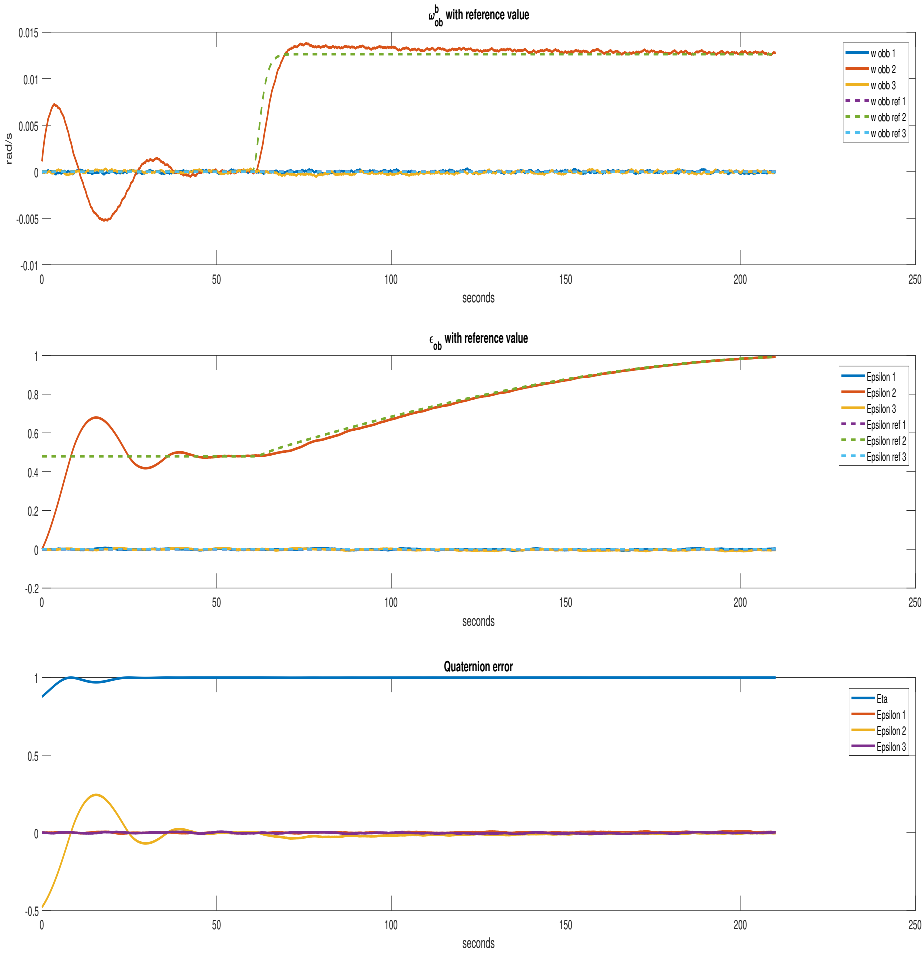


Figure 8.72: PD slewing with delay, with observer. State variables

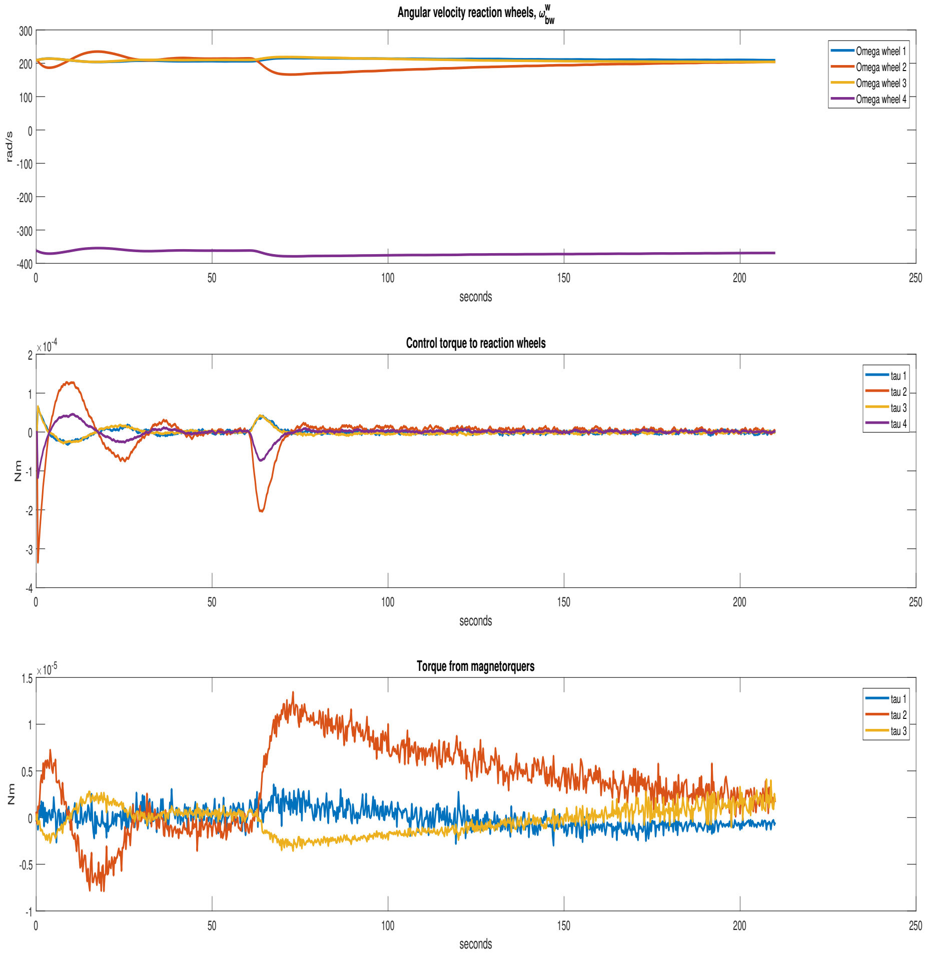


Figure 8.73: PD slewing with delay, with observer. Torques

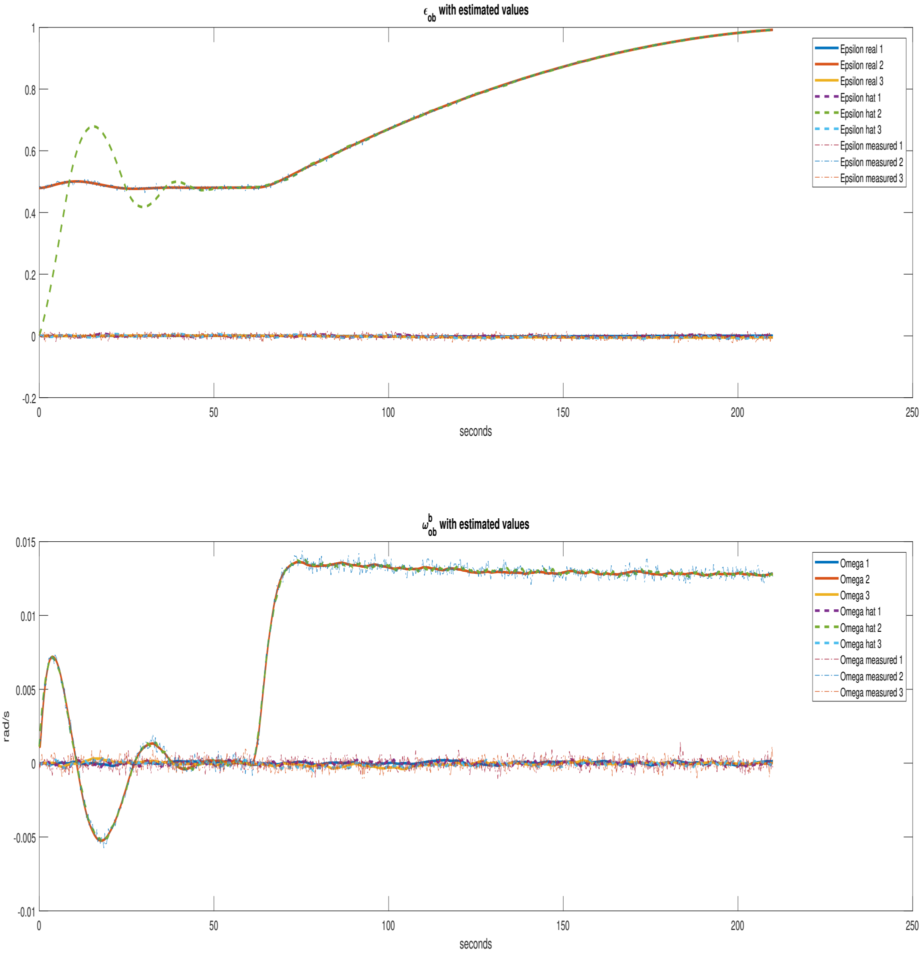


Figure 8.74: PD slewing with delay, with observer. Estimated values

LQR with time delay, with observer. Slewing.

