Master's thesis

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

Sigrid Kjønnø Schaanning

Maximum Hands-Off Control for Attitude Control of a Spacecraft

Master's thesis in Cybernetics and Robotics
Supervisor: Professor Jan Tommy Gravdahl
Co-supervisor: PhD Candidate Bjørn Andreas Kristiansen

June 2021

NTNU
Norwegian University of
Science and Technology

Sigrid Kjønnø Schaanning

Maximum Hands-Off Control for Attitude Control of a Spacecraft

Master's thesis in Cybernetics and Robotics
Supervisor: Professor Jan Tommy Gravdahl
Co-supervisor: PhD Candidate Bjørn Andreas Kristiansen
June 2021

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

NTNU
Norwegian University of
Science and Technology

Abstract

This thesis explores the use of maximum hands-off control for attitude control of a spacecraft actuated by reaction wheels. The maximum hands-off, or L_0-optimal, controller aims to find the sparsest control signal among all admissible control signals. However, L_0-optimal problems are generally hard to solve as L_0-cost functions are discontinuous and nonconvex. Previous research have investigated methods to approximate the L_0-norm in the cost function, for instance, using an L_1-norm. This thesis proposes an approach to the maximum hands-off control problem for spacecraft attitude control involving an L_0-cost function relaxed through complementarity constraints. Then, the controller is applied to the spacecraft attitude control problem, and the sparsity of the maximum hands-off controller is compared to that of the L_1-optimal controller. The maximum hands-off controller is shown to be sparser than the L_1-optimal controller for one of the problems investigated through simulations based on a 6U CubeSat. The simulations were conducted using CasADi as the primary optimization tool, and the L_1- and L_0-optimal control problems were discretized using direct multiple-shooting and solved using the IPOPT solver. In addition to these results, this thesis proposes a new paradigm of control, called *moving* maximum hands-off control, which lets the user specify in which time interval the control should occur and aims to find the sparsest control among all admissible controls based on this information. The moving maximum hands-off controller is shown to be as sparse as the maximum hands-off controller for some spacecraft maneuvers.

Sammendrag

Denne avhandlingen utforsker bruken av "maximum hands-off" regulering for attituderegulering av en satellitt som er drevet av reaksjonshjul. Maximum hands-off, eller den L_0-optimale, regulatoren forsøker å finne pådragssignalet som er mest "sparse" blant alle mulige pådragssignal. L_0-optimale problemer er generelt vanskelige å løse siden en kostfunksjon basert på L_0-normen hverken er kontinuerlig eller konveks. Tidligere forskning har undersøkt metoder for å tilnærme L_0-normen, for eksempel ved bruk av en L_1-norm. Denne oppgaven foreslår å bruke en L_0-kostfunksjon med komplementære begrensninger som en tilnærming til maximum hands-off problemet for attituderegulering av en satellitt. Deretter brukes maximum hands-off regulatoren for å løse attitudereguleringproblemet for en satellitt, og pådragssignalet fra maximum hands-off regulatoren sammenlignes med pådragssignalet fra den L_1-optimale regulatoren. Denne oppgaven viser at maximum hands-off regulatoren er mer sparse enn den L_1-optimale regulatoren for et av problemene som ble undersøkt gjennom simuleringer basert på en 6U CubeSat. Simuleringene ble gjennomført ved bruk av optimeringsverktøyet CasADi, og L_1- og L_0-optimeringsproblemene ble diskretisert ved bruk av "direct multiple-shooting" og løst ved bruk av IPOPT. I tillegg til disse resultatene så foreslår denne oppgaven en ny type regulator, kalt "moving maximum hands-off" regulator, som lar brukeren spesifisere i hvilke tidsintervaller pådragene skal settes på og prøver å finne pådragssignalet som er mest sparse blant alle gyldige pådragssignaler, basert på denne informasjonen. Det viste seg at moving maximum hands-off regulatoren er like sparse som maximum hands-off regulatoren for noen satellittmanøvere.

Preface

This thesis is written and submitted as partial fulfillment of the Master of Science degree in Cybernetics and Robotics at the Norwegian University of Science and Technology (NTNU).

I would like to thank my supervisor Professor Jan Tommy Gravdahl for his excellent advice and for supervising my Master's thesis. A huge thanks also go to my co-supervisor, PhD Candidate Bjørn Andreas Kristiansen, for good discussions and guidance during the work on this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and Background

The work presented in this thesis is related to the HYPSO mission at the NTNU Small Satellite Lab. The HYPer-Spectral smallsat for Ocean observation (HYPSO) seeks to investigate the ocean, as it is of great interest in understanding the effects of climate changes (HYPSO, 2021). Studying the ocean could also increase knowledge of how humans impact the world. More specifically, HYPSO aims to track sporadic algal blooms in the oceans and to provide hyperspectral images to support monitoring of marine environments (Grøtte et al., 2021).

HYPSO is equipped with a hyperspectral camera and advanced on-board processing. A crucial part of the HYPSO mission is to perform a slew maneuver during hyperspectral imaging across a desired target. The subsystem responsible for aiming the satellite, and thereby the camera, towards the target is the Attitude Determination and Control System (ADCS). This thesis aims to solve the spacecraft attitude control problem using a maximum hands-off controller. The controller will be implemented and tested on a configuration similar to that of HYPSO.

The motivation for using maximum hands-off control for the attitude control of a spacecraft is to provide a control signal which is as sparse as possible, i.e., the control signal is most often zero. Even though the maximum hands-off controller yields a control signal that is as sparse as possible, the control input may occur at any time instants within the optimization interval. This motivates the design of the moving maximum hands-off controller, which lets the user specify in which time intervals the control input should occur.

1.2 Contributions of this Thesis

The contributions of this thesis are in the field of optimal attitude control, and more precisely, optimal attitude control with application to spacecraft. The main contribution of this thesis is the use of maximum hands-off control to solve the spacecraft attitude control problem for a spacecraft actuated by reaction wheels. The maximum hands-off controller was tested and simulated on a configuration similar to a 6U CubeSat. Another contribution is the presentation of an example, which demonstrates that using the L_1-norm to approximate the L_0-norm does not always yield the L_0-optimal solution. The moving maximum hands-off controller is designed and implemented as an extension to the maximum hands-off controller. The moving maximum hands-off controller lets the user specify in which time interval the control inputs should occur and is, to the author's best knowledge, a novel concept within the domain of control theory. Finally, this thesis provides a comparison of the responses of the L_1-optimal controller, the maximum hands-off controller, and the moving maximum hands-off controller when solving the spacecraft attitude control problem.

1.3 Research Objectives

The objectives that motivate the research in this thesis are as follows:

i To explore the use of maximum hands-off control for the spacecraft attitude control problem.

ii To develop, design and explore the use of moving maximum hands-off control for the spacecraft attitude control problem.

iii To compare the responses of the L_1-optimal controller, the maximum hands-off controller, and the moving maximum hands-off controller when the controllers are used to solve the spacecraft attitude control problem.

1.4 Outline of the Report

Chapter 1 has presented the motivation and background for this project. The contributions of this thesis have been presented as well as the research objectives.

Chapter 2 presents fundamental theory to derive and understand the equations of motion for a spacecraft. This theory provides useful tools to build a mathematical model for the dynamics of the attitude of a spacecraft. The fields of attitude control and optimal control are also explored, and brief summaries of some of the previous works in these fields are presented.

Chapter 3 introduces the maximum hands-off controller. The mathematical preliminaries required to understand the controller are presented, followed up by the mathematical formulation of the controller. A few examples of the controller, and its expected behavior, are also provided in Chapter 3.

Chapter 4 derives the spacecraft model and the attitude dynamics based on the theory provided in the previous chapters. After completing this chapter, the reader will understand the dynamic equations governing the attitude of a spacecraft.

Chapter 5 presents the design of the maximum hands-off controller, the moving maximum hands-off controller, the L_1-optimal controller, and a PD-controller based on quaternions. The experimental cases are also presented, as well as the simulation setup and software tools used for the experiments.

Chapter 6 presents the results obtained during the experiments and discusses the findings. The results from the simulations of the different controllers and maneuvers are presented and discussed.

Chapter 7 presents the conclusions based on the work in this project. Suggestions for future work are presented based on the experiences obtained during the work with this thesis.

Appendix A presents the plots of the angular velocity of the spacecraft's reaction wheels for the different experiments.

Appendix B includes a copy of a draft paper, which will be submitted for the 2022 American Control Conference (ACC).

Chapter 2

Theory

This chapter provides an overview of the mathematical notation necessary to follow the arguments throughout the thesis. The chapter also gives an introduction to the fields of attitude control and optimal control and an overview of some of the spacecraft-related research within these fields.

2.1 Notation

In this section, mathematical notation related to vectors are presented. Section 2.1.1 gives a general introduction to vector notation, Section 2.1.2 introduces the vector cross product, and Section 2.1.3 presents how to find the time derivative of vectors.

2.1.1 Vectors

Vector notation is applied to describe forces, torques, velocities, and accelerations. A vector $\vec{u}$ can be described by its magnitude $|\vec{u}|$ and its direction. This description of a vector may be said to be coordinate-free, as it does not rely on the definition of any coordinate frame (Egeland and Gravdahl, 2003). Coordinate frames are explored further in Section 2.3.

The vector $\vec{u}$ may also be described in terms of its components in a given coordinate frame, say $\{a\}$. Let the frame $\{a\}$ be defined by three orthogonal unit vectors $\vec{a}_1$, $\vec{a}_2$, $\vec{a}_3$ along the x-, y- and z-axis of $\{a\}$. Then it is possible to express the vector as a linear combination of the orthogonal unit vectors as follows (Egeland and Gravdahl, 2003)

$$\vec{u} = u_1\vec{a}_1 + u_2\vec{a}_2 + u_3\vec{a}_3, \tag{2.1}$$

where the unique components or coordinates of $\vec{u}$ in $\{a\}$ are given as

$$u_i = \vec{u} \cdot \vec{a}_i,\ i \in \{1, 2, 3\}. \tag{2.2}$$

The vector $\vec{u}$ can be expressed in $\{a\}$ by a column vector consisting of the vector's coordinates in the frame. This column vector is also called the coordinate vector form and it is written as (Egeland and Gravdahl, 2003)

$$\mathbf{u}^a = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}. \tag{2.3}$$

2.1.2 Vector cross product

The vector cross product between two coordinate-free vectors $\vec{u}$ and $\vec{v}$ is given as (Egeland and Gravdahl, 2003)

$$\vec{u} \times \vec{v} = \vec{n}|\vec{u}||\vec{v}|\text{sin}(\theta), \tag{2.4}$$

where $\theta \in [0, \pi]$ and $\vec{n}$ is a unit vector orthogonal to $\vec{u}$ and $\vec{v}$, defined such that ($\vec{u}$, $\vec{v}$, $\vec{n}$) forms a right-hand coordinate system.

The skew-symmetric matrix, denoted $\mathbf{S}(\cdot)$, represents the cross product operator and is used to compute the cross product of two coordinate vectors as matrix multiplication as follows (Egeland and Gravdahl, 2003; Fossen, 2021)

$$\mathbf{u} \times \mathbf{v} = \mathbf{S}(\mathbf{u})\mathbf{v}, \tag{2.5}$$

where $\mathbf{S}(\mathbf{u})$ is defined as

$$\mathbf{S}(\mathbf{u}) \triangleq \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}. \tag{2.6}$$

In general, a matrix is said to be skew-symmetric if it satisfies the following

$$\mathbf{S}(\mathbf{u})\mathbf{v} = -\mathbf{S}(\mathbf{v})\mathbf{u}. \tag{2.7}$$

2.1.3 Time derivative of vectors

Differentiation of coordinate vectors

Differentiation of a coordinate vector with respect to time is performed by differentiating the components of the vector with respect to time, as follows (Egeland and Gravdahl, 2003)

$$\dot{\mathbf{u}}^a \triangleq \frac{d}{dt}(\mathbf{u}^a) = \frac{d}{dt}\begin{bmatrix} u_1^a \\ u_2^a \\ u_3^a \end{bmatrix} = \begin{bmatrix} \dot{u}_1^a \\ \dot{u}_2^a \\ \dot{u}_3^a \end{bmatrix}. \tag{2.8}$$

The coordinate vector in frame $\{a\}$ and the coordinate vector in frame $\{b\}$ are related by the equation

$$\mathbf{u}^a = \mathbf{R}^a_b \mathbf{u}^b, \tag{2.9}$$

where $\mathbf{R}^a_b$ is the rotation matrix from frame $\{b\}$ to frame $\{a\}$. Rotation matrices are explored further in Section 2.2.1.

Differentiation of (2.9) yields the following relation between the time derivative in frame $\{a\}$ and the time derivative in frame $\{b\}$ (Egeland and Gravdahl, 2003)

$$\dot{\mathbf{u}}^a = \mathbf{R}^a_b \dot{\mathbf{u}}^b + \dot{\mathbf{R}}^a_b \mathbf{u}^b = \mathbf{R}^a_b [\dot{\mathbf{u}}^b + \mathbf{S}(\boldsymbol{\omega}^b_{ab})\mathbf{u}^b], \tag{2.10}$$

where the property $\dot{\mathbf{R}}^a_b = \mathbf{R}^a_b \mathbf{S}(\boldsymbol{\omega}^b_{ab})$ has been exploited.

Differentiation of vectors

The process of differentiating a vector $\vec{u}$ with respect to time has to be carried out with reference to some reference frame. Differentiating $\vec{u}$ with respect to time in the $\{a\}$-frame yields (Egeland and Gravdahl, 2003)

$$\frac{{}^a d}{dt}\vec{u} \triangleq \dot{u}^a_1 \vec{a}_1 + \dot{u}^a_2 \vec{a}_2 + \dot{u}^a_3 \vec{a}_3, \tag{2.11}$$

where it is assumed that $\vec{u} = u^a_1 \vec{a}_1 + u^a_2 \vec{a}_2 + u^a_3 \vec{a}_3$. The column vector representation of (2.11) is

$$\dot{\mathbf{u}}^a = \begin{bmatrix} \dot{u}^a_1 \\ \dot{u}^a_2 \\ \dot{u}^a_3 \end{bmatrix}. \tag{2.12}$$

The time derivative of $\vec{u}$ with reference to $\{a\}$ may also be found accordingly

$$\frac{{}^a d}{dt}\vec{u} = \frac{{}^b d}{dt}\vec{u} + \vec{\omega}_{ab} \times \vec{u}, \tag{2.13}$$

where $\frac{{}^b d}{dt}\vec{u}$ is the time derivative of $\vec{u}$ with reference to $\{b\}$, and $\vec{\omega}_{ab}$ is the angular velocity of $\{b\}$ relative to $\{a\}$.

2.2 Attitude representation

The attitude of a rigid body may be represented using different parametrizations. In this section, four different parametrizations are presented; the rotation matrix, the Euler angles, the angle-axis, and the unit quaternions.

2.2.1 Rotation matrix

When deriving the equations of motion for a spacecraft, it is often convenient to represent a vector in more than one coordinate frame and to transform vectors between various frames. A vector represented in frame $\{a\}$ may be transformed, or rotated, to a vector in frame $\{b\}$ using a rotation matrix, denoted $\mathbf{R}_a^b$. The rotation matrix $\mathbf{R}_a^b$ has two interpretations; the first is to transform a vector between two reference frames, where $\mathbf{R}_a^b$ acts as a coordinate transformation matrix, and the second is to rotate a vector within a given reference frame, where $\mathbf{R}_a^b$ acts as a rotation matrix. The coordinate transformation of a vector from frame $\{a\}$ to a vector in frame $\{b\}$ is given as (Egeland and Gravdahl, 2003)

$$\mathbf{v}^b = \mathbf{R}_a^b \mathbf{v}^a, \tag{2.14}$$

where the rotation matrix is

$$\mathbf{R}_a^b = \{\vec{b}_i \cdot \vec{a}_j\}, \tag{2.15}$$

and the elements $r_{ij} = \{\vec{b}_i \cdot \vec{a}_j\}$ of $\mathbf{R}_a^b$ are called the direction cosines. The rotation matrix is sometimes also referred to as the orientation matrix, the attitude matrix, or the direction cosine matrix (Wen and Kreutz-Delgado, 1991).

Definition of the rotation matrix

The rotation matrix is an element in the special orthogonal group of order three, i.e., $SO(3)$, defined as (Egeland and Gravdahl, 2003)

$$SO(3) \triangleq \{\mathbf{R} | \mathbf{R} \in \mathbb{R}^{3\times 3},\ \mathbf{R} \text{ is orthogonal}, \quad \det(\mathbf{R}) = 1\}, \tag{2.16}$$

and the orthogonality of a matrix is defined as

$$\mathbf{R}\mathbf{R}^\top = \mathbf{R}^\top \mathbf{R} = \mathbf{I}_{3\times 3}, \tag{2.17}$$

where $\mathbf{I}_{3\times 3}$ is the identity matrix.

Properties of the rotation matrix

Egeland and Gravdahl (2003) presents some useful properties of the rotation matrix. For all $\mathbf{v}^b$ the following holds

$$\mathbf{v}^b = \mathbf{R}_a^b \mathbf{v}^a = \mathbf{R}_a^b \mathbf{R}_b^a \mathbf{v}^b, \tag{2.18}$$

which implies

$$\mathbf{R}_a^b \mathbf{R}_b^a = \mathbf{I}_{3\times 3}, \tag{2.19}$$

and from (2.19) it follows that

$$\mathbf{R}_a^b = (\mathbf{R}_b^a)^{-1}. \tag{2.20}$$

The rotation matrix also satisfies the following

$$\mathbf{R}_a^b = (\mathbf{R}_b^a)^{-1} = (\mathbf{R}_b^a)^\top. \tag{2.21}$$

2.2.2 Euler angles

The 3×3-rotation matrix describes the orientation of a frame $\{b\}$ with respect to a frame $\{a\}$ using nine elements. The rotation matrix is orthogonal, and the orthogonality yields six constraints on the elements of the matrix. From the six constraints, it follows that there are only three independent parameters describing the rotation matrix (Egeland and Gravdahl, 2003). Thus, it is of interest to find a three-parameter representation, i.e., a minimal representation, of the rotation matrix. The Euler angles are often used for this purpose.

The Euler angles consist of three angles, and each angle describes rotation about one of the three principal axes. The description of the rotation matrix using Euler angles, is given as composite rotations about the x, y, and z axes. Several variations of the Euler angle parametrization exist, and two of the most common include the roll-pitch-yaw angles (ZYX) and the classical Euler angles (ZYZ) (Egeland and Gravdahl, 2003; Sciavicco and Siciliano, 2012). The roll-pitch-yaw angles are often used to describe the motion of free moving objects, for instance, spacecraft and satellites, whereas the classical Euler angles are used to describe the rotation of rigid bodies connected to a fixed base, for instance, robotic wrist joints (Egeland and Gravdahl, 2003).

Roll-Pitch-Yaw angles

The rotation from $\{a\}$ to $\{b\}$ can be described by roll-pitch-yaw angles as a rotation ψ about the z_a-axis (yaw), followed by a rotation θ about the rotated y_a-axis (pitch), and then a rotation ϕ about the rotated x_a-axis (roll). The resulting rotation matrix is given as (Egeland and Gravdahl, 2003)

$$\mathbf{R}_a^b = \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi), \tag{2.22}$$

where the rotation matrices are given as (Egeland and Gravdahl, 2003; Fossen, 2021)

$$\mathbf{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}, \tag{2.23a}$$

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, \tag{2.23b}$$

$$\mathbf{R}_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.23c}$$

Kinematic differential equations using Euler angles

It is not possible to integrate the body-fixed angular velocity of an object directly to obtain the Euler angles (Fossen, 2021). Instead, the kinematic relation between the object's angular velocities and the rate of change in the Euler angles are exploited to obtain the Euler angles.

The rotation matrix from $\{d\}$ to $\{a\}$, $\mathbf{R}^a_d$, can be expressed in the roll-pitch-yaw case as (Egeland and Gravdahl, 2003)

$$\mathbf{R}^a_d = \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi), \tag{2.24}$$

where the rotation matrices are given as

$$\mathbf{R}^a_b = \mathbf{R}_z(\psi), \tag{2.25a}$$

$$\mathbf{R}^b_c = \mathbf{R}_y(\theta), \tag{2.25b}$$

$$\mathbf{R}^c_d = \mathbf{R}_x(\phi). \tag{2.25c}$$

The angular velocities associated with the rotations in (2.25) are

$$\boldsymbol{\omega}^a_{ab} = \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}, \tag{2.26a}$$

$$\boldsymbol{\omega}^b_{bc} = \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix}, \tag{2.26b}$$

$$\boldsymbol{\omega}^c_{cd} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}, \tag{2.26c}$$

where ω^a_{ab} denotes the angular velocity of $\{b\}$ with respect to $\{a\}$, expressed in $\{a\}$. The angular velocity of $\{d\}$ with respect to $\{a\}$, expressed in $\{a\}$, can be expressed as a sum of the angular velocities in (2.26)

$$\boldsymbol{\omega}^a_{ad} = \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \mathbf{R}_{z,\psi} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \mathbf{R}_{z,\psi}\mathbf{R}_{y,\theta} \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \triangleq \mathbf{T}_a^{-1}(\boldsymbol{\Theta})\dot{\boldsymbol{\Theta}}, \tag{2.27}$$

where $\boldsymbol{\Theta} = [\phi, \theta, \psi]$ is the vector of Euler angles, $\dot{\boldsymbol{\Theta}} = [\dot{\phi}, \dot{\theta}, \dot{\psi}]$ is the time derivative of the Euler angles, and $\mathbf{T}_a^{-1}(\boldsymbol{\Theta})$ denotes the inverse of the transformation matrix $\mathbf{T}_a(\boldsymbol{\Theta})$. Solving (2.27) for $\dot{\boldsymbol{\Theta}}$ gives the kinematic differential equation

$$\dot{\boldsymbol{\Theta}} = \mathbf{T}_a(\boldsymbol{\Theta})\boldsymbol{\omega}^a_{ad}. \tag{2.28}$$

Expanding (2.27) yields

$$\mathbf{T}_a^{-1}(\boldsymbol{\Theta}) = \begin{bmatrix} c(\psi)c(\theta) & -s(\psi) & 0 \\ s(\psi)c(\theta) & c(\psi) & 0 \\ -s(\theta) & 0 & 1 \end{bmatrix}, \tag{2.29a}$$

$$\mathbf{T}_a(\boldsymbol{\Theta}) = \frac{1}{c(\theta)} \begin{bmatrix} c(\psi) & s(\psi) & 0 \\ -s(\psi)c(\theta) & c(\psi)c(\theta) & 0 \\ c(\psi)s(\theta) & s(\psi)s(\theta) & c(\theta) \end{bmatrix}, \tag{2.29b}$$

where $s(\cdot)$ and $c(\cdot)$ denotes $\sin(\cdot)$ and $\cos(\cdot)$, respectively. From (2.29), it can be seen that $\mathbf{T}_a(\boldsymbol{\Theta})$ is undefined and becomes singular, for $\theta = \pm\frac{\pi}{2}$. This singularity is called the *Euler angle singularity*, and it could be a challenge when representing attitude using Euler angles. For any sequence of Euler angles, i.e., ZYX or ZYZ, the singularity occurs in the middle angle. To avoid the singularity, at least four parameters have to be used to represent the attitude (Fossen, 2021).

2.2.3 Angle-axis

Rotation may also be represented using the angle-axis parametrization. It can be useful to apply the angle-axis parametrization when developing, for instance, kinematic models for use in control systems (Egeland and Gravdahl, 2003). Using angle-axis parametrization, it is possible to describe the rotation from $\{b\}$ to $\{a\}$ as an angle θ about a unit vector $\vec{k}$ fixed in both $\{a\}$ and $\{b\}$. The angle-axis parametrization of the rotation matrix, $\mathbf{R}^a_b$, is given as (Egeland and Gravdahl, 2003)

$$\mathbf{R}^a_b = \cos(\theta)\mathbf{I} + \mathbf{S}(\mathbf{k}^a)\sin(\theta) + (1 - \cos(\theta))\mathbf{k}^a(\mathbf{k}^a)^\top, \tag{2.30}$$

where $\mathbf{k}^a$ is the coordinate vector of $\vec{k}$ in frame $\{a\}$.

2.2.4 Quaternions

The unit quaternions, also known as the Euler parameters, use four parameters to represent attitude. The use of four parameters ensures that the representation is nonsingular for all angles, as opposed to the Euler angles (Fossen, 2021; Egeland and Gravdahl, 2003). It is convenient to use quaternions to describe the attitude of a spacecraft, as opposed to Euler angles, since the spacecraft moves freely in space and would be affected by the Euler angle singularity. Moreover, the quaternions are useful in numerical simulations of rotation, as they are more computationally efficient than the Euler angles.

The quaternion, $\mathbf{q}$, can be written as (Egeland and Gravdahl, 2003; Fossen, 2021; Sola, 2017; Chou, 1992)

$$\mathbf{q} = \begin{bmatrix} \eta \\ \boldsymbol{\epsilon} \end{bmatrix} = \begin{bmatrix} \eta \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix}, \tag{2.31}$$

where η and $\boldsymbol{\epsilon}$ are given in terms of the angle-axis parameters $\mathbf{k}$ and θ, which are discussed in Section 2.2.3, as follows

$$\eta = \cos(\frac{\theta}{2}), \tag{2.32a}$$

$$\boldsymbol{\epsilon} = \mathbf{k}\,\sin(\frac{\theta}{2}). \tag{2.32b}$$

The unit quaternion has several properties. One of the properties is that the unit quaternion satisfies the following condition (Egeland and Gravdahl, 2003; Fossen, 2021)

$$\mathbf{q}^\top \mathbf{q} = 1, \tag{2.33}$$

which may be expanded to

$$\eta^2 + \boldsymbol{\epsilon}^\top \boldsymbol{\epsilon} = \eta^2 + \epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 = 1. \tag{2.34}$$

Another property of the unit quaternion, is the quaternion product between two quaternions, and it is given as (Egeland and Gravdahl, 2003; Fossen, 2021; Sola, 2017; Chou, 1992)

$$\mathbf{q}_1 \otimes \mathbf{q}_2 = \begin{bmatrix} \eta_1\eta_2 - \boldsymbol{\epsilon}_1^\top \boldsymbol{\epsilon}_2 \\ \eta_1\boldsymbol{\epsilon}_2 + \eta_2\boldsymbol{\epsilon}_1 + \mathbf{S}(\boldsymbol{\epsilon}_1)\boldsymbol{\epsilon}_2 \end{bmatrix} = \mathbf{F}(\mathbf{q}_1)\mathbf{q}_2, \tag{2.35}$$

where $\mathbf{F}(\mathbf{q}_1)$ is a matrix defined by

$$\mathbf{F}(\mathbf{q}_1) = \begin{bmatrix} \eta_1 & -\boldsymbol{\epsilon}_1 \\ \boldsymbol{\epsilon}_1 & \eta_1 \mathbf{I}_{3\times 3} + \mathbf{S}(\boldsymbol{\epsilon}_1) \end{bmatrix}. \tag{2.36}$$

Additionally, the inverse quaternion, corresponding to the unit quaternion $\mathbf{q}$ in (2.31), is given as (Egeland and Gravdahl, 2003; Fossen, 2021; Sola, 2017; Chou, 1992)

$$\mathbf{q}^{-1} = \begin{bmatrix} \eta \\ -\boldsymbol{\epsilon} \end{bmatrix}. \tag{2.37}$$

A drawback with the unit quaternions is that they do not represent attitude uniquely since each attitude corresponds to two different quaternion vectors (Chaturvedi et al., 2011). To put it more precisely, a physical attitude $\mathbf{R} \in SO(3)$ is represented by a pair of quaternions $\pm\mathbf{q} \in \mathbb{S}^3$, where $SO(3)$ is the special orthogonal group discussed in Section 2.2.1 and $\mathbb{S}^3$ is the non-Euclidean three-sphere (Chaturvedi et al., 2011).

2.3 Coordinate frames

The description of the position and attitude of a satellite has to be made in relation to some reference frame. A reference frame, or a coordinate frame, is a choice of coordinate system given as $\{r\} = \{O_r, \mathbf{x}_r, \mathbf{y}_r, \mathbf{z}_r\}$, where O_r is the origin and $\mathbf{x}_r, \mathbf{y}_r, \mathbf{z}_r$ are the orthonormal unit vectors. In this thesis, four different coordinate frames are used to describe the attitude of the spacecraft.

2.3.1 Earth-centered inertial (ECI) frame

The Earth-centered inertial frame, denoted $\{i\}$, is considered to be an inertial frame where Newton's laws are valid. The origin of $\{i\}$ is located at the Earth's center of mass, and the z-axis points through the North Pole, the x-axis points towards the vernal equinox, and the y-axis completes the right-hand system (Egeland and Gravdahl, 2003; Fossen, 2021).

2.3.2 Body frame

The body frame, denoted $\{b\}$, is a moving coordinate frame fixed to the spacecraft, with origin at the spacecraft's center of mass (Egeland and Gravdahl, 2003; Fossen, 2021). The body frame axes follow the spacecraft structure.

2.3.3 Orbit frame

The Vehicle Velocity, Local Horizontal (VVLH) frame, or the orbit frame, is denoted $\{o\}$ and has its origin at the spacecraft's center of mass. The z-axis points in the direction of the Earth's center of mass, the x-axis points in the direction of the orbital velocity vector, while the y-axis completes the right-handed coordinate system (Grøtte et al., 2020; Kristiansen et al., 2020). The unit vectors of the orbit frame are defined as (Kristiansen et al., 2020)

$$\hat{\mathbf{z}}^o = -\frac{\mathbf{r}^i}{\|\mathbf{r}^i\|_2},\ \hat{\mathbf{x}}^o = \frac{\mathbf{v}^i}{\|\mathbf{v}^i\|_2},\ \hat{\mathbf{y}}^o = \frac{\hat{\mathbf{z}}^o \times \hat{\mathbf{x}}^o}{\|\hat{\mathbf{z}}^o \times \hat{\mathbf{x}}^o\|_2}, \tag{2.38}$$

where $\mathbf{r}^i$ and $\mathbf{v}^i$ are the distance between the spacecraft and the center of the Earth, and the inertial velocity of the spacecraft, in the ECI frame, respectively.

2.3.4 Wheel frame

The wheel frame is used to specify vectors directly related to the reaction wheels, and it is denoted by $\{w\}$ (Kristiansen et al., 2020). There are two vectors represented in the wheel frame, namely the vector of torques applied to the wheels, $\boldsymbol{\tau}_u^w$, and the angular velocity vector of the wheels, $\boldsymbol{\omega}_{bw}^w$. The dimensions of $\boldsymbol{\tau}_u^w$ and $\boldsymbol{\omega}_{bw}^w$ equals the number of the reaction wheels on the satellite. Each channel of the vectors gives the torque applied or angular velocity about each wheel's axis of rotation.

2.3.5 Coordinate transformations

Orientation is described with respect to different frames. It is necessary to define the transformations between the various frames to be able to convert between them. The transformations are given in terms of rotation matrices, and these rotation matrices can be described using, for instance, Euler angles or quaternions. For a thorough explanation of the Euler angles, quaternions, and other attitude parametrizations, see Section 2.2.

Transformation from $\{i\}$ to $\{o\}$

The transformation from the inertial frame, $\{i\}$, to the orbit frame, $\{o\}$, denoted by $\mathbf{R}_i^o \in \mathbb{R}^{3\times3}$ is given by (Grøtte et al., 2020; Kristiansen et al., 2020)

$$\mathbf{R}_i^o = \left[\hat{\mathbf{x}}^o\ \hat{\mathbf{y}}^o\ \hat{\mathbf{z}}^o\right]^\top, \tag{2.39}$$

where $\hat{\mathbf{x}}^o, \hat{\mathbf{y}}^o$and $\hat{\mathbf{z}}^o$ is defined in (2.38).

Transformation from $\{o\}$ to $\{b\}$

The transformation from the orbit frame, $\{o\}$, to the body frame, $\{b\}$, is given by the rotation matrix, $\mathbf{R}_o^b \in \mathbb{R}^{3\times3}$, in terms of quaternions as (Egeland and Gravdahl, 2003)

$$\begin{aligned}\mathbf{R}_b^o &= \mathbf{R}(\mathbf{q}_b^o) = \mathbf{I}_{3\times 3} + 2\eta_b^o\mathbf{S}(\boldsymbol{\epsilon}_b^o) + 2\mathbf{S}^2(\boldsymbol{\epsilon}_b^o), \\ \mathbf{R}_o^b &= (\mathbf{R}_b^o)^\top,\end{aligned} \tag{2.40}$$

where $\mathbf{S}(\cdot)$ is the skew-symmetric matrix defined in (2.5).

Transformation from $\{w\}$ to $\{b\}$

The matrix $\mathbf{A} \in \mathbb{R}^{3\times n}$ describes the rotation from the wheel frame, $\{w\}$, to the body frame, $\{b\}$ (Grøtte et al., 2020; Kristiansen et al., 2020). The column vectors $\mathbf{a}_j \in \mathbb{R}^3$ of $\mathbf{A}$, are the spinning axes of the j reaction wheels, for $j = \{1, 2, ..., n\}$ such that

$$\mathbf{A} = \begin{bmatrix}\mathbf{a}_1 \; \mathbf{a}_2 \; \cdots \; \mathbf{a}_n\end{bmatrix}. \tag{2.41}$$

The matrix $\mathbf{A} \in \mathbb{R}^{3\times n}$ maps the wheel frame to the body frame as follows

$$\boldsymbol{\tau}_u^b = \mathbf{A}\boldsymbol{\tau}_u^w \implies \boldsymbol{\tau}_u^w = \mathbf{A}^+\boldsymbol{\tau}_u^b, \tag{2.42}$$

where the matrix $\mathbf{A}^+$ is the Moore-Penrose pseudo-inverse of $\mathbf{A}$. Due to a fixed reaction wheel configuration, $\mathbf{A}$ represents a constant mapping between $\{w\}$ and $\{b\}$.

2.4 Attitude control

Attitude control of rigid bodies has important applications in a wide range of fields; from maneuvering a helicopter or a satellite to controlling a robot arm (Wen and Kreutz-Delgado, 1991). Attitude control of rigid bodies is a field in which multiple studies have been conducted, and there exist numerous results on the field. Wen and Kreutz-Delgado (1991) and Chaturvedi et al. (2011) gives a thorough introduction to the attitude control problem and rigid-body attitude control.

In this thesis, the results on attitude control of spacecraft are of particular interest. The attitude of a spacecraft is its orientation in space. Thus, the attitude control of a spacecraft refers to the procedure of controlling its orientation relative to a coordinate frame, for instance, the inertial frame. Attitude control may be divided into two parts: attitude stabilization, which deals with maintaining the current position, and attitude maneuver control, which deals with steering the spacecraft from one orientation to another (Wertz, 2012). Attitude control is required for several operations in a spacecraft, for instance, when it comes to orient spacecraft used for orbit maneuvers or to prevent solar damage to spacecraft components (Wertz, 2012). The work by Meyer (1971), Chobotov (1991), Hughes (2004), and Wertz (2012) are excellent references for the attitude control of spacecraft.

Attitude control is normally studied using different attitude parametrizations (Chaturvedi et al., 2011), and some of these representations are represented in Section 2.2. A challenge related to attitude control is that no parametrization represents attitudes both globally and uniquely (Chaturvedi et al., 2011) For instance, the Euler angles are subject to singularities at certain angles, and thus the derivatives become undefined. It is, therefore, necessary to avoid the singularities when working with Euler angles or any other 3-parameter representation, a consideration that complicates path planning and could constrain admissible attitudes (Wen and Kreutz-Delgado, 1991). Unit quaternions, which represent attitude using four parameters, avoid the previously discussed singularities and are therefore able to represent attitude globally, although they fail to represent attitude uniquely. See Section 2.2.4 for more on quaternions.

Euler angles and unit quaternions are favored in different applications. For surface vessels, the Euler angle singularity is not a problem (Fossen, 2021), as the surface prevents them from rotating to the point where the singularity occurs. When it comes to the dynamics of a flight, the Euler angle representation of attitude is often sufficient, as the singularity can be avoided. For the singularity at $\theta = \pm\frac{\pi}{2}$, where θ is the second angle in the rotation order, to be a problem for the flight attitude, the flight has to point either straight up or straight down, which is seldom a problem under regular flight conditions (Beard and McLain, 2012). On the other hand, the singularity is an issue for acrobatic flight maneuvers and other extreme operations (Beard and McLain, 2012). Underwater vehicles and spacecraft may operate close to the singularity, as they can move freely and are not constrained by a surface (Fossen, 2021). In this case, quaternions could be used as an alternative to the Euler angles to avoid the singularity. The Euler angle singularity is discussed in Section 2.2.2.

Several solutions have been suggested to solve the spacecraft attitude control problem. Some of the most regularly employed attitude control strategies include proportional-derivative (PD) control laws (Wen and Kreutz-Delgado, 1991; Show et al., 2002; Ismail and Varatharajoo, 2010). A PD-controller was also implemented in this thesis, see Section 5.2.4.

Other approaches taken to solve the spacecraft attitude control problem are techniques based on quaternion feedback. A nonlinear control law, which uses feedback from the unit quaternion and measured angular velocities, have been shown to yield global asymptotic stability (Joshi et al., 1995). Moreover, the proposed controller in Joshi et al. (1995) is also robust to modeling errors and can be used for large-angle maneuvers with guaranteed stability. A study conducted on a micro satellite, where attitude control was performed using integrator backstepping based on quaternion feedback, showed that the controller yielded the closed loop equilibrium points asymptotically stable in the sense of Lyapunov (Kristiansen et al., 2008).

Controllers based on the same principle, but utilizing different attitude representation, have been successfully applied for spacecraft attitude control. For instance, control techniques based on sliding mode control (SMC) have been implemented based on modified Rodriquez parameters (Crassidis and Markley, 1996) and based on quaternions (McDuffie and Shtessel, 1997).

Another approach to design the attitude controller is provided by the theory of optimal control (Meyer, 1971). The methods within optimal control could be hard to apply to systems that are both nonlinear and multidimensional, and it could be time consuming to compute control laws for such systems (Meyer, 1971). Nevertheless, the optimal control theory is useful for the analysis of system performance, and the methods based on this theory are elegant and explicit (Meyer, 1971). The theory of optimal control and its applications within attitude control is studied more in-depth in Section 2.5.

2.5 Optimal control

This section provides a brief introduction to the field of optimal control and is primarily based on the works by Athans and Falb (2013), Gros and Diehl (2019), and Nocedal and Wright (2006). Section 2.5.1 introduces some key concepts within optimization, whereas Section 2.5.2 introduces the reader to previous works in the field of optimal spacecraft attitude control.

Optimal control is a field within optimization that deals with the optimization of dynamic systems (Gros and Diehl, 2019). A dynamic system describes processes that change over time, and the evolving processes are often indicated by states x. Control inputs, u, might be used to control the dynamic system. These control inputs can be chosen to optimize some objective function, or cost function, with respect to some constraints, hence the name optimal control. The objective function relates to some requirement put on the system's output, i.e., it can be viewed as a quantitative measure of the performance of the system (Athans and Falb, 2013; Nocedal and Wright, 2006). Depending on what real-world problem the system model describes, the objective could be time, potential energy, money, or any other quantity that can be represented by a single number (Nocedal and Wright, 2006). The constraints represent physical or artificial limitations on the system (Athans and Falb, 2013); for instance, it would not make sense to have quantities such as time or mass be negative.

An example of an optimal control problem is the control of a rocket in outer space (Nagahara, 2020). Note that this is a simple example of rocket control which does not include attitude control of the spacecraft. The position and velocity of the rocket compose the state vector, $\mathbf{x}$, and the rocket engine makes up the propulsion system. The control input, u, represents the engine power that can be chosen at each time instant. A numerical solver might be able to find the control input that minimizes the rocket's travel time from one position to another, subject to a constraint on the magnitude of the control input. The solver minimizes an objective function to find the optimal solution, and for the rocket example, the objective function would involve one or multiple terms related to time. When the objective function has been determined, an engineer set to solve the task may formulate the control problem as follows (Athans and Falb, 2013): determine the control inputs which yield the desired output, satisfies the constraints, and which optimize the objective function. The solution to this problem is called optimal control.

2.5.1 Numerical optimization

In this section, some of the concepts within numerical optimization are explored. The information presented in this section is primarily based on the works by Athans (1963), Gros and Diehl (2019), and Nocedal and Wright (2006).

Optimization is an important tool in problems that deal with the analysis of physical systems (Nocedal and Wright, 2006). Before applying optimization to a problem, it is necessary to identify an objective function, which depends on some characteristics of the system, named variables or unknowns. These variables are often subject to constraints, and the goal of the optimization is to find the variables that optimize the objective function. The objective function, the variables, and the constraints for a given problem have to be identified, a process referred to as modeling (Nocedal and Wright, 2006). It is challenging to find a sufficient mathematical description of a physical system (Athans and Falb, 2013). One reason for this is that mathematical models are not unique, as they are approximations of the real world. Another reason that makes it hard to formulate the mathematical description is that requirements such as reliability and simplicity are hard to formulate using mathematics. Therefore, it is important to note that the mathematical descriptions of physical systems, or other real-life problems, are simplified models of the real world.

A standard optimization problem may be formulated accordingly (Nocedal and Wright, 2006)

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & c_i(x) = 0, \quad i \in \mathcal{E} \\ & c_i(x) \geq 0, \quad i \in \mathcal{I}, \end{aligned} \tag{2.43}$$

where x is a vector that contains the variables, f is the objective function, c_i are the constraint functions, and $\mathcal{E}$ and $\mathcal{I}$ are the sets of equality and inequality constraints, respectively.

To solve the optimization problem, an optimization algorithm is usually applied to the problem. There exist several types of optimization algorithms, and each of the algorithms is specially adapted to solve a particular optimization problem (Nocedal and Wright, 2006). Therefore, it is important to choose the appropriate optimization algorithm when aiming to solve a problem, as it may affect whether the optimal solution is found or not, and if a solution exists - whether it is found fast or slow (Nocedal and Wright, 2006).

An important optimization algorithm, and one of the fundamental tools in optimization and control, is Newton's method. The key idea of Newton's method is as follows (Gros and Diehl, 2019): suppose $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a continuously differentiable function, and that the aim is to solve the nonlinear system equation

$$F(x) = 0, \tag{2.44}$$

where $x \mapsto F(x)$. Starting from an initial guess x_0, Newton's method recursively produces a series of iterates $\{x_k\}_{k=0}^{\infty}$ by linearizing (2.44) at the current iterate

$$F(x_k) + \frac{\partial F}{\partial x}(x_k)(x - x_k) = 0, \tag{2.45}$$

where the term 'iterate' means the improved estimate of the variable x (Nocedal and Wright, 2006). The next iterate can be computed explicitly by solving the linear system, which is Newton's method

$$x_{k+1} = x_k - \left(\frac{\partial F}{\partial x}(x_k)\right)^{-1} F(x_k), \tag{2.46}$$

and the stopping criteria for the method could be to terminate when $|x_{k+1} - x_k| < \delta$, or when $|f(x_{k+1})| < \delta$ (Weerakoon and Fernando, 2000). Here, δ is a user-defined threshold value.

An important assumption in (2.46) is that the Jacobian $J(x_k) \triangleq \frac{\partial F}{\partial x}(x_k)$ is invertible, and a more general formulation of Newton's method can be made using an invertible approximation M_k of $J(x_k)$ (Gros and Diehl, 2019)

$$x_{k+1} = x_k - M_k^{-1} F(x_k). \tag{2.47}$$

The local convergence of the iterates in (2.47) may be fast or slow, or they may not converge at all, depending on how well M_k approximates J_k.

From (2.46) it is clear that the derivative of the function F is needed. Hence, Newton's methods require that the derivative of F exists. Therefore, it is necessary to make sure that when applying Newton's method to a set of equations or a system model, the equations have to be differentiable. For instance, the maximum function, max(x), is neither smooth nor differentiable, and Newton's method is therefore not applicable to the function. The maximum function may be rewritten into a smooth and differentiable function to overcome this issue on the following form

$$\text{max}(x_1, x_2) = \frac{1}{2} \cdot \left((x_1 + x_2) + \sqrt{(x_1 - x_2)^2 + \alpha}\right), \tag{2.48}$$

where α is a parameter to be tuned.

An example of a Newton-based optimization method, which is particularly effective when solving nonlinear problems, is the Sequential Quadratic Programming (SQP) approach (Gros and Diehl, 2019; Nocedal and Wright, 2006). SQP approaches to the next iterate by solving quadratic subproblems and is appropriate for solving both large and small optimization problems (Nocedal and Wright, 2006). Another Newton-based method that is useful for nonlinear problems is the interior-point algorithm named IPOPT (Wächter and Biegler, 2006). Interior-point algorithms are competitive when applied to small optimization problems and without competition when applied to problems of larger scale (Gondzio, 2012).

The optimal control problem in (2.43) can be solved using direct methods, which approximate the continuous problem by a sequence of finite nonlinear programs (NLP) (Gros and Diehl, 2019). Then, the sequence of NLPs can be solved using NLP solvers and reveals why the direct approach is often characterized as "first discretize, the optimize" (Gros and Diehl, 2019). Three approaches used for discretization of continuous optimal control problems are single shooting, multiple shooting, and collocation (Gros and Diehl, 2019). In the two following paragraphs, single shooting and multiple shooting are explained using direct optimization.

Direct single shooting first discretizes the control on a fixed grid, for instance $t_0 < t_1 < \ldots < t_{N-1} < t_N = T_f$, where f denotes the final time (Diehl et al., 2006). Then, a numerical solver is applied to solve an initial value problem with the discretized controls, to obtain the state trajectory over the time horizon (Diehl et al., 2006).

Direct multiple shooting transcribes a continuous optimal control problem into NLPs, by discretizing the control on a chosen grid $t_0 < t_1 < \ldots < t_{N-1} < t_N = T_f$ (Bock and Plitt, 1984), where f denotes the final time. The states are computed at each subinterval, and a matching condition ensures continuity of the solution trajectory (Bock and Plitt, 1984). The matching condition essentially says that the state value at the beginning of control interval m has to equal the state value at the end of control interval $m - 1$, i.e., it closes the shooting gaps and enforces continuity (Bock and Plitt, 1984; Gros and Diehl, 2019). Direct single shooting essentially does the same, only with $N{=}1$ (Bock and Plitt, 1984).

Continuous and discrete time optimization

Dynamic systems evolve over time, and time can be expressed using two different concepts, namely continuous time and discrete time (Gros and Diehl, 2019). Physical time is continuous, and it is common to express the dynamics of many biological and technical systems using continuous time (Gros and Diehl, 2019). The use of discrete time often makes sense for digital systems (Gros and Diehl, 2019).

In continuous optimization problems, the variables used in the objective function and the constraints are continuous, i.e., the variables are real numbers (Nocedal and Wright, 2006). On the other hand, some optimization problems may contain variables that only make sense if they take on integer, or even binary, values. Such optimization problems are called discrete optimization problems, and the defining feature of such problems is that the unknown variables are drawn from a finite, but often very large, set (Nocedal and Wright, 2006). Normally, it is easier to solve continuous optimization problems due to the smoothness of the objective function and constraint functions. This characteristic enables the use of information from the objective and constraints at a particular point x to gather information about the function's behaviour at every point near x (Nocedal and Wright, 2006). Conversely, in discrete optimization problems, the behaviour of the objective function and constraint functions may differ significantly, moving from one feasible point to the next, even though the two points are "close" to one another (Nocedal and Wright, 2006).

Constrained and unconstrained optimization

Optimization problems may be classified according to the linearity, or nonlinearity, of the objective function (Nocedal and Wright, 2006). Moreover, the problems can be classified according to the number of variables present in the problem (Nocedal and Wright, 2006). One essential distinction between optimization problems is between those having constraints on the variables and those that do not.

Constrained optimization problems model problems in which there are explicit constraints on the variables, and these constraints play an important role (Nocedal and Wright, 2006). The optimization problem in (2.43) is a constrained optimization problem, with constraint functions $c_i(x)$. The constraint functions could have varying complexity, ranging from simple linear bounds to nonlinear inequalities that represent complex relations between the system variables. It is possible to remove the constraints from a constrained optimization problem by replacing them with a penalizing term which is added to the objective function, thereby discouraging constraint violations (Nocedal and Wright, 2006). Removing the constraints from a constrained optimization problem yields an unconstrained optimization problem. In unconstrained optimization problems, there are no constraints on the variables, i.e., for (2.43) this means that $\mathcal{E} = \mathcal{I} = \emptyset$ (Nocedal and Wright, 2006).

Global and local optimization

The global solution to an optimization problem refers to the point which yields the optimal objective function value among all feasible points, whereas the local solution yields an objective function value that is smaller than all feasible nearby points (Nocedal and Wright, 2006). For convex optimization problems, and especially linear problems, a local solution is also a global solution - but for many problems, it is difficult to recognize a global solution and even harder to locate the solution. For nonlinear optimization problems, there may exist local solutions that are not global solutions (Nocedal and Wright, 2006).

Indirect approach to optimal control problems

Direct and indirect approaches are two different methods used to solve optimal control problems (Gros and Diehl, 2019), and the direct approach was discussed previously in this section. Indirect methods use necessary optimality conditions to eliminate the controls from the optimal control problem and derive a boundary value problem (BVP) using ordinary differential equations (ODE) (Gros and Diehl, 2019). The necessary optimality conditions for continuous time optimal control problems are described by Pontryagin's Maximum (Minimum) Principle (Gros and Diehl, 2019; Pontryagin, 1962). After the BVP has been formulated, it has to be solved numerically, which could be done using shooting techniques or collocation (Gros and Diehl, 2019). Because the indirect approach first identifies the optimality conditions of a problem in continuous time and then discretizes the problem to compute a numerical solution, it is often referred to as "first optimize, then discretize" (Gros and Diehl, 2019). There are two major disadvantages of the indirect approach (Gros and Diehl, 2019). The first relates to the underlying differential equations, which are often hard to solve due to strong nonlinearities and instabilities. The second disadvantage relates to the changes in control structure, which could be difficult to handle as they often require a completely new setup of the problem. One important advantage of direct approaches compared to indirect approaches is that the direct approaches can easily treat all sorts of constraints (Gros and Diehl, 2019).

2.5.2 Optimal spacecraft attitude control

Optimal control has been applied for spacecraft attitude control in multiple cases and using different cost functions. For instance, a time optimal attitude control problem has been studied, where the objective is to minimize the time it takes to rotate a rigid body to a desired attitude and angular velocity while subject to control input constraints (Lee et al., 2008). Other time optimal attitude control problems have been studied, one where the objective is to orient a spacecraft from an initial attitude and angular velocity to a final attitude and angular velocity in minimum time, using two independent control torques (Shen and Tsiotras, 1999), and a second where the objective is the same as before only using reaction wheels (Zhang, 2010).

A cost function based on angular velocity has been used to optimize the attitude motion planning of a spacecraft, with pointing and actuator constraints

(Biggs and Colley, 2016). Control Lyapunov functions, applied together with optimal sliding mode controllers, have been studied in the regards of attitude tracking of spacecraft (Pukdeboon and Zinober, 2012).

In small satellites, the available power is limited due to the little surface area to excess heat, and the electrical power that a small satellite can produce is limited (Schaub and Lappas, 2009). Motivated by the limited power in a small satellite, a power-optimal reaction wheel motor torque distribution strategy has been developed (Schaub and Lappas, 2009). This distribution strategy aims to minimize the instantaneous requirements for electrical power in the small satellite and yields a power-optimal attitude control.

During the operation of a spacecraft, several attitude maneuvers are necessary to perform tasks such as scientific observation and communication (Wu and Han, 2019). Quite a few constraints limit the feasible region of a spacecraft during attitude maneuvers. For instance, the direction of the solar array must satisfy the necessary energy supply, sensitive components and sensors might be damaged by light and radiation, and the path may be limited by the bounds on angular velocities and actuator input torque. Motivated by these restrictions, an objective has been studied that aims to find an energy-optimal attitude maneuver path for rigid body spacecraft under complex constraints (Wu and Han, 2019).

A spacecraft is subject to a wide range of external disturbances during its maneuvering phase, which motivates the use of a robust controller when designing an attitude control system for a spacecraft (Banerjee et al., 2019; Park, 2005). In addition to being robust, the controller should satisfy the constraints put on the spacecraft, i.e., it should be optimal (Banerjee et al., 2019). A robust and optimal control method has been suggested for three-axis attitude control of a spacecraft with external disturbances, in which the control law is based on the minimax approach and the inverse optimal approach (Park, 2005). However, Banerjee et al. (2019) argues that integrating both robustness and optimality in the same control law poses a serious challenge, as these two terms conflict one another in a control system. Therefore, these challenges have been solved in two steps. First, a technique called the Pseudospectral method (PSM) has been applied to obtain the optimal control part of the control system (Banerjee et al., 2019). Secondly, the optimal control has been integrated with sliding mode control (SMC) to ensure an optimal-robust controller for the attitude control of a spacecraft (Banerjee et al., 2019).

Chapter 3

Maximum Hands-Off Control

3.1 Introduction

Maximum hands-off control is a type of control algorithm with control values which are most often zero, i.e., the control values are *sparse*, but still manage to achieve the control objectives (Chatterjee et al., 2016). A hands-off control holds the control values at exactly zero over a time interval, and the *maximum* hands-off control maximizes the time interval over which the control input is exactly zero (Nagahara et al., 2015). The works presented in Nagahara et al. (2013), Nagahara et al. (2015), and Nagahara (2020) gives a thorough introduction to the maximum hands-off controller, and the work in Nagahara et al. (2015) is an extension of the work in Nagahara et al. (2013). Nagahara (2020) presents the maximum hands-off controller within a textbook dedicated to sparsity methods and summarizes the works in Nagahara et al. (2013) and Nagahara et al. (2015). This thesis will primarily refer to Nagahara et al. (2015) which essentially also cover the theoretical material presented in Nagahara et al. (2013) and Nagahara (2020).

Nagahara et al. (2015) argues that applications within the automotive industry, railway vehicles, wireless communication, and networked control could benefit from hands-off control. For instance, a start-stop system in a car that shuts down the engine automatically to prevent it from idling exhibits hands-off behavior (Kirchhoff et al., 2010). A similar feature is used in hybrid cars; when the car is not moving, or the speed is below a given threshold, the fuel engine is stopped, and the electric motor is used (Chan, 2007; Shakouri et al., 2013). Using hands-off control in vehicles could reduce fuel consumption and reduce the CO and CO2 emissions (Nagahara et al., 2015). Moreover, hands-off control is used to reduce energy consumption in railway vehicles (Liu and Golovitcher, 2003; Khmelnitsky, 2000). Application of hands-off control could reduce the control effort dramatically, as the control value is held exactly zero over a time interval (Nagahara et al., 2015).

Maximum hands-off control, or what is frequently called sparsity control, is often approximated through the L_1-optimal control problem, and it is of interest in several applications (Feng et al., 2016). Using L_1-optimal control to approximate maximum hands-off control is what is done in Nagahara et al. (2015). Applications within machine learning and image processing, including compressed sensing (Donoho, 2006), feature selection (Ng, 2004), image restoration (Dong and Zhang, 2013), and trend filtering (Kim et al., 2009) seek to solve sparsity minimization problems. The use of the L_1-norm to approximate the maximum hands-off controller is discussed further in Section 3.4.

The rest of this chapter is organized as follows. Section 3.2 contains the mathematical preliminaries required for understanding the concepts behind maximum hands-off control. Section 3.3 formulates the maximum hands-off control problem using the notation from Section 3.2. Section 3.4 gives a brief summary of L_1-optimal control, whereas Section 3.5 discusses the theoretical relation between maximum hands-off control and L_1-optimal control. Section 3.6 presents various reformulations of the maximum hands-off control problem. To become more familiar with the maximum hands-off controller, some examples are presented in Section 3.7.

3.2 Mathematical preliminaries

Some mathematical preliminaries need to be in place in order to understand the concept and theoretical derivation of maximum hands-off control. Nagahara et al. (2015) gives a detailed review of the mathematics behind maximum hands-off control, and the content presented in this section is based on the review by Nagahara et al. (2015).

The L_1-, L_2-, and L_∞-norm of a vector $\mathbf{x} \in \mathbb{R}^n$ are defined as

$$\|\mathbf{x}\|_1 \triangleq \sum_{i=1}^{n} |x_i|, \tag{3.1a}$$

$$\|\mathbf{x}\|_2 \triangleq \sqrt{\sum_{i=1}^{n} |x_i|^2}, \tag{3.1b}$$

$$\|\mathbf{x}\|_\infty \triangleq \max_{i=1,\dots,n} |x_i|, \tag{3.1c}$$

respectively. The L_p-norm, with $p \in [1, \infty)$, for a vector of continuous-time signals $\mathbf{u}(t)$ over the time interval $[0, T)$ is defined as

$$\|\mathbf{u}\|_p \triangleq \left(\int_0^T \|\mathbf{u}(t)\|^p dt \right)^{\frac{1}{p}}. \tag{3.2}$$

The norm $\|\cdot\|$ inside the integral in (3.2) can be any norm p-norm for $p \in [1, \infty]$ (Khalil, 2002), i.e., all norms presented in (3.1) could be used in the definition of $\|\cdot\|_p$. If $p \in (0, 1)$, in (3.2), then $\|\cdot\|_p$ is not a norm as it fails to satisfy the triangle inequality (Nagahara et al., 2015).

The support of a function is the set of points where the function takes on nonzero values (Royden and Fitzpatrick, 1988), and the support set of a function $u(t)$, is defined on the time interval $t \in [0, T]$ as (Nagahara et al., 2015)

$$\text{supp}(u(t)) \triangleq \{t \in [0, T] : u(t) \neq 0\}. \tag{3.3}$$

By using (3.3), the L_0-norm for a vector of continuous-time signals $\mathbf{u}(t)$ can be defined accordingly:

$$\|\mathbf{u}\|_0 \triangleq \mu(\text{supp}(\mathbf{u}(t))), \tag{3.4}$$

where $\mu(\cdot)$ is the Lebesgue measure.

The Lebesgue measure is explained thoroughly in Royden and Fitzpatrick (1988), and it can be summarized as follows: the length of an interval I bounded by the endpoints a and b is defined to be $\ell(I) = b - a$, whereas if the interval I is unbounded its length is defined to be $\ell(I) = \infty$. The Lebesgue measure extends the concept of "length" to subsets of the n-dimensional Euclidean space. For n=1, the Lebesgue measure corresponds to the standard measure of length. While the Lebesgue measure gives the length of which a signal is defined, the second part of the L_0-norm, the support, returns a positive value when the signal is different from zero, and zero otherwise. Following, the signal's L_0-norm is the total length for which a continuous-time signal takes on nonzero values, i.e., the sparsity of the signal, which is clear from the definition of the L_0-norm in (3.4). In (3.4) the Lebesgue measure "counts" the length of the support of the signal $\mathbf{u}(t)$, i.e., the Lebesgue measure sums up the parts where the control signal is nonzero. Note that the L_0-"norm" is not an actual norm because it is not positive homogeneous (Nagahara et al., 2015).

3.3 Formulation of the Maximum Hands-Off Control Problem

The maximum hands-off control problem is formulated in this section. Briefly explained, the maximum hands-off control is the control that maximizes the time interval over which the control input is exactly zero. To put it more precisely, the controller minimizes the Lebesgue measure of the support, or the L_0-norm, which was defined in Section 3.2.

Nagahara et al. (2015) considers a general nonlinear system when introducing the maximum hands-off controller

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \sum_{i=1}^{m} \mathbf{g}_i(\mathbf{x}(t))u_i(t), \quad t \in [0, T], \tag{3.5}$$

where $\mathbf{x}(t)$ is the n-dimensional state vector, $\dot{\mathbf{x}}(t) = \frac{d\mathbf{x}(t)}{dt}$, $u_i(t)$ are the control inputs, m is the total number of control inputs, and $\mathbf{f}(\cdot)$ and $\mathbf{g}_i(\cdot)$ are functions on $\mathbb{R}^n$.

The vector representation $\mathbf{u}(t)$ denotes the vector of the control inputs $u_i(t)$, and it aims to drive the state, $\mathbf{x}(t)$, from an initial state, $\mathbf{x}(0) = \mathbf{x}_0$, to a final state, $\mathbf{x}(T) = \mathbf{0}$, at time $t = T$. Moreover, the magnitude of the control input, $\mathbf{u}(t)$, is constrained by

$$\|\mathbf{u}(t)\|_\infty \leq u_{\text{limit}}, \ t \in [0, T], \tag{3.6}$$

where u_{limit} is the constraint on the magnitude of the control input. Nagahara et al. (2015) use $u_{\text{limit}} = 1$, i.e., $\|\mathbf{u}(t)\|_\infty \leq 1$, in their work.

If a control $\{\mathbf{u}(t) : t \in [0, T]\}$ satisfies (3.6), and drives the state, $\mathbf{x}(t)$, from the initial state to the final state within the desired time, the control is called *admissible* (Nagahara et al., 2015). The set of all admissible controls is denoted by $\mathcal{U}$. Another way to formulate the purpose of the maximum hands-off controller is that it aims to find the sparsest control among all admissible controls in $\mathcal{U}$ (Nagahara et al., 2015).

To find the sparsest control among all admissible controls for the system in (3.5), the L_0-cost function has to be minimized, and the L_0-cost function is defined using the L_0-norm as (Nagahara et al., 2015)

$$J_0(\mathbf{u}) \triangleq \sum_{i=1}^{m} \lambda_i \|u_i\|_0, \tag{3.7}$$

where λ_i are positive weights. The control that minimizes (3.7) is called the *maximum hands-off control*, or the *L_0-optimal control*, and it is the sparsest control among all admissible controls (Nagahara et al., 2015).

The L_0-cost function in (3.7) is discontinuous and nonconvex (Nagahara, 2020). Solving discontinous and nonconvex optimization problems are generally hard (Chatterjee et al., 2016), and solving the L_0-optimal control problem is NP-hard (Candes and Tao, 2005; Feng et al., 2016). Several relaxation methods and reformulations have been suggested to (3.7). Nagahara et al. (2015) replaces the L_0-norm by the L_1-norm, which is studied more thoroughly in Section 3.4. Feng et al. (2016), on the other hand, reformulates the L_0-optimal control problem using a set of complementary constraints, an idea that is explored further in Section 3.6.

3.4 L_1-Optimal Control

The L_0-optimal control problem is NP-hard, and several reformulations of the problem have been suggested, as mentioned in Section 3.3. One reformulation that has been suggested replaces the L_0-norm, $\|u\|_0$, in (3.7), with the L_1-norm, $\|u\|_1$ (Candes and Tao, 2005; Nagahara et al., 2015). This approximation yields an L_1-optimal control problem. L_1-optimal control is discussed in detail in Athans (1963) and Athans and Falb (2013), and a brief review of these works are given in Nagahara et al. (2015), which forms the basis for this section. Note that the approach used in Nagahara et al. (2015) to solve the L_1-optimal control problem is the indirect approach, which is discussed in Section 2.5.

The L_1-optimal control problem is formulated in Nagahara et al. (2015): for the system in (3.5), find an admissible control $\{\mathbf{u}(t) : t \in [0, T]\} \in \mathcal{U}$ that minimizes the L_1-cost function, which is defined as

$$J_1(\mathbf{u}) \triangleq \sum_{i=1}^{m} \lambda_i \|u_i\|_1 = \sum_{i=1}^{m} \lambda_i \int_0^T |u_i(t)| \mathrm{d}t, \tag{3.8}$$

where λ_i are positive weights and m is the total number of control inputs. The L_1-optimal control problem based on (3.8) can be formulated as follows

$$\underset{\mathbf{u}}{\text{minimize}} \quad J_1(\mathbf{u}) \triangleq \sum_{i=1}^{m} \lambda_i \|u_i\|_1. \tag{3.9}$$

The relation between the maximum hands-off control and the L_1-optimal control can be established by first considering the Hamiltonian function for the L_1-optimal control problem, which is defined as (Nagahara et al., 2015)

$$H(\mathbf{x}, \mathbf{p}, \mathbf{u}) = \sum_{i=1}^{m} \lambda_i |u_i| + \mathbf{p}^\top \left(\mathbf{f}(\mathbf{x}) + \sum_{i=1}^{m} \mathbf{g}_i(\mathbf{x}) u_i \right), \tag{3.10}$$

where $\mathbf{p}$ is the costate vector, λ_i are positive weights, m is the number of control inputs, and $\mathbf{g}_i(\cdot)$ are functions on $\mathbb{R}^n$. Pontryagin's minimum principle (Pontryagin, 1962) claims that there exists a costate $\mathbf{p}^*$ such that an optimal control $\mathbf{u}^*$ satisfies

$$H(\mathbf{x}^*, \mathbf{p}^*, \mathbf{u}^*) \leq H(\mathbf{x}^*, \mathbf{p}^*, \mathbf{u}), \tag{3.11}$$

for all admissible $\mathbf{u}$, where $\mathbf{x}^*$ is the optimal state resulting from the optimal control $\mathbf{u}^*$. The costate $\mathbf{p}^*$ and the optimal state $\mathbf{x}^*$ satisfies the following canonical equations (Nagahara et al., 2015)

$$
\begin{aligned}
\dot{\mathbf{x}}^* &= \mathbf{f}(\mathbf{x}^*(t)) + \sum_{i=1}^{m} \mathbf{g}_i(\mathbf{x}^*(t))u_i^*(t) \\
\dot{\mathbf{p}} &= -\mathbf{f}'(\mathbf{x}^*(t))^\top \mathbf{p}^*(t) - \sum_{i=1}^{m} u_i^*(t)\mathbf{g}_i'(\mathbf{x}^*(t))^\top \mathbf{p}^*(t),
\end{aligned}
\tag{3.12}
$$

where $\dot{\mathbf{x}}^* = \frac{d\mathbf{x}^*}{dt}$ and $\dot{\mathbf{p}}^* = \frac{d\mathbf{p}^*}{dt}$, and $\mathbf{f}'(\cdot)$ and $\mathbf{g}_i'(\cdot)$ are the Jacobians of $\mathbf{f}(\cdot)$ and $\mathbf{g}_i(\cdot)$, respectively. The canonical equations are subject to the boundary conditions $\mathbf{x}^*(0) = \mathbf{x}_0$ and $\mathbf{x}^*(T) = \mathbf{0}$.

The optimal control input vector $\mathbf{u}^* = [u_1^*, \ldots, u_m^*]$ of (3.10) is given by (Nagahara et al., 2015) as

$$
u_i^*(t) = -\mathcal{D}_{\lambda_i}(\mathbf{g}_i(\mathbf{x}^*(t))^\top \mathbf{p}^*(t)), \tag{3.13}
$$

for $t \in [0, T]$, where $\mathcal{D}_\lambda(\cdot)$ is the dead-zone function. The graph of the dead-zone function can be seen in Fig. 3.1, and the function is defined as (Nagahara et al., 2015)

$$
\begin{aligned}
&\mathcal{D}_\lambda(z) = \begin{cases} -1, & \text{if } z < -\lambda \\ 0, & \text{if } -\lambda < z < \lambda, \\ 1, & \text{if } \lambda < z, \end{cases} \\
&\mathcal{D}_\lambda(z) \in [-1, 0], \text{if } z = -\lambda, \\
&\mathcal{D}_\lambda(z) \in [0, 1], \text{if } z = \lambda.
\end{aligned}
\tag{3.14}
$$

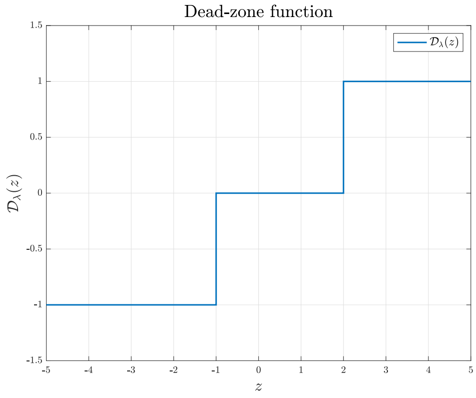


Figure 3.1: Dead-zone function, $\mathcal{D}_\lambda(z)$, for $z \in [-5, 5]$.

According to Nagahara et al. (2015), the L_1-optimal control problem, in (3.8), is called *normal* if the set

$$
\mathcal{T}_i \triangleq \{t \in [0, T] : |\lambda_i^{-1}\mathbf{g}_i(\mathbf{x}^*(t))^\top \mathbf{p}^*(t)| = 1\} \tag{3.15}
$$

is countable for $i = 1, \ldots, m$. If the L_1-optimal control problem is normal, then the components of the optimal control, $\mathbf{u}^*(t)$, are piecewise constant. The components are also ternary, which means that they take on the values -1, 1 or 0 for almost all $t \in [0, T]$. The ternary property of the optimal control, $\mathbf{u}^*(t)$, is called "bang-bang", and it connects the L_1-optimal control and the maximum hands-off control (Nagahara et al., 2015). The connection between L_1-optimal control and the maximum hands-off control is explored further in Section 3.5.

3.4.1 Slack formulation

This section presents a method to solve the L_1-optimal control problem using numerical solvers, i.e., using a direct approach to optimization. A cost function that includes an L_1-norm is non-differentiable, and to obtain a reasonably fast and guaranteed convergence when deploying Newton algorithms on non-differentiable problems; it is necessary to be careful (Gros and Diehl, 2019). Gros and Diehl (2019) suggests a reformulation of a cost function involving the L_1-norm, which removes the non-smoothness from the cost function and place it in the inequality constraints instead. The following example illustrates how an L_1-optimal control problem can be reformulated. First, a minimization problem is formulated using a cost function based on the L_1-norm

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & \|\mathbf{u}\|_1 \\ \text{subject to} \quad & \mathbf{g}(\mathbf{u}) = 0, \end{aligned} \tag{3.16}$$

where $\mathbf{g}(\cdot)$ denotes a constraint function of $\mathbf{u}$. According to Gros and Diehl (2019), (3.16) can be reformulated by introducing an additional set of *slack variables*, $\mathbf{s} \in \mathbb{R}^n$. The slack variables have the same dimension as the vector subjected to the L_1-cost function; in this case the control input vector $\mathbf{u}$. Then, the L_1-norm is reformulated by confining the control input $\mathbf{u}$ between the slack variables $-\mathbf{s}$ and $\mathbf{s}$ accordingly

$$- s_i \leq u_i \leq s_i, \tag{3.17}$$

where the subscript i denotes each element of $\mathbf{s}$ and $\mathbf{u}$. If all constraints are active, then

$$|u_i| = s_i, \tag{3.18}$$

and

$$\|\mathbf{u}\|_1 = \sum_{i=0}^{n} s_i = \mathbf{1}^\top \mathbf{s}. \tag{3.19}$$

Using (3.17), (3.18), and (3.19) , the optimal control problem in (3.16) can be rewritten as

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & \mathbf{1}^\top \mathbf{s} \\ \text{subject to} \quad & -\mathbf{s} \leq \mathbf{u} \leq \mathbf{s}, \\ & \mathbf{g}(\mathbf{u}) = 0. \end{aligned} \tag{3.20}$$

The reformulation of the L_1-optimal control problem using slack variables is also discussed and used by Candes and Tao (2005).

3.5 Connection between L_1-optimal control and L_0-optimal control

The L_1-optimal control problem has been frequently used as a convex relaxation of the L_0-optimal control problem (Chatterjee et al., 2016; Nagahara et al., 2015; Candes and Tao, 2005; Yuan and Ghanem, 2016; Zhang, 2010). This section studies the connection between the two optimal control problems and starts by presenting a theorem that is fundamental to prove the connection between L_0- and L_1-optimality, and the theorem is given in the work by Nagahara et al. (2015).

Theorem 1. Assume that the L_1-optimal control problem, presented in (3.9), has at least one solution and that the problem is normal. Let the optimal solution to (3.8) be a bang-bang control, and the set of optimal solutions be denoted by $\mathcal{U}_1^*$. Let $\mathcal{U}_0^*$ denote the set optimal solutions to the L_0-optimal control problem, presented in (3.7). Then $\mathcal{U}_1^*$ is also L_0-optimal, and $\mathcal{U}_1^* = \mathcal{U}_0^*$ (Nagahara et al., 2015).

Proof. From the assumption, the optimal control $\mathcal{U}_1^*$ is non-empty. Since $\mathcal{U}_1^*$ is an admissible control and non-empty, the set of admissible controls $\mathcal{U}(T, \mathbf{x}_0)$ is also non-empty. Moreover, $\mathcal{U}_0^* \subset \mathcal{U}(T, \mathbf{x}_0)$. This proof first shows that $\mathcal{U}_0^*$ is non-empty. Then, it is shown that $\mathcal{U}_1^* = \mathcal{U}_0^*$.

For any admissible control $\mathbf{u} \in \mathcal{U}(T, \mathbf{x}_0)$, the following holds

$$J_1(\mathbf{u}) = \sum_{i=1}^{m} \lambda_i \int_0^T |u_i(t)| dt = \sum_{i=1}^{m} \lambda_i \int_{supp(u_i)} |u_i(t)| dt \leq \sum_{i=1}^{m} \lambda_i \int_{supp(u_i)} 1 dt = J_0(\mathbf{u}). \tag{3.21}$$

Consider an arbitrary $\mathbf{u}_1^* \in \mathcal{U}_1^*$. From the assumption, the L_1-optimal control problem is normal and the optimal control values $u_{1,i}^*$ in $\mathbf{u}_1$ takes on values ± 1 and 0 for almost every $t \in [0, T]$, which implies the following

$$J_1(\mathbf{u}_1^*) = \sum_{i=1}^{m} \lambda_i \int_0^T |u_{1,i}^*(t)| dt = \sum_{i=1}^{m} \lambda_i \int_{supp(u_{1,i}^*)} 1 dt = J_0(\mathbf{u}_1^*). \tag{3.22}$$

(3.21) and (3.22) shows that $\mathbf{u}_1^*$ minimizes J_0, and that $\mathbf{u}_1^* \in \mathcal{U}_0^*$. This shows that $\mathcal{U}_0^*$ is non-empty and that $\mathcal{U}_1^* \subset \mathcal{U}_0^*$.