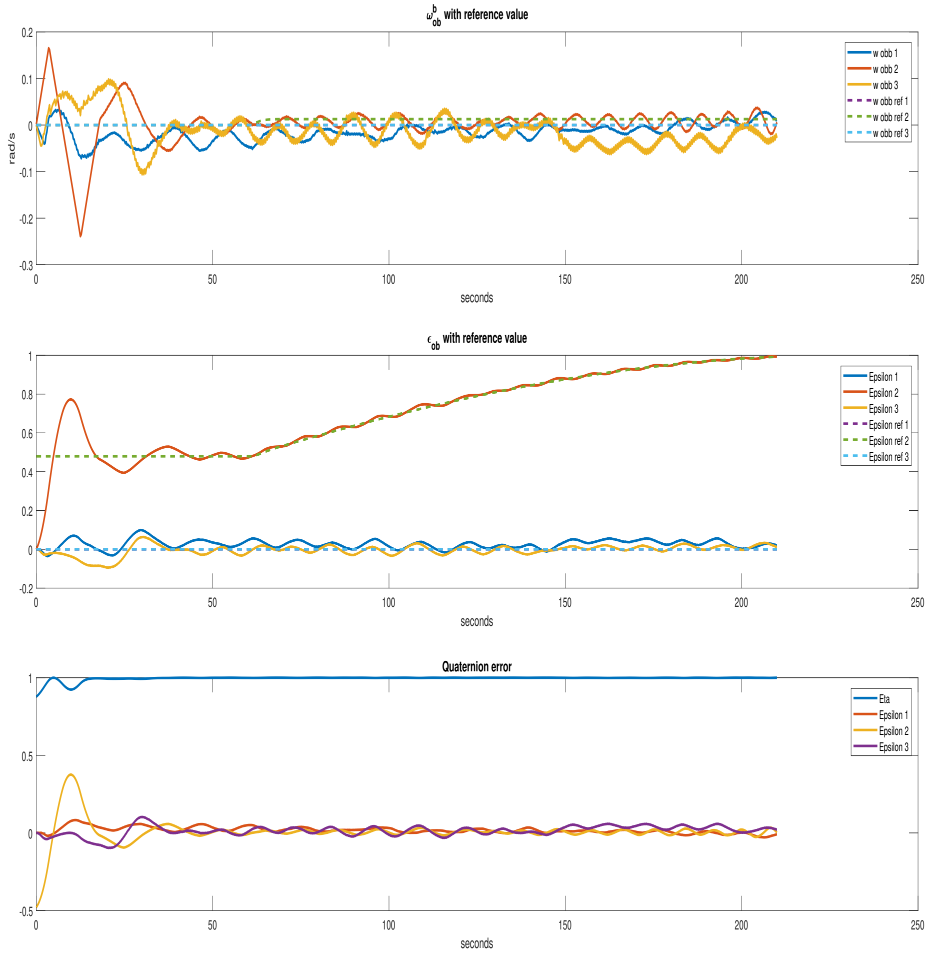


Figure 8.75: LQR slewing with delay, with observer. State variables

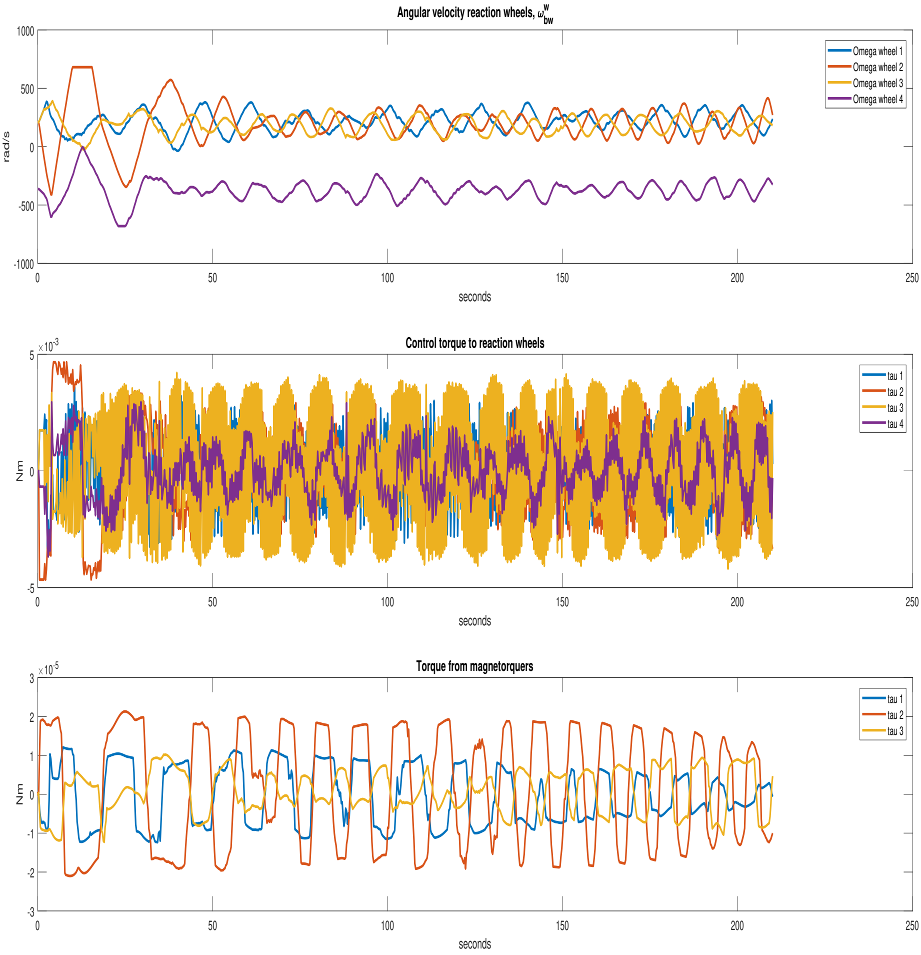


Figure 8.76: LQR slewing with delay, with observer. Torques

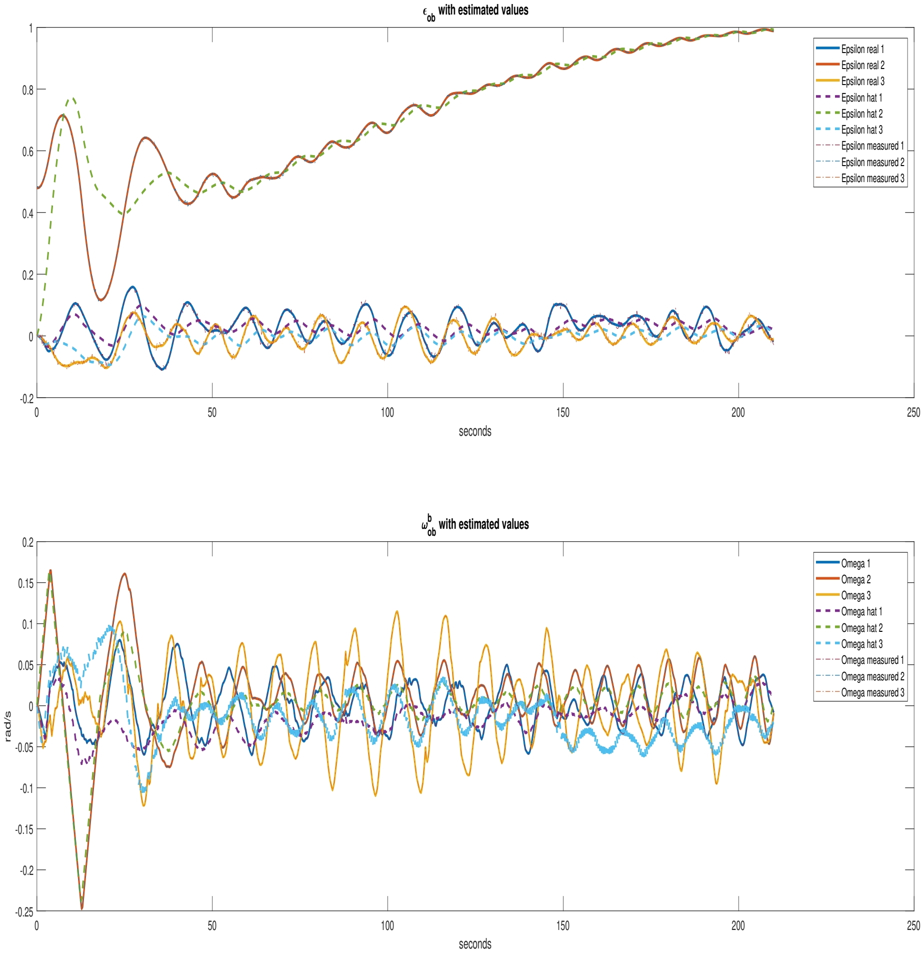


Figure 8.77: LQR slewing with delay, with observer. Estimated values

Sliding mode control with time delay, with observer. Slewing.

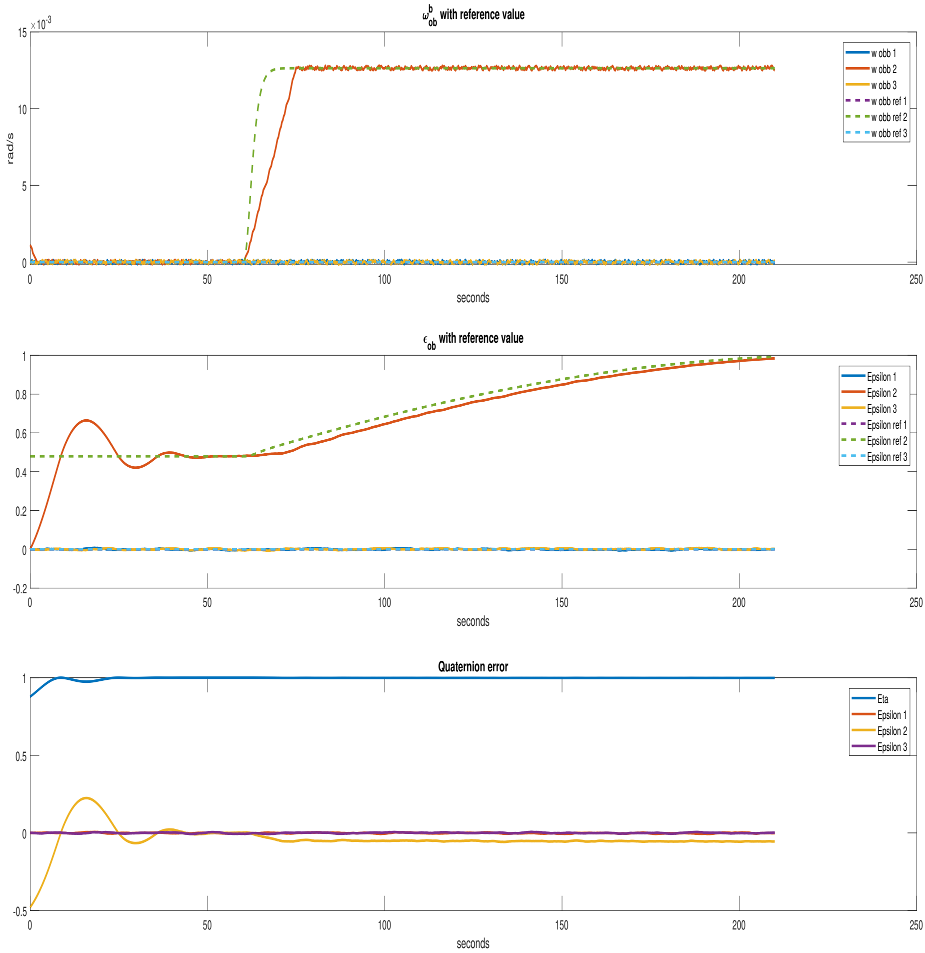


Figure 8.78: SMC slewing with delay, with observer. State variables

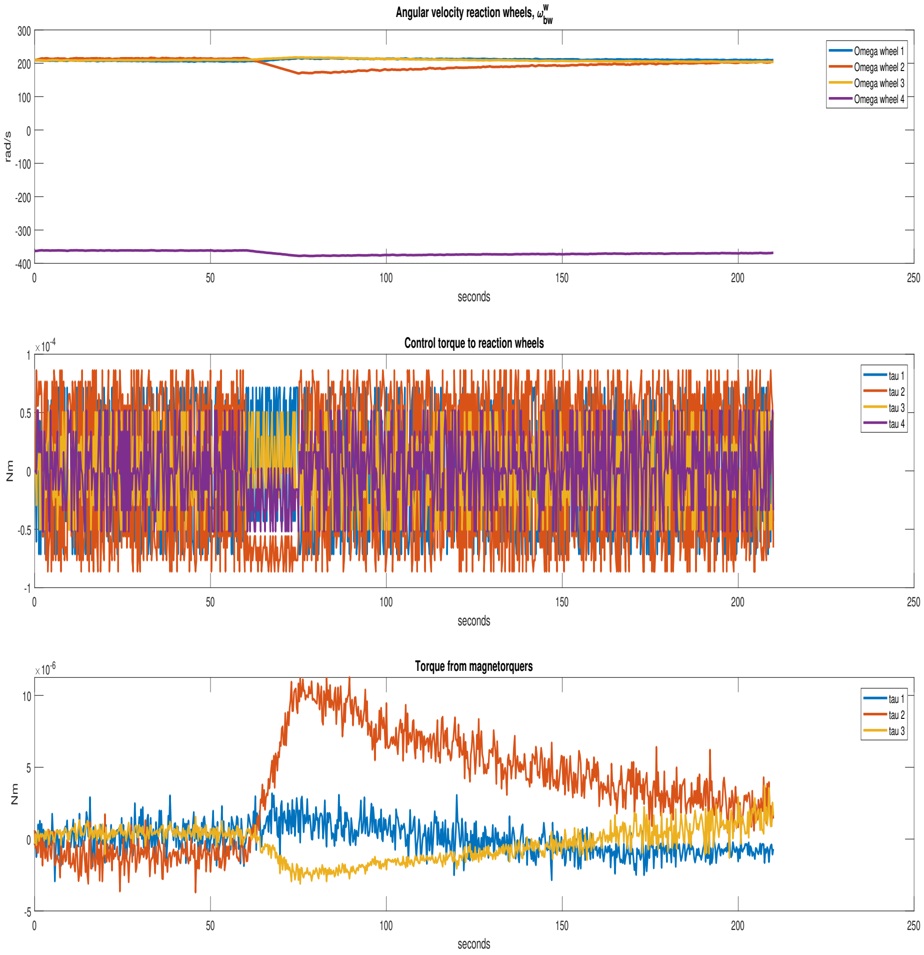


Figure 8.79: SMC slewing with delay, with observer. Torques

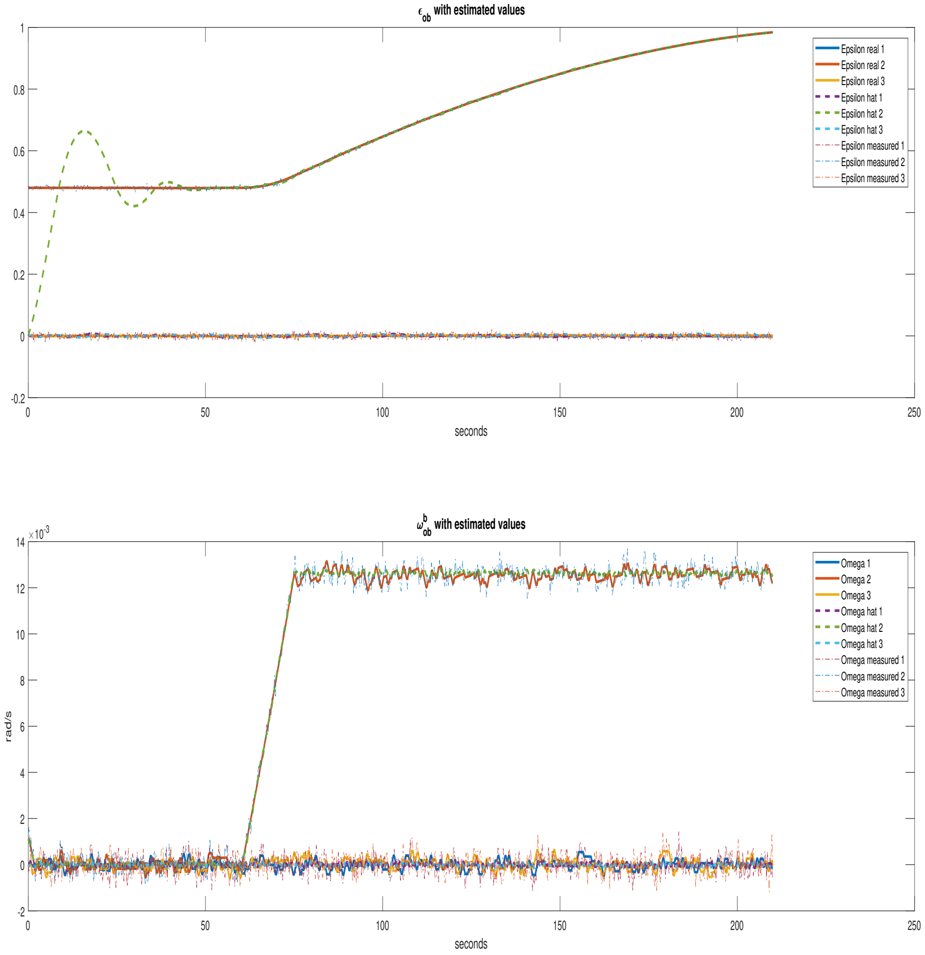


Figure 8.80: SMC slewing with delay, with observer. Estimated values

GSTA with time delay, with observer. Slewing.