On the other hand, let $\mathbf{u}_0^* \in \mathcal{U}_0^* \subset \mathcal{U}(T, \mathbf{x}_0)$, and let $\mathbf{u}_1^* \in \mathcal{U}_1^* \subset \mathcal{U}(T, \mathbf{x}_0)$. The L_1-optimality of $\mathbf{u}_1^*$ and (3.22) yields

$$J_0(\mathbf{u}_1^*) = J_1(\mathbf{u}_1^*) \leq J_1(\mathbf{u}_0^*). \tag{3.23}$$

The L_0-optimality of $\mathbf{u}_0^*$ and (3.21) yields

$$J_1(\mathbf{u}_0^*) \leq J_0(\mathbf{u}_0^*) \leq J_0(\mathbf{u}_1^*). \tag{3.24}$$

(3.23) and (3.24) reveals that $J_1(\mathbf{u}_1^*) = J_1(\mathbf{u}_0^*)$. Thus, $\mathbf{u}_0^*$ minimizes J_1. Consequently, $\mathbf{u}_0^* \in \mathcal{U}_1^*$ and $\mathcal{U}_0^* \subset \mathcal{U}_1^*$ (Nagahara et al., 2015). $\square$

Theorem 1 proposes the use of L_1-optimal control to find the L_0-optimal solution. Using an L_1-cost to obtain a sparse solution in the control input is a well-known result, under "nonsingularity" assumptions on the control system (Chatterjee et al., 2016). However, it is uncertain if L_1-optimization yields sparse results when the problem is singular (Chatterjee et al., 2016). Feng et al. (2016) argues that, even though theoretical results exist that provide conditions under which an optimal solution to the L_1-optimal control problem is also an optimal solution to the L_0-optimal control problem, optimal solutions to the L_1-problem yields suboptimal solutions to the L_0-problem.

3.6 Different formulations for approximating the L_0-optimal control problem

Using L_1-optimal control to solve the L_0-optimal control problem is only one among several approaches. Table 3.1 summarizes the main existing algorithms used to solve the L_0-norm minimization problem.

Method (reference)	**Description**
greedy descent methods (Mallat and Zhang, 1993)	only for smooth, typically quadratic, objective
L_1 norm relaxation (Candes and Tao, 2005)	$\|\mathbf{x}\|_0 \approx \|\mathbf{x}\|_1$
k-support norm relaxation (Argyriou et al., 2012)	$\|\mathbf{x}\|_0 \approx \|\mathbf{x}\|_{\text{k-sup}} \triangleq \max_{\mathbf{0}<\mathbf{v}\leq\mathbf{1},\langle\mathbf{v},\mathbf{1}\rangle\leq k} (\sum_i x)_i^2/\mathbf{v}_i)^{1/2}$
k-largest norm relaxation (Yu et al., 2014)	$\|\mathbf{x}\|_0 \approx \|\mathbf{x}\|_{\text{k-sup}} \triangleq \max_{\mathbf{0}<\mathbf{v}\leq\mathbf{1},\|\mathbf{v}\|_1\leq k} \langle\mathbf{v},\mathbf{x}\rangle$
SOCP convex relaxation (Chan et al., 2007)	$\|\mathbf{x}_0 \leq k \implies \|\mathbf{x}\|_1 \leq \sqrt{k}\|x\|_2$
SDP convex relaxation (Chan et al., 2007)	$\|\mathbf{x}_0 \leq k \implies \mathbf{X} = \mathbf{x}\mathbf{x}^\top, \|\mathbf{X}\|_1 \leq ktr(\mathbf{X})$
Schatten L_p approximation (Ge et al., 2011)	$\|\mathbf{x}\|_0 \approx \|\mathbf{x}\|_p$
re-weighted L_1 approximation (Candes et al., 2008)	$\|\mathbf{x}\|_0 \approx \langle\mathbf{1}, \ln(\|\mathbf{x}\| + \epsilon)\rangle$
L_{1-2} DC approximation (Yin et al., 2015)	$\|\mathbf{x}\|_0 \approx \|\mathbf{x}\|_1 - \|\mathbf{x}\|_2$
0-1 mixed integer programming (Bienstock, 1996)	$\{\|\mathbf{x}\|_0 : \|\mathbf{x}\|_\infty \leq \lambda\} \Leftrightarrow \{\min_{\mathbf{v}\in\{0,1\}}\langle\mathbf{1},\mathbf{v}\rangle : \|\mathbf{x}\| \leq \lambda\mathbf{v}\}$
iterative hard shreadholding MPEC (Beck and Eldar, 2013)	$0.5\|\mathbf{x} - \mathbf{x}'\|_2^2, s.t.\|\mathbf{x}\|_0 \leq k$
non-separable MPEC (Yuan and Ghanem, 2016)	$\|\mathbf{x}\|_0 = \min_{\mathbf{u}}\|\mathbf{u}\|_1, s.t.\|\mathbf{x}\|_1 = \langle\mathbf{x},\mathbf{u}\rangle,\ -\mathbf{1} \leq \mathbf{u} \leq \mathbf{1}$
separable MPEC (Yuan and Ghanem, 2015, 2016)	$\|\mathbf{x}\|_0 = \min_{\mathbf{v}}\langle\mathbf{1},\mathbf{1}-\mathbf{v}\rangle, s.t.\|\mathbf{x}\| \odot \mathbf{v} = \mathbf{0}, \mathbf{0} \leq \mathbf{v} \leq \mathbf{1}$

Table 3.1: L_0-norm optimization techniques (Yuan and Ghanem, 2016).

Another approach to solve the L_0-optimal control problem is presented in Feng et al. (2016). The work presented by Feng et al. (2016) is of particular interest, as they aim to find improved solutions to the L_0-optimal control problem by reformulating the problem in terms of complementarity constraints and show that their solutions are sparser than the L_1-approximation. The rest of this section discusses the reformulations presented in Feng et al. (2016), and the following L_0-problem is considered

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & \|\mathbf{x}\|_0 \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ & \mathbf{C}\mathbf{x} = \mathbf{d}, \end{aligned} \tag{3.25}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector, $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{C} \in \mathbb{R}^{k \times n}$ are matrices, and $\mathbf{b} \in \mathbb{R}^m$ and $\mathbf{d} \in \mathbb{R}^k$ are given vectors.

3.6.1 Full complementarity

One way to reformulate the L_0-norm is through the use of complementarity constraints. First, the state vector x is divided into its non-negative parts, $\mathbf{x}^+$, and non-positive parts, $\mathbf{x}^-$, as follows, $\mathbf{x} = \mathbf{x}^+ - \mathbf{x}^-$. Then, a vector $\boldsymbol{\xi} \in [0, 1]^n$ is defined, being the complementary vector to $|\mathbf{x}|$, the absolute value vector of x. This leads to the full complementarity formulation of (3.25) (Feng et al., 2016)

$$\begin{aligned} \underset{\mathbf{x}, \mathbf{x}^{\pm}, \boldsymbol{\xi}}{\text{minimize}} \quad & \mathbf{1}_N^\top (\mathbf{1}_N - \boldsymbol{\xi}) = \sum_{j=1}^{N} (1 - \xi_j) \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ & \mathbf{C}\mathbf{x} = \mathbf{d} \\ & \mathbf{x} = \mathbf{x}^+ - \mathbf{x}^- \\ & 0 \leq \boldsymbol{\xi} \perp \mathbf{x}^+ + \mathbf{x}^- \geq 0 \\ & \mathbf{x}^{\pm} \geq 0 \\ & \boldsymbol{\xi} \leq \mathbf{1}_N, \end{aligned} \tag{3.26}$$

with $\mathbf{1}_N$ being the N-vector of ones, N is the number of control intervals, the scalar $1 - \xi_j$ indicates the support of the state x_j, and $\mathbf{1}_N - \boldsymbol{\xi}$ is the support vector of x. The support $1 - \xi_j$ of the state x_j in (3.26) essentially plays the same role as the support, supp$(\cdot)$, in (3.3). Moreover, the sum-sign used to add the supports $1 - \xi_j$ in (3.26) corresponds to the Lebesgue measure $\mu(\cdot)$ in (3.4). If x is an optimal solution to the problem presented in (3.25), and $\mathbf{x}^{\pm} \triangleq$ max(0, $\pm$x), and

$$\xi_j \triangleq \begin{cases} 0, & \text{if } x_j \neq 0 \\ 1, & \text{if } x_j = 0, \end{cases} \tag{3.27}$$

for $j = 1, \ldots, N$, then ($\mathrm{x}^{\pm}$, $\boldsymbol{\xi}$) is an optimal solution to the minimization problem in (3.26) with objective function equal to $\|\mathrm{x}_0\|$ (Feng et al., 2016).

The complementarity condition, $0 \leq \boldsymbol{\xi} \perp \mathrm{x}^{+} + \mathrm{x}^{-} \geq 0$, in (3.26), can be reformulated to provide a smooth and continuous NLP formulation. When reformulating the complementarity condition, the notation $0 \leq y \perp z \geq 0$ is used for simplicity purposes. One reformulation of the complementarity condition, called the componentwise or Hadamard complementarity, is given in Feng et al. (2016) as

$$(y, z) \geq 0 \text{ and } y \circ z \leq 0,$$

where $y \circ z$ denotes the Hadamard product of the vectors y and z, i.e., the componentwise product.

3.6.2 Half complementarity

The full complementarity problem, in (3.26), may be formulated more simply. This formulation is called the half complementarity formulation (Feng et al., 2016)

$$\begin{aligned} \underset{\mathbf{x},\boldsymbol{\xi}}{\text{minimize}} \quad & \mathbf{1}_N^\top(\mathbf{1}_N - \boldsymbol{\xi}) = \sum_{j=1}^{N}(1 - \xi_j) \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ & \mathbf{C}\mathbf{x} = \mathbf{d} \\ & 0 \leq \boldsymbol{\xi} \leq \mathbf{1}_N \\ & \boldsymbol{\xi} \circ \mathbf{x} = 0. \end{aligned} \tag{3.28}$$

The definition of $\boldsymbol{\xi}$, from (3.27), leads to the equivalence between the minimization problem in (3.25), and the half complementarity formulation in (3.28). As there is no non-negativity requirement on the variable x in (3.28), the constraints are not truly of the complementarity type. Nevertheless, the constraint $\boldsymbol{\xi} \circ \mathbf{x} = 0$ ensures that either $\xi_i = 0$ or $x_i = 0$, or both equals zero, for all i (Feng et al., 2016).

3.6.3 A general L_0-norm minimization problem

The L_0-norm and the complementarity formulations, presented in (3.26) and (3.28), makes it possible to formulate several minimization problems as programs with complementarity constraints (Feng et al., 2016). A general NLP model on this form is formulated as (Feng et al., 2016)

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) + \gamma \|\mathbf{x}\|_0 \\ \text{subject to} \quad & c_i(\mathbf{x}) = 0,\ i \in \mathcal{E} \\ & c_i(\mathbf{x}) \leq 0,\ i \in \mathcal{I}, \end{aligned} \tag{3.29}$$

where $\mathcal{E}$ and $\mathcal{I}$ are two finite index sets, γ is a positive scalar, $f(\cdot)$ is the continuously differentiable objective function and c_i the continuously differentiable constraint functions.

The NLP in (3.29) may be rewritten using an equivalent complementarity constrained formulation, similar to the full complementarity formulation in (3.26) (Feng et al., 2016)

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) + \boldsymbol{\gamma}^\top(\mathbf{1}_N - \boldsymbol{\xi}) \\ \text{subject to} \quad & c_i(\mathbf{x}) = 0,\ i \in \mathcal{E} \\ & c_i(\mathbf{x}) \leq 0,\ i \in \mathcal{I} \\ & \mathbf{x} = \mathbf{x}^+ - \mathbf{x}^- \\ & 0 \leq \boldsymbol{\xi} \perp \mathbf{x}^+ + \mathbf{x}^- \geq 0 \\ & 0 \leq \mathbf{x}^+ \perp \mathbf{x}^- \geq 0 \\ & \boldsymbol{\xi} \leq \mathbf{1}_N, \end{aligned} \tag{3.30}$$

where $\boldsymbol{\gamma}$ is a positive vector. The complementarity constrained formulation in (3.30) is a Mathematical Program with Complementarity Constraints (MPCC) because the objective function and the constraint functions are nonlinear. The interested reader is referred to Fletcher and Leyffer (2004) for an introduction of how to solve mathematical programs with complementarity constraints as nonlinear programs.

The NLP in (3.29) may also be reformulated using an approach similar to the half complementarity formulation in (3.28) (Feng et al., 2016)

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) + \boldsymbol{\gamma}^\top(\mathbf{1}_N - \boldsymbol{\xi}) \\ \text{subject to} \quad & c_i(\mathbf{x}) = 0,\ i \in \mathcal{E} \\ & c_i(\mathbf{x}) \leq 0,\ i \in \mathcal{I} \\ & 0 \leq \boldsymbol{\xi} \leq \mathbf{1}_N \\ & \boldsymbol{\xi} \circ \boldsymbol{x} = 0. \end{aligned} \tag{3.31}$$

3.6.4 Relaxed formulations

In general, exact reformulations of an MPCC does not lead to a well-posed NLP (Feng et al., 2016). The complementarity constraints in the MPCC are therefore relaxed, which result in NLPs that have better properties. A relaxation of (3.31) is given as (Feng et al., 2016)

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) + \boldsymbol{\gamma}^\top(\mathbf{1}_N - \boldsymbol{\xi}) \\ \text{subject to} \quad & c_i(\mathbf{x}) = 0, \ i \in \mathcal{E} \\ & c_i(\mathbf{x}) \le 0, \ i \in \mathcal{I} \\ & \boldsymbol{\xi} \le \mathbf{1}_N \\ & \boldsymbol{\xi} \circ \mathbf{x} \le \epsilon \mathbf{1}_N \\ -& \boldsymbol{\xi} \circ \mathbf{x} \le \epsilon \mathbf{1}_N \\ & \boldsymbol{\xi} \ge 0. \end{aligned} \tag{3.32}$$

It is desirable to investigate the properties of the relaxed problem when ϵ approaches zero because then the complementarity constraints would equal zero. For $\epsilon = 0$, the relaxed problem in (3.32) equals the problem in (3.31).

3.7 Examples

A few test examples were carried out to become familiar with the maximum hands-off controller and the L_1-norm reformulation of the controller. Two systems, one linear and one nonlinear system, were investigated for each controller.

3.7.1 Linear example problem

The linear example problem is used by Nagahara (2020) when investigating the L_1-norm approach to the maximum hands-off control problem. The linear system is sometimes referred to as the double integrator and defined as

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= u(t), \end{aligned} \tag{3.33}$$

with $t \in [0, T]$, where $T = 5$. The initial state is set to $x_1(0) = x_2(0) = 1$, the final state is $x_1(T) = x_2(T) = 0$ and the control input is bounded by $|u(t)| \le 1$.

3.7.2 Nonlinear example problem

The nonlinear example is inspired by the problem presented in Andersson et al. (2016). The system model is given as

$$\begin{aligned} \dot{x}_1(t) &= (1 - x_2(t)^2)x_1(t) - x_2(t) + u(t) \\ \dot{x}_2(t) &= x_1(t), \end{aligned} \tag{3.34}$$

with $t \in [0, T]$ where $T = 10$. The initial state is set to $x_1(0) = 0$ and $x_2(0) = 1$, the final state is $x_1(T) = x_2(T) = 0$ and the control input is bounded by $|u(t)| \leq 1$. The state x_2 is constrained by $x_2(t) \geq -0.25$.

3.7.3 L_1-optimal control: linear example

In this section, the linear example given in Section 3.7.1 was solved using L_1-optimal control. The L_1-optimal control problem was formulated in MATLAB and solved using CasADi, see Section 5.4.

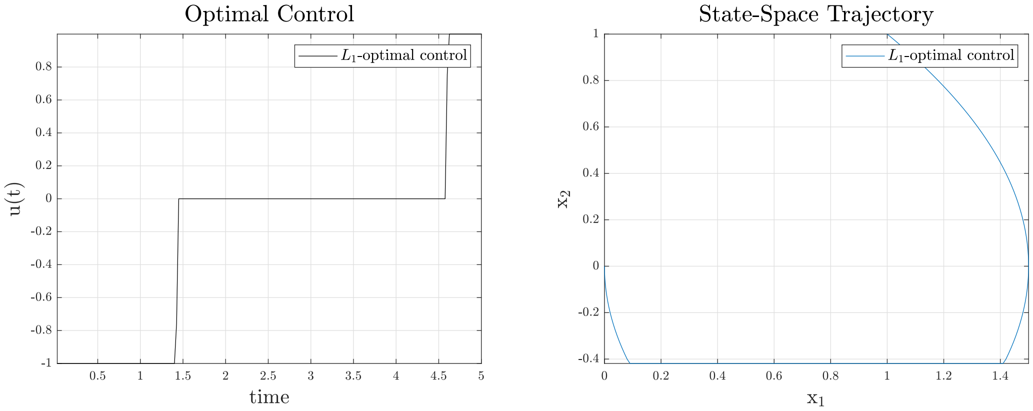
A cost function that includes an L_1-norm is non-differentiable, which was discussed in Section 3.4. Therefore, the L_1-optimal control problem was formulated according to Gros and Diehl (2019), using a vector of slack variables s and the notation presented in Section 3.4

$$\begin{aligned} \underset{u,x}{\text{minimize}} \quad & \mathbf{1}^\top \mathbf{s} \\ \text{subject to} \quad & \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}\mathbf{u}, \\ & |u_i(t)| \leq 1, \\ & \mathbf{x}(0) = [1, 1]^\top, \\ & \mathbf{x}(T) = [0, 0]^\top, \\ & -s_i \leq u_i(t) \leq s_i, \end{aligned} \tag{3.35}$$

where $\dot{\mathbf{x}}$ denotes the linear system dynamics in (3.33), $\mathbf{x}$ denotes the state vector, $\mathbf{u}$ denotes the control input vector, u_i denotes each element in $\mathbf{u}$, and s_i each element in the vector $\mathbf{s}$. The matrix $\mathbf{A}$ and the vector $\mathbf{b}$ are given as

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \\ \mathbf{b} &= \begin{bmatrix} 0 & 1 \end{bmatrix}^\top. \end{aligned} \tag{3.36}$$

Fig. 3.2 shows the resulting control input and state-space trajectory, which resembles the results from Nagahara (2020). The shape of the optimal control signal in Fig. 3.2.a, is very similar to the shape of a "bang-bang" control, as the control signal changes almost instantaneously from -1 to 0 at $t \approx 1.5$, and from 0 to 1 at $t \approx 4.5$, which characterizes L_1-optimal control, as discussed in Section 3.4. If the value of the control signal had changed instantaneously it would have been on the exact form of a "bang-bang" control. From Fig. 3.2.b it is clear that the optimal control signal moves the states from the initial state $[1, 1]^\top$ to the final state $[0, 0]^\top$.



(3.2.a) L_1-optimal control.

(3.2.b) State-space trajectory.

Figure 3.2: L_1-optimal control on the linear example.

3.7.4 L_1-optimal control: nonlinear example

In this section, the nonlinear system presented in Section 3.7.2, was solved as an L_1-optimal control problem. The L_1-optimal control problem used to solve (3.34), was formulated as follows

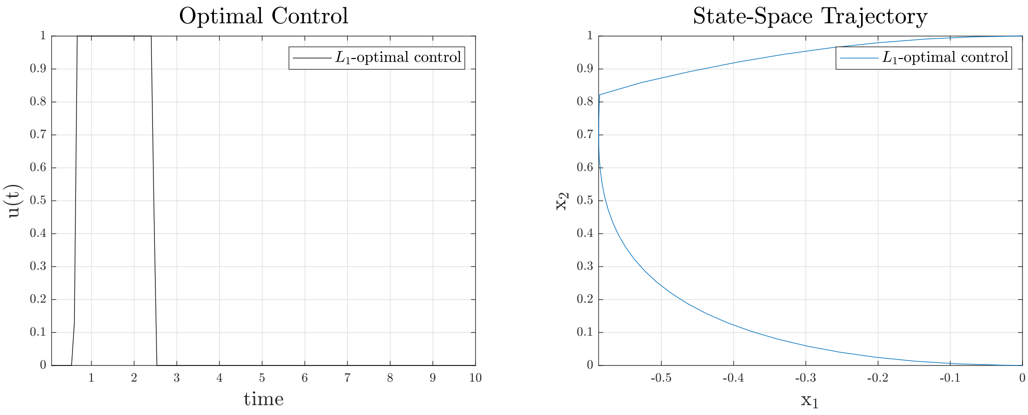
$$\begin{aligned} \underset{u,x}{\text{minimize}} \quad & \mathbf{1}^\top \mathbf{s} \\ \text{subject to} \quad & \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \\ & |u_i(t)| \leq 1, \\ & \mathbf{x}(0) = [0, 1]^\top, \\ & \mathbf{x}(T) = [0, 0]^\top, \\ & \mathbf{x}(t) \geq -0.25, \\ & -s_i \leq u_i(t) \leq s_i, \end{aligned}$$

where $\dot{\mathbf{x}}$ denotes the nonlinear system dynamics in (3.34), $\mathbf{x}$ denotes the state vector, $\mathbf{u}$ denotes the control input vector, u_i denotes each element in $\mathbf{u}$, and s_i each element in the vector of slack variables $\mathbf{s}$.

Fig. 3.3 shows the resulting control input and state trajectory. Fig. 3.3.a illustrates the signals "bang-bang" characteristics, and Fig. 3.3.b shows that the states change from the initial state to the final state while the constraints are not violated.

3.7.5 L_0-optimal control: linear example

In this section, the L_0-optimal control approach to the problem formulated in Section 3.7.1, is presented. The L_0-optimal control problem was formulated using the relaxed approach presented in Section 3.6, and is formulated as follows



(3.3.a) L_1-optimal control. **(3.3.b)** State-space trajectory.

Figure 3.3: L_1-optimal control on the nonlinear example.

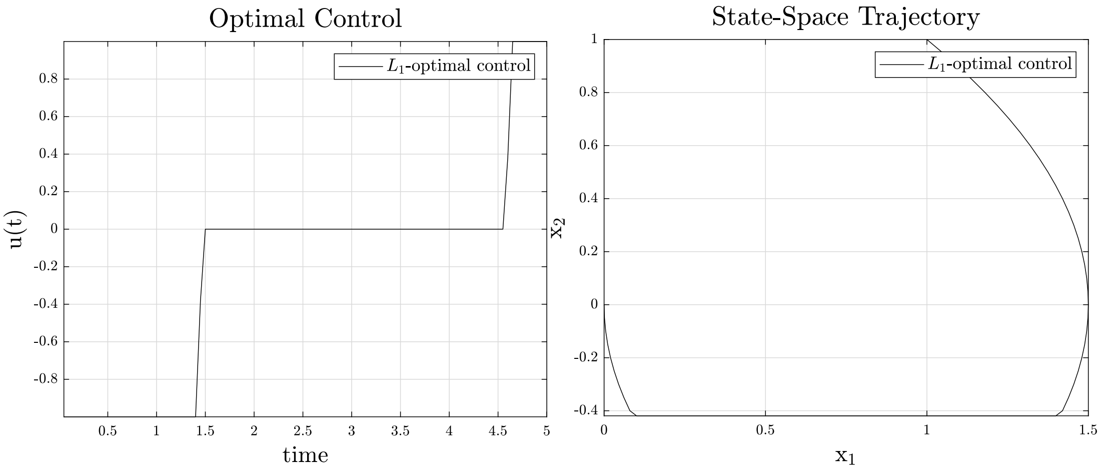
$$
\begin{aligned}
\underset{\mathbf{u},\boldsymbol{\xi}}{\text{minimize}} \quad & (\mathbf{1}_N - \boldsymbol{\xi})\mathbf{1}_N^\top \\
\text{subject to} \quad & \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}\mathbf{u}, \\
& |u_i(t)| \leq 1, \\
& \mathbf{x}(0) = [1, 1]^\top, \\
& \mathbf{x}(T) = [0, 0]^\top, \\
& \boldsymbol{\xi} \leq \mathbf{1}_N, \\
& \boldsymbol{\xi} \circ \mathbf{u} \leq \epsilon \mathbf{1}_N, \\
& -\boldsymbol{\xi} \circ \mathbf{u} \leq \epsilon \mathbf{1}_N, \\
& \boldsymbol{\xi} \geq 0,
\end{aligned}
$$

where $\dot{\mathbf{x}}$ denotes the linear system dynamics in (3.33), $\mathbf{x}$ denotes the state vector, $\mathbf{u}$ denotes the control input vector, and u_i each element in $\mathbf{u}$, $\epsilon = 1 \cdot 10^{-6}$, and $\mathbf{A}$ and $\mathbf{b}$ are defined in (3.36).

Fig. 3.4 shows the resulting control input and state-space trajectory, and it is clear that the L_0-optimal control signal, or maximum hands-off control signal, in Fig. 3.4.a, and the L_1-optimal control signal in Fig. 3.2.a yield similar results. Fig. 3.4.b shows the state-space trajectory resulting from the L_0-controller. The state-space trajectories resulting from the two controllers are similar, which becomes clear comparing the trajectories in Fig. 3.4.b and Fig. 3.2.b.

3.7.6 L_0-optimal control: nonlinear example

The nonlinear problem, from Section 3.7.2, was solved using L_0-optimal control. Similar to the optimal control problem formulated for the linear problem, presented in Section 3.7.5, the relaxed formulation of the L_0-problem was used



(3.4.a) L_0-optimal control. **(3.4.b)** State-space trajectory.

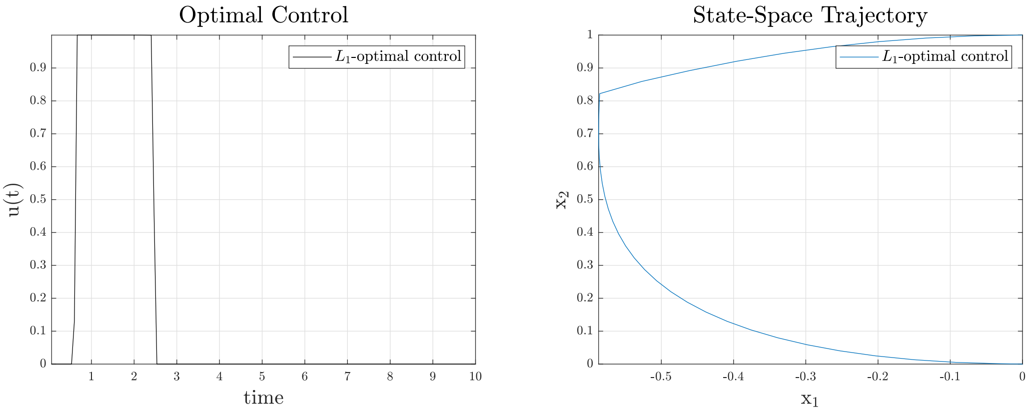
Figure 3.4: L_0-optimal control on the linear example.

to solve the nonlinear problem in this section. The optimal control problem is formulated as follows

$$\begin{aligned} \underset{\mathbf{u},\boldsymbol{\xi}}{\text{minimize}} \quad & (\mathbf{1}_N - \boldsymbol{\xi})\mathbf{1}_N^\top \\ \text{subject to} \quad & |u_i(t)| \leq 1, \\ & \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \\ & \mathbf{x}(0) = [0, 1]^\top, \\ & \mathbf{x}(T) = [0, 0]^\top, \\ & \boldsymbol{\xi} \leq \mathbf{1}_N, \\ & \boldsymbol{\xi} \circ \mathbf{u} \leq \epsilon \mathbf{1}_N, \\ & -\boldsymbol{\xi} \circ \mathbf{u} \leq \epsilon \mathbf{1}_N, \\ & \boldsymbol{\xi} \geq 0, \end{aligned}$$

where $\dot{\mathbf{x}}$ denotes the nonlinear system dynamics in (3.34), $\mathbf{x}$ denotes the state vector, $\mathbf{u}$ denotes the control input vector, and u_i denotes each element in $\mathbf{u}$, and $\epsilon = 1 \cdot 10^{-6}$.

Fig. 3.5 shows the resulting L_0-optimal control and the state-space trajectory. More specifically, the optimal control signal is shown in Fig. 3.5.a and the state-space trajectory is shown in Fig. 3.5.b. By comparing the results in Fig. 3.5 to the results in Fig. 3.3, the similarities between the L_0-optimal control and the L_1-optimal control become apparent as the figures are almost identical.



(3.5.a) L_0-optimal control. **(3.5.b)** State-space trajectory.

Figure 3.5: L_0-optimal control on the nonlinear example.

Chapter 4

Spacecraft Model

In this chapter, the dynamic spacecraft model is deduced. First, the attitude representation used for the model dynamics is presented, followed by an overview of the angular velocity. Then, the spacecraft attitude dynamics, the system perturbations, the reaction wheel dynamics, and the error variables are presented. Finally, the total system dynamics are deduced and presented. The structure of this chapter and the deduction of the dynamic model are both inspired by the work presented in Kristiansen et al. (2020).

4.1 Attitude representation

Unit quaternions, presented in Section 2.2.4, are used to describe the attitude of the spacecraft. The unit quaternion, $\mathbf{q}_b^o$, denotes the attitude of $\{b\}$ relative to $\{o\}$. The rotation matrix from $\{o\}$ to $\{b\}$, $\mathbf{R}_o^b$, is defined in (2.40), and the time derivative of $\mathbf{R}_o^b$ is given as (Egeland and Gravdahl, 2003)

$$\dot{\mathbf{R}}_o^b = \mathbf{S}(\boldsymbol{\omega}_{bo}^b)\mathbf{R}_o^b = -\mathbf{S}(\boldsymbol{\omega}_{ob}^b)\mathbf{R}_o^b, \tag{4.1}$$

where the angular velocity of $\{b\}$ relative to $\{o\}$ and represented in $\{b\}$ is given by $\boldsymbol{\omega}_{ob}^b$.

The differential equation for $\mathbf{q}_b^o$, meaning the kinematic differential equation for the spacecraft's attitude, is given by (Egeland and Gravdahl, 2003)

$$\dot{\mathbf{q}}_b^o = \begin{bmatrix} \dot{\eta}_b^o \\ \dot{\boldsymbol{\epsilon}}_b^o \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -\boldsymbol{\epsilon}_b^{o\top} \\ \eta_b^o\mathbf{I}_{3\times 3} + \mathbf{S}(\boldsymbol{\epsilon}_b^o) \end{bmatrix}\boldsymbol{\omega}_{ob}^b = \frac{1}{2}\mathbf{T}(\mathbf{q}_b^o)\boldsymbol{\omega}_{ob}^b, \tag{4.2}$$

where $\mathbf{T}(\cdot)$ denotes the angular velocity transformation matrix.

4.2 Angular velocity

When analysing the attitude of the spacecraft, three different angular velocities are of interest, namely $\boldsymbol{\omega}_{ob}^{b}$, which is the angular velocity of $\{b\}$ relative to $\{o\}$, $\boldsymbol{\omega}_{io}^{b}$, which is the angular velocity of $\{o\}$ relative to $\{i\}$, and $\boldsymbol{\omega}_{ib}^{b}$, which is the angular velocity of $\{b\}$ relative to $\{i\}$. The angular velocities relate to one another as follows

$$\boldsymbol{\omega}_{ob}^{b} = \boldsymbol{\omega}_{ib}^{b} - \boldsymbol{\omega}_{io}^{b} = \boldsymbol{\omega}_{ib}^{b} - \mathbf{R}_{o}^{b}\boldsymbol{\omega}_{io}^{o}. \tag{4.3}$$

The angular velocity $\boldsymbol{\omega}_{io}^{o}$ is defined as (Oland and Schlanbusch, 2009)

$$\boldsymbol{\omega}_{io}^{o} = \mathbf{R}_{i}^{o}\frac{\mathbf{S}(\mathbf{r}^{i})\mathbf{v}^{i}}{(\mathbf{r}^{i})^{\top}\mathbf{r}^{i}}. \tag{4.4}$$

Moreover, the inertial acceleration of the spacecraft is defined as

$$\dot{\mathbf{v}}^{i} = -\frac{\mu}{\|\mathbf{r}^{i}\|_{2}^{3}}\mathbf{r}^{i}, \tag{4.5}$$

where μ is the standard gravitational parameter of the Earth and $\|\mathbf{r}^{i}\|_{2}$ denotes the 2-norm of $\mathbf{r}^{i}$. The rate of change for $\mathbf{r}^{i}$ is given as

$$\dot{\mathbf{r}}^{i} = \mathbf{v}^{i}. \tag{4.6}$$

4.3 Attitude dynamics

The total system inertia of the spacecraft rigid body, $\mathbf{J} \in \mathbb{R}^{3\times 3}$, is defined as

$$\mathbf{J} = \mathbf{J}_{s} + \mathbf{A}\mathbf{J}_{w}\mathbf{A}^{\top}, \tag{4.7}$$

where $\mathbf{J}_{s} \in \mathbb{R}^{3\times 3}$ denotes the inertia of the spacecraft rigid body excluding the inertia about the spinning axes of the reaction wheels, and $\mathbf{J}_{w} \in \mathbb{R}^{n\times n}$ denotes the inertia matrix of the reaction wheels about the spinning axes, where n is the number of reaction wheels.

The total angular momentum of the spacecraft is defined as (Krogstad and Gravdahl, 2006)

$$\mathbf{H}_{s}^{b} = \mathbf{J}\boldsymbol{\omega}_{ib}^{b} + \mathbf{A}\mathbf{J}_{w}\boldsymbol{\omega}_{bw}^{w}. \tag{4.8}$$

Applying Euler's second axiom, $\frac{{}^{i}d}{dt}\vec{\mathbf{H}} = \sum\vec{\boldsymbol{\tau}}$, to (4.8) yields the rigid body dynamics

$$\frac{{}^{i}d}{dt}\mathbf{H}_{s}^{b} = \mathbf{J}\frac{{}^{b}d}{dt}\boldsymbol{\omega}_{ib}^{b} + \mathbf{A}\mathbf{J}_{w}\frac{{}^{w}d}{dt}\boldsymbol{\omega}_{bw}^{w} + \mathbf{S}(\boldsymbol{\omega}_{ib}^{b})\mathbf{H}_{s}^{b} = \boldsymbol{\tau}_{\text{ext}}^{b}, \tag{4.9}$$

where $\boldsymbol{\tau}^b_{\text{ext}}$ is the external perturbation torque acting on the spacecraft. The torque vector, $\boldsymbol{\tau}^b_{\text{ext}}$, would also include torque produced by actuators on the spacecraft, for instance thrusters or magnetorquers, if such equipment is available.

4.4 Perturbations

The total external perturbation torque acting on the spacecraft is given as

$$\boldsymbol{\tau}^b_{\text{ext}} = \boldsymbol{\tau}^b_{\text{gravity}} + \boldsymbol{\tau}^b_{\text{noise}}, \tag{4.10}$$

where $\boldsymbol{\tau}^b_{\text{gravity}}$ is the external torque caused by the gravity gradient, and $\boldsymbol{\tau}^b_{\text{noise}}$ represents disturbances present in the system. Note that (4.10) is a simplification of the total external perturbation torque that acts on a real-life spacecraft operating in space.

The torque due to the gravity gradient occurs because the spacecraft is an elongated body in the rigid body dynamic model, and it is defined by (Hughes, 2004)

$$\boldsymbol{\tau}^b_{\text{gravity}} = 3\frac{\mu}{\|\mathbf{r}^i\|^3}\mathbf{S}(\mathbf{c}_3)\mathbf{J}\mathbf{c}_3, \tag{4.11}$$

where μ is the standard gravitational parameter of the Earth, $\mathbf{r}^i$ is the distance between the spacecraft and the center of the Earth, and $\mathbf{c}_3$ denotes the third column of the rotation matrix $\mathbf{R}^b_o$.

The torque caused by the noise affecting the system is modelled by a sinusoidal wave

$$\boldsymbol{\tau}^b_{\text{noise}} = A\sin(t)\mathbf{1}_3, \tag{4.12}$$

where A is the amplitude of the wave, t is time, and $\mathbf{1}_3$ is a 3×1-vector of all ones. Note that the torque $\boldsymbol{\tau}^b_{\text{noise}}$ is a simplification of noise that would affect a real spacecraft, and the simplification is not rooted in physical concepts.

4.5 Reaction wheel dynamics

The reaction wheel angular momentum is defined as (Krogstad and Gravdahl, 2006)

$$\begin{aligned}\mathbf{H}^b_w &= \mathbf{A}\mathbf{J}_w\mathbf{A}^\top(\boldsymbol{\omega}^b_{ib} + \mathbf{A}\boldsymbol{\omega}^w_{bw})\\ &= \mathbf{A}\mathbf{J}_w\mathbf{A}^\top(\boldsymbol{\omega}^b_{ib} + \boldsymbol{\omega}^b_{bw})\\ &= \mathbf{A}\mathbf{J}_w\boldsymbol{\omega}^w_{iw}\\ &= \mathbf{A}\mathbf{J}_w\mathbf{A}^\top\boldsymbol{\omega}^b_{iw}.\end{aligned} \tag{4.13}$$

Applying Euler's second axiom to (4.13) yields the reaction wheel dynamics

$$
\begin{aligned}
\frac{{}^{i}d}{dt}\mathbf{H}_w^b &= \frac{{}^{b}d}{dt}(\mathbf{A}\mathbf{J}_w\mathbf{A}^\top\boldsymbol{\omega}_{iw}^b) + \mathbf{S}(\boldsymbol{\omega}_{iw}^b)\mathbf{H}_w^b \\
&= \mathbf{A}\mathbf{J}_w\mathbf{A}^\top\frac{{}^{b}d}{dt}\boldsymbol{\omega}_{iw}^b + \mathbf{S}(\mathbf{A}\boldsymbol{\omega}_{iw}^w)(\mathbf{A}\mathbf{J}_w\boldsymbol{\omega}_{iw}^w) \\
&= \mathbf{A}\mathbf{J}_w\mathbf{A}^\top\frac{{}^{b}d}{dt}\boldsymbol{\omega}_{iw}^b \\
&= \mathbf{A}\mathbf{J}_w\mathbf{A}^\top(\frac{{}^{b}d}{dt}\boldsymbol{\omega}_{ib}^b + \mathbf{A}\frac{{}^{w}d}{dt}\boldsymbol{\omega}_{bw}^w) \\
&= \boldsymbol{\tau}_u^b = \mathbf{A}\boldsymbol{\tau}_u^w,
\end{aligned}
\tag{4.14}
$$

where $\mathbf{S}(\mathbf{A}\boldsymbol{\omega}_{iw}^w)(\mathbf{A}\mathbf{J}_w\boldsymbol{\omega}_{iw}^w) = 0$ since the inertia of the reactions wheels are identical.