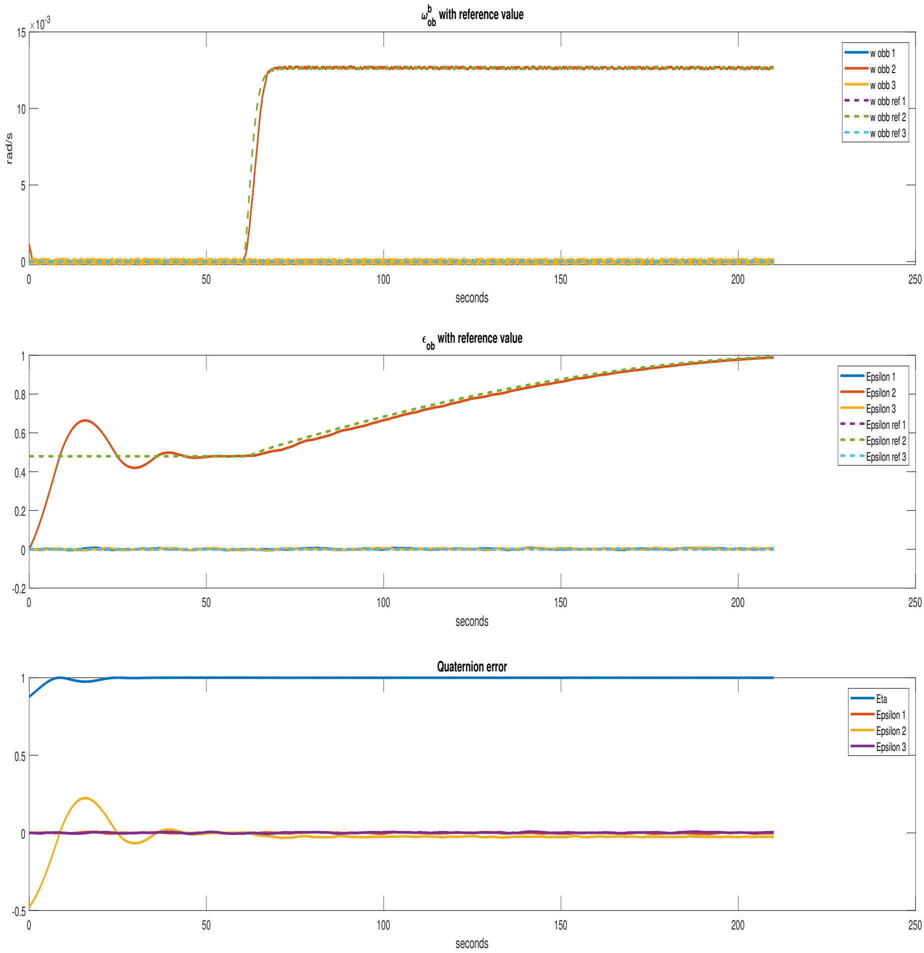


Figure 8.81: GSTA slewing with delay, with observer. State variables

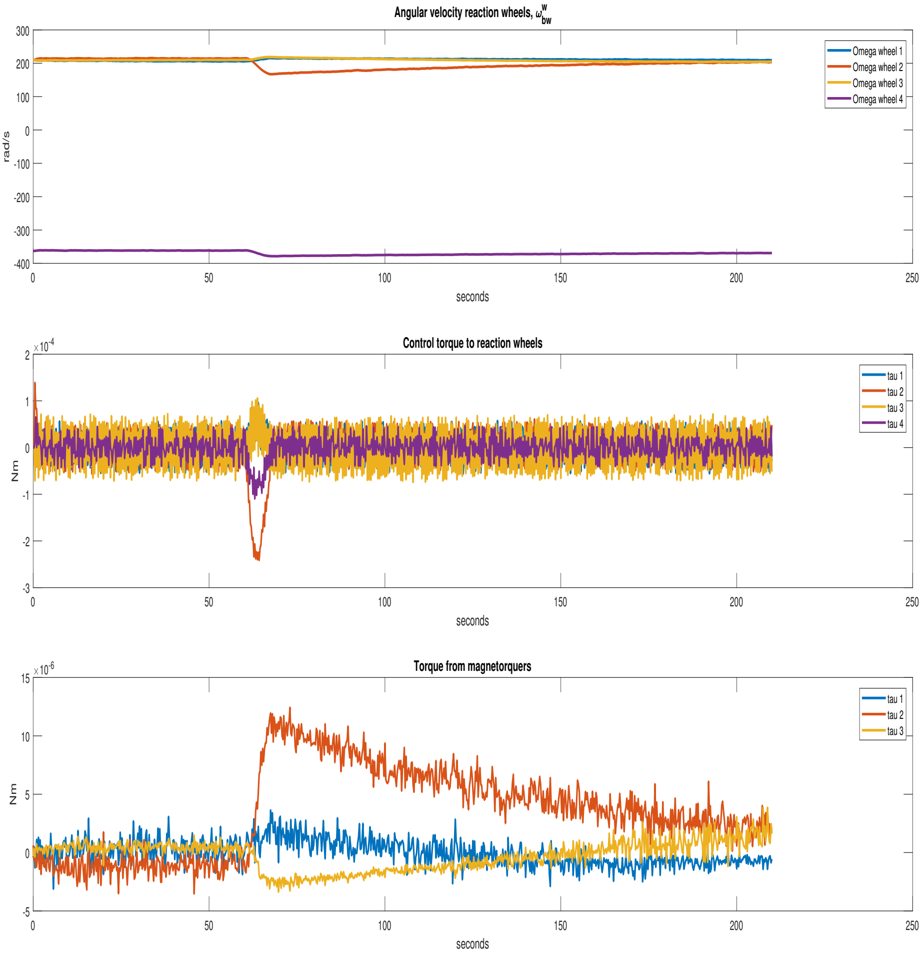


Figure 8.82: GSTA slewing with delay, with observer. Torques

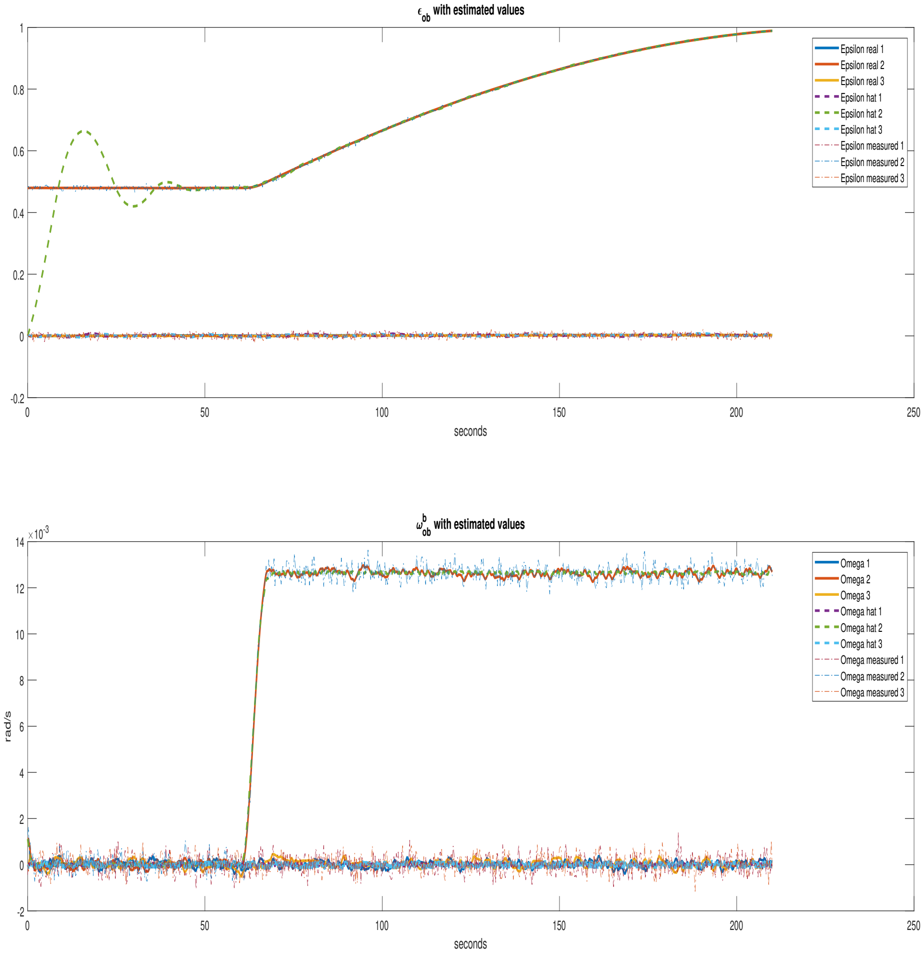


Figure 8.83: GSTA slewing with delay, with observer. Estimated values

Comparison of the different controllers based on settling time, accuracy, and steady-state error.

	Errors								
	Settling time[s]			RMSE $[\frac{rad}{s}]$			Steady-state error$[\frac{rad}{s}]$		
	ω_1	ω_2	ω_3	ω_1	ω_2	ω_3	ω_1	ω_2	ω_3
PD	-1	62	-1	0.787e-4	2.344e-4	1.313e-4	1.283e-4	1.73e-4	0.04e-4
LQR	-	-	-	-	-	-	8.3e-3	2.17e-2	8.6e-3
SMC	102.75	146.75	144.5	1.588e-4	2.369e-4	1.235e-4	1.468e-4	3.87e-4	0.34e-4
GSTA	-1	6.5	10.25	1.408e-4	1.369e-4	1.45e-4	1.123e-4	0.52e-4	0.82e-4

Table 8.8: Error comparison. Slewing, with delay, with observer

By introducing the time delay, the LQR combined with the observer seems to be performing worse than it was before with the observer. The SMC looks to work well, but when the time delay is added, the controller makes the response jump out of the error band more, and thus the settling time is increased. The PD and the GSTA works the same as they did without the time delay.

8.5 Controller discussion

This section is a discussion on the results from sections 8.3 and 8.4.

Throughout the tests, the PD control with gyroscopic compensation and the GSTA show consistent performance. The LQR and the SMC vary more depending on the change in control objective, observer, and time delay, and thus look like the less robust of the four controllers.

For pointing, the GSTA has consistently the best RMSE values, and the same kind of settling times at every step. But they are not the fastest settling times, as both the LQR and the SMC originally reach the reference faster. The GSTA settling time is often the best anyway, as the LQR and SMC may escape the error band. For slewing, the RMSE values of the controllers that managed to reach a settling time are similar to each other, though ranking the controllers by their settling times is possible. The GSTA is the fastest, the SMC is the second fastest whenever the observer is not connected with a time delay, and the PD is slow compared to the sliding mode controllers. The PD controller settles slowly due to it having a small overshoot in the angular velocity, due to the desire to track the epsilon values as well. The LQR does not always reach a good slew rate, and even when it does, the angular velocity is visibly contaminated by noise. A way to try to fix this could be to change the calculated LQR gain used for slewing to only include the angular velocity error, not the epsilon error.

The LQR and SMC work worse with the observer when there is a time delay. In fact, the observer does not seem to add any performance to the system, but rather slightly decrease the performance compared to the system without observer. In some cases, such as the time delayed system with the LQR or SMC, using the observer would make the system less stable. Because of this, it is reasonable to suggest that the observer should not be used as it is. A different observer should be developed to take advantage of whichever of the measurements, from the IMU or from the star tracker, are the most accurate. An example of this is the observer in equation 6.14.

8.6 Momentum dumping

Without any measurement noise, and with no observer, it becomes easier to compare and to see the importance of the momentum dumping control the magnetorquers provide. The satellite is fed two references, which are switching at an interval in between each new reference in order for the system to settle. The GSTA controller is used.

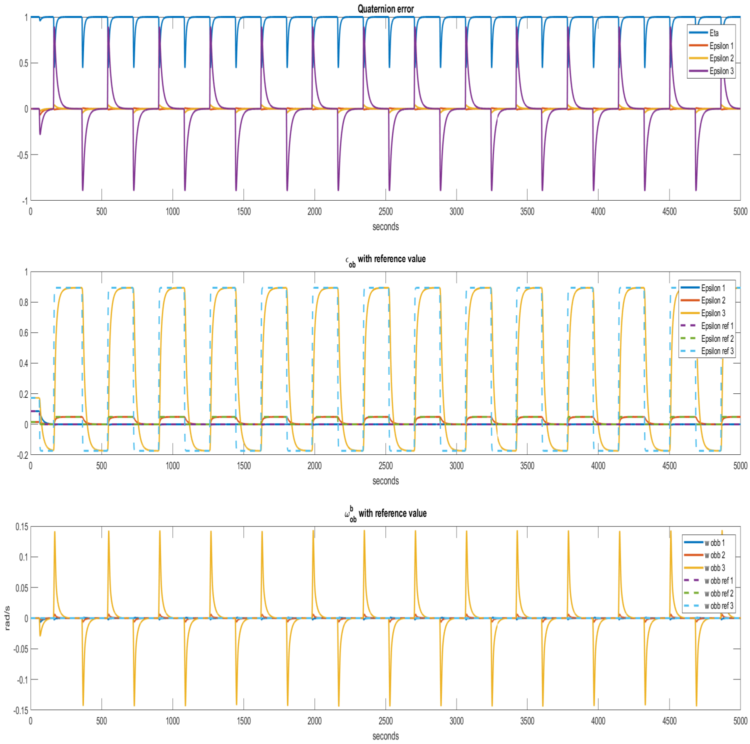


Figure 8.84: With momentum dumping, reference switching. State variables

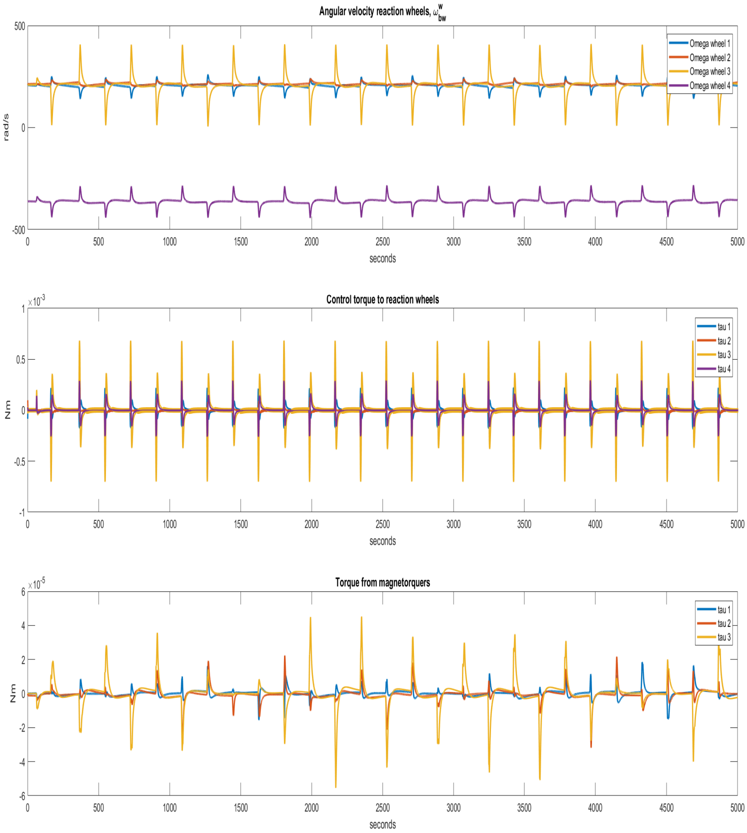


Figure 8.85: With momentum dumping, reference switching. Torques and reaction wheel angular velocity