4.6 Attitude error dynamics

The quaternion error is defined as (Sola, 2017)

$$
\mathbf{q}_e = \mathbf{q}_d^{-1} \otimes \mathbf{q}_b^o = \begin{bmatrix} \eta_d & -\boldsymbol{\epsilon}_d^\top \\ -\boldsymbol{\epsilon}_d & \eta_d\mathbf{I}_{3\times 3} + \mathbf{S}(\boldsymbol{\epsilon}_d) \end{bmatrix} \mathbf{q}_b^o,
\tag{4.15}
$$

where η_d and $\boldsymbol{\epsilon}_d$ composes the desired quaternion $\mathbf{q}_d = \begin{bmatrix} \eta_d \\ \boldsymbol{\epsilon}_d \end{bmatrix}$.

The angular velocity error is defined as

$$
\boldsymbol{\omega}_e^b = \boldsymbol{\omega}_{ob}^b - \boldsymbol{\omega}_d^b = \boldsymbol{\omega}_{ib}^b - \mathbf{R}_o^b\boldsymbol{\omega}_{io}^o - \boldsymbol{\omega}_d^b,
\tag{4.16}
$$

where $\boldsymbol{\omega}_d^b$ represents the desired angular velocity. Together, the quaternion error and the angular velocity error forms the attitude error variables.

4.7 Total system dynamics

The rate of change in the angular velocity $\boldsymbol{\omega}_{ob}^b$, given in (4.3), can be found by first differentiating the term involving a rotation matrix in (4.3) as follows

$$
\begin{aligned}
\frac{{}^{b}d}{dt}(\mathbf{R}_o^b\boldsymbol{\omega}_{io}^o) &= \left(\frac{{}^{b}d}{dt}\mathbf{R}_o^b\right)\boldsymbol{\omega}_{io}^o + \mathbf{R}_o^b\left(\frac{{}^{b}d}{dt}\boldsymbol{\omega}_{io}^o\right) \\
&= -\mathbf{S}(\boldsymbol{\omega}_{ob}^b)\mathbf{R}_o^b\boldsymbol{\omega}_{io}^o + \mathbf{R}_o^b\frac{{}^{o}d}{dt}(\boldsymbol{\omega}_{io}^o) + \mathbf{R}_o^b\mathbf{S}(\boldsymbol{\omega}_{bo}^o)\boldsymbol{\omega}_{io}^o \\
&= -\mathbf{S}(\boldsymbol{\omega}_{ob}^b)\mathbf{R}_o^b\boldsymbol{\omega}_{io}^o - \mathbf{R}_o^b\mathbf{S}(\boldsymbol{\omega}_{ob}^o)\boldsymbol{\omega}_{io}^o \\
&= -\mathbf{S}(\boldsymbol{\omega}_{ob}^b)\mathbf{R}_o^b\boldsymbol{\omega}_{io}^o - \mathbf{R}_o^b\mathbf{S}(\mathbf{R}_b^o\boldsymbol{\omega}_{ob}^b)\boldsymbol{\omega}_{io}^o,
\end{aligned}
\tag{4.17}
$$

where $\frac{{}^o d}{dt}(\boldsymbol{\omega}^o_{io})=0$ since the orbit is assumed to be constant over time. Then, differentiating (4.3) using (4.17) yields

$$\begin{aligned} \frac{{}^b d}{dt}\boldsymbol{\omega}^b_{ob} &= \frac{{}^b d}{dt}\boldsymbol{\omega}^b_{ib} - \frac{{}^b d}{dt}(\mathbf{R}^b_o\boldsymbol{\omega}^o_{io}) \\ &= \frac{{}^b d}{dt}\boldsymbol{\omega}^b_{ib} + \mathbf{S}(\boldsymbol{\omega}^b_{ob})\mathbf{R}^b_o\boldsymbol{\omega}^o_{io} + \mathbf{R}^b_o\mathbf{S}(\mathbf{R}^o_b\boldsymbol{\omega}^b_{ob})\boldsymbol{\omega}^o_{io}. \end{aligned} \tag{4.18}$$

Rewriting (4.9) using (4.7) and (4.14) gives

$$\begin{aligned} \mathbf{J}\frac{{}^b d}{dt}\boldsymbol{\omega}^b_{ib} &= -\mathbf{A}\mathbf{J}_w\frac{{}^w d}{dt}\boldsymbol{\omega}^w_{bw} - \mathbf{S}(\boldsymbol{\omega}^b_{ib})\mathbf{H}^b_s + \boldsymbol{\tau}^b_{\text{ext}} \Longrightarrow \\ \mathbf{J}_s\frac{{}^b d}{dt}\boldsymbol{\omega}^b_{ib} &= -(\mathbf{A}\mathbf{J}_w\mathbf{A}^\top\frac{{}^b d}{dt}\boldsymbol{\omega}^b_{ib} + \mathbf{A}\mathbf{J}_w\frac{{}^w d}{dt}\boldsymbol{\omega}^w_{bw}) - \mathbf{S}(\boldsymbol{\omega}^b_{ib})\mathbf{H}^b_s + \boldsymbol{\tau}^b_{\text{ext}} \\ &= -\mathbf{A}\boldsymbol{\tau}^w_u - \mathbf{S}(\boldsymbol{\omega}^b_{ib})\mathbf{H}^b_s + \boldsymbol{\tau}^b_{\text{ext}}. \end{aligned} \tag{4.19}$$

Using (4.2), (4.3), (4.14), (4.17), (4.18) and (4.19) the total spacecraft dynamics are found to be

$$\dot{\mathbf{q}}^o_b = \frac{1}{2}\mathbf{T}(\mathbf{q}^o_b)\boldsymbol{\omega}^b_{ob} \tag{4.20a}$$

$$\dot{\boldsymbol{\omega}}^b_{ib} = \mathbf{J}_s{}^{-1}(-\mathbf{A}\boldsymbol{\tau}^w_u - \mathbf{S}(\boldsymbol{\omega}^b_{ib})\mathbf{H}^b_s + \boldsymbol{\tau}^b_{\text{ext}}) \tag{4.20b}$$

$$\dot{\boldsymbol{\omega}}^b_{ob} = \dot{\boldsymbol{\omega}}^b_{ib} + \mathbf{S}(\boldsymbol{\omega}^b_{ob})\mathbf{R}^b_o\boldsymbol{\omega}^o_{io} + \mathbf{R}^b_o\mathbf{S}(\mathbf{R}^o_b\boldsymbol{\omega}^b_{ob})\boldsymbol{\omega}^o_{io} \tag{4.20c}$$

$$\dot{\boldsymbol{\omega}}^w_{bw} = \mathbf{J}_w{}^{-1}\boldsymbol{\tau}^w_u - \mathbf{A}^\top\dot{\boldsymbol{\omega}}^b_{ib}, \tag{4.20d}$$

where $\dot{\boldsymbol{\omega}}^b_{ib} = \frac{{}^b d}{dt}\boldsymbol{\omega}^b_{ib}$, $\dot{\boldsymbol{\omega}}^b_{ob} = \frac{{}^b d}{dt}\boldsymbol{\omega}^b_{ob}$, and $\dot{\boldsymbol{\omega}}^w_{bw} = \frac{{}^b d}{dt}\boldsymbol{\omega}^w_{bw}$.

Chapter 5

Control Design

In this chapter, the designs of four controllers are presented, namely the maximum hands-off controller, the novel moving maximum hands-off controller, the L_1-controller, and the proportional-derivative (PD) controller. The three former controllers are optimal controllers; the maximum hands-off controller and the moving maximum hands-off controller minimize the L_0-norm, whereas the L_1-controller minimizes the L_1-norm. For all controllers, attitude is represented using quaternions.

One of the main contributions of this thesis is to use maximum hands-off control for attitude control of a spacecraft, and the maximum hands-off controller presented in this chapter is designed for this specific purpose. Moreover, the design and implementation of the moving maximum hands-off controller constitute an important contribution. The moving maximum hands-off controller is also designed for the purpose of spacecraft attitude control.

The rest of this chapter is organized as follows. First, the control objectives are presented in Section 5.1, followed by the designs of the controllers in Section 5.2. The experimental cases are presented in Section 5.3, and the simulation setup used for the experiments is presented in Section 5.4. Finally, in Section 5.5, the design choices for the maximum hands-off controller and the moving maximum hands-off controller are discussed, as well as the different parameters used in the optimal control problems.

5.1 Control objective

The control objective of the maximum hands-off controller is to find the sparsest control among all admissible controls, as discussed in Chapter 3. This thesis explores how the maximum hands-off controller works for attitude control of a spacecraft. Note that for the remaining parts of this study, the term *sparsity* is defined accordingly:

Definition 5.1.1 (Sparsity)**.** The *sparsity* of a control signal refers to the number of control intervals for which the control signal takes on nonzero values.

The number of control intervals for which the control signal takes on nonzero values refers to the number of control intervals for which the absolute value of the control signal is larger than a threshold value chosen to be $1 \cdot 10^{-6}$.

Although the maximum hands-off controller yields the sparsest control signal, the control torques might not occur at the most favorable instants of time, which motivates the design of a controller that inherits the maximum hands-off controller's sparsity and lets the user specify at which time intervals the control input should occur. A novel controller, called the moving maximum hands-off controller, has been designed as an extension of the maximum hands-off controller. The term *moving* refers to the characteristic of the controller, which lets the user move the sparse control according to a desired set of preferences, for instance, environmental constraints.

The maximum hands-off controller, the moving maximum hands-off controller, and the L_1-controller are all instances of optimal control. The optimal control problems seek to find the optimal control signals, i.e., an optimal sequence of control torques that move the spacecraft to the desired orientation. The control torques are produced by the spacecraft's reaction wheels. Optimization of the control input could therefore be conducted either directly in the wheel-frame $\{w\}$, or in the body-frame $\{b\}$, and afterward be transformed to $\{w\}$. The attitude of the satellite is given as the orientation of the body-frame $\{b\}$ relative to the orbit-frame $\{o\}$. For the satellite to move to a desired attitude, the body-frame $\{b\}$ has to be rotated relative to the orbit-frame $\{o\}$. Torque is applied in $\{b\}$ and makes the satellite rotate, i.e., $\{b\}$ rotates relative to $\{o\}$. The optimization is carried out in $\{b\}$ because this yields the optimal amount of control torque about each axis necessary to reach the desired orientation. The optimal control input is then transformed from $\{b\}$ to $\{w\}$ using the Moore-Penrose pseudo-inverse of $\mathbf{A}$ so that the desired control input is distributed among the reaction wheels. The conversion between $\{b\}$ and $\{w\}$ are studied in Section 2.3.4.

5.2 Mathematical formulations of the controllers

In this section, the mathematical formulations of the controllers are presented. First, the control design of the maximum hands-off controller for the attitude control problem with attitude represented using quaternions is presented, followed by an introduction to the design and implementation of the moving maximum hands-off controller. Then, the implementation of the L_1-controller is presented, and last, the classical PD-controller is presented. The designs of the maximum hands-off controller, the moving maximum hands-off controller, and the L_1-controller, are based on the theory presented in Chapter 3.

5.2.1 Maximum hands-off controller

The maximum hands-off controller, or the L_0-optimal controller, aims to minimize the L_0-norm of the control input, as discussed in Chapter 3. The mathematical preliminaries necessary to understand the controller, and the various reformulations of the L_0-control problem, are presented in Chapter 3. The maximum hands-off controller for the spacecraft attitude control problem is given as

$$
\begin{aligned}
\underset{\boldsymbol{\tau}_u^b}{\text{minimize}} \quad & k_1 f(\boldsymbol{\omega}_{ob}^b) + k_2 g(\mathbf{q}_b^o) + k_3 \|\boldsymbol{\tau}_u^b\|_0 && \text{(5.1a)} \\
\text{subject to} \quad \dot{\mathbf{q}}_b^o &= \frac{1}{2}\mathbf{T}(\mathbf{q}_b^o)\boldsymbol{\omega}_{ob}^b && \text{(5.1b)} \\
\dot{\boldsymbol{\omega}}_{ib}^b &= \mathbf{J}_s^{\text{-1}}(-\mathbf{A}\boldsymbol{\tau}_u^w - \mathbf{S}(\boldsymbol{\omega}_{ib}^b)\mathbf{H}_s^b + \boldsymbol{\tau}_{\text{ext}}^b) && \text{(5.1c)} \\
\dot{\boldsymbol{\omega}}_{bw}^w &= \mathbf{J}_w^{\text{-1}}\boldsymbol{\tau}_u^w - \mathbf{A}^\top \dot{\boldsymbol{\omega}}_{ib}^b && \text{(5.1d)} \\
\dot{\boldsymbol{\omega}}_{ob}^b &= \dot{\boldsymbol{\omega}}_{ib}^b + \mathbf{S}(\boldsymbol{\omega}_{ob}^b)\mathbf{R}_o^b\boldsymbol{\omega}_{io}^o + \mathbf{R}_o^b\mathbf{S}(\mathbf{R}_b^o\boldsymbol{\omega}_{ob}^b)\boldsymbol{\omega}_{io}^o && \text{(5.1e)} \\
\dot{\mathbf{r}}^i &= \mathbf{v}^i && \text{(5.1f)} \\
\dot{\mathbf{v}}^i &= -\frac{\mu}{\|\mathbf{r}^i\|_2^3}\mathbf{r}^i && \text{(5.1g)} \\
\boldsymbol{\tau}_u^w &\leq \boldsymbol{\tau}_{\text{limit}} && \text{(5.1h)} \\
\boldsymbol{\tau}_u^w &\geq -\boldsymbol{\tau}_{\text{limit}} && \text{(5.1i)} \\
\mathbf{q}_b^o(0) &= \mathbf{q}_{b,0}^o && \text{(5.1j)} \\
\boldsymbol{\omega}_{ib}^b(0) &= \boldsymbol{\omega}_{ib,0}^b && \text{(5.1k)} \\
\boldsymbol{\omega}_{bw}^w(0) &= \boldsymbol{\omega}_{bw,0}^w && \text{(5.1l)} \\
\boldsymbol{\omega}_{ob}^b(0) &= \boldsymbol{\omega}_{ob,0}^b && \text{(5.1m)} \\
\mathbf{r}^i(0) &= \mathbf{r}_0^i && \text{(5.1n)} \\
\mathbf{v}^i(0) &= \mathbf{v}_0^i, && \text{(5.1o)}
\end{aligned}
$$

where k_1, k_2, and k_3 are positive constants, $\boldsymbol{\tau}_u^b$ is the vector of control inputs in $\{b\}$, $\boldsymbol{\tau}_u^w$ is the vector of control inputs in $\{w\}$, and $\boldsymbol{\tau}_{\text{limit}}$ is the torque limit on the reaction wheels. The torque vectors $\boldsymbol{\tau}_u^b$ and $\boldsymbol{\tau}_u^w$ are related through (2.42). The kinematic equation, (5.1b), and the dynamic equations, (5.1c) to (5.1e), are derived in Chapter 4, and $\mathbf{q}_{b,0}^o$, $\boldsymbol{\omega}_{ib,0}^b$, $\boldsymbol{\omega}_{bw,0}^w$, and $\boldsymbol{\omega}_{ob,0}^b$ are the state initial values. (5.1f) and (5.1g) denotes the orbital dynamics derived in Section 4.2, and $\mathbf{r}_0^i$ and $\mathbf{v}_0^i$ are the initial values for (5.1f) and (5.1g). The functions $f(\cdot)$ and $g(\cdot)$ are designed to steer $\boldsymbol{\omega}_{ob}^b$ and $\mathbf{q}_b^o$, respectively, to their desired final states. More specifically,

$$\begin{aligned} f(\boldsymbol{\omega}_{ob}^{b}) &= \sum_{i=1}^{n} (\omega_{ob,i}^{b}(T) - \omega_{ob,\mathrm{ref},i}^{b})^2 \\ g(\mathbf{q}_b^o) &= 1 - \left|(\mathbf{q}_b^o(T))^\top \mathbf{q}_{b,\mathrm{ref}}^o\right| , \end{aligned} \tag{5.2}$$

where T denotes the final time, $\boldsymbol{\omega}_{ob,\mathrm{ref}}^{b}$ and $\mathbf{q}_{b,\mathrm{ref}}^o$ are the reference angular velocity and reference quaternion, n denotes the number of entries in $\boldsymbol{\omega}_{ob}^{b}(T)$ and $\boldsymbol{\omega}_{ob,\mathrm{ref}}^{b}$, while $\omega_{ob,i}^{b}(T)$ and $\omega_{ob,\mathrm{ref},i}^{b}$ denotes the i-th component of $\boldsymbol{\omega}_{ob}^{b}(T)$ and $\boldsymbol{\omega}_{ob,\mathrm{ref}}^{b}$, respectively. The function $f(\cdot)$ yields the sum of the squared elements of a vector, and is implemented using the SUMSQR-function in MATLAB (MATLAB, 2020). The function $g(\cdot)$ is a pseudometric on the unit quaternion, but a metric on $SO(3)$ (Huynh, 2009). The absolute value, $|(\mathbf{q}_b^o(T))^\top \mathbf{q}_{b,\mathrm{ref}}^o|$, is implemented using the smooth maximum function in (2.48) (Kristiansen et al., 2021).

The L_0-cost function is discontinuous and nonconvex, and solving optimization problems with these characteristics are generally hard, as discussed in Section 3.3. Therefore, the maximum hands-off control problem in (5.1) is rewritten according to the L_0-controller given in Feng et al. (2016), using the relaxed formulation presented in (3.32) in Section 3.6.4. This formulation is different from the one used in the work by Nagahara et al. (2015), where the L_1-norm was used as an approximation to the L_0-norm. The reformulation of (5.1) results in the following minimization problem

$$\underset{\boldsymbol{\tau}^b_u, \boldsymbol{\xi}}{\text{minimize}} \quad k_1 f(\boldsymbol{\omega}^b_{ob}) + k_2 g(\mathbf{q}^o_b) + k_3(\mathbf{1}_N - \boldsymbol{\xi})\mathbf{1}_N^\top \tag{5.3a}$$

$$\text{subject to} \quad \dot{\mathbf{q}}^o_b = \frac{1}{2}\mathbf{T}(\mathbf{q}^o_b)\boldsymbol{\omega}^b_{ob} \tag{5.3b}$$

$$\dot{\boldsymbol{\omega}}^b_{ib} = \mathbf{J}_s{}^{\text{-1}}(-\mathbf{A}\boldsymbol{\tau}^w_u - \mathbf{S}(\boldsymbol{\omega}^b_{ib})\mathbf{H}^b_s + \boldsymbol{\tau}^b_{\text{ext}}) \tag{5.3c}$$

$$\dot{\boldsymbol{\omega}}^w_{bw} = \mathbf{J}_w{}^{\text{-1}}\boldsymbol{\tau}^w_u - \mathbf{A}^\top\dot{\boldsymbol{\omega}}^b_{ib} \tag{5.3d}$$

$$\dot{\boldsymbol{\omega}}^b_{ob} = \dot{\boldsymbol{\omega}}^b_{ib} + \mathbf{S}(\boldsymbol{\omega}^b_{ob})\mathbf{R}^b_o\boldsymbol{\omega}^o_{io} + \mathbf{R}^b_o\mathbf{S}(\mathbf{R}^o_b\boldsymbol{\omega}^b_{ob})\boldsymbol{\omega}^o_{io} \tag{5.3e}$$

$$\dot{\mathbf{r}}^i = \mathbf{v}^i \tag{5.3f}$$

$$\dot{\mathbf{v}}^i = -\frac{\mu}{\|\mathbf{r}^i\|_2^3}\mathbf{r}^i \tag{5.3g}$$

$$\boldsymbol{\tau}^w_u \leq \boldsymbol{\tau}_{\text{limit}} \tag{5.3h}$$

$$\boldsymbol{\tau}^w_u \geq -\boldsymbol{\tau}_{\text{limit}} \tag{5.3i}$$

$$\mathbf{q}^o_b(0) = \mathbf{q}^o_{b,0} \tag{5.3j}$$

$$\boldsymbol{\omega}^b_{ib}(0) = \boldsymbol{\omega}^b_{ib,0} \tag{5.3k}$$

$$\boldsymbol{\omega}^w_{bw}(0) = \boldsymbol{\omega}^w_{bw,0} \tag{5.3l}$$

$$\boldsymbol{\omega}^b_{ob}(0) = \boldsymbol{\omega}^b_{ob,0} \tag{5.3m}$$

$$\mathbf{r}^i(0) = \mathbf{r}^i_0 \tag{5.3n}$$

$$\mathbf{v}^i(0) = \mathbf{v}^i_0 \tag{5.3o}$$

$$\boldsymbol{\xi} \leq \mathbf{1}_N \tag{5.3p}$$

$$\boldsymbol{\xi} \circ \boldsymbol{\tau}^b_{u,1} \leq \epsilon\mathbf{1}_N \tag{5.3q}$$

$$-\boldsymbol{\xi} \circ \boldsymbol{\tau}^b_{u,1} \leq \epsilon\mathbf{1}_N \tag{5.3r}$$

$$\boldsymbol{\xi} \circ \boldsymbol{\tau}^b_{u,2} \leq \epsilon\mathbf{1}_N \tag{5.3s}$$

$$-\boldsymbol{\xi} \circ \boldsymbol{\tau}^b_{u,2} \leq \epsilon\mathbf{1}_N \tag{5.3t}$$

$$\boldsymbol{\xi} \circ \boldsymbol{\tau}^b_{u,3} \leq \epsilon\mathbf{1}_N \tag{5.3u}$$

$$-\boldsymbol{\xi} \circ \boldsymbol{\tau}^b_{u,3} \leq \epsilon\mathbf{1}_N \tag{5.3v}$$

$$\boldsymbol{\xi} \geq \mathbf{0}, \tag{5.3w}$$

where ϵ is a positive constant, $\boldsymbol{\tau}^b_{u,1}$, $\boldsymbol{\tau}^b_{u,2}$, and $\boldsymbol{\tau}^b_{u,3}$ denotes the components of $\boldsymbol{\tau}^b_u$ about the x-, y- and z-axis in $\{b\}$, respectively, and $\boldsymbol{\tau}^w_u$ is the vector of control inputs in $\{w\}$. The torque vectors $\boldsymbol{\tau}^b_u$ and $\boldsymbol{\tau}^w_u$ are related through (2.42). $\mathbf{1}_N$ is the N-dimensional vector of all ones, where N is the number of control intervals, $\boldsymbol{\xi}$ is the complementarity vector to the control input $\boldsymbol{\tau}^b_u$, and $\mathbf{1}_N - \boldsymbol{\xi}$ is the support vector of $\boldsymbol{\tau}^b_u$. The concepts complementarity vector and support vector was presented in Section 3.6. The notation $\mathbf{a} \circ \mathbf{b}$ denotes the componentwise product between the vectors $\mathbf{a}$ and $\mathbf{b}$.

The complementarity constraints in (5.3q) to (5.3v) imply that either ξ_i or $\tau^b_{u,1,i}$, or both, must be 0. Here, ξ_i and $\tau^b_{u,1,i}$ denotes the components of the vectors $\boldsymbol{\xi}$ and $\boldsymbol{\tau}^b_{u,1}$, respectively, and $\boldsymbol{\tau}^b_{u,1}$ is chosen as an example; the complementarity constraints imply the same for all components of $\boldsymbol{\tau}^b_u$, i.e., they imply the same for $\boldsymbol{\tau}^b_{u,2}$ and $\boldsymbol{\tau}^b_{u,3}$ as well. The complementarity constraints in (5.3) were implemented using the same complementarity vector $\boldsymbol{\xi}$ for all three components of $\boldsymbol{\tau}^b_u$. The reason for this design choice was to ensure that control input about the x-, y- and z-axis in $\{b\}$ could occur at the same time instants without extra cost. The positive constant ϵ in (5.3q) to (5.3v) imposes slack on the complementarity constraints, in the sense that neither ξ_i nor $\tau^b_{u,1,i}$ (or $\tau^b_{u,2,i}$ or $\tau^b_{u,3,i}$) have to be exactly 0; they have to be close to 0 depending on the value of ϵ.

Note that for the results and discussion parts in Chapter 6, the maximum hands-off controller will be referred to as the L_0-controller or the L_0-optimal controller.

5.2.2 Moving maximum hands-off controller

The moving maximum hands-off controller inherits the optimal sparsity from the maximum hands-off controller and also lets the user specify in which time interval the control should occur. To put it more precisely, the moving maximum hands-off controller aims to find the sparsest control among all admissible controls based on the preferences of the user.

The moving maximum hands-off controller lets the user specify in which time interval the control should occur and aims to find the sparsest control among all admissible controls based on this information.

Considering attitude control of a spacecraft, it could, for example, be interesting to perform a maneuver, which is as sparse as possible, i.e., provide the sparsest control input, within time intervals for which it is possible to observe the satellite from a specific location on the Earth.

An advantage of the moving maximum hands-off controller is the opportunity to apply control within a user specified time interval and observe if the system responds as expected. If the system does not respond as expected within the time interval specified by the user, i.e., the system differs from the model, or the actuators do not work as expected, it could be possible to correct the system response by applying new control torque.

One disadvantage related to the novel controller is that it might require more energy than the maximum hands-off controller, which yields the L_0-optimal control. The user of the moving maximum hands-off controller may specify that the control torques should occur within a time interval of, for instance, $\Delta t = 30$ s. Contrary, for the maximum hands-off controller, the control torques may occur anywhere on the total time interval, specified by the simulation time of, for instance, $\Delta t = 100$ s. One could imagine that a control torque must be applied at two time instants to steer the spacecraft to the desired attitude;

one control torque pushes the spacecraft towards the desired attitude, and one control input stops the spacecraft's rotation. To reach the desired attitude within a short time interval, the control torques applied to the spacecraft have to be larger than for a longer time interval. The control torques could be smaller for a larger time interval as the spacecraft would have more time to rotate towards the desired orientation after the initial control input has been applied. It requires more energy to produce a large control torque than a small control torque. Therefore, it might be possible that the moving maximum hands-off controller uses more energy than the maximum hands-off controller.

The design of the moving maximum hands-off controller is similar to that of the maximum hands-off controller in (5.1), and it is formulated as

$$\underset{\boldsymbol{\tau}_u^b}{\text{minimize}} \quad k_1 f(\boldsymbol{\omega}_{ob}^b) + k_2 g(\mathbf{q}_b^o) + k_3 h(t) \|\boldsymbol{\tau}_u^b\|_0 \tag{5.4a}$$

$$\text{subject to} \quad \dot{\mathbf{q}}_b^o = \frac{1}{2}\mathbf{T}(\mathbf{q}_b^o)\boldsymbol{\omega}_{ob}^b \tag{5.4b}$$

$$\dot{\boldsymbol{\omega}}_{ib}^b = \mathbf{J}_s{}^{\text{-1}}(-\mathbf{A}\boldsymbol{\tau}_u^w - \mathbf{S}(\boldsymbol{\omega}_{ib}^b)\mathbf{H}_s^b + \boldsymbol{\tau}_{\text{ext}}^b) \tag{5.4c}$$

$$\dot{\boldsymbol{\omega}}_{bw}^w = \mathbf{J}_w{}^{\text{-1}}\boldsymbol{\tau}_u^w - \mathbf{A}^\top \dot{\boldsymbol{\omega}}_{ib}^b \tag{5.4d}$$

$$\dot{\boldsymbol{\omega}}_{ob}^b = \dot{\boldsymbol{\omega}}_{ib}^b + \mathbf{S}(\boldsymbol{\omega}_{ob}^b)\mathbf{R}_o^b\boldsymbol{\omega}_{io}^o + \mathbf{R}_o^b\mathbf{S}(\mathbf{R}_b^o\boldsymbol{\omega}_{ob}^b)\boldsymbol{\omega}_{io}^o \tag{5.4e}$$

$$\dot{\mathbf{r}}^i = \mathbf{v}^i \tag{5.4f}$$

$$\dot{\mathbf{v}}^i = -\frac{\mu}{\|\mathbf{r}^i\|_2^3}\mathbf{r}^i \tag{5.4g}$$

$$\boldsymbol{\tau}_u^w \leq \boldsymbol{\tau}_{\text{limit}} \tag{5.4h}$$

$$\boldsymbol{\tau}_u^w \geq -\boldsymbol{\tau}_{\text{limit}} \tag{5.4i}$$

$$\mathbf{q}_b^o(0) = \mathbf{q}_{b,0}^o \tag{5.4j}$$

$$\boldsymbol{\omega}_{ib}^b(0) = \boldsymbol{\omega}_{ib,0}^b \tag{5.4k}$$

$$\boldsymbol{\omega}_{bw}^w(0) = \boldsymbol{\omega}_{bw,0}^w \tag{5.4l}$$

$$\boldsymbol{\omega}_{ob}^b(0) = \boldsymbol{\omega}_{ob,0}^b \tag{5.4m}$$

$$\mathbf{r}^i(0) = \mathbf{r}_0^i \tag{5.4n}$$

$$\mathbf{v}^i(0) = \mathbf{v}_0^i, \tag{5.4o}$$

where k_1, k_2, and k_3 are positive constants, $\boldsymbol{\tau}_u^b$ is the vector of control inputs in $\{b\}$, $\boldsymbol{\tau}_u^w$ is the vector of control inputs in $\{w\}$, and $\boldsymbol{\tau}_{\text{limit}}$ is the torque limit on the reaction wheels. The torque vectors $\boldsymbol{\tau}_u^b$ and $\boldsymbol{\tau}_u^w$ are related through (2.42). The kinematic equation, (5.4b), and the dynamic equations, (5.4c) to (5.4e) are derived in Chapter 4, and $\mathbf{q}_{b,0}^o$, $\boldsymbol{\omega}_{ib,0}^b$, $\boldsymbol{\omega}_{bw,0}^w$, and $\boldsymbol{\omega}_{ob,0}^b$ are the state initial values. The functions $f(\cdot)$ and $g(\cdot)$ are given in (5.2), and the function $h(\cdot)$ specifies where the control torques should occur. In a similar way as for the maximum hands-off controller, the moving maximum hands-off controller in (5.4) is reformulated as

$$
\begin{align}
\underset{\boldsymbol{\tau}^b_u,\boldsymbol{\xi}}{\text{minimize}} \quad & k_1 f(\boldsymbol{\omega}^b_{ob}) + k_2 g(\mathbf{q}^o_b) + k_3(\mathbf{1}_N - \boldsymbol{\xi} \circ \mathbf{h}_N)\mathbf{1}_N^\top \tag{5.5a} \\
\text{subject to} \quad \dot{\mathbf{q}}^o_b &= \frac{1}{2}\mathbf{T}(\mathbf{q}^o_b)\boldsymbol{\omega}^b_{ob} \tag{5.5b} \\
\dot{\boldsymbol{\omega}}^b_{ib} &= \mathbf{J}_s{}^{\text{-}1}(-\mathbf{A}\boldsymbol{\tau}^w_u - \mathbf{S}(\boldsymbol{\omega}^b_{ib})\mathbf{H}^b_s + \boldsymbol{\tau}^b_{\text{ext}}) \tag{5.5c} \\
\dot{\boldsymbol{\omega}}^w_{bw} &= \mathbf{J}_w{}^{\text{-}1}\boldsymbol{\tau}^w_u - \mathbf{A}^\top\dot{\boldsymbol{\omega}}^b_{ib} \tag{5.5d} \\
\dot{\boldsymbol{\omega}}^b_{ob} &= \dot{\boldsymbol{\omega}}^b_{ib} + \mathbf{S}(\boldsymbol{\omega}^b_{ob})\mathbf{R}^b_o\boldsymbol{\omega}^o_{io} + \mathbf{R}^b_o\mathbf{S}(\mathbf{R}^o_b\boldsymbol{\omega}^b_{ob})\boldsymbol{\omega}^o_{io} \tag{5.5e} \\
\dot{\mathbf{r}}^i &= \mathbf{v}^i \tag{5.5f} \\
\dot{\mathbf{v}}^i &= -\frac{\mu}{\|\mathbf{r}^i\|_2^3}\mathbf{r}^i \tag{5.5g} \\
\boldsymbol{\tau}^w_u &\leq \boldsymbol{\tau}_{\text{limit}} \tag{5.5h} \\
\boldsymbol{\tau}^w_u &\geq -\boldsymbol{\tau}_{\text{limit}} \tag{5.5i} \\
\mathbf{q}^o_b(0) &= \mathbf{q}^o_{b,0} \tag{5.5j} \\
\boldsymbol{\omega}^b_{ib}(0) &= \boldsymbol{\omega}^b_{ib,0} \tag{5.5k} \\
\boldsymbol{\omega}^w_{bw}(0) &= \boldsymbol{\omega}^w_{bw,0} \tag{5.5l} \\
\boldsymbol{\omega}^b_{ob}(0) &= \boldsymbol{\omega}^b_{ob,0} \tag{5.5m} \\
\mathbf{r}^i(0) &= \mathbf{r}^i_0 \tag{5.5n} \\
\mathbf{v}^i(0) &= \mathbf{v}^i_0 \tag{5.5o} \\
\boldsymbol{\xi} &\leq \mathbf{1}_N \tag{5.5p} \\
\boldsymbol{\xi} \circ \boldsymbol{\tau}^b_{u,1} &\leq \epsilon\mathbf{1}_N \tag{5.5q} \\
-\boldsymbol{\xi} \circ \boldsymbol{\tau}^b_{u,1} &\leq \epsilon\mathbf{1}_N \tag{5.5r} \\
\boldsymbol{\xi} \circ \boldsymbol{\tau}^b_{u,2} &\leq \epsilon\mathbf{1}_N \tag{5.5s} \\
-\boldsymbol{\xi} \circ \boldsymbol{\tau}^b_{u,2} &\leq \epsilon\mathbf{1}_N \tag{5.5t} \\
\boldsymbol{\xi} \circ \boldsymbol{\tau}^b_{u,3} &\leq \epsilon\mathbf{1}_N \tag{5.5u} \\
-\boldsymbol{\xi} \circ \boldsymbol{\tau}^b_{u,3} &\leq \epsilon\mathbf{1}_N \tag{5.5v} \\
\boldsymbol{\xi} &\geq \mathbf{0}, \tag{5.5w}
\end{align}
$$

where ϵ is a positive constant, $\boldsymbol{\tau}^b_{u,1}$, $\boldsymbol{\tau}^b_{u,2}$, and $\boldsymbol{\tau}^b_{u,3}$ denotes the components of $\boldsymbol{\tau}^b_u$ about the x-, y- and z-axis in $\{b\}$, respectively, and $\boldsymbol{\tau}^w_u$ is the vector of control inputs in $\{w\}$. The torque vectors $\boldsymbol{\tau}^b_u$ and $\boldsymbol{\tau}^w_u$ are related through (2.42). $\mathbf{1}_N$ is the N-dimensional vector of all ones, where N is the number of control intervals. $\boldsymbol{\xi}$ is the complementarity vector to the control input $\boldsymbol{\tau}^b_u$, and $\mathbf{1}_N - \boldsymbol{\xi}$ is the support vector of $\boldsymbol{\tau}^b_u$. The N-dimensional vector $\mathbf{h}_N$ represents the function $h(\cdot)$ in (5.4) as a function of control intervals instead of time. By default, the vector $\mathbf{h}_N$ is a vector of all ones. The user might change the values of $\mathbf{h}_N$ to values between 0 and 1, to indicate for which time intervals the control input should occur. For instance, if the value of $\mathbf{h}_N$ is set to 0.1 for $N = 10, \ldots, 20$, it could yield a more optimal solution if the control occur between these control intervals as sparsity comes at a lower cost in this interval than the rest.

Note that for the results and discussion parts in Chapter 6, the moving maximum hands-off controller will be referred to as the moving L_0-controller or the moving L_0-optimal controller.

5.2.3 L_1-optimal controller

The L_1-optimal controller is implemented according to the theory presented in Section 3.4. The design of the L_1-optimal control problem is similar to that of the maximum hands-off control problem in (5.1), except that the L_0-norm is replaced by the L_1-norm which yields the following design

$$
\begin{aligned}
\underset{\boldsymbol{\tau}_u^b}{\text{minimize}} \quad & k_1 f(\boldsymbol{\omega}_{ob}^b) + k_2 g(\mathbf{q}_b^o) + k_3 \|\boldsymbol{\tau}_u^b\|_1 && \text{(5.6a)}\\
\text{subject to} \quad & \dot{\mathbf{q}}_b^o = \frac{1}{2}\mathbf{T}(\mathbf{q}_b^o)\boldsymbol{\omega}_{ob}^b && \text{(5.6b)}\\
& \dot{\boldsymbol{\omega}}_{ib}^b = \mathbf{J}_s^{-1}(-\mathbf{A}\boldsymbol{\tau}_u^w - \mathbf{S}(\boldsymbol{\omega}_{ib}^b)\mathbf{H}_s^b + \boldsymbol{\tau}_{\text{ext}}^b) && \text{(5.6c)}\\
& \dot{\boldsymbol{\omega}}_{bw}^w = \mathbf{J}_w^{-1}\boldsymbol{\tau}_u^w - \mathbf{A}^\top \dot{\boldsymbol{\omega}}_{ib}^b && \text{(5.6d)}\\
& \dot{\boldsymbol{\omega}}_{ob}^b = \dot{\boldsymbol{\omega}}_{ib}^b + \mathbf{S}(\boldsymbol{\omega}_{ob}^b)\mathbf{R}_o^b\boldsymbol{\omega}_{io}^o + \mathbf{R}_o^b\mathbf{S}(\mathbf{R}_b^o\boldsymbol{\omega}_{ob}^b)\boldsymbol{\omega}_{io}^o && \text{(5.6e)}\\
& \dot{\mathbf{r}}^i = \mathbf{v}^i && \text{(5.6f)}\\
& \dot{\mathbf{v}}^i = -\frac{\mu}{\|\mathbf{r}^i\|_2^3}\mathbf{r}^i && \text{(5.6g)}\\
& \boldsymbol{\tau}_u^w \leq \boldsymbol{\tau}_{\text{limit}} && \text{(5.6h)}\\
& \boldsymbol{\tau}_u^w \geq -\boldsymbol{\tau}_{\text{limit}} && \text{(5.6i)}\\
& \mathbf{q}_b^o(0) = \mathbf{q}_{b,0}^o && \text{(5.6j)}\\
& \boldsymbol{\omega}_{ib}^b(0) = \boldsymbol{\omega}_{ib,0}^b && \text{(5.6k)}\\
& \boldsymbol{\omega}_{bw}^w(0) = \boldsymbol{\omega}_{bw,0}^w && \text{(5.6l)}\\
& \boldsymbol{\omega}_{ob}^b(0) = \boldsymbol{\omega}_{ob,0}^b && \text{(5.6m)}\\
& \mathbf{r}^i(0) = \mathbf{r}_0^i && \text{(5.6n)}\\
& \mathbf{v}^i(0) = \mathbf{v}_0^i, && \text{(5.6o)}
\end{aligned}
$$

where k_1, k_2, and k_3 are positive constants, $\boldsymbol{\tau}_u^b$ is the vector of control inputs in $\{b\}$, $\boldsymbol{\tau}_u^w$ is the vector of control inputs in $\{w\}$, and $\boldsymbol{\tau}_{\text{limit}}$ is the torque limit on the reaction wheels. The torque vectors $\boldsymbol{\tau}_u^b$ and $\boldsymbol{\tau}_u^w$ are related through (2.42). The kinematic equation, (5.6b), and the dynamic equations, (5.6c) to (5.6e) are derived in Chapter 4, and $\mathbf{q}_{b,0}^o$, $\boldsymbol{\omega}_{ib,0}^b$, $\boldsymbol{\omega}_{bw,0}^w$, and $\boldsymbol{\omega}_{ob,0}^b$ are the state initial values. The functions $f(\cdot)$ and $g(\cdot)$ are given in (5.2).

Using the approach based on Gros and Diehl (2019) and presented in Section 3.4, the L_1-optimal control problem is reformulated as follows

$$\underset{s,\boldsymbol{\tau}_u^b}{\text{minimize}} \quad k_1 f(\boldsymbol{\omega}_{ob}^b) + k_2 g(\mathbf{q}_b^o) + k_3 \sum_{i=0}^{N} s_k \tag{5.7a}$$

$$\text{subject to} \quad \dot{\mathbf{q}}_b^o = \frac{1}{2}\mathbf{T}(\mathbf{q}_b^o)\boldsymbol{\omega}_{ob}^b \tag{5.7b}$$

$$\dot{\boldsymbol{\omega}}_{ib}^b = \mathbf{J}_s^{\text{-1}}(-\mathbf{A}\boldsymbol{\tau}_u^w - \mathbf{S}(\boldsymbol{\omega}_{ib}^b)\mathbf{H}_s^b + \boldsymbol{\tau}_{\text{ext}}^b) \tag{5.7c}$$

$$\dot{\boldsymbol{\omega}}_{bw}^w = \mathbf{J}_w^{\text{-1}}\boldsymbol{\tau}_u^w - \mathbf{A}^\top\dot{\boldsymbol{\omega}}_{ib}^b \tag{5.7d}$$

$$\dot{\boldsymbol{\omega}}_{ob}^b = \dot{\boldsymbol{\omega}}_{ib}^b + \mathbf{S}(\boldsymbol{\omega}_{ob}^b)\mathbf{R}_o^b\boldsymbol{\omega}_{io}^o + \mathbf{R}_o^b\mathbf{S}(\mathbf{R}_b^o\boldsymbol{\omega}_{ob}^b)\boldsymbol{\omega}_{io}^o \tag{5.7e}$$

$$\dot{\mathbf{r}}^i = \mathbf{v}^i \tag{5.7f}$$

$$\dot{\mathbf{v}}^i = -\frac{\mu}{\|\mathbf{r}^i\|_2^3}\mathbf{r}^i \tag{5.7g}$$

$$\boldsymbol{\tau}_u^w \leq \boldsymbol{\tau}_{\text{limit}} \tag{5.7h}$$

$$\boldsymbol{\tau}_u^w \geq -\boldsymbol{\tau}_{\text{limit}} \tag{5.7i}$$

$$\mathbf{q}_b^o(0) = \mathbf{q}_{b,0}^o \tag{5.7j}$$

$$\boldsymbol{\omega}_{ib}^b(0) = \boldsymbol{\omega}_{ib,0}^b \tag{5.7k}$$

$$\boldsymbol{\omega}_{bw}^w(0) = \boldsymbol{\omega}_{bw,0}^w \tag{5.7l}$$

$$\boldsymbol{\omega}_{ob}^b(0) = \boldsymbol{\omega}_{ob,0}^b \tag{5.7m}$$

$$\mathbf{r}^i(0) = \mathbf{r}_0^i \tag{5.7n}$$

$$\mathbf{v}^i(0) = \mathbf{v}_0^i \tag{5.7o}$$

$$-\mathrm{s}_N \leq \boldsymbol{\tau}_{u,1}^b \leq \mathrm{s}_N \tag{5.7p}$$

$$-\mathrm{s}_N \leq \boldsymbol{\tau}_{u,2}^b \leq \mathrm{s}_N \tag{5.7q}$$

$$-\mathrm{s}_N \leq \boldsymbol{\tau}_{u,3}^b \leq \mathrm{s}_N, \tag{5.7r}$$

where ϵ is a positive constant, $\boldsymbol{\tau}_{u,1}^b$, $\boldsymbol{\tau}_{u,2}^b$, and $\boldsymbol{\tau}_{u,3}^b$ denotes the components of $\boldsymbol{\tau}_u^b$ about the x-, y- and z-axis in $\{b\}$, respectively, and $\boldsymbol{\tau}_u^w$ is the vector of control inputs in $\{w\}$. The torque vectors $\boldsymbol{\tau}_u^b$ and $\boldsymbol{\tau}_u^w$ are related through (2.42). The N-vector of slack variables is denoted by s_N, where N is the number of control intervals, and s_k denotes each slack variable in the slack vector.

5.2.4 Proportional-derivative (PD) controller

A PD-controller was implemented to test if the satellite dynamics worked as intended. The PD-controller is defined in Wen and Kreutz-Delgado (1991)

$$\boldsymbol{\tau}_u^b = \mathbf{K}_d\boldsymbol{\omega}_e^b + \mathbf{K}_p\boldsymbol{\epsilon}_e, \tag{5.8}$$

where $\boldsymbol{\omega}_e^b$ is the angular velocity error, given in (4.16), and $\boldsymbol{\epsilon}_e$ is the vector part of the error quaternion given in (4.15). $\mathbf{K}_d$ and $\mathbf{K}_p$ are constant and positive definite controller gain matrices. The numerical values of the gain matrices were chosen while tuning the PD-controller.

5.3 Experimental cases

The performances of the maximum hands-off controller, the moving maximum hands-off controller, and the L_1-optimal controller were tested and compared through the following experiments based on the simulations of a 6U CubeSat. First, a single-axis maneuver was tested, i.e., the spacecraft orients towards a setpoint involving rotation about one single axis in the body-frame $\{b\}$. Next, an experiment involving rotation about all three axes in $\{b\}$, a multiple-axis maneuver, was conducted. Then, the spacecraft controllers' abilities to perform a path-following maneuver were investigated, and finally, the performances of the controllers when there are known perturbations present in the system, were tested. All experiments were conducted in a simulation environment, which is explored in Section 5.4. Note that rotations about the body-frame axes are given as an angle ϕ about the x-axis, an angle θ about the y-axis, and an angle ψ about the z-axis. All rotations are given in degrees [° deg]. For instance, a maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$ means that the spacecraft rotates an angle $\phi = 90°$ about the x-axis, an angle $\theta = 45°$ about the y-axis, and an angle $\psi = 15°$ about the z-axis.

5.3.1 Single-axis maneuver

The single-axis maneuver tests the controllers' abilities to steer the spacecraft to a desired position which involves rotation about one single axis in the body-frame $\{b\}$. The spacecraft was rotated from an initial position $(\phi = 0°, \theta = 0°, \psi = 0°)$ to a final position $(45°, 0°, 0°)$, which means a rotation of an angle $\phi = 45°$ about the x-axis. For the moving maximum hands-off controller, the vector $\mathbf{h}_N$ was chosen such that it is cheaper for control input to occur between $N = 20$ and $N = 30$, which corresponds to $t = 28$ s and $t = 42$ s, respectively. The cost was set to 0.001 within this interval. The cost outside the interval specified by $\mathbf{h}_N$ is by default 1.

5.3.2 Multiple-axis maneuver

Two multiple-axis maneuvers were tested; one maneuver rotates the spacecraft from an initial position $(0°, 0°, 0°)$ to a final position $(90°, 45°, 15°)$; the second maneuver rotates the spacecraft from $(0°, 0°, 0°)$ to $(75°, 50°, 15°)$. For the moving maximum hands-off controller, the vector $\mathbf{h}_N$ was chosen such that it is cheaper for control input to occur between $N = 20$ and $N = 30$, which corresponds to $t = 28$ s and $t = 42$ s, respectively. The cost was set to 0.001 within this interval. The cost outside the interval specified by $\mathbf{h}_N$ is by default 1.

5.3.3 Path-following maneuver

The path-following experiments rotate the spacecraft from an initial orientation $(0^\circ, 0^\circ, 0^\circ)$ through $(30^\circ, 45^\circ, 15^\circ)$ and back to $(0^\circ, 0^\circ, 0^\circ)$.

For the path-following maneuver, two different approaches were tested. The first approach iterates through a list of setpoints and computes the optimal control signal from one setpoint to the next. The second method utilizes slightly modified versions of the cost functions (5.3a), (5.5a) and (5.7a) including an additional quaternion term. The cost functions are modified as follows

$$J_{L0} = \quad a_1 f(\boldsymbol{\omega}_{ob}^b) + a_2 g_1(\mathbf{q}_b^o) + +a_3 g_2(\mathbf{q}_b^o) + a_4(\mathbf{1}_N - \boldsymbol{\xi})\mathbf{1}_N^\top \tag{5.9a}$$

$$J_{mov,L0} = a_1 f(\boldsymbol{\omega}_{ob}^b) + a_2 g_1(\mathbf{q}_b^o) + +a_3 g_2(\mathbf{q}_b^o) + a_4(\mathbf{1}_N - \boldsymbol{\xi} \circ \mathbf{h}_N)\mathbf{1}_N^\top \tag{5.9b}$$

$$J_{L1} = a_1 f(\boldsymbol{\omega}_{ob}^b) + a_2 g_1(\mathbf{q}_b^o) + +a_3 g_2(\mathbf{q}_b^o) + a_4 \sum_{i=0}^{N} s_k, \tag{5.9c}$$

where (5.9a) is the modified cost function for the maximum hands-off controller, (5.9b) is the modified cost function for the moving maximum hands-off controller, and (5.9c) is the modified cost function for the L_1-controller. a_1, a_2, a_3, and a_4 are positive constants, the function $f(\cdot)$ is defined in (5.2), and $g_1(\cdot)$ and $g_2(\cdot)$ are functions of the quaternion, $\mathbf{q}_b^o$. The function $g_1(\cdot)$ is equal to $g(\cdot)$ in (5.2), and $g_2(\cdot)$ is given as:

$$g(\mathbf{q}_b^o) = 1 - \left| (\mathbf{q}_b^o(T'))^\top \mathbf{q}_b^{o'} \right|, \tag{5.10}$$

where T' denotes the time for which it is desirable that the spacecraft reaches the intermediate attitude, which is denoted by the intermediate quaternion $\mathbf{q}_b^{o'}$, and the absolute value, $\left|(\mathbf{q}_b^o(T'))^\top \mathbf{q}_b^{o'}\right|$, is implemented using the smooth maximum function in (2.48).

For the path-following maneuver where multiple optimizations are performed, the vector $\mathbf{h}_N$ in the moving maximum hands-off controller was chosen such that it is cheaper for control input to occur between $N = 20$ and $N = 30$, which corresponds to $t = 28$ s and $t = 42$ s, respectively. The cost was set to 0.001 within this interval. For the path-following maneuver where an extra quaternion-term is added to the cost functions, the value of the vector $\mathbf{h}_N$ is set to 0.001 between $N = 7$ and $N = 14$, and $N = 35$ and $N = 42$, which corresponds to $t \approx 10$ s and $t \approx 20$ s, and $t = 49$ and $t = 59$ s, respectively, for the moving maximum hands-off controller. The cost outside the intervals specified by $\mathbf{h}_N$ is by default 1.

5.3.4 Multiple-axis maneuver with known perturbations

The last experiment aims to test the performance of the controllers when known perturbations are present in the system. The perturbations added to the system are the sum of the gravity gradient torque and a sine-wave. The implementation of the gravity gradient torque is shown in (4.11), and the implementation of the sine-wave is shown in (4.12). The multiple-axis maneuver rotates the spacecraft from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$.

For the moving maximum hands-off controller, the vector $\mathbf{h}_N$ was chosen such that it is cheaper for control input to occur between $N = 20$ and $N = 30$, which corresponds to $t = 28$ s and $t = 42$ s, respectively. The cost was set to 0.001 within this interval. The cost outside the interval specified by $\mathbf{h}_N$ is by default 1.

5.4 Simulation setup

The experiments were conducted using MATLAB R2020b (MATLAB, 2020) as the scripting language and CasADi as the optimization tool. CasADi is an open-source software framework for numerical optimization, which can be used to model and solve optimization problems with a large degree of flexibility (Andersson et al., 2019). CasADi also generates derivatives automatically using algorithmic differentiation, i.e., it automatically calculates the derivative of functions represented as algorithms (Andersson et al., 2019). The NLP-solver IPOPT was used to solve the optimization problems, using the solver's default options. See Section 2.5.1 for a more in-depth description of IPOPT. The optimal control problems in (5.3), (5.5) and (5.7) were discretized using direct multiple-shooting, and the dynamics of the spacecraft were discretized and implemented using Runge-Kutta 4 integration. Initial guesses for the starting point were given to IPOPT at the beginning of the optimization. The output from the PD-controller was given as the initial guess for the L_1-optimal controller, and more specifically as initial guess on the spacecraft's states $\boldsymbol{q}_b^o$, $\boldsymbol{\omega}_{ib}^b$, and $\boldsymbol{\omega}_{bw}^w$. The output from the L_1-optimal controller was given as initial guesses for the maximum hands-off controller and the moving maximum hands-off controller, on the states $\boldsymbol{q}_b^o$, $\boldsymbol{\omega}_{ib}^b$, and $\boldsymbol{\omega}_{bw}^w$. No initial guesses were applied to the state $\boldsymbol{\omega}_{ob}^b$ and the control torque $\boldsymbol{\tau}_u^b$, i.e., the default initial guess used by CasADi was applied. CasADi's default initial guess is a vector of all zeros. The experiments reported in this thesis were conducted using a 2 GHz Intel Core i7-9700T CPU computer running Windows.

The simulations of the experiments were conducted using the parameters for a 6U M6P CubeSat as the spacecraft rigid body, and it is assumed to orbit in low-Earth orbit (LEO). The CubeSat uses an M6P platform from NanoAvionics, and four reaction wheels are used to control the attitude of the CubeSat. This is part of the setup that is going to be used for the HYPSO-1 mission, which motivates this thesis, as discussed in Section 1.1. Three of the reaction wheels are placed orthogonally along the three axes of the body frame. The fourth reaction wheel is placed such that its torque yields equal components about each of the body-frame axes. The torque distribution matrix **A** is given as (Grøtte et al., 2020; Kristiansen et al., 2020)

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & \frac{1}{\sqrt{3}} \\ 0 & 1 & 0 & \frac{1}{\sqrt{3}} \\ 0 & 0 & 1 & \frac{1}{\sqrt{3}} \end{bmatrix}. \tag{5.11}$$

The total inertia matrix for the spacecraft rigid body is given as (Kristiansen et al., 2020)

$$\mathbf{J} = \begin{bmatrix} 0.0775 & 0.0002 & -0.0002 \\ 0.0002 & 0.1067 & 0.0005 \\ -0.0002 & 0.0005 & 0.0389 \end{bmatrix} \text{kg} \cdot \text{m}^2, \tag{5.12}$$

and the inertia matrix of the reaction wheels is (Kristiansen et al., 2020)

$$\mathbf{J}_w = J_w \mathbf{I}_{4\times 4}, \tag{5.13}$$

where $J_w = 2.1 \cdot 10^{-4}$ kg·m^2 is the inertia of a single reaction wheel. The inertia of all four reaction wheels are set to be identical. The gains for the controllers in (5.3), (5.5), (5.7) and (5.8), and the path-following cost functions in (5.9a) to (5.9c), are shown in Table 5.1, and the parameters required for the optimization are shown in Table 5.2.

The optimization is performed over N control intervals, and the simulation of the system stops at the final control interval. Thus, if the desired states are reached at control interval no. N, it is not possible to see from the simulations whether the states remain at the desired value or not. Therefore, the system is propagated for an additional number of control intervals, denoted N_{prop}, to visualize what happens after the optimization finishes. The final state from the optimization serves as the initial state for the propagation, and the control input is set to zero for the whole propagation. The orbit of the spacecraft is initialized using the orbital parameters in Table 5.3, which are transformed into ECI coordinates using the RANDV-function from Vallado (2001).

5.5 Discussion

In this section, the controller designs and the parameter values are discussed. First, Section 5.5.1 discusses the design of the maximum hands-off controller, whereas Section 5.5.2 discusses the design of the moving maximum hands-off

Controller gain	**Value**	**Unit**
k_1 (maximum hands-off controller)	$1 \cdot 10^6$	s^2
k_2 (maximum hands-off controller)	$1 \cdot 10^2$	-
k_3 (maximum hands-off controller)	$1 \cdot 10^1$	-
k_1 (moving maximum hands-off controller)	$1 \cdot 10^6$	s^2
k_2 (moving maximum hands-off controller)	$1 \cdot 10^2$	-
k_3 (moving maximum hands-off controller)	$1 \cdot 10^1$	-
k_1 (L_1-controller)	$1 \cdot 10^6$	s^2
k_2 (L_1-controller)	$1 \cdot 10^2$	-
k_3 (L_1-controller)	$1 \cdot 10^1$	-
a_1 (path-following)	$1 \cdot 10^6$	s^2
a_2 (path-following)	$1 \cdot 10^2$	-
a_3 (path-following)	$1 \cdot 10^2$	-
a_4 (path-following)	$1 \cdot 10^1$	-
$\mathbf{K}_p$ (PD-controller)	$3 \cdot \mathbf{J}_s$	N·m
$\mathbf{K}_d$ (PD-controller)	$2.7 \cdot \mathbf{J}_s$	N·m·s

Table 5.1: Controller gains.

Variable	**Value**	**Unit**
Simulation time (T)	70	s
Control intervals (N)	50	-
Step size (h)	1.4	s
Control intervals for propagation (N_{prop})	20	-
ϵ (maximum hands-off controller)	$1 \cdot 10^{-8}$	-
ϵ (moving maximum hands-off controller)	$1 \cdot 10^{-8}$	-
τ_{limit}	$\pm 3 \cdot 10^{-3}$	N·m
Standard grav. parameter, Earth (μ)	$3.986 \cdot 10^{14}$	m^3/s^2

Table 5.2: Optimization constants.

controller. Then, Section 5.5.3 discusses the values of the parameters used in the optimal control problems. Note that, in this section, the *sparsity* of a control signal refers to the number of nonzero control intervals in the signal, as defined in definition 5.1.1.

5.5.1 Design choice for the maximum hands-off controller

The maximum hands-off controller used to solve the spacecraft attitude control problem was presented in Section 5.2.1. During the design phase of the controller, different ways to formulate the maximum hands-off control problem were explored before the final design choice was made. This section compares

Orbital elements	**Value**	**Unit**
Semi-major axis	6852.2	km
Eccentricity	0.002	-
Inclination	97	°
Right ascension of the ascending node	280	°
Argument of periapsis	0	°
True anomaly	0	°

Table 5.3: Orbital parameters.

the different design choices to one another and explains why the final design choice was made. The different formulations of the maximum hands-off control problem were all inspired by the relaxed formulation in the work by Feng et al. (2016), which was presented in Section 3.6.4.

The formulation of the maximum hands-off control problem presented in (5.3) was chosen because it yielded the best results compared to the other designs that were tested. For a single-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 0°, 0°)$, the computation time, the sparsity, and the total number of iterations for (5.3) are shown in Table 5.4 in the row named 'Design 1'. The orientation trajectories and the optimal control signals represented in the body-frame are shown in Fig. 5.1 and Fig. 5.2, respectively.

A different design of the maximum hands-off control problem was considered, where three complementarity vectors, $\boldsymbol{\xi}_1$, $\boldsymbol{\xi}_2$ and $\boldsymbol{\xi}_3$, were related to each of the three components of $\boldsymbol{\tau}_u^b$, yielding the following optimal control problem

$$
\begin{aligned}
\underset{\boldsymbol{\tau}_u^b,\boldsymbol{\xi}}{\text{minimize}} \quad & k_1 f(\boldsymbol{\omega}_{ob}^b) + k_2 g(\mathbf{q}_b^o) + k_3((\mathbf{1}_N - \boldsymbol{\xi}_1)\mathbf{1}_N^\top + (\mathbf{1}_N - \boldsymbol{\xi}_2)\mathbf{1}_N^\top + (\mathbf{1}_N - \boldsymbol{\xi}_3)\mathbf{1}_N^\top) && \text{(5.14a)}\\
\text{subject to} \quad & \dot{\mathbf{q}}_b^o = \frac{1}{2}\mathbf{T}(\mathbf{q}_b^o)\boldsymbol{\omega}_{ob}^b && \text{(5.14b)}\\
& \dot{\boldsymbol{\omega}}_{ib}^b = \mathbf{J}_s^{\text{-1}}(-\mathbf{A}\boldsymbol{\tau}_u^w - \mathbf{S}(\boldsymbol{\omega}_{ib}^b)\mathbf{H}_s^b + \boldsymbol{\tau}_{\text{mtq}}^b + \boldsymbol{\tau}_{\text{ext}}^b) && \text{(5.14c)}\\
& \dot{\boldsymbol{\omega}}_{bw}^w = \mathbf{J}_w^{\text{-1}}\boldsymbol{\tau}_u^w - \mathbf{A}^\top\dot{\boldsymbol{\omega}}_{ib}^b && \text{(5.14d)}\\
& \dot{\boldsymbol{\omega}}_{ob}^b = \dot{\boldsymbol{\omega}}_{ib}^b + \mathbf{S}(\boldsymbol{\omega}_{ob}^b)\mathbf{R}_o^b\boldsymbol{\omega}_{io}^o + \mathbf{R}_o^b\mathbf{S}(\mathbf{R}_b^o\boldsymbol{\omega}_{ob}^b)\boldsymbol{\omega}_{io}^o && \text{(5.14e)}\\
& \dot{\mathbf{r}}^i = \mathbf{v}^i && \text{(5.14f)}\\
& \dot{\mathbf{v}}^i = -\frac{\mu}{\|\mathbf{r}^i\|_2^3}\mathbf{r}^i && \text{(5.14g)}\\
& \boldsymbol{\tau}_u^w \leq \boldsymbol{\tau}_{\text{limit}} && \text{(5.14h)}\\
& \boldsymbol{\tau}_u^w \geq -\boldsymbol{\tau}_{\text{limit}} && \text{(5.14i)}\\
& \mathbf{q}_b^o(0) = \mathbf{q}_{b,0}^o && \text{(5.14j)}\\
& \boldsymbol{\omega}_{ib}^b(0) = \boldsymbol{\omega}_{ib,0}^b && \text{(5.14k)}\\
& \boldsymbol{\omega}_{bw}^w(0) = \boldsymbol{\omega}_{bw,0}^w && \text{(5.14l)}\\
& \boldsymbol{\omega}_{ob}^b(0) = \boldsymbol{\omega}_{ob,0}^b && \text{(5.14m)}\\
& \mathbf{r}^i(0) = \mathbf{r}_0^i && \text{(5.14n)}\\
& \mathbf{v}^i(0) = \mathbf{v}_0^i && \text{(5.14o)}\\
& \boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3 \leq \mathbf{1}_N && \text{(5.14p)}\\
& \boldsymbol{\xi}_1 \circ \boldsymbol{\tau}_{u,1}^b \leq \epsilon\mathbf{1}_N && \text{(5.14q)}\\
& -\boldsymbol{\xi}_1 \circ \boldsymbol{\tau}_{u,1}^b \leq \epsilon\mathbf{1}_N && \text{(5.14r)}\\
& \boldsymbol{\xi}_2 \circ \boldsymbol{\tau}_{u,2}^b \leq \epsilon\mathbf{1}_N && \text{(5.14s)}\\
& -\boldsymbol{\xi}_2 \circ \boldsymbol{\tau}_{u,2}^b \leq \epsilon\mathbf{1}_N && \text{(5.14t)}\\
& \boldsymbol{\xi}_3 \circ \boldsymbol{\tau}_{u,3}^b \leq \epsilon\mathbf{1}_N && \text{(5.14u)}\\
& -\boldsymbol{\xi}_3 \circ \boldsymbol{\tau}_{u,3}^b \leq \epsilon\mathbf{1}_N && \text{(5.14v)}\\
& \boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3 \geq \mathbf{0}, && \text{(5.14w)}
\end{aligned}
$$

where the constants, functions, and dynamics are the same as in (5.3).

The optimal control problem in (5.14) was tested for a single-axis maneuver from $(0^\circ, 0^\circ, 0^\circ)$ to $(90^\circ, 0^\circ, 0^\circ)$, with the controller gains in Table 5.1 and the output from the L_1-controller as initial guess for the optimization. The computation time for the optimization, the sparsity, and the number of iterations are shown in Table 5.4, in the row named 'Design 2'. The orientation trajectory is shown in Fig. 5.1, and the optimal control signal represented in the body-frame is shown in Fig. 5.2.

The design choice in (5.14) would make the optimal control inputs more independent from one another, but it would also result in extra cost when trying to provide control inputs about different axes simultaneously. Because of this, the optimal control inputs would most likely occur at more places, e.g., a control input about the z-axis would not occur at the same time instant as a control input about the y-axis, which is confirmed in Fig. 5.2. In Fig. 5.2 it can be seen that for Design 2, the first control torque is applied about the x-axis, followed by a control torque about the z-axis. This contrasts the findings for Design 1, where the first occurrence of control input has components about the x-, y-, and z-axes. Control input at more time instants yields a less sparse control signal, as shown in Table 5.4 where Design 2 gives a control signal which is less sparse than the control signal resulting from Design 1. These results favor an optimal control problem on the form of (5.3) over (5.14).

A third design choice also considers three complementarity vectors, $\boldsymbol{\xi}_1$, $\boldsymbol{\xi}_2$ and $\boldsymbol{\xi}_3$, related to each of the three components of $\boldsymbol{\tau}_u^b$, resulting in the following optimal control problem

$$\underset{\boldsymbol{\tau}_u^b, \boldsymbol{\xi}}{\text{minimize}} \quad k_1 f(\boldsymbol{\omega}_{ob}^b) + k_2 g(\mathbf{q}_b^o) + k_3(\mathbf{1}_N - (\boldsymbol{\xi}_1 + \boldsymbol{\xi}_2 + \boldsymbol{\xi}_3))\mathbf{1}_N^\top, \tag{5.15}$$

where the constraints on (5.15) are identical to (5.14b) to (5.14w), and the constants, functions, and dynamics are the same as in (5.3). The formulation in (5.15) provides a design where control torque about different axes could occur separately, which could lead to a less sparse signal.

The optimal control problem in (5.15), called 'Design 3', was tested for a single-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 0°, 0°)$. The computation time for the optimization, the sparsity of the control signal, and the number of iterations are shown in Table 5.4 in the row named Design 3. The orientation trajectory of the spacecraft is shown in Fig. 5.1, and the optimal control signal in the body-frame is shown in Fig. 5.2.

The orientation trajectory and optimal control signal resulting from Design 3 is similar to the orientation trajectory and optimal control signal resulting from Design 2, as shown in Fig. 5.1 and Fig. 5.2. The optimal control problems with Design 2 and Design 3 uses the same number of iterations, 1025, to find the optimal control signal, compared to 553 iterations for Design 1. The complementarity constraints in Design 1 force the control torques to occur simultaneously, as opposed to the complementarity constraints in Design 2 and Design 3, which yield the control inputs more independent from one another. It might be easier for the solver to find the optimal solution in the case for Design 1, since the control inputs have to appear simultaneously due to the design of the complementarity constraints, and may explain why the solver uses fewer iterations to find the optimal solution for Design 1 than for the two other designs. This reasoning may also apply to the computation times shown in Table 5.4, where it is clear Design 1 uses less time than the other two designs. Another difference in the design choices becomes apparent in Fig. 5.1 where θ_{L0} deviates from zero both throughout the simulations and also at the end of the simulations for Design 2 and Design 3, as opposed to for Design 1. From Fig. 5.2 it can be seen that Design 2 and Design 3 only provide control torque about the y-axis at the end of the simulation, whereas Design 1 provides control torque about the y-axis both at the start and the end of the simulation. This could be the reason for the different responses in θ_{L0} for the different designs.

Table 5.4 shows that Design 1 provides a control signal with a sparsity of 2, whereas Design 2 and 3 yield control signals with sparsities of 3. The reason why Design 1 yields a sparser control signal is because this design makes it possible for control inputs to occur simultaneously at no extra cost. For instance, if a large control torque occurs at one time instant, then a smaller control torque could be applied at the same instant without increasing the value of the cost function. This is possible because the optimal control problem in (5.3), Design 1, is defined using only one complementarity vector $\boldsymbol{\xi}$ which relates to all three components of $\boldsymbol{\tau}_u^b$. For Design 2 and Design 3 there are three complementarity vectors, $\boldsymbol{\xi}_1$, $\boldsymbol{\xi}_2$ and $\boldsymbol{\xi}_3$, that relates to each of the three components of $\boldsymbol{\tau}_u^b$. This design provides for control inputs that are more independent from one another, but, as shown in Table 5.4, may also result in control signals that are less sparse compared to those produced by Design 1.

The results and discussions presented in this section show why the optimal control problem in (5.3) were chosen over (5.14) and (5.15) for the design of the maximum hands-off control problem.

Control design	**CPU time NLP, [s]**	**CPU time IPOPT, [s]**	**Sparsity**	**Iterations**
Design 1	70.571	6.279	2	553
Design 2	133.854	12.225	3	1025
Design 3	132.411	12.070	3	1025

Table 5.4: Computation time, sparsity and iterations for Design 1, Design 2, and Design 3 for a single-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 0°, 0°)$.

5.5.2 Design choice for the moving maximum hands-off controller

During the design phase of the moving maximum hands-off controller, two different formulations of the moving maximum hands-off optimal control problem were explored. The first optimal control problem that was tested is given in (5.5). The second optimal control problem explores a design similar to that in (5.5), but the cost function is formulated slightly different resulting in the following minimization problem

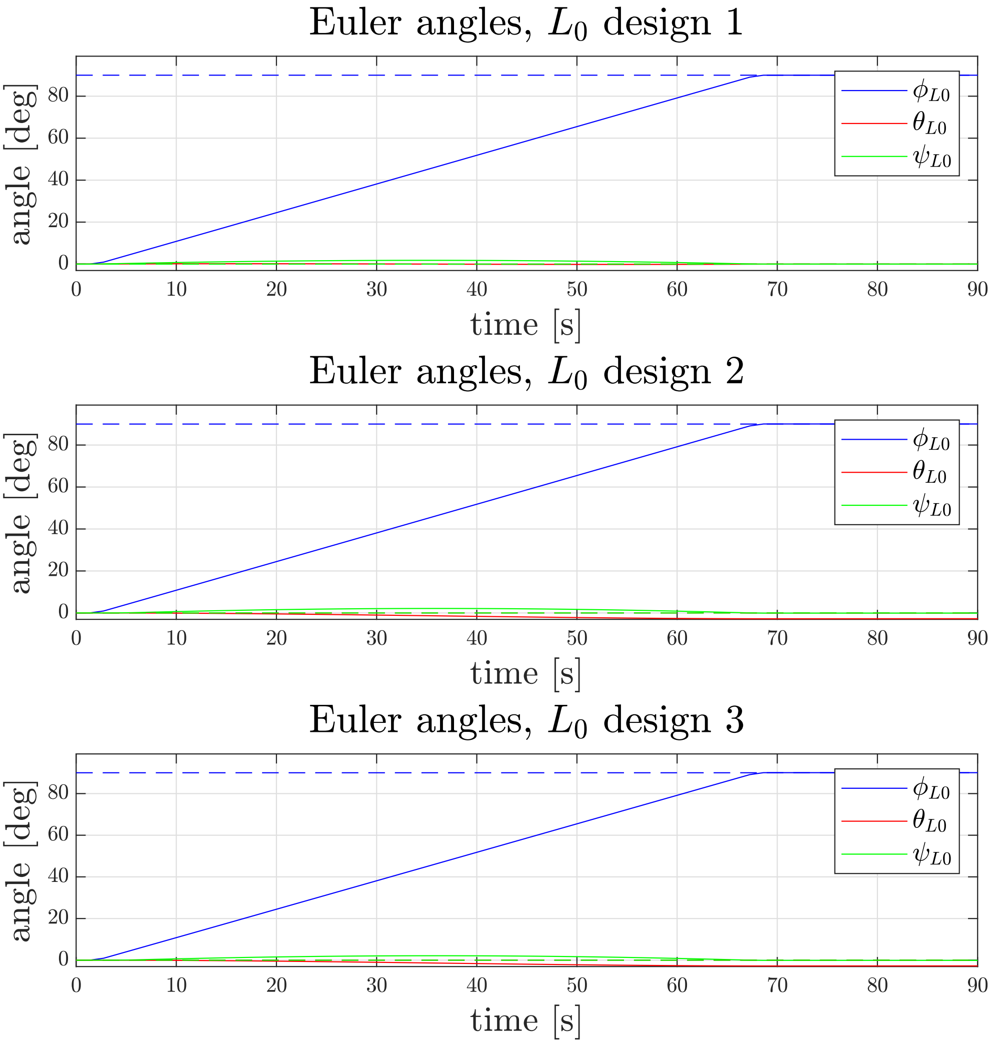


Figure 5.1: Euler angles for Design 1, Design 2, and Design 3 for a single-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 0°, 0°)$.

$$\underset{\boldsymbol{\tau}_u^b, \boldsymbol{\xi}}{\text{minimize}} \quad k_1 f(\boldsymbol{\omega}_{ob}^b) + k_2 g(\mathbf{q}_b^o) + k_3(\mathbf{h}_N \circ (\mathbf{1}_N - \boldsymbol{\xi})\mathbf{1}_N^\top), \tag{5.16}$$

where the controller gains are given as in Table 5.1, the functions $f(\cdot)$ and $g(\cdot)$ are given in (5.2), and the constraints on the minimization problems are identical to those in (5.5b) to (5.5w) but omitted here for the sake of simplicity. The two different formulations of the moving maximum hands-off control problem gave the same results, and the first formulation was chosen.