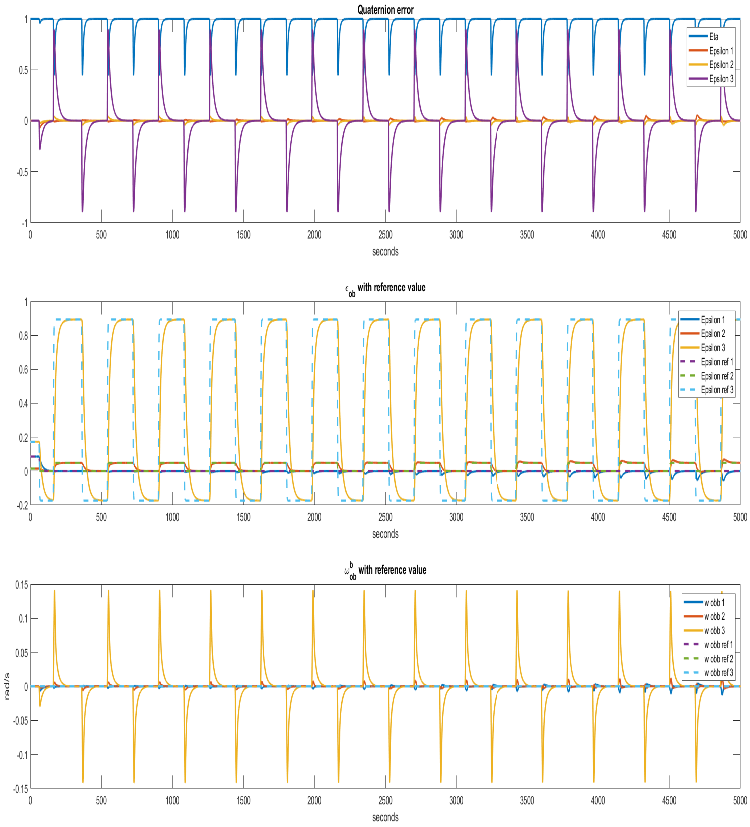


Figure 8.86: Without momentum dumping, reference switching. State variables

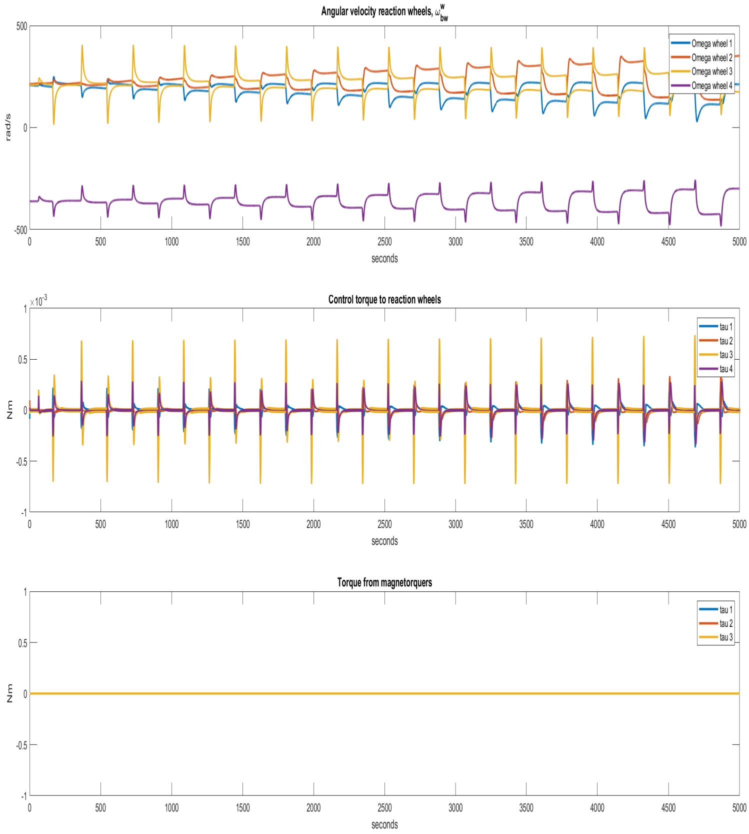


Figure 8.87: Without momentum dumping, reference switching. Torques and reaction wheel angular velocity

The figures show the difference the momentum dumping control makes. The angular velocity of the reaction wheels are kept at an almost constant level when the momentum dumping controller is active, whereas the lines clearly diverge when the controller is missing. If this simulation went on indefinitely, the impact of this would have become more clear: The reaction wheels would saturate without the momentum dumping controller, and the satellite would because of it not be able to be controlled toward the desired attitude.

8.7 Uncertainty in assembly of reaction wheels

The nominal torque distrubution matrix describing the placement of the reaction wheels is shown in equation 7.2, here repeated for convenience

$$\mathbf{A}_{nom} = \begin{bmatrix} 1 & 0 & 0 & \sqrt{\frac{1}{3}} \\ 0 & 1 & 0 & \sqrt{\frac{1}{3}} \\ 0 & 0 & 1 & \sqrt{\frac{1}{3}} \end{bmatrix} \tag{8.1}$$

The real placement of the wheels is in this section given as a perturbed version of the torque distribution matrix, given as

$$\mathbf{A} = \begin{bmatrix} 0.9879 & 0.0902 & -0.1098 & 0.5442 \\ -0.1098 & 0.9918 & -0.1098 & 0.6385 \\ -0.1098 & 0.0902 & 0.9879 & 0.5442 \end{bmatrix} \tag{8.2}$$

The controllers are compared with the results from section 8.3.1.1, the section where the controllers are used for pointing without observer and no time delay is inserted into the system. Note that there is no uncertainty in the reaction wheel assembly in section 8.3.1.1.

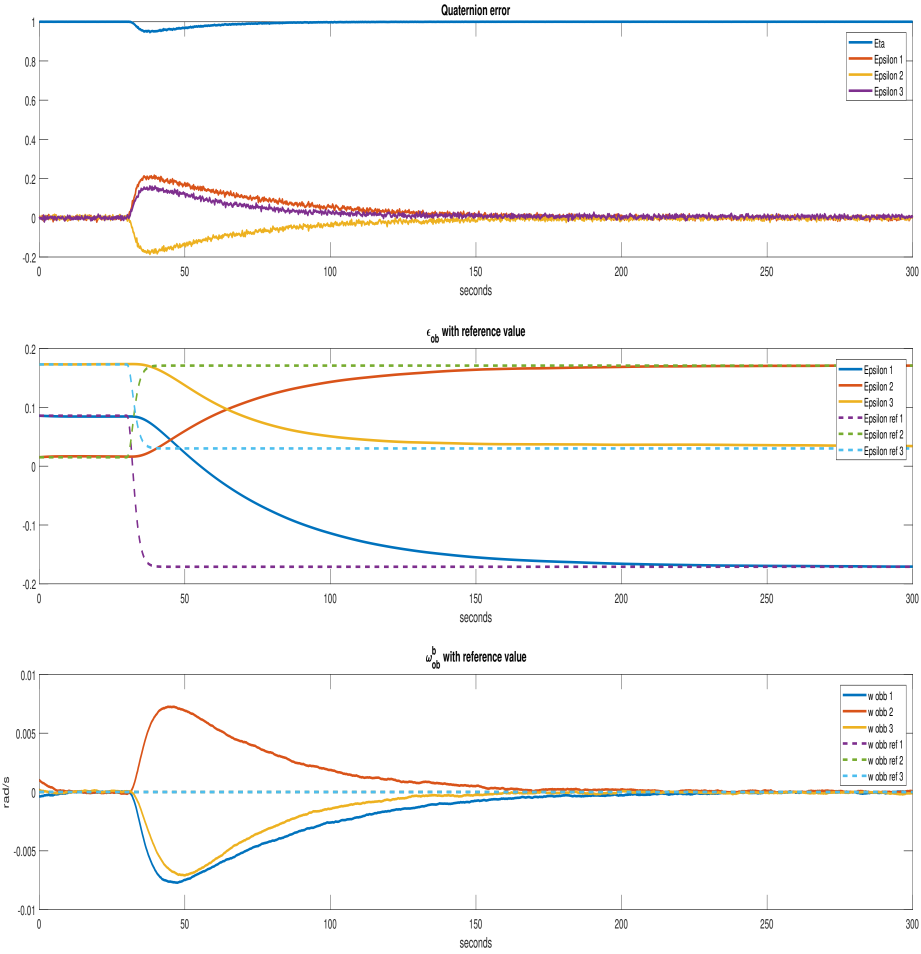


Figure 8.88: PD pointing without delay, without observer. With uncertain assembly. State variables

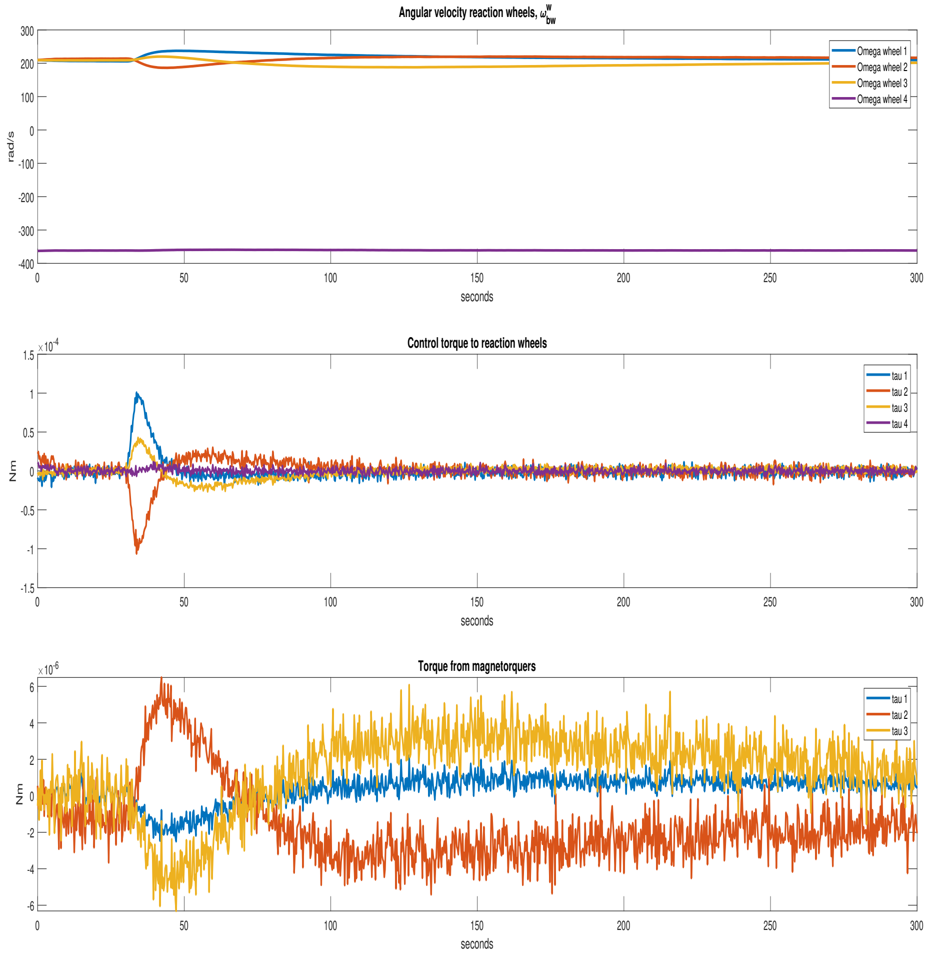


Figure 8.89: PD pointing without delay, without observer. With uncertain assembly. Torques

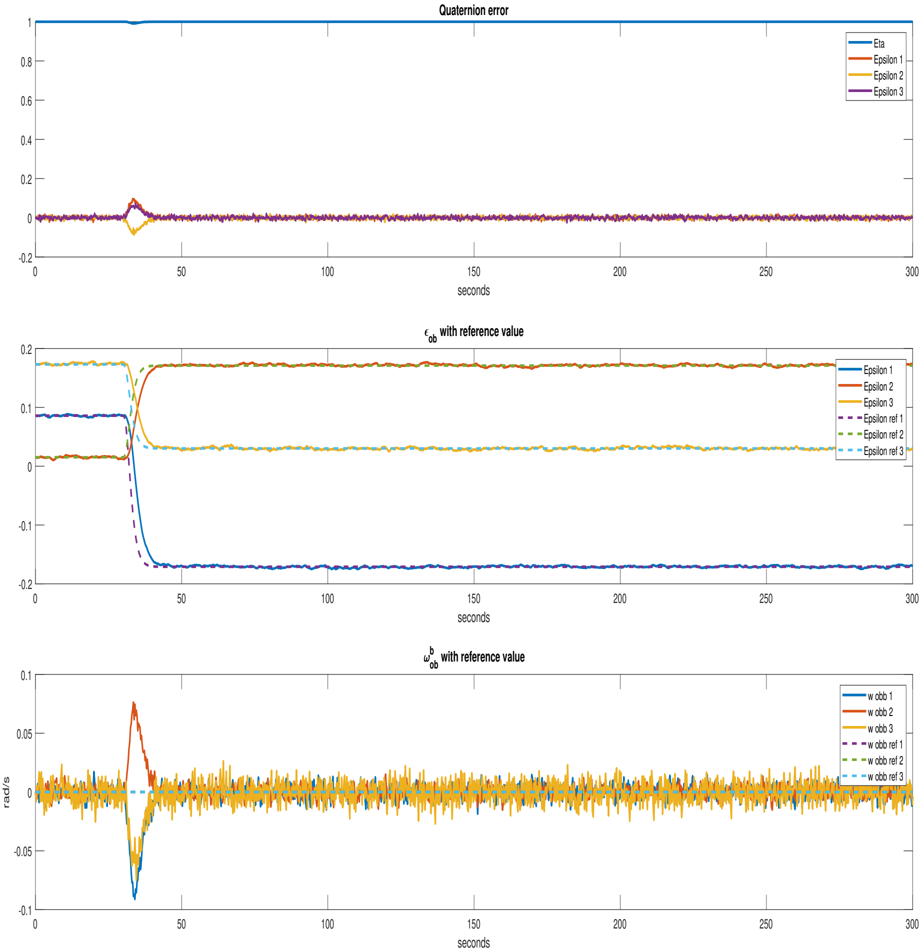


Figure 8.90: LQR pointing without delay, without observer. With uncertain assembly. State variables

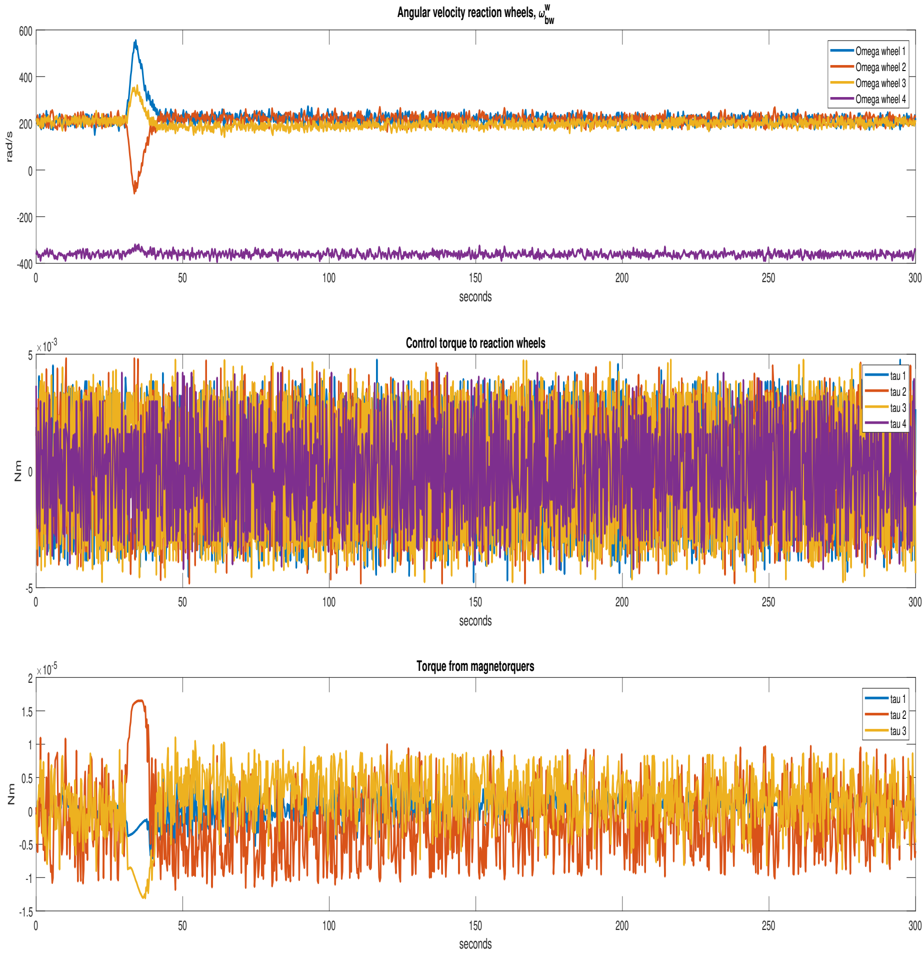


Figure 8.91: LQR pointing without delay, without observer. With uncertain assembly. Torques

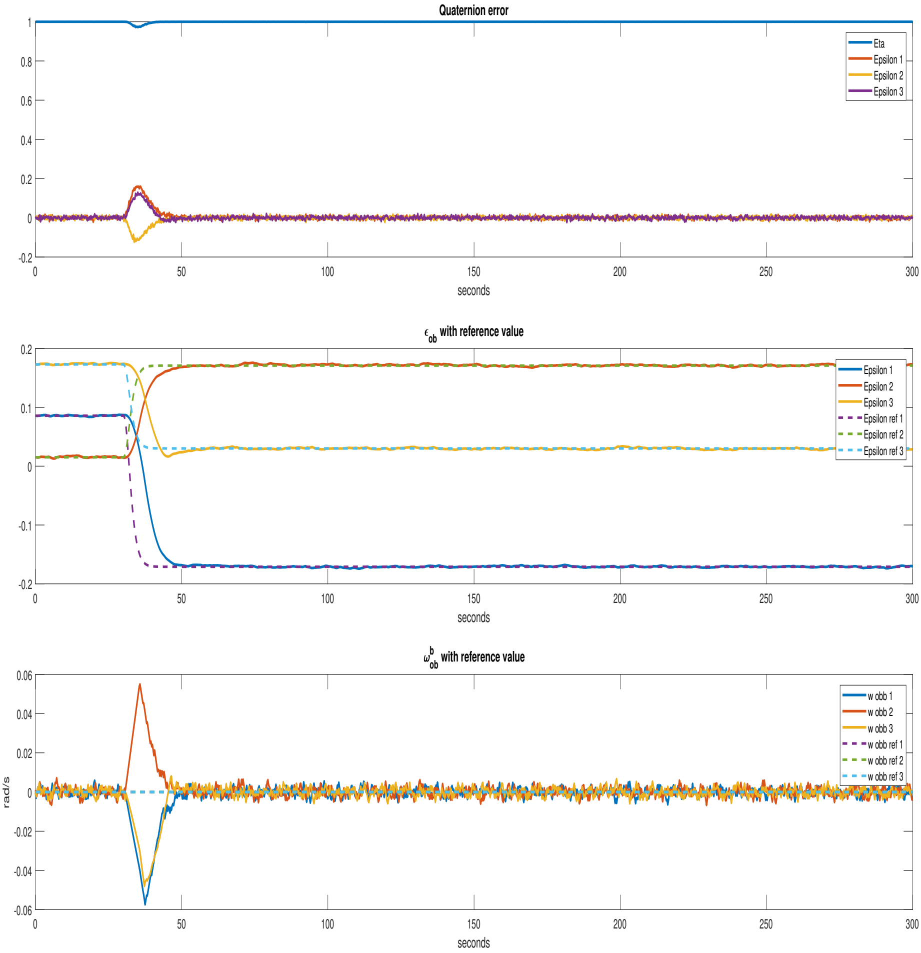


Figure 8.92: SMC pointing without delay, without observer. With uncertain assembly. State variables

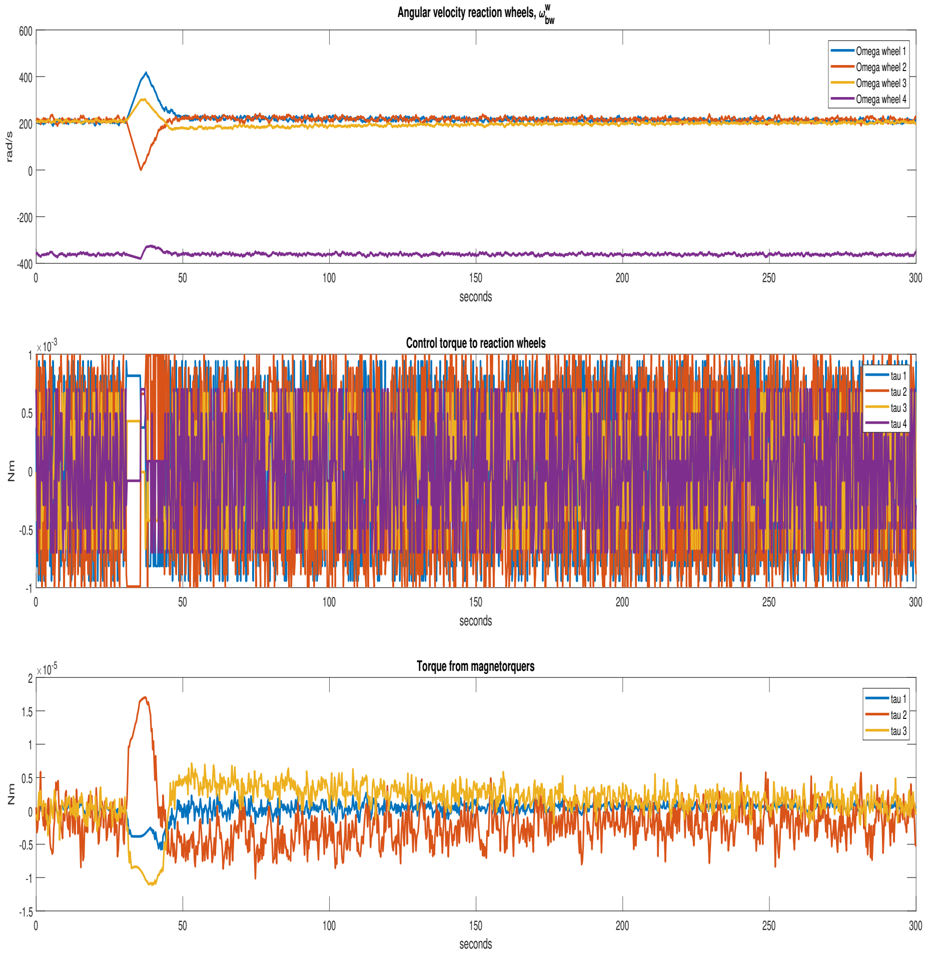


Figure 8.93: SMC pointing without delay, without observer. With uncertain assembly. Torques

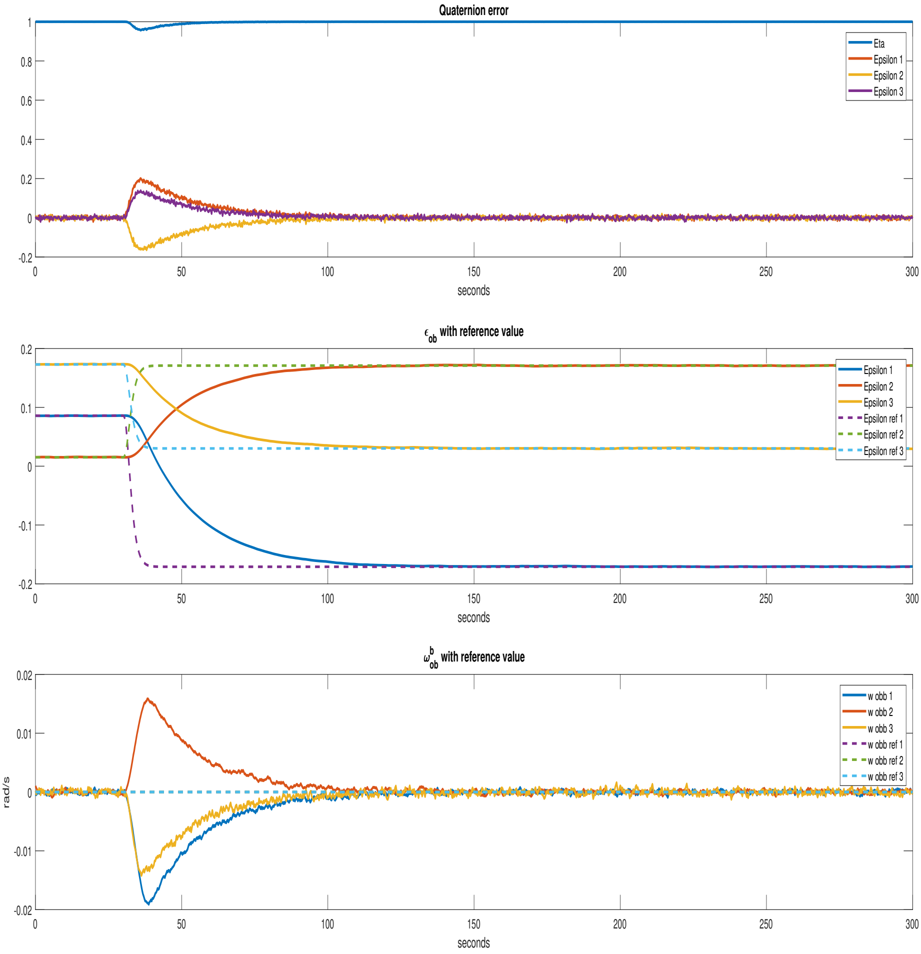


Figure 8.94: GSTA pointing without delay, without observer. With uncertain assembly. State variables