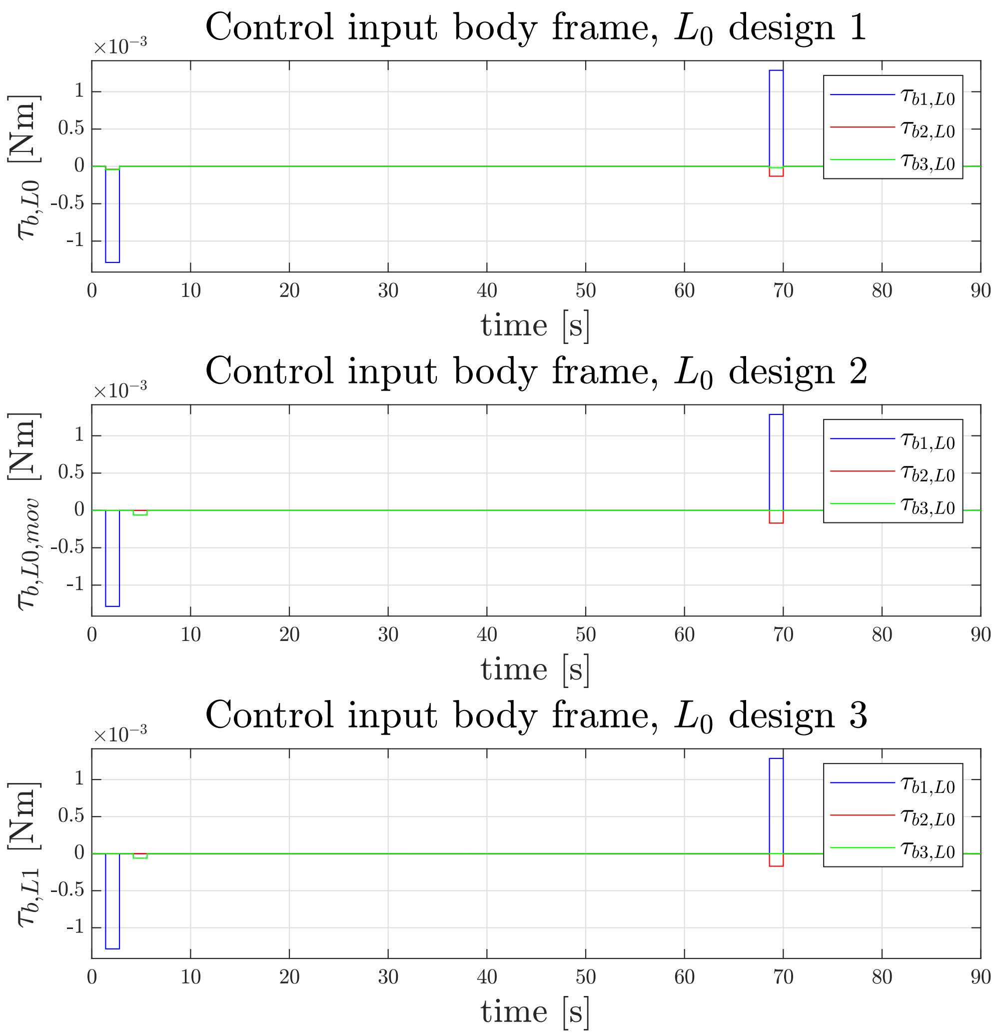


Figure 5.2: Control input body frame for Design 1, Design 2, and Design 3 for a single-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 0°, 0°)$.

5.5.3 Parameter values in the optimal control problems

The values for the different controller gains are showed in Table 5.1, and their values are discussed in this section.

The PD-controller gains, $\mathbf{K}_p$ and $\mathbf{K}_d$, were found through tuning as mentioned in Section 5.2.4. Experiments with the PD-controller have confirmed that the controller works well, but to make the PD-controller perform even better, additional tuning of the controller gains could be done. The PD-controller was used to test whether the spacecraft attitude dynamics worked as intended and to produce initial guesses for the L_1-controller.

The positive constants, k_1, k_2, and k_3 in (5.3a), (5.5a), and (5.7a) were found through tuning and they represent the weights on the different terms in the cost functions. The values of k_1, k_2, and k_3 are identical for the three controllers to provide identical conditions for all of them. The results of the optimization are sensitive to changes in the weights, because k_1, k_2, and k_3 specifies how much 'effort' should be put into optimizing the term related to the respective controller gain. For instance, if there is a large value on the controller gain in front of the term dealing with the final state of $\boldsymbol{\omega}_{ob}^b$, and a small value on the controller gain in front of the term dealing with the final state of $\mathbf{q}_b^o$, it essentially means that the optimization algorithm considers it more important to reach the desired final state of $\boldsymbol{\omega}_{ob}^b$. For the path-following cost functions in (5.9a) to (5.9c), the constant a_1 was set to the same value as k_1 since both constants determine the cost on the term involving $\boldsymbol{\omega}_{ob}^b$, a_2 and a_3 were both chosen to equal k_2 since all three constants regard terms involving $\mathbf{q}_b^o$, and a_4 were chosen to equal k_3 since both constants regard the L_0- and L_1-norm.

The simulation time T is set to 70 s, and the number of control intervals N is set to 50, which yields a step size $h = \frac{T}{N} = 1.4$ s. A larger number of control intervals, i.e., N approaches ∞, would yield a smaller step size. For a smaller step size, the control torques have to become larger at each step in order to obtain the same effect as before. The total amount of torque required to perform a maneuver is the same regardless of the step size. Therefore, when the step size becomes smaller, the control torque will be applied over a shorter time, and the value of the torque needs to be larger to obtain the same total torque as for a larger time step. At one point, the control torques cannot be larger due to saturation on the actuators, which in this case are reaction wheels. If the control torque needed at one time step is larger than the saturation limit, it could be necessary to apply an additional control torque at the following time step to get the same amount of torque. However, if the saturation limit was higher, the reaction wheels may not saturate even though the step size decreases, and it would not be necessary to apply torque at an additional time step. Thus, decreasing the step size while keeping the saturation limit constant could yield a control signal which is less sparse than if the saturation limits were higher.

The spacecraft state variables and the control input that are optimized using CasADi, are of different orders of magnitude and are therefore scaled before the optimization. IPOPT does not scale the optimization variables automatically, even though it does so for the objective function and the constraints (CasADi, 2018). Therefore, the optimization variables had to be scaled before the optimization. The scaling variables were found by studying the final values of the state and input before the scaling was applied, followed by some additional tuning. An observation was made during the tuning that the solution to the optimization is very sensitive to the values chosen for the scaling variables.

In Section 5.2.1 it was mentioned that the value ϵ, used in the complimentary constraints of the controllers (5.3) and (5.5), is a positive constant which imposes slack on the complementarity constraints. (5.3) and (5.5) are relaxed formulations of the general L_0-norm minimization problem discussed in Feng et al. (2016) and Section 3.6. (3.31) shows that the complementarity constraint on the minimization problem is equal to zero for the half complementarity formulation of the general L_0-norm minimization problem, i.e., $\boldsymbol{\xi} \circ \mathrm{x} = 0$. The relaxed formulation presented in Section 3.6.4, and used in the optimal control problems (5.3) and (5.5), relaxes the complementarity constraints using ϵ. The value of ϵ determines how strict the complementarity constraint is, i.e. if $\epsilon = 0$ it implies that the value of either $\boldsymbol{\xi}$ or x, or both, has to be exactly zero, and it could be hard to solve an optimization problem having such strict constraints. On the other hand, if ϵ is a nonzero and small value then the values of $\boldsymbol{\xi}$ or x, or both, do not have to exactly zero, they just have to be small. This would provide a less strict constraint on the optimal control problem, and the problem could be easier to solve. During the design process of the maximum hands-off controller and the moving maximum hands-off controller, different values of ϵ were tested, resulting in various performance of the controllers. Eventually, the value for ϵ was chosen to be $1 \cdot 10^{-8}$ because it provided good results during the tests. Also, $\epsilon = 1 \cdot 10^{-8}$ is the value used for the experiments in the work by Feng et al. (2016).

Chapter 6

Results and Discussion

This chapter presents the results from the simulations and discusses the findings. First, Section 6.1 provides a general discussion related to the implementation of the L_0-controller and to whether the control torques should be optimized in the body-frame $\{b\}$ or in the wheel-frame $\{w\}$. The rest of the chapter presents and discusses the results from each of the experiments considered in Section 5.3. Section 6.2 presents and discusses the results from the experiment involving rotation around one single axis, and Section 6.3 deals with the experiments involving rotation around multiple axes. In Section 6.4, the results from the two path-following schemes are presented and discussed. The last section, Section 6.5, presents and discusses the results from a multiple-axis maneuver where known perturbations are present in the system. For each experiment, the optimal control inputs and spacecraft state trajectories resulting from the maximum hands-off controller, the moving maximum hands-off controller, and the L_1-optimal controller are compared. For the rest of this chapter, the maximum hands-off controller will be referred to as the L_0-optimal controller or the L_0-controller, whereas the moving maximum hands-off controller will be referred to as the moving L_0-optimal controller or the moving L_0-controller. Note that the *sparsity* of a control signal refers to the number of nonzero control intervals in the signal, as defined in Definition 5.1.1. The number of control intervals for which the control signal takes on nonzero values refers to the number of control intervals for which the absolute value of the control signal is larger than a threshold value chosen to be $1 \cdot 10^{-6}$.

6.1 General discussion

6.1.1 Initial guesses for the IPOPT solver used to solve the L_0-optimal control problem

This section presents and discusses the findings from the tests where different initial guesses were provided for the IPOPT solver used to solve the L_0-optimal control problem. Table 6.1 presents the computation time and the number of iterations used to find the optimal solution, for different initial guesses for the IPOPT solver, when the spacecraft performs a single-axis maneuver from $(0°, 0°, 0°)$ to $(45°, 0°, 0°)$. Fig. 6.1 and Fig. 6.2 show the spacecraft's orientation trajectories and the control signals resulting from the different initial guesses, respectively. In this section, variants of the phrases "initial guesses for the L_0-controller" and "initial guesses for the solver" are used interchangeably, and both refer to the initial guesses for the IPOPT solver used to solve the L_0-optimal control problem.

Initial guess for the L_0-controller	CPU time NLP, [s]	CPU time IPOPT, [s]	Iterations
Sat. PD	122.923	10.794	953
Unsat. PD	71.794	5.949	559
No init	1.831	0.145	14
All ones	227.425	19.587	1757
PD (45, 30, 15)	192.134	16.798	1473
L_0 init	130.639	11.199	1001

Table 6.1: Computation time and iterations for different initial guesses on the L_0-controller, for a single-axis maneuver from $(0°, 0°, 0°)$ to $(45°, 0°, 0°)$.

The initial guesses for IPOPT are of significant importance when trying to find the optimal solutions to the optimal control problems. Different initial guesses cause the optimization of the different optimal control problems to iterate fast or slow towards an optimal solution, and it affects the quality of the solution. As mentioned in Section 5.4, the output from the PD-controller serves as an initial guess for the L_1-controller, and more specifically as an initial guess for the spacecraft's states $\boldsymbol{q}_b^o$, $\boldsymbol{\omega}_{ib}^b$, and $\boldsymbol{\omega}_{bw}^w$. The output from the L_1-controller serves as an initial guess for the L_0- and moving L_0-controller, on the states $\boldsymbol{q}_b^o$, $\boldsymbol{\omega}_{ib}^b$, and $\boldsymbol{\omega}_{bw}^w$. No initial guesses were applied to the state $\boldsymbol{\omega}_{ob}^b$ and the control torque $\boldsymbol{\tau}_u^b$, i.e., the default initial guess used by CasADi was applied. CasADi's default initial guess is a vector of all zeros. Note that the different initial guesses, presented in this section, were all tested on a single-axis maneuver from $(0°, 0°, 0°)$ to $(45°, 0°, 0°)$.

Table 6.3 shows the number of iterations used to find the optimal control signal using the output from the L_1-controller as an initial guess for the L_0- and

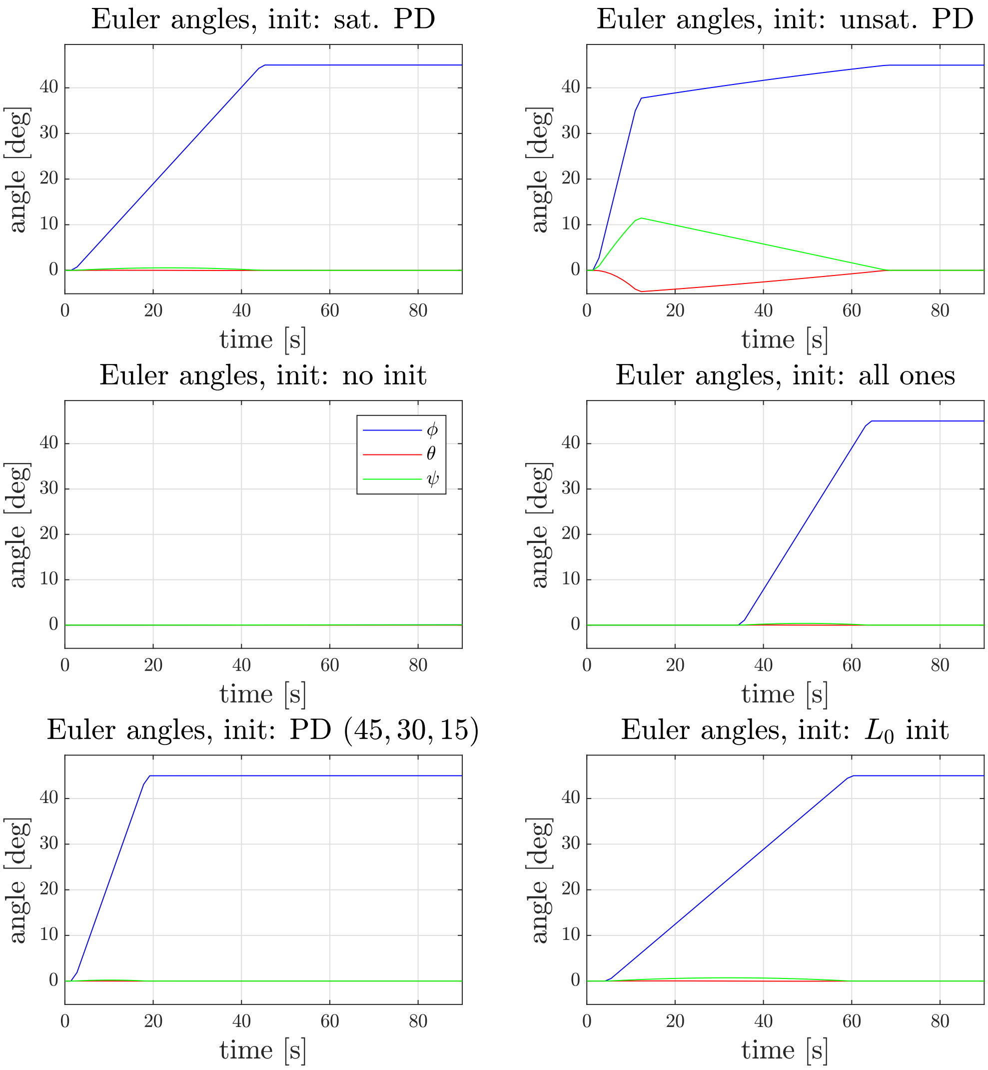


Figure 6.1: Euler angles for different initial guesses on the L_0-controller, for a single-axis maneuver from $(0°, 0°, 0°)$ to $(45°, 0°, 0°)$. Note that the legend describing the colors of the trajectories for ϕ, θ, ψ in the subplot named "Euler angles, init: no init" applies to all the subplots in this figure.

moving L_0-controllers. Initially, the output from the PD-controller served as an initial guess for both the L_1- and the L_0-controller, but after testing different initial guesses, it turned out the better solution was to use the output from the L_1-controller as an initial guess for the L_0- and moving L_0-controllers.

The control torques produced by the PD-controller saturate due to the limitations of the reaction wheels, and the saturated control input gives a different spacecraft response than the response following from an unsaturated control input. The output from both the saturated and unsaturated PD-controller were used as initial guesses for the (saturated) L_0-controller to study the response of

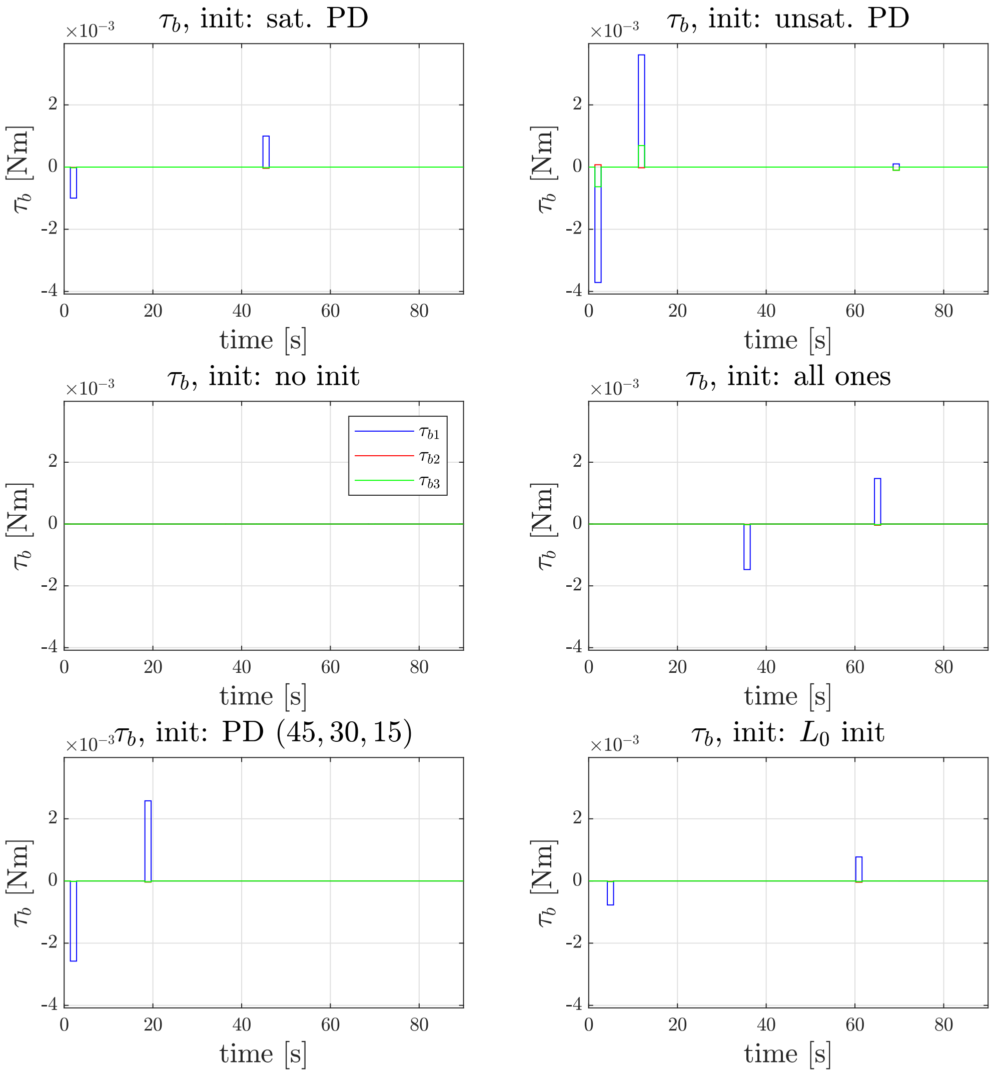


Figure 6.2: Control input in body-frame for different initial guesses on the L_0-controller, for a single-axis maneuver from $(0°, 0°, 0°)$ to $(45°, 0°, 0°)$. Note that the legend describing the colors of control torques for $\tau_{b,1}, \tau_{b,2}, \tau_{b,3}$ in the subplot named "Euler angles, init: no init" applies to all the subplots in this figure.

the L_0-controller resulting from different initial guesses. The computation time and the number of iterations used to find the L_0-optimal control signal, with initial guesses from both the saturated and unsaturated PD-controller, are shown in Table 6.1. The orientation trajectory is shown in Fig. 6.1, and the resulting optimal control signal is shown in Fig. 6.2, where 'sat. PD' and 'unsat. PD' denotes the results from the saturated and unsaturated PD-controller, respectively. From Table 6.1, it is clear that using the initial guess from the unsaturated PD-controller requires fewer iterations to find the L_0-optimal solution than using the initial guess from the saturated PD-controller. Although the initial guess from the unsaturated PD-controller yields fewer iterations, Fig. 6.2 reveals that the optimal control signal found using the initial guess from the unsaturated PD-controller is less sparse than that of the saturated PD-controller.

A test was conducted with no initial guess on the L_0-controller, which made the optimization procedure using only 14 iterations to find the optimal solution, as shown in Table 6.1. The optimal solution found in this case was not even close to reaching the desired orientation, as shown in Fig. 6.1. The abbreviation 'no init' denotes the case for no initial guess. Running the optimization with no initial guess gives the same result as applying a vector of all zeros as an initial guess, which is the default initial guess in CasADi.

A vector of all ones of dimension N, where N is the number of control intervals, was applied as an initial guess for the L_0-controller. The computation time for the optimization, and the number of iterations, when applying this initial guess are shown in Table 6.1, in the row named 'all ones'. The vector of all ones yields the highest computation time and the highest number of iterations among all the initial guesses that were tested. Although the computation time and number of iterations are relatively high, the control input shown in Fig. 6.2 is as sparse as the L_0-optimal control input shown in Fig. 6.5, where the output from the L_1-controller is used as initial guess. The spacecraft reaches the desired orientation for the initial guess of all ones, as can be seen in Fig. 6.1.

An experiment was conducted to investigate the performance of the L_0-controller for a single-axis maneuver from $(0°, 0°, 0°)$ to $(45°, 0°, 0°)$, when it is provided an initial guess from the PD-controller performing a multiple-axis maneuver from $(0°, 0°, 0°)$ to $(45°, 30°, 15°)$. Table 6.1 shows that this initial guess causes the optimization procedure to use 1473 iterations, compared to 953 iterations when using an initial guess from the PD-controller performing the same maneuver as the L_0-controller. Fig. 6.1 shows that the spacecraft reaches the desired orientation for this initial guess, denoted 'PD (45, 30, 15)', and Fig. 6.2 shows that the optimal control signal is as sparse as for the initial guess from the saturated PD-controller performing a maneuver from $(0°, 0°, 0°)$ to $(45°, 0°, 0°)$.

Another initial guess was tested for the L_0-controller, based on the output from the previous optimization. The output from the previous optimization refers to the output from the L_0-controller performing a single-axis maneuver from $(0°, 0°, 0°)$ to $(45°, 0°, 0°)$ using the output from the L_1-controller as the initial guess. The output of the previous optimization was applied to the L_0-controller performing a single-axis maneuver from $(0°, 0°, 0°)$ to $(45°, 0°, 0°)$. The computation time and number of iterations can be seen in Table 6.1, in the row denoted 'L_0 init'. Fig. 6.1 and Fig. 6.2 show the change in orientation and the optimal control input.

By comparing the number of iterations in Table 6.1 and the number of iterations used by the L_0-controller in Table 6.3, it is clear that using the output from the L_1-controller as an initial guess for the L_0-controller yields an optimization procedure that in most cases iterates faster than for the other initial guesses that were tested. Table 6.1 reveals that the L_0-controller uses only 14 iterations to find the optimal solution, when there are no initial guesses applied, compared to the 25 iterations in Table 6.3. However, considering the Euler angles in Fig. 6.1 it is clear that the spacecraft does not reach the desired position when there are no initial guesses applied. Conversely, Fig. 6.3 shows that the spacecraft reaches the desired orientation using the L_0-controller with initial guesses provided by the L_1-optimal controller. The output from the L_1-optimal controller was chosen as input for the L_0-optimal controller for the remaining experiments, after considering the different cases explained in this section and comparing the computation time, the number of iterations, the optimal control signals, and the different spacecraft orientation trajectories.

Additionally, a few tests were carried out to explore how the L_1-controller responds to different initial guesses. The L_1-controller is less sensitive to different initial guesses than the L_0-controller. However, the L_1-controller provides better results when the output from the PD-controller is applied as an initial guess for the controller, compared to no initial guess at all. Here, better results refer to the optimization procedure using fewer iterations to find the optimal solution, as the spacecraft attitude trajectories were identical for both initial guesses.

6.1.2 Optimization in body-frame versus optimization in wheel-frame

Section 5.1 describes that the optimization of the control input is performed in the body-frame $\{b\}$, instead of in the wheel-frame $\{w\}$. Tests were conducted to investigate in which frame the optimization should be conducted, and the computation times and the number of iterations found for the different tests are shown in Table 6.2. The tests were conducted for a single-axis maneuver from $(0°, 0°, 0°)$ to $(45°, 0°, 0°)$, using the L_0-controller with an initial guess from the L_1-controller. From Table 6.2 it is clear that optimization conducted in $\{b\}$ uses less computation time and fewer iterations than optimization in $\{w\}$. A possible explanation for this is that optimization in $\{w\}$ could add complexity to the optimization procedure as the distribution of control would have to be optimized among four reaction wheels, instead of only distributing the control about the three $\{b\}$-axes. The comparison between optimization in the two frames revealed that the optimal control signals had the same sparsity. Thus, since the optimization conducted in $\{b\}$ uses less computation time and fewer iterations than optimization in $\{w\}$, a decision was made to perform the optimization in $\{b\}$. The control torques in $\{b\}$ are then transformed to $\{w\}$ using the Moore-Penrose pseudo-inverse of $\mathbf{A}$ so that the desired control torques are distributed among the four reaction wheels. The Moore-Penrose pseudo-inverse of $\mathbf{A}$ is the optimal mapping from $\{b\}$ to $\{w\}$, and the conversion between $\{b\}$ and $\{w\}$ are studied in Section 2.3.4.

Frame used for optimization	**CPU time NLP, [s]**	**CPU time IPOPT, [s]**	**Iterations**
Body-frame $\{b\}$	3.277	0.225	25
Wheel-frame $\{w\}$	5.979	0.499	41

Table 6.2: Computation time and iterations in $\{b\}$ and $\{w\}$, for a single-axis maneuver from $(0°, 0°, 0°)$ to $(45°, 0°, 0°)$ the L_0-controller.

6.2 Single-axis maneuver

This section presents and discusses the results from the experiment discussed in Section 5.3.1, in which the spacecraft is rotated an angle $\phi = 45°$ about the x-axis. There is no rotation about the remaining axes, i.e., $\theta = 0°$, and $\psi = 0°$. First, the results are presented, followed by a discussion of the results in Section 6.2.1.

Fig. 6.3 shows the change in the spacecraft's attitude over time, and reveals that all three controllers are able to steer the spacecraft to the desired orientation of $(\phi, \theta, \psi) = (45°, 0°, 0°)$ and keep the spacecraft at this attitude. The dotted lines ϕ_d, θ_d, and ψ_d denotes the angles of the desired attitude, whereas the solid lines ϕ, θ, and ψ corresponds to the actual states. The angular velocity of $\{b\}$ relative to $\{o\}$, represented in $\{b\}$, $\boldsymbol{\omega}_{ob}^{b}$, is shown in Fig. 6.4 and reveals that the spacecraft stops rotating, i.e., the body-frame stops rotating relative to the orbit-frame, when the desired orientation is reached. The optimal control input is computed in the body-frame $\{b\}$, and the optimal control signal can be seen in Fig. 6.5. The torque vector in $\{b\}$, $\boldsymbol{\tau}_{u}^{b}$, is transformed into the wheel-frame $\{w\}$, $\boldsymbol{\tau}_{u}^{w}$, before the torques are applied to the reaction wheels. The transformation between $\{b\}$ and $\{w\}$ is discussed in Section 2.3.4. The optimal control signals represented in $\{w\}$ are shown in Fig. 6.6. The angular velocity of the reaction wheels, $\boldsymbol{\omega}_{bw}^{w}$, can be seen in Fig. 6.7, and illustrates the dynamical response of the reaction wheels. The computation times for the different control scenarios, the sparsity, i.e., the number of nonzero control intervals, for each of the control signals, and the number of iterations used to find the optimal solution for each of the three controllers are shown in Table 6.3.

6.2.1 Discussion

Fig. 6.3, Fig. 6.4, Fig. 6.5, and Fig. 6.6 show that the changes in the spacecraft's states over time and the control signals are identical for the L_0-controller and the L_1-controller. These results are in agreement with the findings in Nagahara

Controller	**CPU time NLP, [s]**	**CPU time IPOPT, [s]**	**Sparsity**	**Iterations**
L_0-optimal control	3.235	0.253	2	25
Moving L_0-optimal control	46.914	3.828	2	371
L_1-optimal control	2.479	0.204	2	19

Table 6.3: Computation time, sparsity and number of iterations for a single-axis maneuver from $(0°, 0°, 0°)$ to $(45°, 0°, 0°)$.

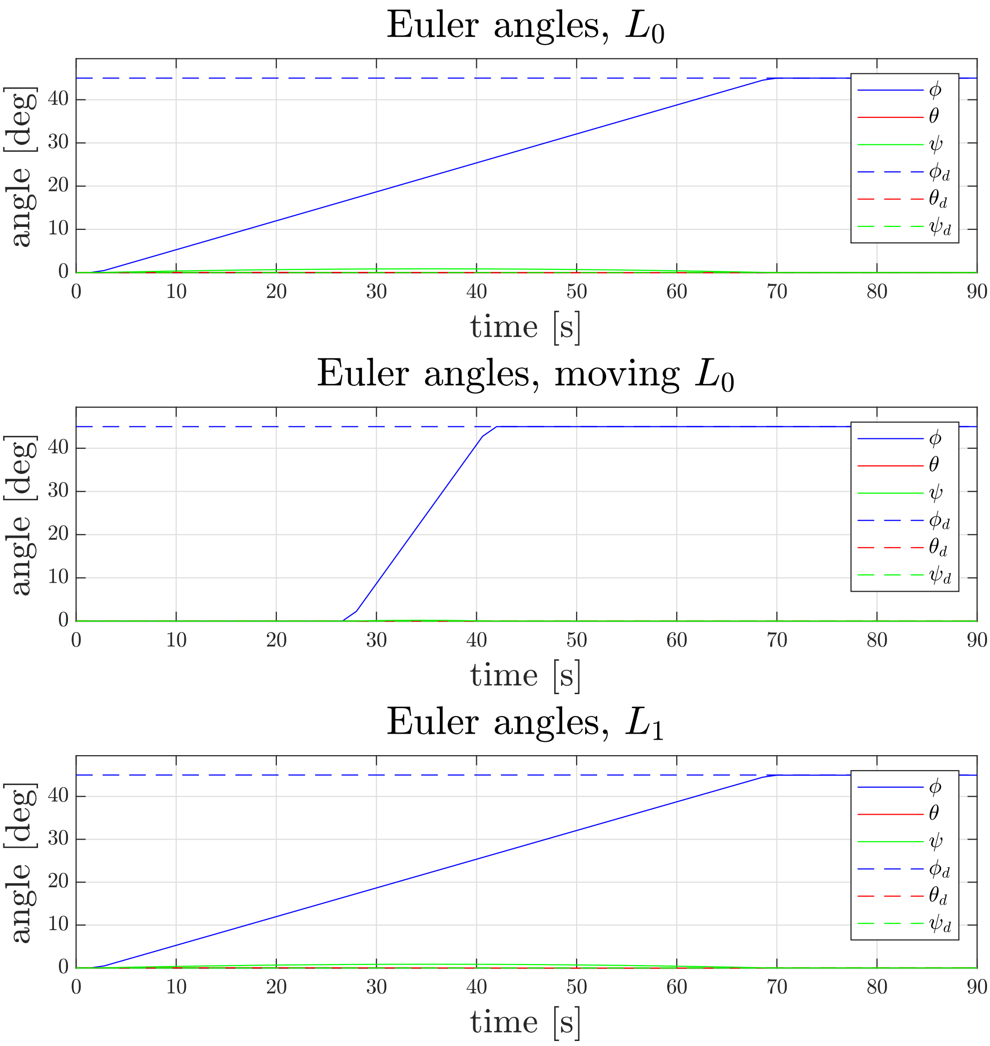


Figure 6.3: Euler angles for a single-axis maneuver from $(0°, 0°, 0°)$ to $(45°, 0°, 0°)$.

et al. (2015) and suggest that the L_1-norm may be used as an approximation to the L_0-norm.

On the other hand, Fig. 6.3, Fig. 6.4, Fig. 6.5, and Fig. 6.6 show that the responses and control signals resulting from the moving L_0-controller differs from the two other controllers. The differences can be explained by comparing the cost functions of the three controllers in (5.3a), (5.5a), and (5.7a). For the

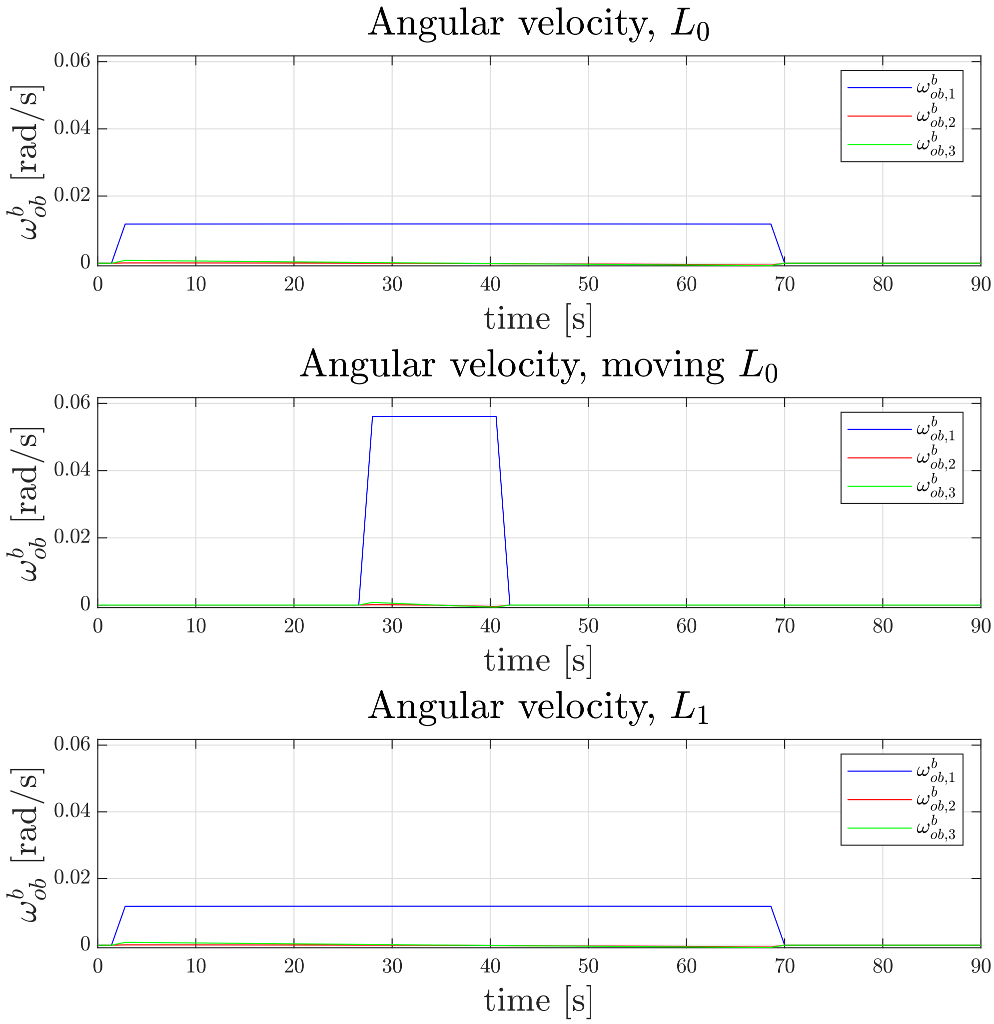


Figure 6.4: Angular velocity, $\boldsymbol{\omega}_{ob}^b$, for a single-axis maneuver from $(0°, 0°, 0°)$ to $(45°, 0°, 0°)$.

moving L_0-controller, the vector $\mathbf{h}_N$ was chosen such that it would cost less for the control inputs to occur between $t = 28$ s and $t = 42$ s, whereas for the two other controllers, it is equally expensive for the control inputs to occur over the whole time interval. Fig. 6.5 and Fig. 6.6 show that the control inputs produced by the moving L_0-controller occur at $t = 28$ s and $t = 42$ s, and Fig. 6.3 and Fig. 6.4 show that the spacecraft's states change within this interval. Because it is cheaper for the control input to occur within the time interval specified by $\mathbf{h}_N$, it could make sense that the responses and control signal from the moving L_0-controller are different from the responses and control signals for the two other controllers. This finding suggest that the moving L_0-controller is able to move control inputs to a user-defined interval.