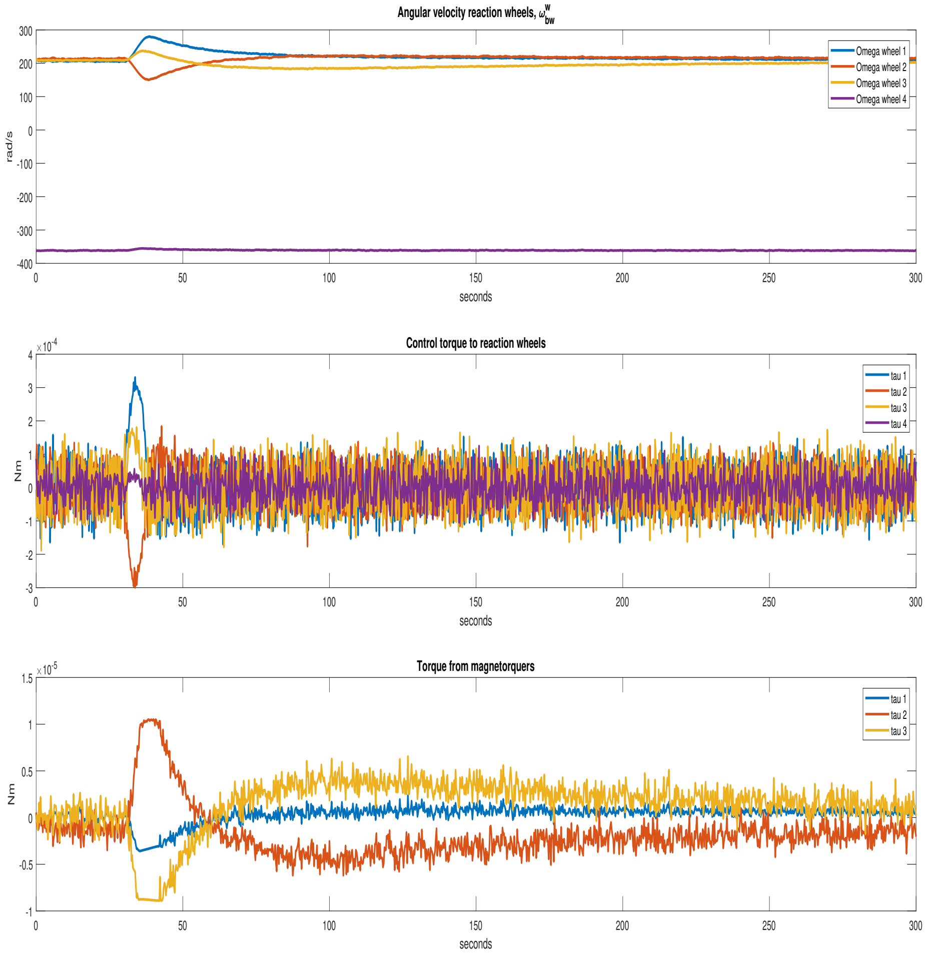


Figure 8.95: GSTA pointing without delay, without observer. With uncertain assembly. Torques

	Errors								
	Settling time[s]			RMSE [°]			Steady-state error[°]		
	ϵ_1	ϵ_2	ϵ_3	ϵ_1	ϵ_2	ϵ_3	ϵ_1	ϵ_2	ϵ_3
PD	176.5	144.75	265.5	0.2304	0.1896	0.4760	0.0299	0.0023	0.4589
LQR	14	140.25	249.75	0.1740	0.1725	0.2202	0.1084	0.1995	0.0103
SMC	17.5	44.75	22	0.1388	0.1638	0.1488	0.0858	0.1824	0.1724
GSTA	79.5	67.25	72.25	0.1177	0.1069	0.1239	0.0441	0.0298	0.0797

Table 8.9: Error comparison. Pointing, without delay, without observer. Uncertainty in assembly of reaction wheels

The performance of the PD control drops when the uncertainty in the reaction wheel assembly is introduced. The controllers based on sliding mode, the SMC and the GSTA, as well as the LQR seem to be robust with respect to uncertainty in the placement of the reaction wheels. The settling times for SMC improves along the first and third channel. The settling time and RMSE improve for the LQR. For both the SMC and the GSTA, the RMSE values go up slightly. This may be due to the fact that the current initial reaction wheel angular velocities are set to create zero total angular velocity in total. When the configuration is changed, i.e. the torque distribution matrix is changed, this is no longer the case and the system reacts slower. The response of the momentum dumping controller is better seen when a switching reference scheme for target pointing is used. The GSTA is used, without time delay, observer or measurement noise.

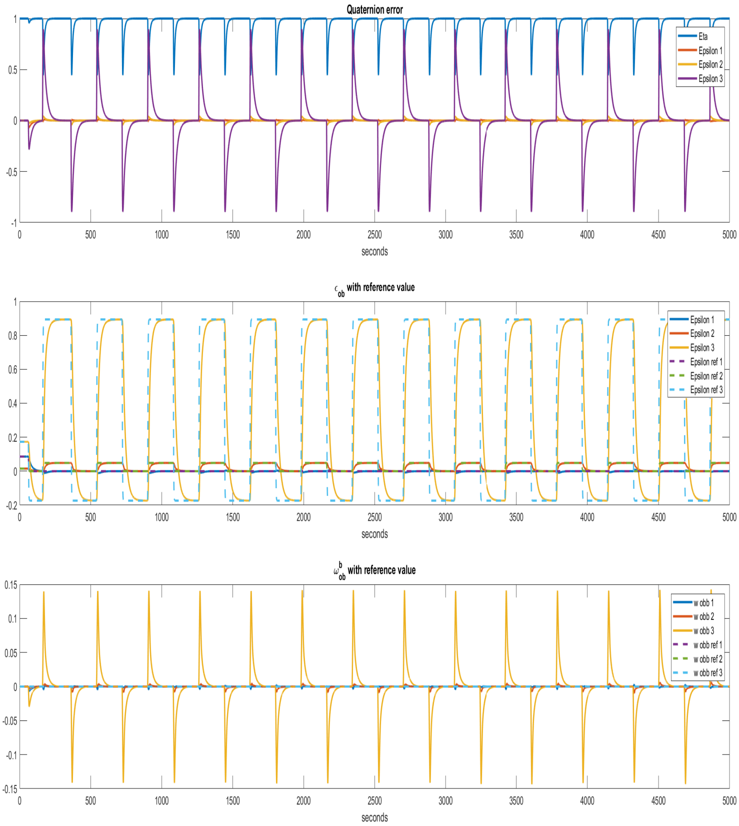


Figure 8.96: GSTA pointing without delay, without observer. With uncertain assembly. Reference switching. State variables

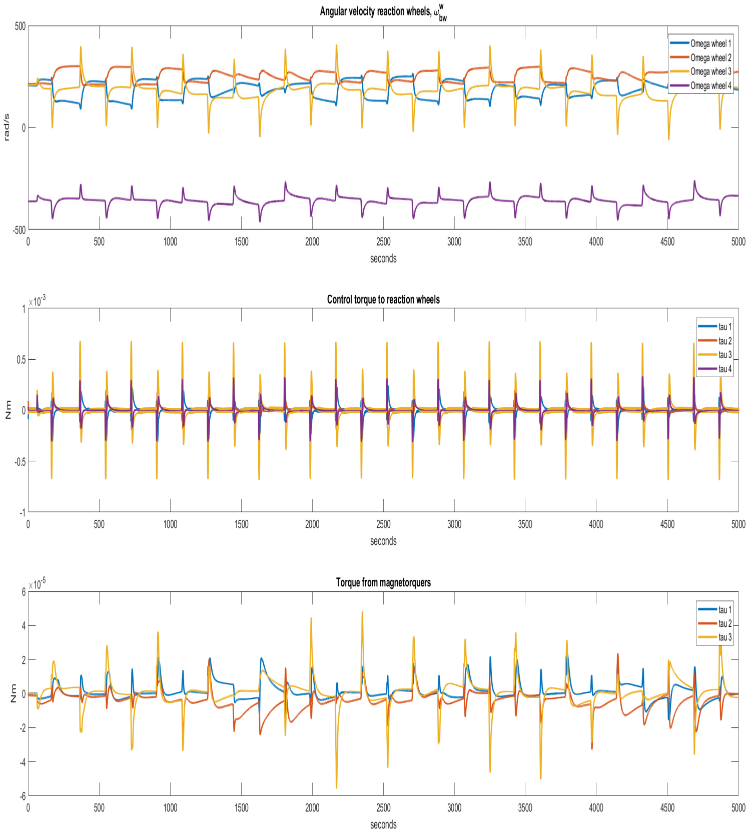


Figure 8.97: GSTA pointing without delay, without observer. With uncertain assembly. Reference switching. Torques

The momentum dumping controller deviates more compared to figures 8.86 and 8.87, but it still manages to control the momentum of the wheels. Note that the momentum dumping control is dependent on the torque distribution matrix, see equation 5.5.

8.8 Full mission

The following section show the result for the nominal operations with HSI for the HYPSO satellite. The satellite begins the scenario with pointing nadir. Then, it points [0;-20°;0] in the orbit frame. After that it slews at the rate 0.7238°/s along the second axis for 65 seconds. When the slewing is over, the satellite returns to pointing nadir.

The full mission is performed with the GSTA controller, using the combination of the third order HOSMO for quaternions and first order SMO for angular velocity. The simulations include a 0.25 second time delay, uncertainty in the reaction wheel assembly, and measurement noise. Sampling rate is 4Hz.

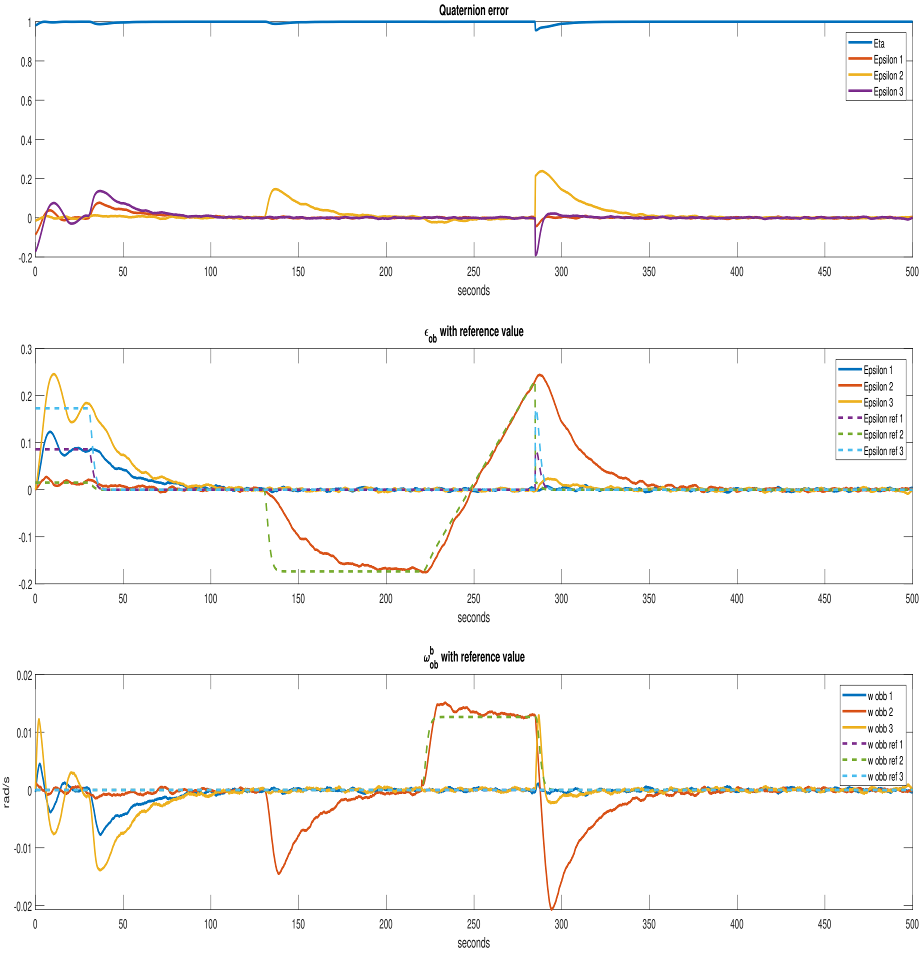


Figure 8.98: Tracking for the full mission

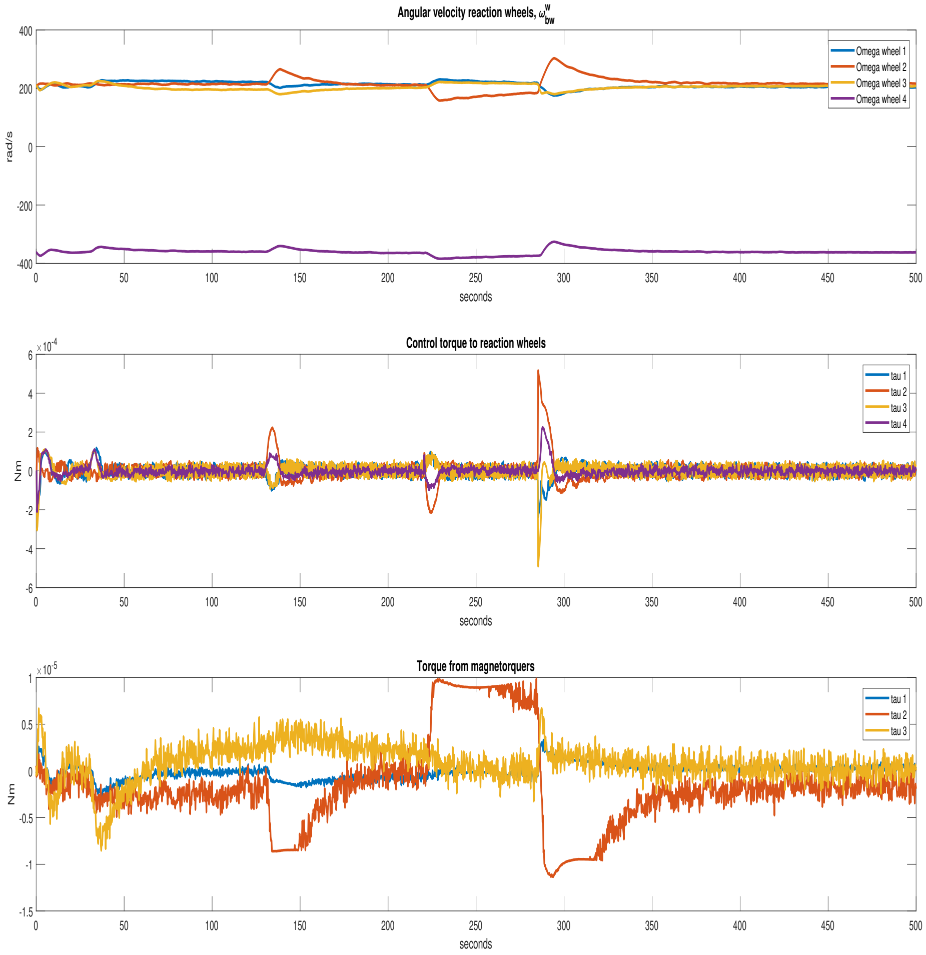


Figure 8.99: Full mission control torques and reaction wheel angular velocity

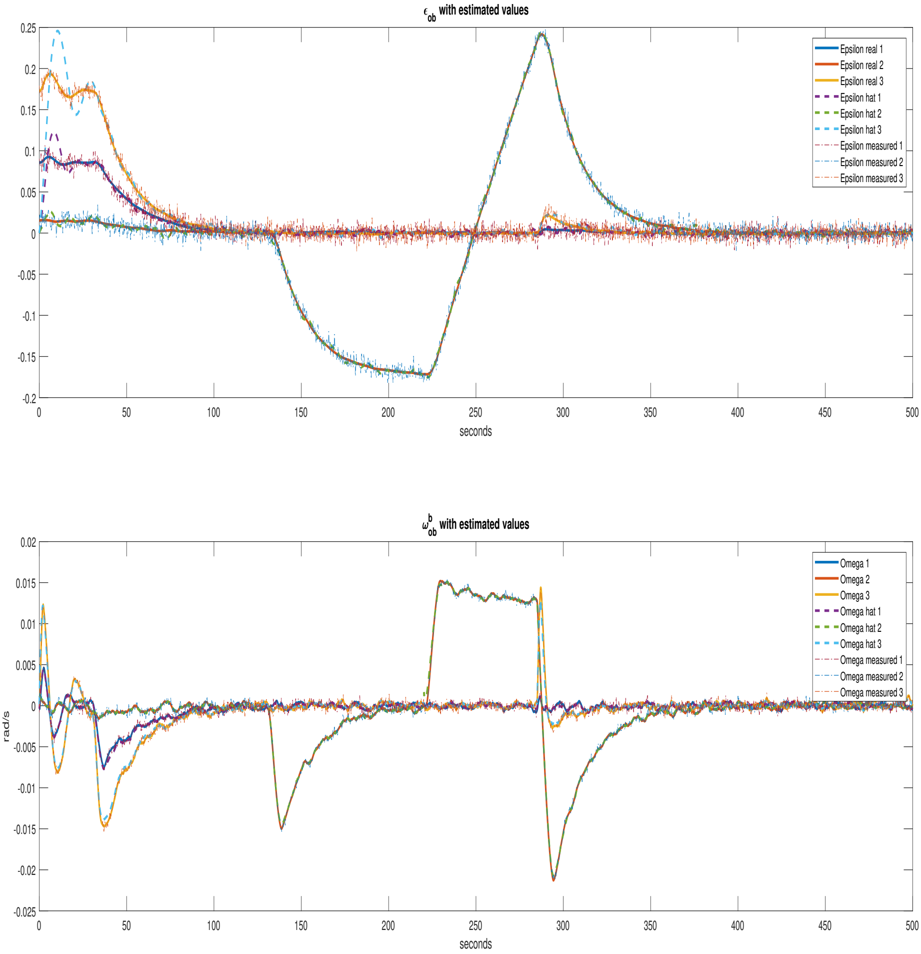


Figure 8.100: Full mission observed values

Note the small overshoot in slewing rate that was not present without the perturbed torque distribution matrix.

The only parameter that has not been evaluated as to it affect the performance of the system in this report is the sampling rate. A sampling rate of 4Hz is not very high for a modern computer. Below are the full mission simulations, run with a 40 Hz sampling rate. The time delay is still 0.25 seconds.

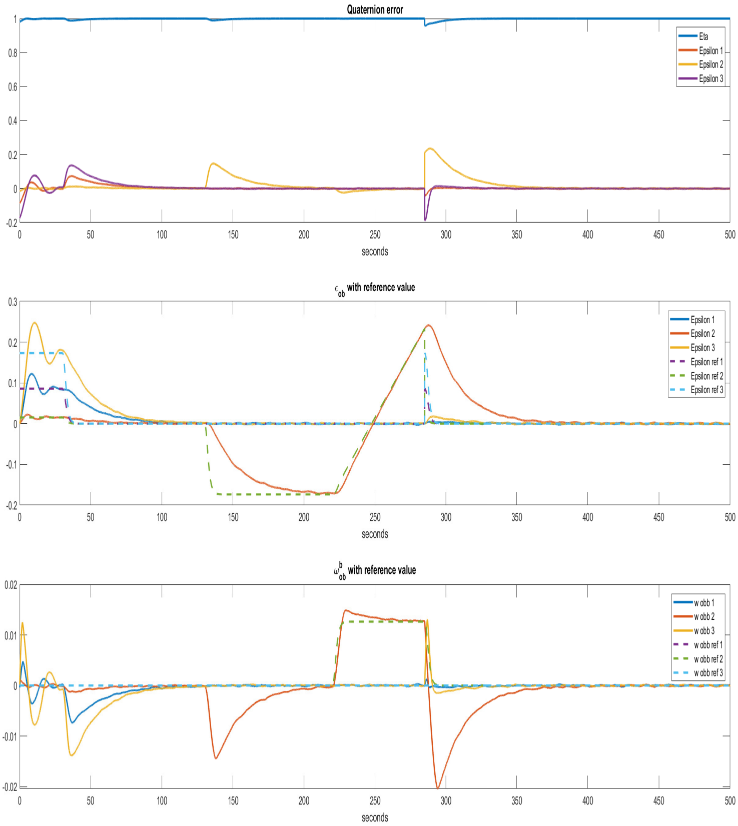


Figure 8.101: Tracking for the full mission, 40 Hz

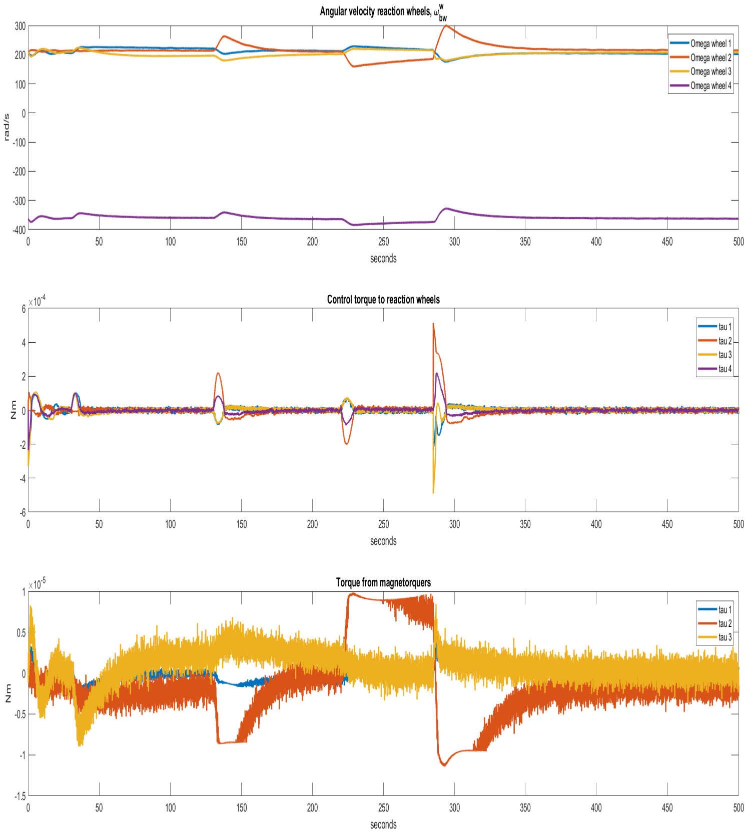


Figure 8.102: Full mission control torques and reaction wheel angular velocity, 40 Hz

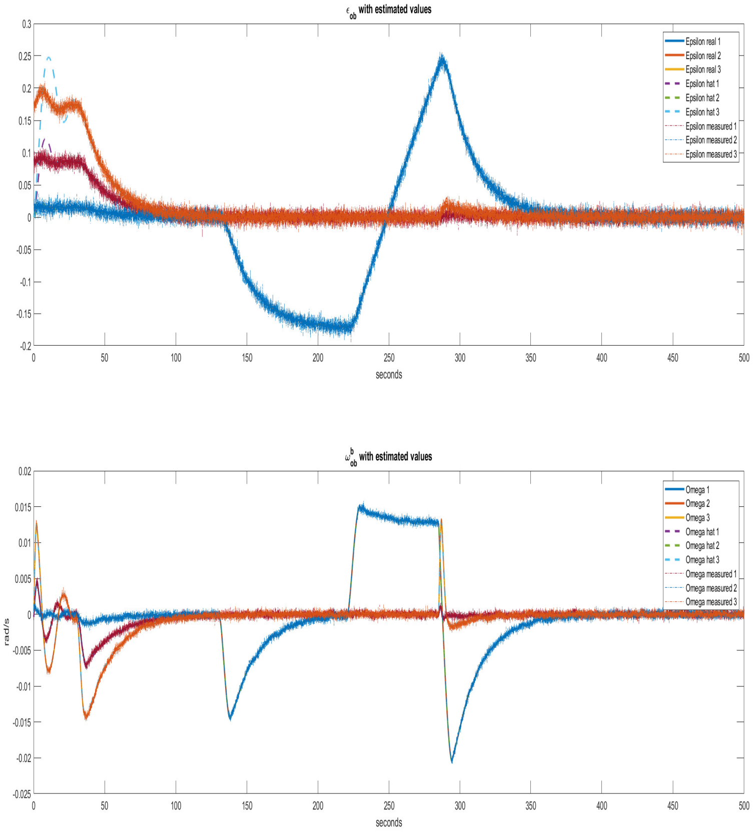


Figure 8.103: Full mission observed values, 40 Hz

As can be seen from the state variable plots, the response is smoother when the sampling rate is increased.

Chapter 9

Conclusion

9.1 Findings

The GSTA seems to be the most robust of the controllers tested. It worked well with the observer, with time delay, and with the uncertainty in the assembly of the reaction wheels. The PD control with gyroscopic compensation worked well with the observer and with time delay, but was not robust to the assembly uncertainty. The SMC and LQR both seem to do fine with uncertainty in the assembly of the reaction wheels, but struggle when connected to the observer. The SMC performs well without the observer. The LQR implementation used is not ideal for slewing. By most performance measures, the GSTA is the most desirable control law, aside from settling time during target pointing, where both LQR and SMC at times was able to produce a faster response.

As for the HYPSO mission, it would be necessary to gather more information about the specifications on the $\boldsymbol{\omega}_{ib}^{b}$ measurements. Should the system not be any worse than what is simulated, then it can be seen that the GSTA is good enough for slewing. It has a fast settling time, and a low enough RMSE. For pointing, no conclusion can be made. The required RMSE value for HYPSO is 0.01°, which is about 10% of what the GSTA manages to produce here.

9.2 Future work

Future work related to the observers would be to solve the quaternion reconstruction problem explained in section 8.2 and find a way around the singularity in the stability proof for the higher order sliding mode observer in section 6.2. Observing both states of interest based on one measurement could be desirable, as mentioned in section 8.5.

Further analysis on pointing accuracy to reach the desired RMSE values for the HYPSO mission can be done in two ways: increasing the efficacy of the controller, or decreasing the effect of noise. The first can be attempted with tuning, should the chosen controller be GSTA for its robustness. The latter can be done either by using high quality sensors or by using an effective observer.

Appendix A

Mission overview

A.1 HYPSO

HYPer-spectral Smallsat for ocean Observation (HYPSO) is a small satellite project at NTNU Centre for Autonomous Marine Operations and Systems (AMOS). It is part of the MASSIVE project (Mission-oriented autonomous systems with small satellites for maritime sensing, surveillance and communication), under the NTNU SmallSat lab. The purpose of HYPSO is to observe ocean color off the coast of Norway and to detect and monitor algae blooms using a hyperspectral imager (HSI).

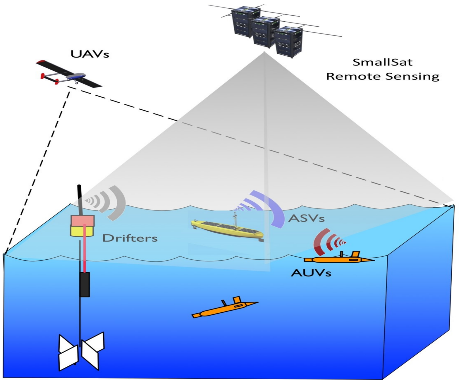


Figure A.1: SmallSats in the MASSIVE project. Credit: Mariusz Eivind Grøtte.

A.2 The satellite

The HYPSO satellite is a CubeSat of size 6U. HYPSO uses the M6P 6U nano-satellite bus from nanoAvionics [41]. Sun sensors, an inertial measurement unit (IMU), gyros, magnetometers, and a star tracker are included in the setup. The hyper spectral camera will be mounted along the z-axis of the satellite's body frame. HYPSO has two types of actuators: magnetorquers and reaction wheels. The magnetorquers are a set of electromagnetic coils that set up a magnetic dipole when current runs through them. As the Earth has a magnetic field, this dipole can be used to control the satellite's attitude. The reaction wheels are set of wheels that each spin about their own axis. The axes are fixed. The reaction wheels are included because of their high accuracy, torque, and momentum storage capability, which is required to satisfy the control objectives in section A.3.