Fig. 6.5 and Fig. 6.6 show that the optimal control signal computed by the

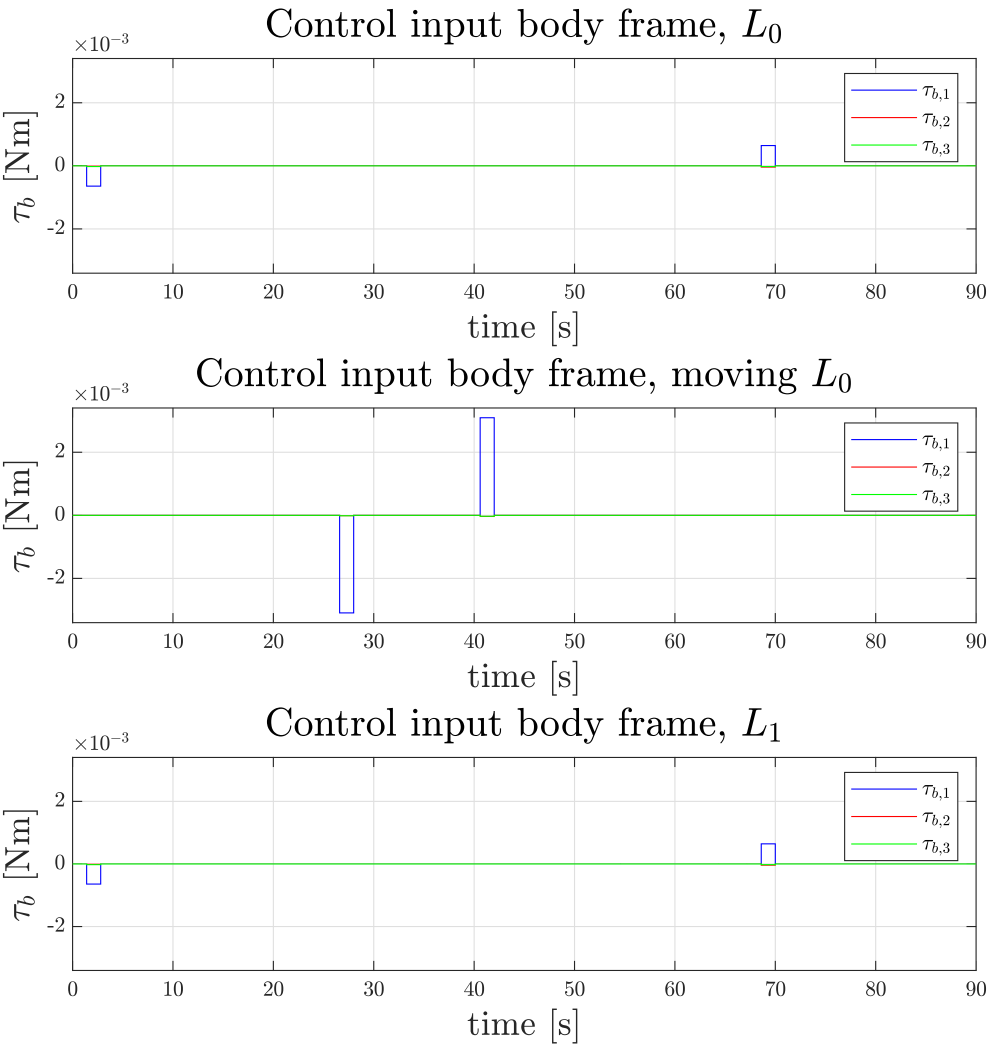


Figure 6.5: Control input body frame for a single-axis maneuver from $(0°, 0°, 0°)$ to $(45°, 0°, 0°)$.

moving L_0-controller has larger amplitude than the control signals produced by the two other controllers. A reasonable explanation for this may be that since the moving L_0-control torques occur closer in time, the torque applied at each of the two time instants has to be larger in order to steer the spacecraft to the desired orientation within a smaller time interval. The control input that occurs at $t = 28$ s pushes the spacecraft towards the desired orientation, whereas the control input at $t = 42$ s slows the spacecraft down so that it stays at the desired orientation and stops spinning, i.e., the body-frame stops rotating relative to the orbit-frame. To reach the desired position in a short time interval, the control torques applied to the spacecraft have to be larger than if the time interval was longer. However, if the time interval was longer, the control inputs could be smaller as the spacecraft would have more time to rotate towards the

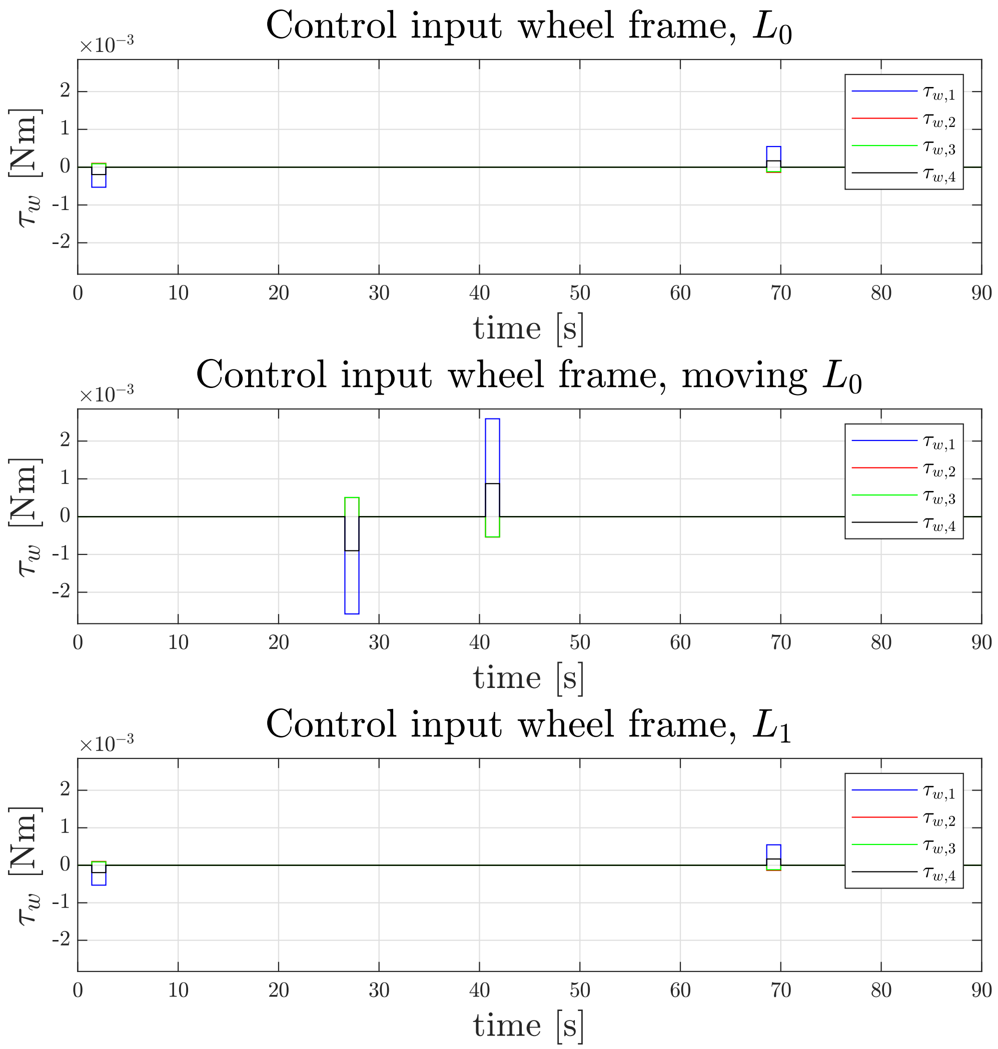
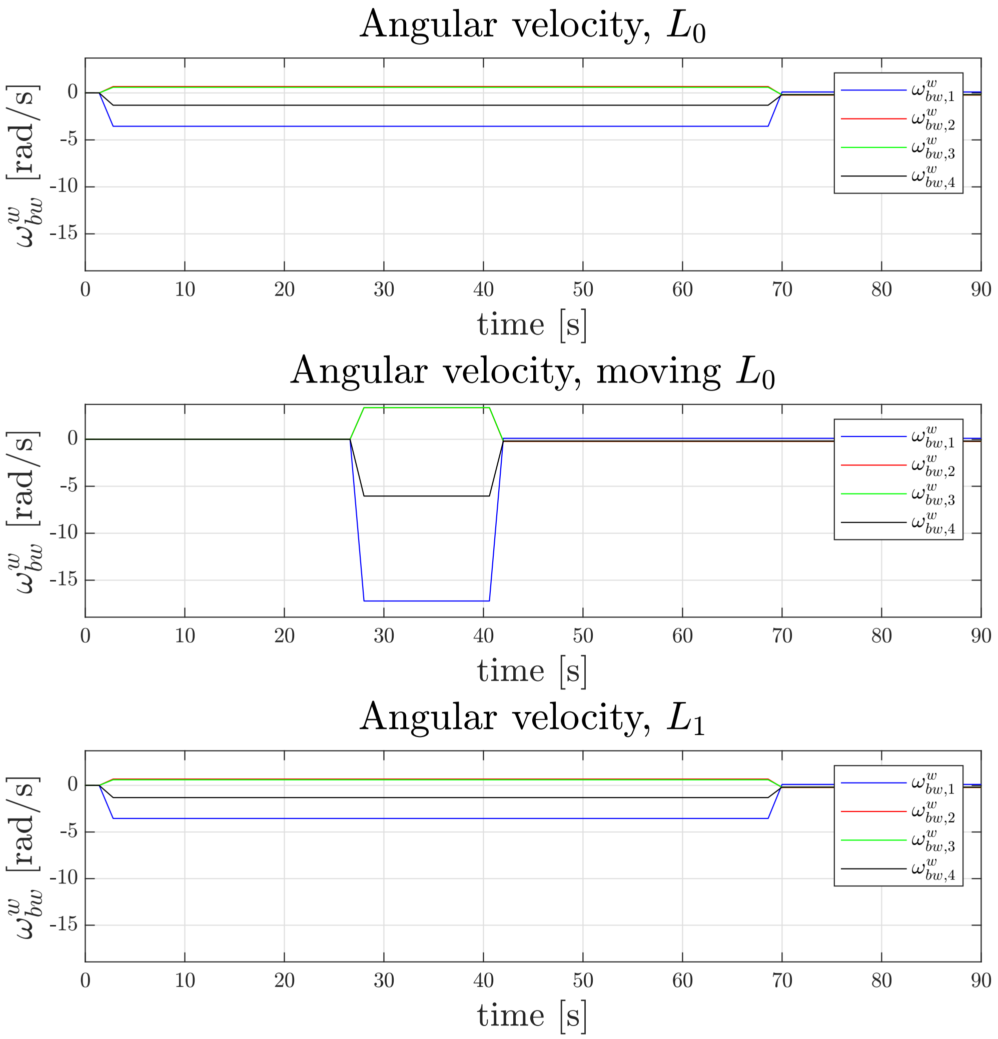


Figure 6.6: Control input wheel frame for a single-axis maneuver from $(0°, 0°, 0°)$ to $(45°, 0°, 0°)$.

desired orientation after the initial control input has been applied. This finding indicates that the moving L_0-controller may require more energy than the two other controllers, because it requires more energy to produce a large control torque than a small control torque.

Fig. 6.5 and Fig. 6.6 show that the control inputs produced by the moving L_0-controller occur at $t = 28$ s and $t = 42$ s, which means that they occur exactly at the boundaries of the time interval specified by $\mathbf{h}_N$. The saturation limits for the control torque are $\tau_{\text{limit}} = \pm 3 \cdot 10^{-3}$ N·m, and Fig. 6.6 shows that the control torques produced by the moving L_0-controller are relatively close to the saturation limits. As outlined previously, for the spacecraft to reach the desired attitude in a short time interval, the control torques applied to it have to be larger than if the time interval was longer. Thus, if the control torques had


Angular velocity, L_0

Angular velocity, moving L_0

Angular velocity, L_1

Figure 6.7: Angular velocity, $\boldsymbol{\omega}_{bw}^{w}$, for a single-axis maneuver from $(0°, 0°, 0°)$ to $(45°, 0°, 0°)$.

occurred at other time instants in the interval specified by $\mathbf{h}_N$, they would have occurred closer in time, and the torques would therefore have larger values. The control torques are already close to the saturation limits when they occur at $t = 28$ s and $t = 42$ s, and if the torque values were to increase, the reaction wheels might saturate. If the reaction wheels saturate, an additional control torque may be required to perform the spacecraft maneuver, as discussed in Section 5.5.3, and an additional control torque would yield a less sparse control signal. It is cheaper for the control input to occur between $t = 28$ s and $t = 42$ s, but if the control torques are too close in time, they may saturate, and the controller would have to apply an additional control torque. Hence, the most optimal solution may be to apply the control torques as far apart in time as possible, but still within the time interval specified by $\mathbf{h}_N$. This is because it is cheaper for the control inputs to occur within this interval and they are not so close in time that they will saturate. Therefore, it makes sense that the control torques occur at the borders of the time interval specified by $\mathbf{h}_N$, i.e., at $t = 28$ s and $t = 42$ s.

Table 6.3 shows that all three controllers yield optimal control signals which have the same sparsity, i.e., the number of nonzero control intervals are 2 for all controllers. This finding suggests that all three controllers are able to find the sparsest solution. For a spacecraft single-axis maneuver, it may not be possible to find a control signal which is sparser than 2, as one control torque has to push the spacecraft towards the desired attitude and one control input has to stop the spacecraft rotation. If the spacecraft was operating inside the Earth's atmosphere, it might be possible to obtain a sparser control signal, as it may be possible to apply control torque to the spacecraft at only one instant. This control torque would push the spacecraft towards its desired attitude, and instead of applying a second control torque to stop the spacecraft, it could be possible to rely on the air resistance to slow the spacecraft down. Relying on the air resistance to slow the spacecraft down would require a relatively long time horizon since the air resistance does not affect the spacecraft much in orbit. Air resistance is less important as the altitude of the spacecraft increases, and therefore a control torque would be needed to stop the spacecraft from rotating when the spacecraft is operating at a higher altitude. Hence, a control signal in which control torques occur at two different time instants provides the sparsest control signal for the spacecraft single-axis maneuver in this thesis.

When trying to maneuver the spacecraft an angle $\phi = 45°$ about the x-axis, one might expect an optimal controller to yield control torque simply about the body-frame x-axis. Fig. 6.5 shows that control torques are applied about all three axes, although the control torque about the x-axis is the most prominent. The optimal control algorithms yield torque about all three axes because the spacecraft rotates relative to its orbit at the same time as it orbits the Earth. To put it more mathematically, the body-frame $\{b\}$ rotates relative to the orbit-frame $\{o\}$, at the same time as the orbit-frame $\{o\}$ also rotates relative to the inertial frame $\{i\}$. If the spacecraft is not operating in orbit, which would be the case if the spacecraft operated in deep-space, then the orbit-frame $\{o\}$ would not be well-defined. Then, the angular acceleration of the spacecraft would be represented by the angular acceleration of $\{b\}$ relative to $\{i\}$, and (4.20c) may be rewritten accordingly

$$\dot{\boldsymbol{\omega}}_{ob}^{b} = \dot{\boldsymbol{\omega}}_{ib}^{b}. \tag{6.1}$$

A spacecraft operating in LEO, such as the satellite used for the simulations in this thesis, would rotate relative to its orbit, i.e., $\{b\}$ rotates relative to $\{o\}$. This results in the angular acceleration in (4.20c). For the single-axis maneuver from $(0°, 0°, 0°)$ to $(45°, 0°, 0°)$, one might expect zero rotation about the y- and z-

axis, since θ_d and ϕ_d are both zero. However, Fig. 6.3 shows that θ and ϕ are not entirely zero during the simulations, which confirms that there is rotation about the y- and z-axis. Because of the spacecraft's rotation around the Earth, there will be rotation about the y- and z-axis throughout the optimization horizon, in addition to the control effort made about the x-axis. Therefore, torque from the reaction wheels must be applied to compensate for the drift about the y- and z-axis.

The total inertia matrix for the spacecraft rigid body, $\mathbf{J}$, in (5.12) also contributes to rotation about multiple axes. Due to the nonlinearity of the spacecraft dynamics, the terms are coupled, which results in torques about all three axes, even though motion is only needed about one axis. The total system inertia matrix is not diagonal and is given in (5.12). If a diagonal $\mathbf{J}$-matrix were used instead of the one in (5.12), the states would be less coupled, which is clear from (4.7), (4.8), and (4.20b). Less coupling of the dynamics would yield less torque about the y- and z-axis when a single-axis maneuver is performed about the x-axis.

6.3 Multiple-axis maneuver

Two multiple-axis maneuvers were tested, as discussed in Section 5.3.2, and the results from the experiments are presented and discussed in this section. Section 6.3.1 presents the results for the multiple-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$, and Section 6.3.2 discusses these results. Section 6.3.3 presents the results from the multiple-axis maneuver from $(0°, 0°, 0°)$ to $(75°, 50°, 15°)$, and Section 6.3.4 discusses these results.

6.3.1 Multiple-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$

The first multiple-axis maneuver rotates the spacecraft an angle $\phi = 90°$ about the x-axis, $\theta = 45°$ about the y-axis, and $\psi = 15°$ about the z-axis.

Fig. 6.8 shows the change in attitude over the simulation time, and reveals that the spacecraft reaches the desired attitude. The angular velocity, $\boldsymbol{\omega}_{ob}^{b}$, for the three controllers are shown in Fig. 6.9, and reveals that the spacecraft stops rotating when it reaches the desired attitude. The optimal control input torques in $\{b\}$ and $\{w\}$ are shown in Fig. 6.10 and Fig. 6.11, respectively. Note that the angular velocity of the reaction wheels can be found in Fig. A.1 in Appendix A. Table 6.4 presents the computation times for the different control scenarios, the sparsity for each of the three control signals, and the number of iterations used to find the optimal solution for each of the three control scenarios.

Controller	CPU time NLP, [s]	CPU time IPOPT, [s]	Sparsity	Iterations
L_0-optimal control	66.216	6.106	2	497
Moving L_0-optimal control	90.175	7.706	2	699
L_1-optimal control	5.131	0.360	2	39

Table 6.4: Computation time, sparsity and iterations for a multiple-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$.

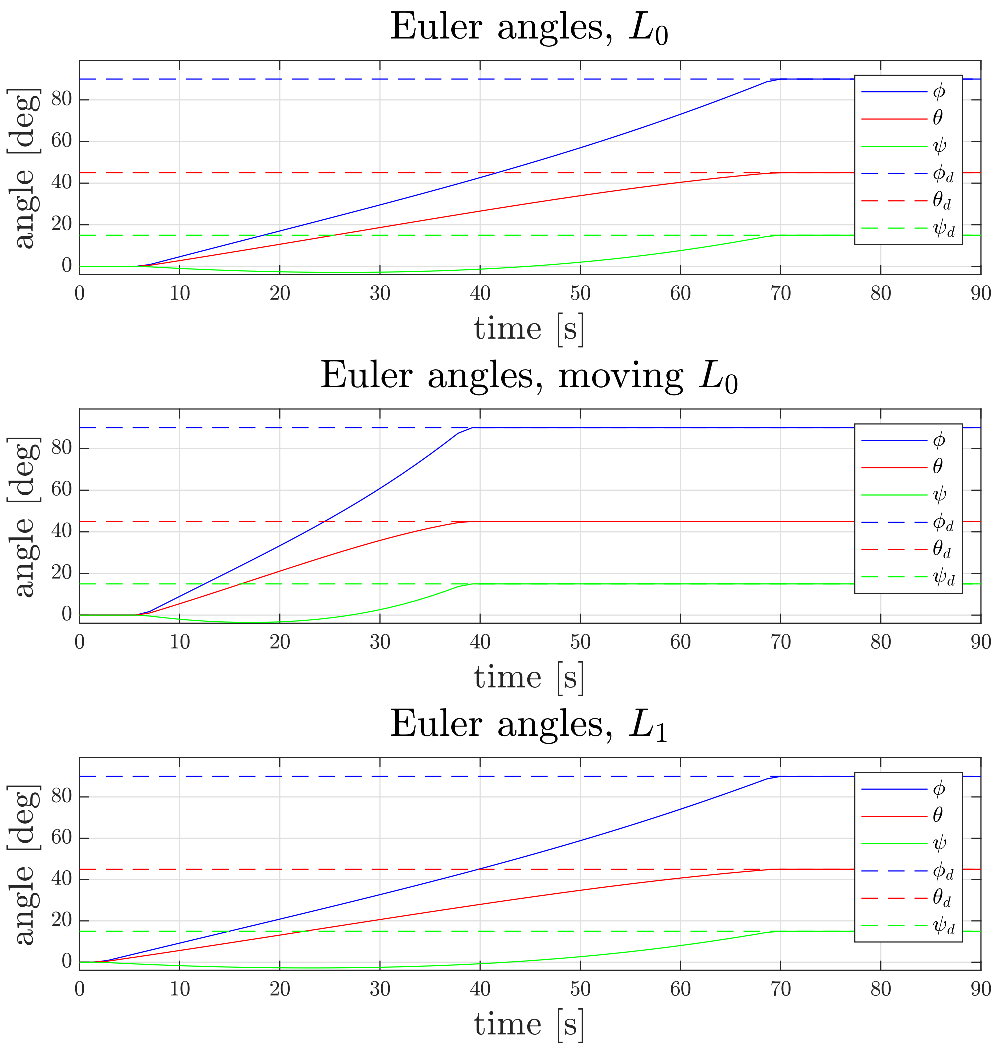


Figure 6.8: Euler angles for a multiple-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$.

6.3.2 Discussion of the multiple-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$

Fig. 6.8 shows that all three controllers are able to steer the spacecraft to the desired orientation of $(90°, 45°, 15°)$. Fig. 6.9 shows that the angular velocity of

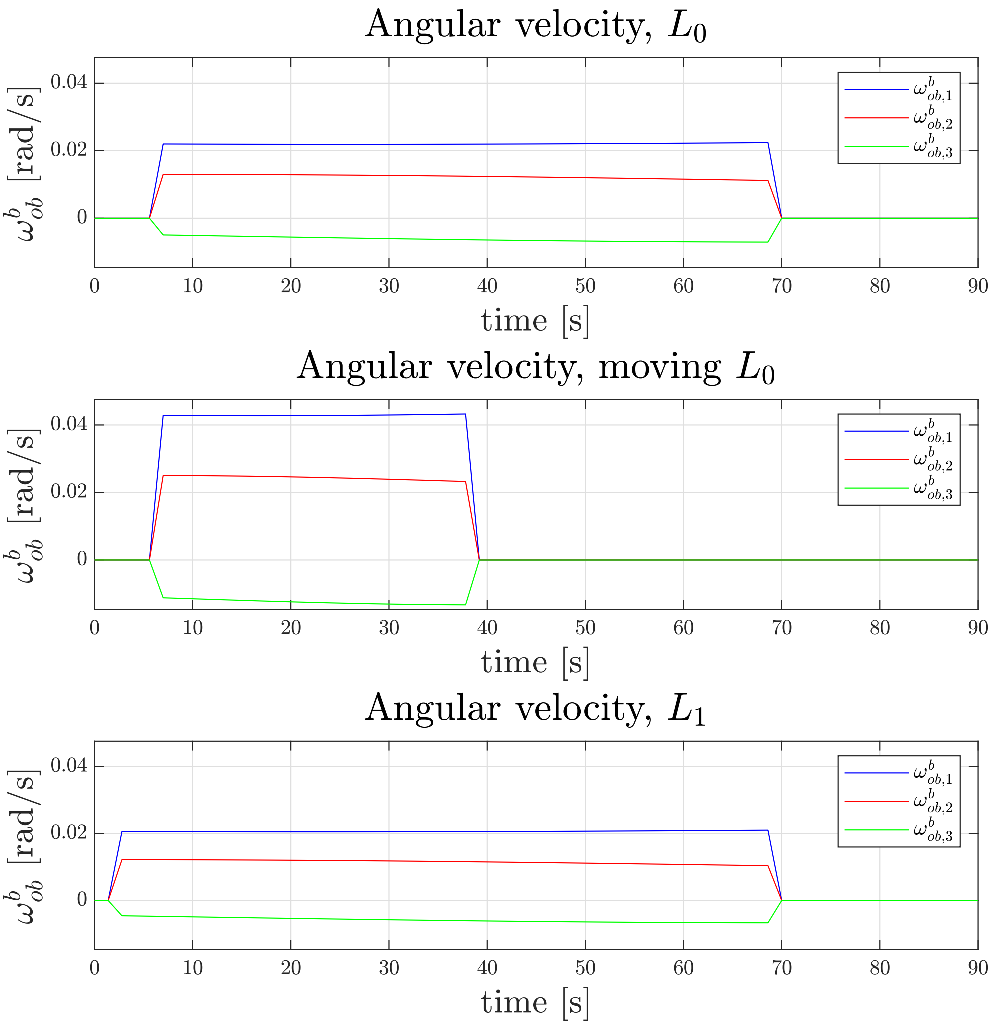


Figure 6.9: Angular velocity, ω^b_{ob}, for a multiple-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$.

$\{b\}$ relative to $\{o\}$ becomes zero, for all three controllers, when the spacecraft reaches the desired attitude. The results in this section support the findings from Section 6.2, which suggested that all three controllers work as intended.

Fig. 6.10 and Fig. 6.11 show that the first control torque from the moving L_0-controller occurs after about $t = 8$ s, and the second control torque occurs close to $t = 40$ s. The second control torque occurs within the interval specified by $\mathbf{h}_N$. The vector $\mathbf{h}_N$ was chosen such that it would cost less for the control inputs to occur between $t = 28$ s and $t = 42$ s. One might have expected all control inputs to occur within this time interval. However, one control input occurs outside this interval. The reason for this is that there are no constraints on where the control input should *not* occur; it only costs less between $t = 28$ s and $t = 42$ s. The optimization procedure aims to satisfy the constraints and reach the final

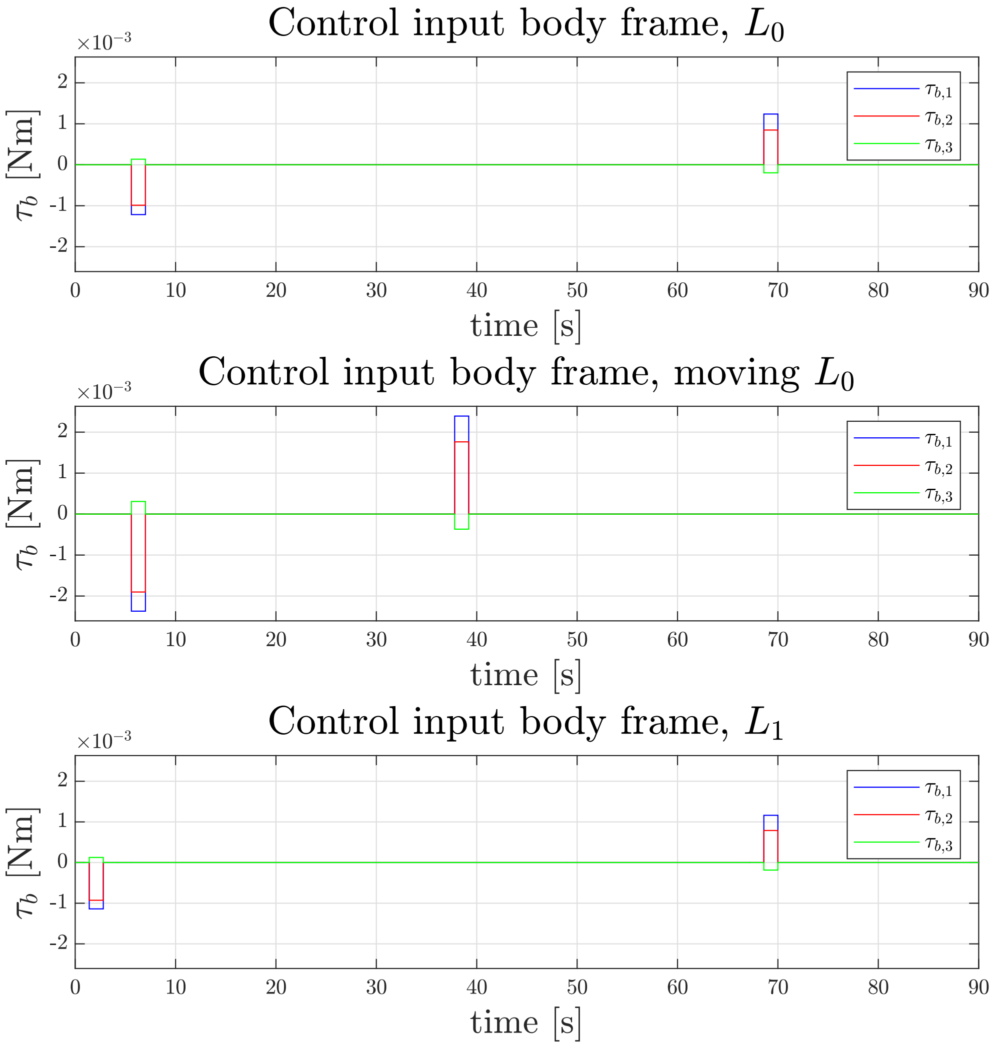


Figure 6.10: Control input body frame for a multiple-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$.

state values while minimizing the cost function. If it is not possible to reach this goal by applying control input within the cheap interval specified by $\mathbf{h}_N$, some or all of the control input will occur outside this interval. Therefore, control inputs may occur outside the interval defined by $\mathbf{h}_N$.

The L_1-optimal control signal and the L_0-optimal control signal are not identical, which can be seen in Fig. 6.10 and Fig. 6.11. The first control input produced by the L_1-controller occurs at $t \approx 2$ s, whereas the first control input produced by the L_0-controller occurs at $t \approx 8$ s. The two last control inputs occur simultaneously for both controllers. The amplitudes of the control torques from the L_1-controller are smaller than those from the L_0-controller. A possible explanation for the difference in amplitudes is that since there is more time between the two L_1-control inputs, the spacecraft will have more time to rotate to

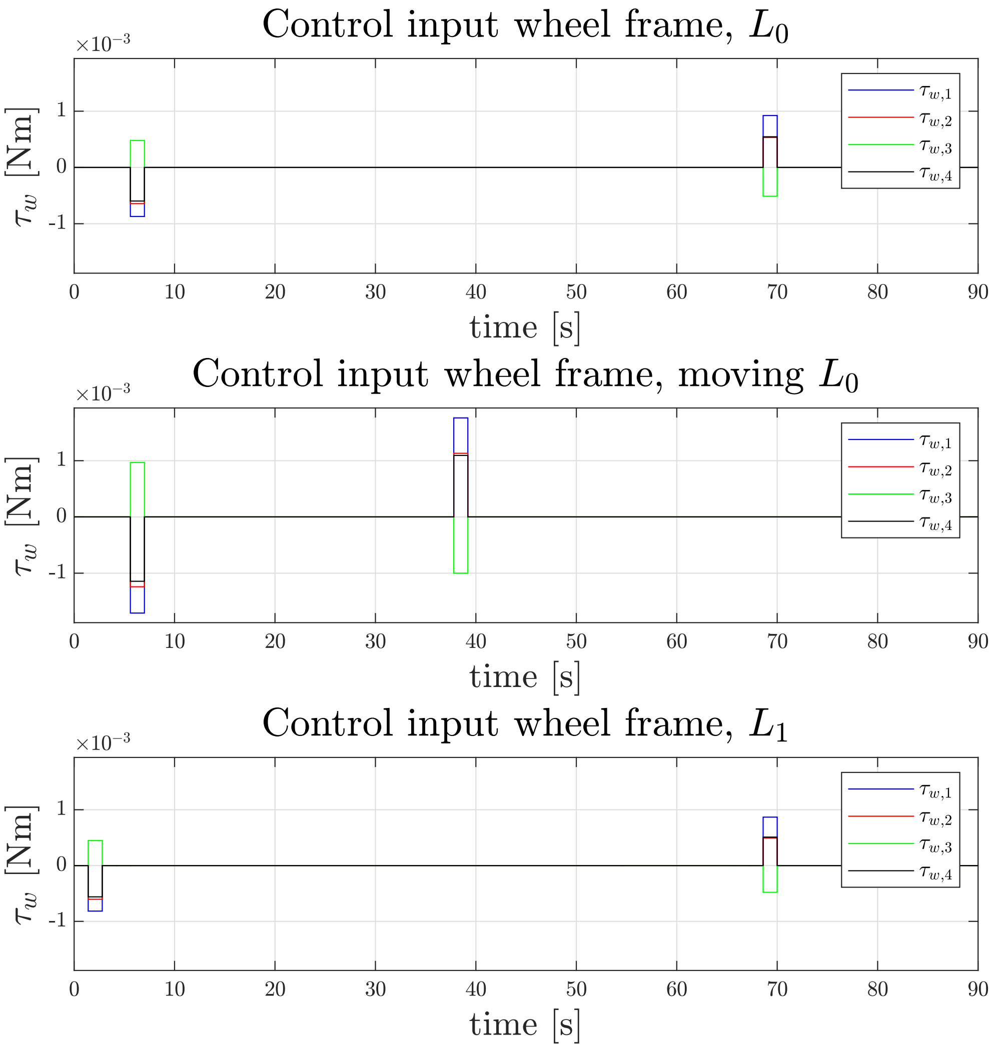


Figure 6.11: Control input wheel frame for a multiple-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$.

the desired orientation, and thus less torque would need to be applied. Therefore, it makes sense that the control torques produced by the L_1-controller are smaller than the control torques produced by the L_0-controller. These results suggest that the L_1-optimal solution does not always equal the L_0-optimal solution. On the other hand, they suggest that the L_1-optimal control problem could be a rather acceptable approximation to the L_0-optimal control problem.

6.3.3 Multiple-axis maneuver $(0°, 0°, 0°)$ to $(75°, 50°, 15°)$

The second multiple-axis maneuver rotates the spacecraft from an initial orientation $(\phi, \theta, \psi) = (0°, 0°, 0°)$ to a final orientation $(75°, 50°, 15°)$.

Table 6.5 gives the computation times for each of the optimal control problems, the sparsity of the optimal control signals in $\{b\}$ for each of the three controllers, and the number of iterations used to solve the optimal control problems using IPOPT. The spacecraft's change in attitude for the different controllers is shown in Fig. 6.12. The angular velocity of the body-frame relative to the orbit-frame is shown in Fig. 6.13. Fig. 6.14 shows the optimal control signals in the body-frame, whereas Fig. 6.15 shows the optimal control signals in the wheel-frame. The angular velocity of the reaction wheels can be found in Fig. A.2 in Appendix A.

Controller	**CPU time NLP, [s]**	**CPU time IPOPT, [s]**	**Sparsity**	**Iterations**
L_0-optimal control	115.960	10.171	2	913
Moving L_0-optimal control	40.436	3.438	2	315
L_1-optimal control	2.947	0.221	4	23

Table 6.5: Computation time, sparsity and iterations for a multiple-axis maneuver from $(0°, 0°, 0°)$ to $(75°, 50°, 15°)$.

6.3.4 Discussion of the multiple-axis maneuver from $(0°, 0°, 0°)$ to $(75°, 50°, 15°)$

The L_1-optimal control signal is less sparse than the L_0-optimal control signal for the for the multiple-axis maneuver from $(0°, 0°, 0°)$ to $(75°, 50°, 15°)$, as shown in Table 6.5. This is an interesting result as it confirms what was suggested in Feng et al. (2016), i.e., that L_1-optimal solutions provide suboptimal solutions to the L_0-optimal control problem. Although this finding is interesting, the difference in sparsity for the two control signals might occur due to the implementation of the optimal control problems and the numerical approach taken to solve them. The results presented in this section were found using direct optimization, whereas indirect optimization was used in the work by Nagahara et al. (2015), which suggested that L_1-optimal control could be used to approximate L_0-optimal control.

For the moving L_0-controller, only one of the two control torques occur within the time interval specified by $\mathbf{h}_N$ from $t = 28$ s to $t = 42$ s, as shown in Fig. 6.14 and Fig. 6.15. As discussed before, this finding illustrates that even though it is cheaper for the control inputs to occur within the specified interval, this does not guarantee that they will occur here. Additional constraints may be required in the moving L_0-optimal control problem to prevent the control inputs from occurring outside the desired time interval.

By comparing the moving L_0-optimal control signals in Fig. 6.10 and Fig. 6.14, it is clear that for both control signals only one control torque occur with the time interval specified by $\mathbf{h}_N$ from $t = 28$ s to $t = 42$ s, whereas the remaining control torques occur outside this interval. This finding supports what has been

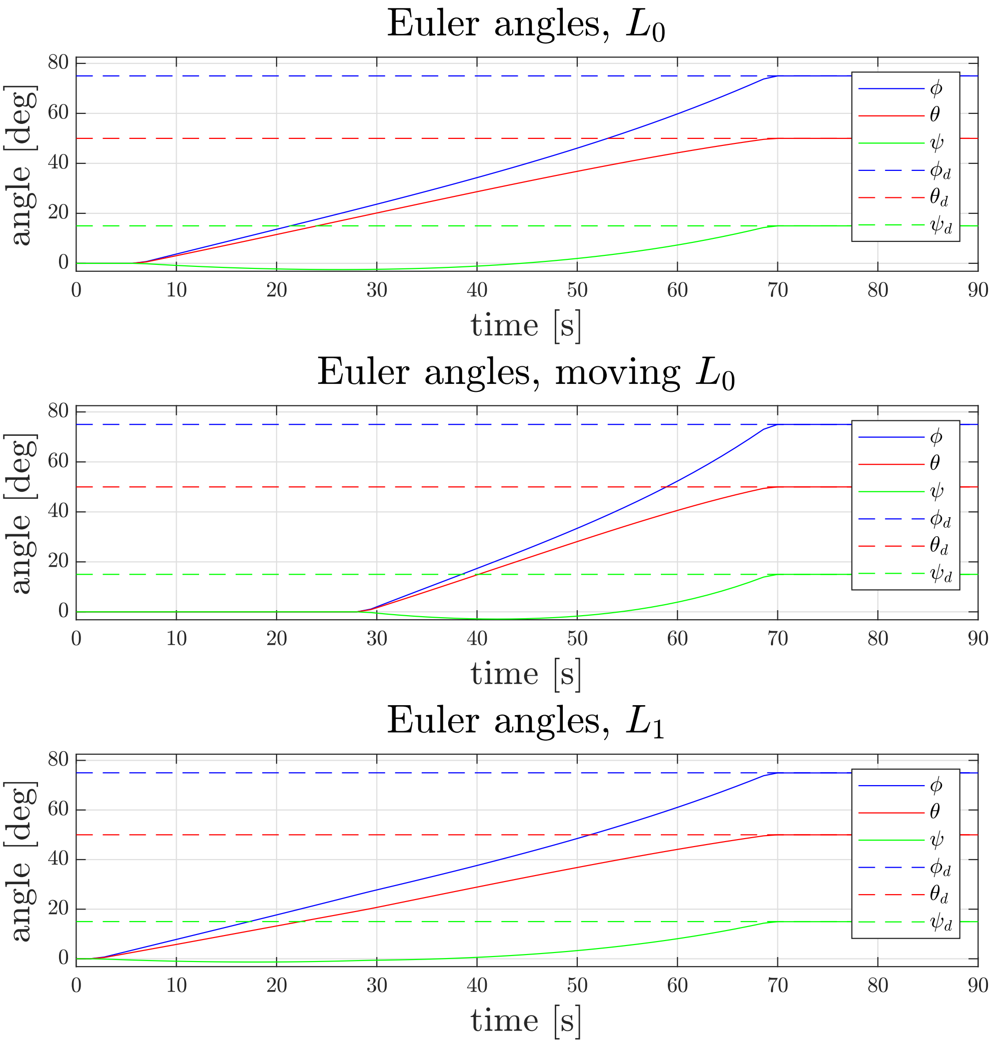


Figure 6.12: Euler angles for a multiple-axis maneuver from $(0°, 0°, 0°)$ to $(75°, 50°, 15°)$.

stated previously in that even though it is cheaper for the control inputs to occur within the specified interval, this does not guarantee that they will occur here. Fig. 6.10 reveals that for the maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$, the control torque occurring outside the interval specified by $\mathbf{h}_N$ is applied at the beginning of the simulation, at about $t = 8$ s. Contrary, Fig. 6.14 shows that for the maneuver from $(0°, 0°, 0°)$ to $(75°, 50°, 15°)$ the control torque occurring outside the interval specified by $\mathbf{h}_N$ is applied at the end of the simulation, close to $t = 70$ s. This finding suggests that if two control torques are required to perform a maneuver, it is possible to move both the first and the second control torque to within the time interval specified by $\mathbf{h}_N$.