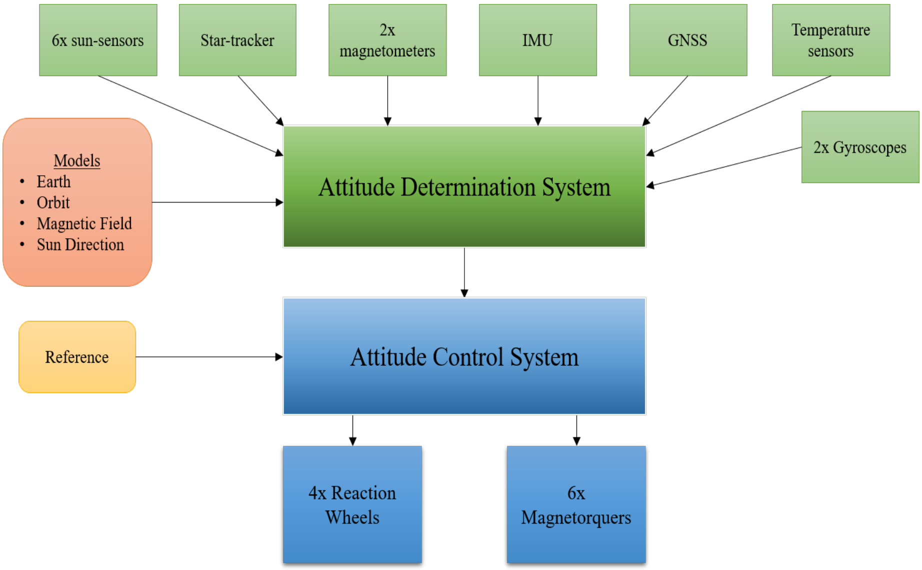


Figure A.2: ADCS, HYPSO. Credit: Mariusz Eivind Grøtte.

A.3 Control objectives

The control objectives that the attitude determination and control system (ADCS) has to perform in order to meet the mission requirements of the HYPSO mission can be summarized in four points.

- Pointing
  - Pointing refers to the objective of aiming the HSI at a desired target, for example nadir (straight down).
- Slewing maneuver
  - Slewing over an area at a specific angular velocity. The full maneuver consists of pointing towards the starting point, slew at the given angular velocity until the desired area has been covered, and then return to pointing

nadir. The motivation behind this maneuver is to produce overlapping pixels.

- Achieve high accuracy for pointing and slewing
  - High accuracy is crucial to fulfil the mission requirements. The RMSE requirement for target pointing accuracy is 0.01°. For this reason, the HYPSO satellite has reaction wheels. See section A.2.
- Detumbling
  - Slowing down the satellite's spin when the spin gets above a set threshold.

Appendix B

Stability Theory

The following paragraph is based on [23].
Stability of a system is defined in equilibrium points. A point is stable if there for every chosen region ϵ there exists a region $\delta(\epsilon)$ where the states may start and end up inside the ϵ region. The point is asymptotically stable if it is stable and converges to the equilibrium point as time goes to infinity, given that the states start within some domain. It is exponential stable if it is stable and converges to the equilibrium point at an exponential rate, given that the states start within some domain. Global asymptotic stability or global exponential stability is achieved if the domain is the whole of $\mathbb{R}$.

Finite-time stability implies convergence to the equilibrium point in finite time. Finite-time stability and exponential stability are both stronger properties than asymptotic stability, and either of global finite-time stability or global exponential stability implies global asymptotic stability.

Appendix C

LQR in Simulink

Implementing the LQR from a state-space representation of a system may be done easily in Matlab. The function *lqr(·)* gives the optimal gain used for the controller.

To produce a state-space representation, linearization is required for nonlinear systems. The motivation for implementing the LQR in a Matlab Function block in Simulink was that the ideal point of linearization would vary. The problem was that *lqr(·)* was not available for code generation at the time (spring 2019), so it could not be used directly.

From there other options were explored: *care(·)* solves the algebraic Riccati equation used to find $\mathbf{P}$ in $\mathbf{u} = -\mathbf{K}\mathbf{x}, \mathbf{K} = \mathbf{R}^{-1}\mathbf{B}\mathbf{P}$, but like *lqr(·)* it is not supported for code generation. Defining the Hamiltonian as

$$\mathbf{Z} = \begin{bmatrix} \mathbf{A} & -\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^{\intercal} \\ -\mathbf{Q} & -\mathbf{A}^{\intercal} \end{bmatrix} \tag{C.1}$$

the solution $\mathbf{P}$ can be found from the Schur decomposition of the $\mathbf{Z}$ matrix, as shown in [42]. There exists a function available for code generation that produces a Schur factorization in Matlab. However, for this method to work to solve the algebraic Riccati equation, the Schur decomposition needs to have leading negative eigenvalues.

Matlab's *schur(·)* cannot guarantee the order of the eigenvalues along the diagonal. To solve the problem this way in Matlab, it is required to use *schur(·)*, then *ordschur(·)* to have the desired matrices. *ordschur(·)* is not available for code generation. Attempts at implementing an alternative to the *ordschur(·)* function were abandoned as the alternative algorithms (such as the algorithm presented in [43]) were too complicated and time-demanding for the purpose of this thesis. Instead, the *lqr(·)* function were used offline, before the simulation in Simulink were started.

Bibliography

[1] L. Moreira, T. Fossen, and C. Guedes Soares, “Path following control system for a tanker ship model,” *Ocean Engineering*, vol. 34, pp. 2074–2085, Oct. 2007.

[2] S. K. Mudge and R. J. Patton, “Analysis of the technique of robust eigenstructure assignment with application to aircraft control,” *IEE Proceedings D - Control Theory and Applications*, vol. 135, no. 4, pp. 275–281, Jul. 1988.

[3] R. Kristiansen, P. J. Nicklasson, and J. T. Gravdahl, “Satellite attitude control by quaternion-based backstepping,” *IEEE Transactions on Control Systems Technology*, vol. 17, no. 1, pp. 227–232, Jan. 2009.

[4] I.-L. G. Borlaug, “Higher-Order Sliding Mode Control - Stability analysis and application to underwater snake robots,” 2017.

[5] Y. Shtessel, I. Shkolnikov, and A. Levant, “Smooth second-order sliding modes: Missile guidance application,” *Automatica*, vol. 43, pp. 1470–1476, Aug. 2007.

[6] H. Alwi and C. Edwards, “Fault detection and fault-tolerant control of a civil aircraft using a sliding-mode-based scheme,” *IEEE Transactions on Control Systems Technology*, vol. 16, no. 3, pp. 499–510, May 2008.

[7] L. Derafa, L. Fridman, A. Benallegue, and A. Ouldali, "Super twisting control algorithm for the four rotors helicopter attitude tracking problem," in *2010 11th International Workshop on Variable Structure Systems (VSS)*, Jun. 2010, pp. 62–67.

[8] A. SI MOHAMMED, B. SEBA, Y. Bentoutou, N. Taleb, A. Boudjemai, and A. Bellar, "Three axis attitude control using sliding mode based on the artificial neural network for low earth orbit microsatellite.," *INTERNATIONAL JOURNAL OF SYSTEMS APPLICATIONS, ENGINEERING & DEVELOPMENT*, vol. 6, pp. 223–233, Jan. 2012.

[9] Pong, Christopher Masaru and Miller, David W., "High-Precision Pointing and Attitude Estimation and Control Algorithms for Hardware-Constrained Spacecraft," 2014.

[10] P. Tsiotras, H. Shen, and C. Hall, "Satellite attitude control and power tracking with energy/momentum wheels," *Journal of Guidance, Control, and Dynamics*, vol. 24, pp. 23–34, Jan. 2001.

[11] R. Wisniewski, "Linear time-varying approach to satellite attitude control using only electromagnetic actuation," *Journal of Guidance, Control, and Dynamics*, vol. 23, pp. 640–647, Jul. 2000.

[12] S. Kowalchuk and C. Hall, "Spacecraft attitude sliding mode controller using reaction wheels," Aug. 2008.

[13] E. Abdulhamitbilal and E. M. Jafarov, "Performances comparison of linear and sliding mode attitude controllers for flexible spacecraft with reaction wheels," in *International Workshop on Variable Structure Systems, 2006. VSS'06.*, Jun. 2006, pp. 351–358.

[14] U. Jørgensen and J. T. Gravdahl, "Observer Based Sliding Mode Attitude Control: Theoretical and Experimental Results," *Modeling, Identification and Control*, vol. 32, no. 3, pp. 113–121, 2011.

[15] O. Egeland and J. .-.-. Godhavn, "Passivity-based adaptive attitude control of a rigid spacecraft," *IEEE Transactions on Automatic Control*, vol. 39, no. 4, pp. 842–846, Apr. 1994.

[16] J. Gravdahl, "Magnetic attitude control for satellites," vol. 1, Jan. 2005, 261–266 Vol.1.

[17] J. T. Gravdahl, E. Eide, A. Skavhaug, K. M. Fauske, K. Svartveit, and F. M. Indregaard, "Three axis attitude determination and control system for a picosatellite: Design and implementation," in *54th International Astronautical Congress of the International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law*. eprint: `https://arc.aiaa.org/doi/pdf/10.2514/6.IAC-03-A.5.07`.

[18] F. Alvenes, "Attitude Controller-Observer Design for the NTNU Test Satellite," 2013.

[19] G. Bråthen, "Design of Attitude Control System of a Double CubeSat," 2013.

[20] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Wiley, 2014.

[21] O. Egeland and T. Gravdahl, *Modeling and Simulation for Automatic Control*. Tapir Trykkeri, 2003.

[22] E. Oland and R. Schlanbusch, "Reaction wheel design for cubesats," Jun. 2009.

[23] H. K. Khalil, *Nonlinear Systems, Third edition*. Prentice hall, 2002.

[24] A. Isidori, *Nonlinear Control Systems, Third edition*. Springer Science & Business Media, 1995.

[25] H. D. Young and R. A. Freedman, *University Physics with Modern Physics. Scandinavian Edition, Volume One*. Dover Publications, Inc., 2016.

[26] O.-E. Fjellstad and T. I. Fossen, “Singularity-free tracking of unmanned underwater vehicles in 6 dof,” in *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, vol. 2, Dec. 1994, 1128–1133 vol.2.

[27] T. R. Krogstad, *Attitude synchronization in spacecraft formations, Theoretical and experimental results*. Department of Engineering Cybernetics, Norwegian University of Science and Technology, 2010.

[28] D. S. Bernstein, *Matrix Mathematics: Theory, Facts, and Formulas (Second Edition)*. Princeton University Press, 2009.

[29] P. C. Hughes, *Spacecraft Attitude Dynamics*. Dover Publications, Inc., 1993.

[30] Y. Shtessel, L. Fridman, C. Edwards, and A. Levant, *Sliding mode control and observation*. Springer New York, 2014.

[31] J. H. McDuffie and Y. B. Shtessel, “A de-coupled sliding mode controller and observer for satellite attitude control,” in *Proceedings The Twenty-Ninth Southeastern Symposium on System Theory*, Mar. 1997, pp. 92–97.

[32] I. Castillo, L. Fridman, and J. A. Moreno, “Super-twisting algorithm in presence of time and state dependent perturbations,” *International Journal of Control*, vol. 91, no. 11, pp. 2535–2548, 2018. eprint: `https://doi.org/10.1080/00207179.2016.1269952`.

[33] nanoAvionics, “M6P Attitude controller,”

[34] B. Wie, *Spacecraft attitude dynamics and control*. 1998.

[35] Ulrik Jørgensen, *Synchronization and Control of Attitude for Spacecrafts, Design, Analysis and Experiments*. Department of Engineering Cybernetics, Norwegian University of Science and Technology, 2010.

[36] R. Patera, "Attitude propagation for a slewing angular rate vector," *Journal of Guidance, Control, and Dynamics*, vol. 33, p. 1847, Nov. 2010.

[37] I.-L. G. Borlaug, K. Y. Pettersen, and J. T. Gravdahl, "Tracking control of an articulated intervention AUV in 6DOF using generalized super-twisting:, Theory and Experiments," *IEEE Transactions on Control Systems Technology*, 2019, (Submitted).

[38] L. Fridman, Y. Shtessel, C. Edwards, and X.-G. Yan, "Higher-order sliding-mode observer for state estimation and input reconstruction in nonlinear systems," *International Journal of Robust and Nonlinear Control*, vol. 18, pp. 399–412, Mar. 2008.

[39] National Oceanic and Atmospheric Administration. (). International Geomagnetic Reference Field, [Online]. Available: `https://www.ngdc.noaa.gov/IAGA/vmod/igrf.html` (visited on 05/31/2019).

[40] Jeremy Davis. (). Mathematical Modeling of Earth's Magnetic Field, Technical Note, [Online]. Available: `https://hanspeterschaub.info/Papers/UnderGradStudents/MagneticField.pdf` (visited on 05/31/2019).

[41] nanoAvionics. (2018). Innovative Ocean Research from NTNU to Ride on the NanoAvionics M6P Nano-Satellite Bus, [Online]. Available: `https://n-avionics.com/innovative-ocean-research-from-ntnu-to-ride-on-the-nanoavionics-m6p-nano-satellite-bus/` (visited on 05/31/2019).

[42] A. J. Laub, "A schur method for solving algebraic riccati equations," in *1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*, Jan. 1978, pp. 60–65.

[43] Z. Bai and J. W. Demmel, "On swapping diagonal blocks in real schur form," *Linear Algebra and its Applications*, vol. 186, pp. 75–95, Jun. 1993.

Bjørn Andreas Kristiansen

Nonlinear Attitude Control of a Pointing and Slewing Spacecraft

NTNU

Norwegian University of
Science and Technology