The computation times for the L_1-controller are different for the two multiple-axis maneuvers, and smaller in the case for the maneuver from $(0°, 0°, 0°)$ to

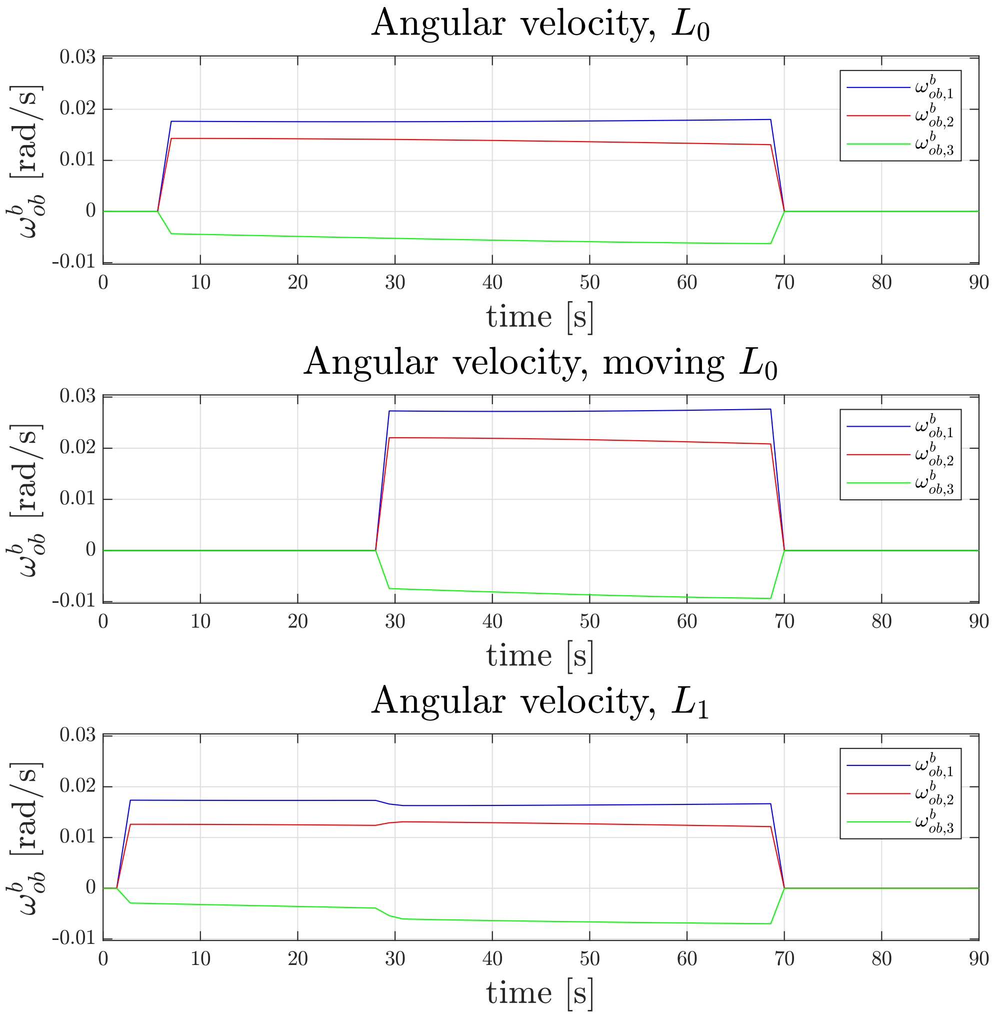


Figure 6.13: Angular velocity, ω^b_{ob}, for a multiple-axis maneuver from $(0°, 0°, 0°)$ to $(75°, 50°, 15°)$.

$(75°, 50°, 15°)$, which can be seen in Table 6.4 and Table 6.5. One possible explanation to why the computation time is less for the maneuver from $(0°, 0°, 0°)$ to $(75°, 50°, 15°)$ is that the optimization solver gets stuck in a local minimum and stops searching for a more optimal solution. Comparing Table 6.5 and Table 6.4 reveals that the L_1-optimal control signal for the maneuver from $(0°, 0°, 0°)$ to $(75°, 50°, 15°)$ is less sparse than the L_1-optimal control signal for the maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$. This finding may also suggest that the solver has stopped in a local minimum while searching for the optimal solution to the L_1-optimal control problem for the maneuver from $(0°, 0°, 0°)$ to $(75°, 50°, 15°)$. Another possible explanation to why the computation time is less for the maneuver from $(0°, 0°, 0°)$ to $(75°, 50°, 15°)$ is that the initial guesses provided for the optimization is closer to the optimal solution or the local minimum, com-

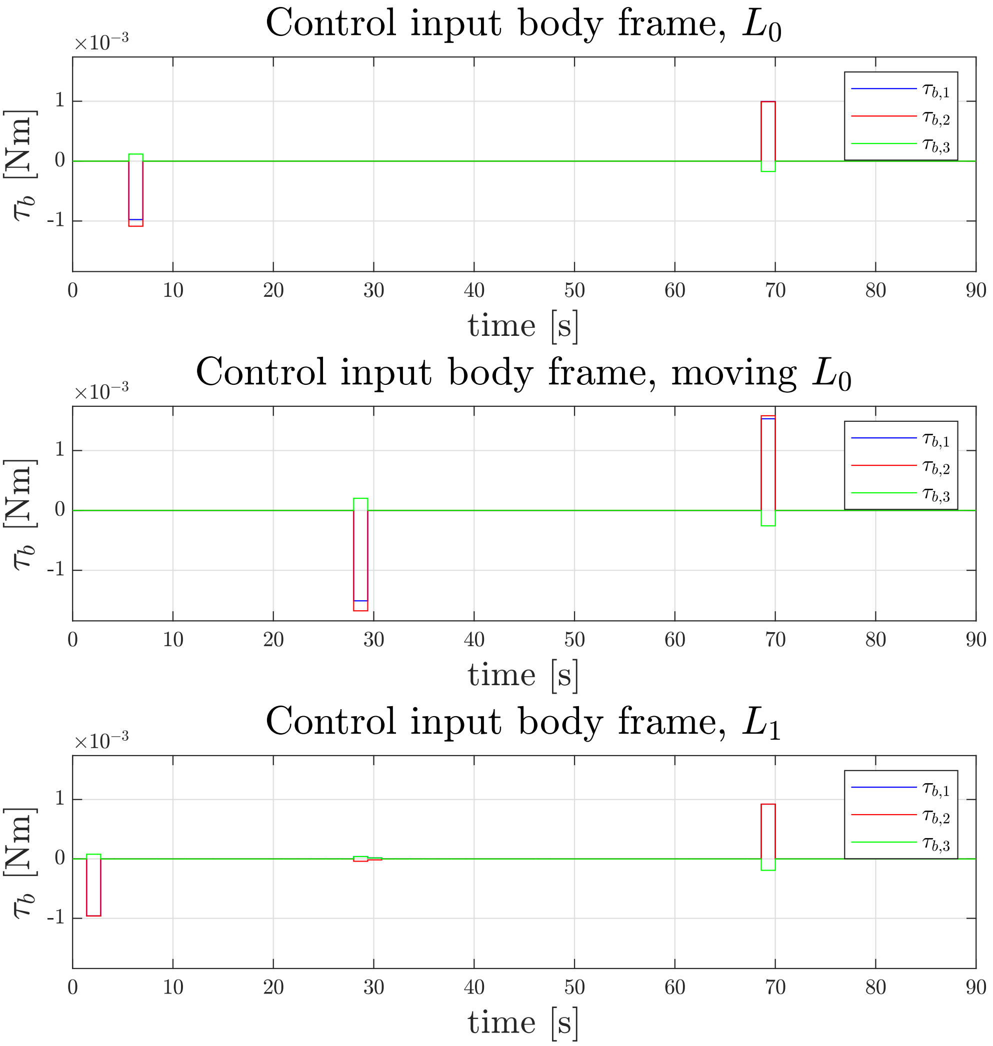


Figure 6.14: Control input body frame for a multiple-axis maneuver from $(0°, 0°, 0°)$ to $(75°, 50°, 15°)$.

pared to the initial guesses provided for the optimization of the maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$.

6.4 Path-following maneuver

This section presents and discusses the results from the experiments dealing with path-following, as explained in Section 5.3.3. First, Section 6.4.1 presents the results from the path-following maneuver where multiple optimizations are used to find the optimal solutions. The results presented in Section 6.4.1 are discussed in Section 6.4.2. Then, in Section 6.4.3, the results from the path-following maneuver where an extra quaternion term is added to the cost func-

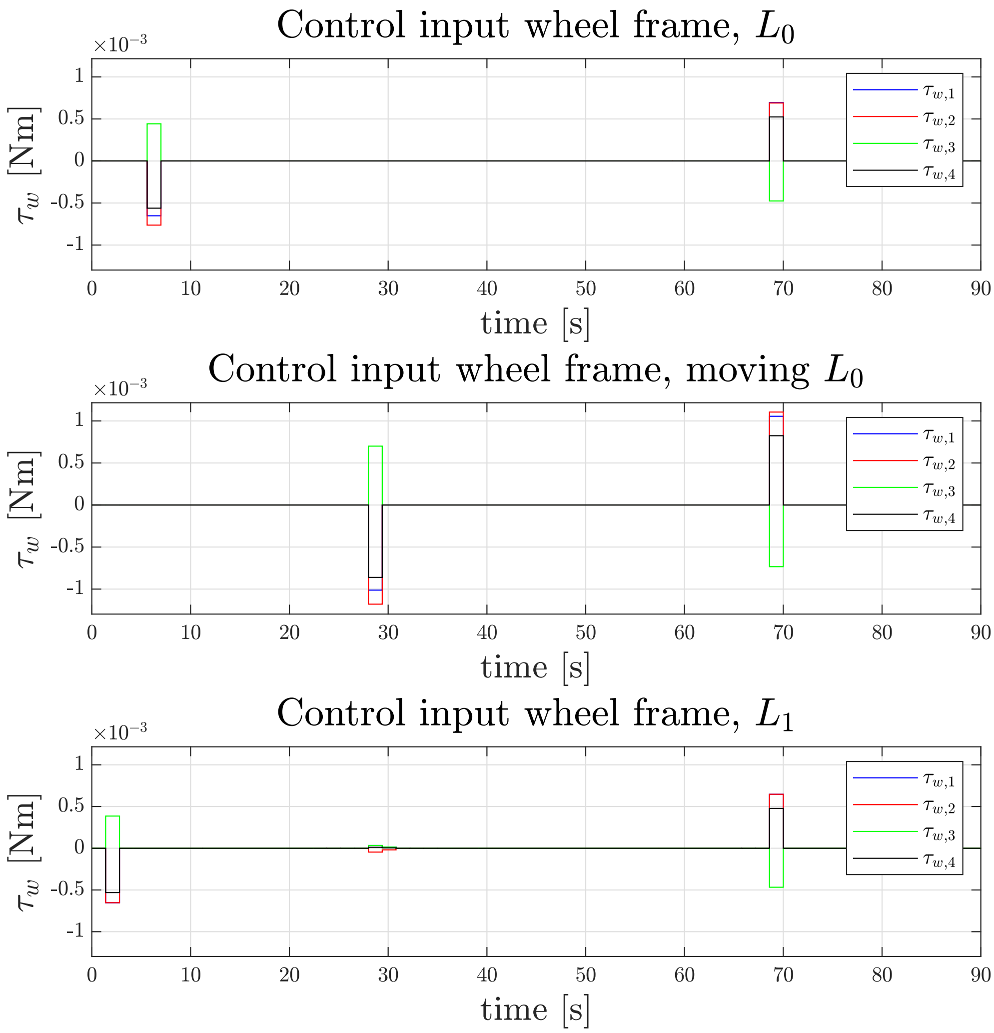


Figure 6.15: Control input wheel frame for a multiple-axis maneuver from $(0°, 0°, 0°)$ to $(75°, 50°, 15°)$.

tions are presented, followed by a discussion of these results in Section 6.4.4. For both path-following approaches, a maneuver from $(\phi, \theta, \psi) = (0°, 0°, 0°)$ through $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$ is performed.

6.4.1 Path-following using multiple optimizations

The results for the path-following maneuver from $(\phi, \theta, \psi) = (0°, 0°, 0°)$ through $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$, where multiple optimizations are used to find the optimal solutions are presented in this section. Using multiple optimizations to solve the optimal control problem means that first, the maneuver from $(0°, 0°, 0°)$ to $(30°, 45°, 15°)$ is optimized, and then the maneuver from $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$ is optimized.

The sparsity of the optimal control signals are shown in Table 6.6, whereas the computation times used for each of the three controllers, for each of the two optimizations, are shown in Table 6.7. The number of iterations used to find the optimal solution are also shown in Table 6.7. Fig. 6.16 shows the attitude trajectories of the spacecraft, and illustrates that the spacecraft is able to follow the predefined path. Fig. 6.17 shows the angular velocity of the body-frame relative to the orbit-frame, $\boldsymbol{\omega}_{ob}^{b}$. The control signals computed in the body-frame are found in Fig. 6.18, whereas the control signals computed in the wheel-frame are shown in Fig. 6.19. The angular velocity of the reaction wheels can be found in Fig. A.3 in Appendix A.

Controller	**Sparsity**
L_0-optimal control	4
Moving L_0-optimal control	6
L_1-optimal control	4

Table 6.6: Sparsity for path-following using multiple optimizations, from $(0°, 0°, 0°)$ through $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$.

Controller	**Optim. no.**	**CPU time NLP, [s]**	**CPU time IPOPT, [s]**	**Iterations**
L_0-optimal control	1	49.505	4.108	358
L_0-optimal control	2	97.399	8.897	717
Total:	-	146.904	13.005	1075
Moving L_0-optimal control	1	88.086	7.775	642
Moving L_0-optimal control	2	147.574	13.009	1073
Total:	-	235.660	20.784	1715
L_1-optimal control	1	2.959	0.171	22
L_1-optimal control	2	3.139	0.265	24
Total:	-	6.098	0.436	46

Table 6.7: Computation time and iterations for path-following using multiple optimizations, from $(0°, 0°, 0°)$ through $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$.

6.4.2 Discussion for the path-following using multiple optimizations

At first, the initial guesses for the L_0- and moving L_0-controller were the output from the L_1-controller performing a path-following maneuver from $(0°, 0°, 0°)$ through $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$, but these initial guesses yielded poor results. In other words, the spacecraft did not reach the desired attitude and angular velocity when applying the output from the L_1-controller as initial guesses for the L_0- and moving L_0-controller. Therefore, different initial guesses than what were first used were applied to the L_0- and moving L_0-controller. The new

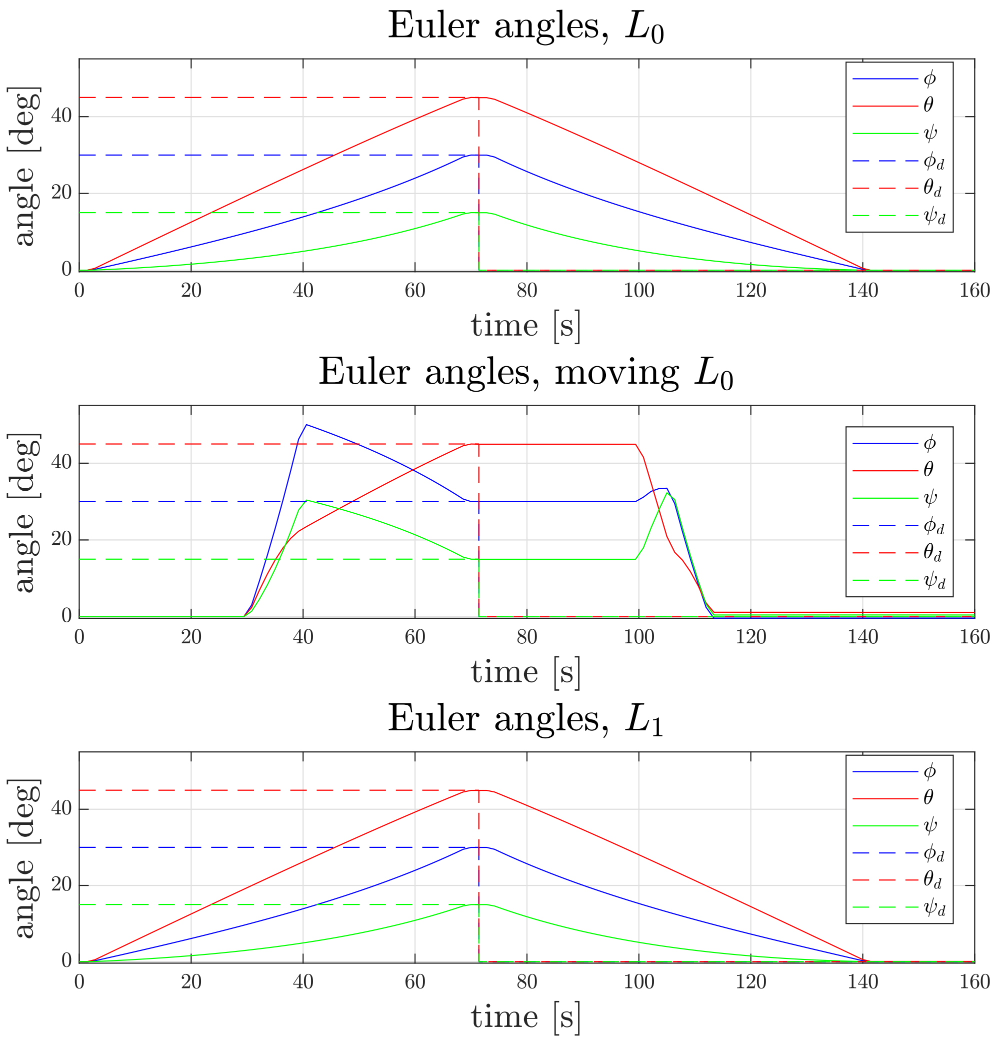


Figure 6.16: Euler angles for path-following using multiple optimizations, from $(0°, 0°, 0°)$ through $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$.

initial guess applied for the L_0-controller was the output from the L_1-controller performing a single-axis maneuver from $(0°, 0°, 0°)$ to $(75°, 0°, 0°)$. The new initial guess applied for the moving L_0-controller was the output from the L_1-controller performing a single-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 0°, 0°)$. From Fig. 6.16 and Fig. 6.17 it is clear that the spacecraft follows the path and stops rotating for these initial guesses, although they were not expected to yield better results than the original initial guesses. A possible explanation to why the first initial guesses did not yield satisfactory results is that the optimization solver might have become stuck in a local minimum, and is therefore not able to find the optimal solution. The first initial guesses may have provided a starting point close to a local minimum, and the optimization stops at this point. Contrary, the second initial guesses may have provided a starting point

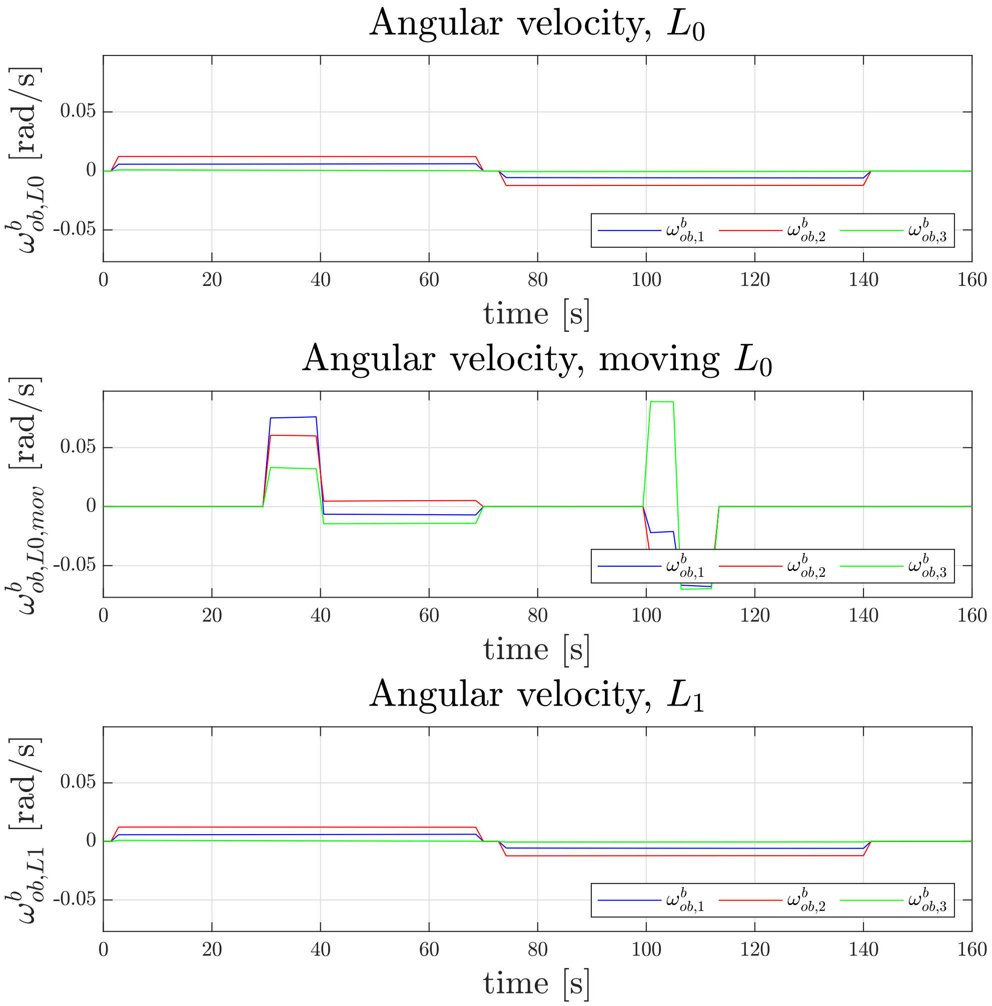


Figure 6.17: Angular velocity, ω^b_{ob}, for path-following using multiple optimizations, from $(0°, 0°, 0°)$ through $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$.

closer to a better solution. This confirms that the optimization solver, IPOPT, is sensitive to the choice of initial guesses. The results also illustrates the importance of choosing good initial guesses for the L_0- and moving L_0-optimal control problems. The sensitivity of the L_0-optimal control problem to different initial guesses was explored in Section 6.1.1.

Each of the two subsequent optimizations lasts for $t = 70$ s, and the orientation trajectories resulting from the L_1-controller and L_0-controller are symmetrical about $t = 70$ s, excluding the propagation time at the end of the second optimization. The same goes for the control signals computed by the two controllers. One explanation to why the symmetrical results make sense, is because the same amount of torque is required to move the spacecraft from $(0°, 0°, 0°)$ to $(30°, 45°, 15°)$, and from $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$. Fig. 6.18 and Fig. 6.19 show that the control torques for the two optimizations have the same size, and

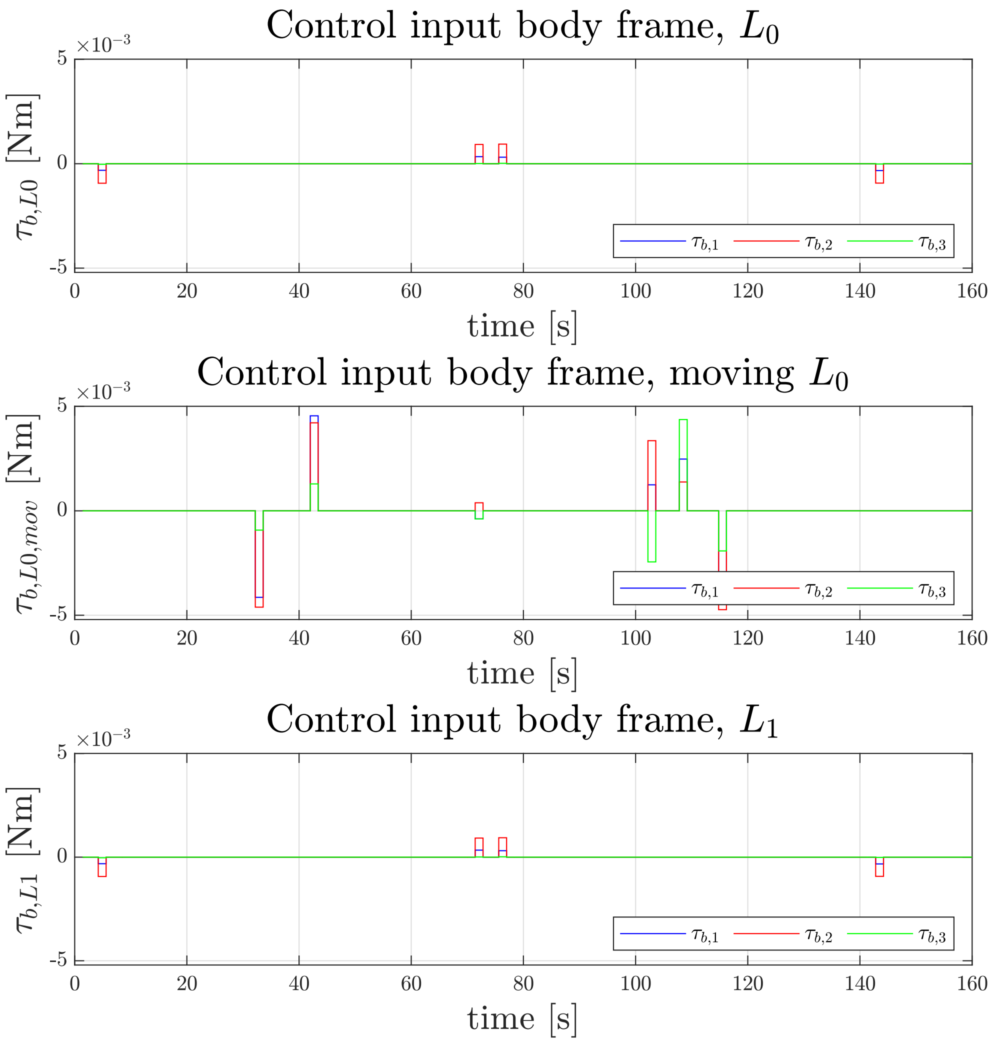


Figure 6.18: Control input body frame for path-following using multiple optimizations, from $(0°, 0°, 0°)$ through $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$.

thus supports the suggested explanation.

On the other hand, the results for the moving L_0-controller are rather asymmetrical. One way to explain the asymmetrical results for the moving L_0-controller could be that the choice of $\mathbf{h}_N$ makes it cheaper to apply control torques if they occur within $t = 28$ s and $t = 42$ s for the moving L_0-controller for each of the two optimizations. This means within $t = 28$ s to $t = 42$ s and $t = 98$ s to $t = 112$ s in the plots. For the first maneuver from $(0°, 0°, 0°)$ to $(30°, 45°, 15°)$, two control inputs occur within this interval and one occur at the end of the optimization interval. A close-up study of the moving L_0-control signals in Fig. 6.18 and Fig. 6.19 reveals that the first and second control torques have slightly different magnitudes. Therefore an additional control input has to be applied to compensate for the difference between the two first control torques, i.e., the spacecraft

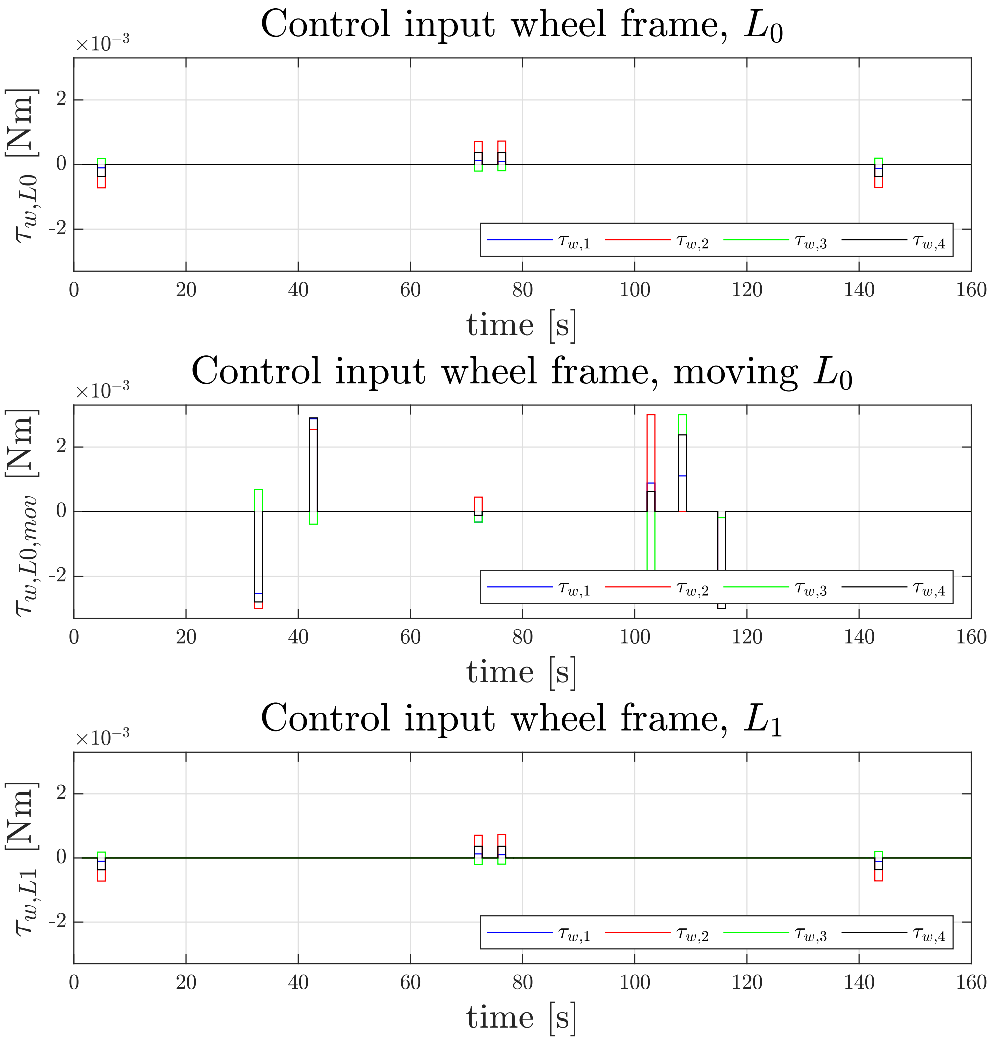


Figure 6.19: Control input wheel frame for path-following using multiple optimizations, from $(0°, 0°, 0°)$ through $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$.

requires the same amount of torque to stop spinning as it requires to start. For the second optimization, i.e., the maneuver from $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$, the control torques occur at three instants, all within the interval $t = 98$ s to $t = 112$, which differs from the first maneuver. One might have expected that the control inputs would occur at the same time instants for the optimizations for both maneuvers, considering they are opposite maneuvers and the optimization is performed over the same number of control intervals. Contrary to this, the moving L_0-control signals in Fig. 6.18 and Fig. 6.19 are asymmetrical. For each of the optimizations, i.e., the optimization of the maneuver from $(0°, 0°, 0°)$ to $(30°, 45°, 15°)$ and the optimization from $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$, the optimization procedure aims to satisfy the constraints and reach the final state values while minimizing the cost function. If it is not possible to reach this goal by applying control input within the cheap interval specified by $\mathbf{h}_N$, some or all of the control input will occur outside this interval. Therefore, control inputs may occur outside the interval defined by $\mathbf{h}_N$. This, together with the fact that the two maneuvers are different, could explain why the control signals differs for the first and second part of the path-following maneuver for the moving L_0-controller. Another possible explanation to why the first and second part of the path-following maneuver for the moving L_0-controller differ, is that the controller behave differently based on initial guesses, as discussed in Section 6.1.1.

For the moving L_0-controller and the second optimization, i.e., the maneuver from $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$, control torques occur at three instants, all within the interval $t = 98$ s to $t = 112$ as outlined above. The control torques that occur at $t \approx 102$ s and the control torques that occur at $t \approx 112$ s make the reaction wheels saturate, as shown in Fig. 6.19. An explanation to why the actuators saturated could be that all the control torques occur within an interval of $\Delta t \approx 14$ s for the second optimization, compared to an interval of $\Delta t \approx 40$ s for the first optimization. As discussed in Section 6.2.1; if the control torques occur within a small time interval the values of the torques have to be larger than if there was more time between each control torque, in order to perform a specified maneuver. For the first optimization, i.e., for the maneuver from $(0°, 0°, 0°)$ to $(30°, 45°, 15°)$, the two first control inputs are close to the saturation limits of $\tau_{\text{limit}} = \pm 3 \cdot 10^{-3}$ N·m. However, for the first optimization, a third control input occurs at the end of the optimization interval. Although the control torques occur within a relatively large time interval for the first maneuver, two of them are close to the saturation limits. The spacecraft has to rotate the same distance for the two maneuvers, i.e., from $(0°, 0°, 0°)$ to $(30°, 45°, 15°)$, and from $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$. Since the control torques applied to steer the spacecraft from $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$ occurs closer in time than for the first maneuver, it could make sense that the actuators saturate. If the control torques were even closer together, yet another control torque might be required to perform the maneuver as discussed in Section 5.5.3.

6.4.3 Path-following using an extra quaternion in the cost function

The results for the path-following maneuver from $(\phi, \theta, \psi) = (0°, 0°, 0°)$ through $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$, where an extra quaternion term is added to the cost functions, are presented in this section.

The computation times for the different control scenarios, the sparsity for each of the control signals, and the number of iterations used to find the optimal solution for each of the three controllers are shown in Table 6.8.

Fig. 6.20 shows the change in the spacecraft's attitude over time, and reveals that all three controllers are able to steer the spacecraft through the predefined path. The angular velocity of $\{b\}$ relative to $\{o\}$, represented in $\{b\}$, $\boldsymbol{\omega}^b_{ob}$, is shown in Fig. 6.21. The optimal control input in the body-frame can be seen in Fig. 6.22, and the torques represented in the wheel-frame are shown in Fig. 6.23. The angular velocity of the reaction wheels, $\boldsymbol{\omega}^w_{bw}$, can be seen in Fig. A.4 in Appendix A.

Controller	**CPU time NLP, [s]**	**CPU time IPOPT, [s]**	**Sparsity**	**Iterations**
L_0-optimal control	81.583	6.937	3	627
Moving L_0-optimal control	104.913	9.173	3	829
L_1-optimal control	3.705	0.275	3	29

Table 6.8: Computation time, sparsity and iterations for path-following using an extra quaternion in the cost function, from $(0°, 0°, 0°)$ through $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$.

6.4.4 Discussion for the path-following using an extra quaternion in the cost function

The initial guess for the L_0-controller were the output from the L_1-controller for a single axis maneuver from $(0°, 0°, 0°)$ to $(90°, 0°, 0°)$. This initial guess yielded better results than using an initial guess from the L_1-controller for a path-following maneuver from $(0°, 0°, 0°)$ through $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$, similarly to the observations made for the path-following maneuver in Section 6.4.1. As mentioned previously, a possible explanation to why the initial guess from the path-following maneuver did not yield satisfactory results may be that the optimization solver could have become stuck in a local minimum, and is therefore not able to find the optimal solution.

For the moving L_0-controller, the vector $\mathbf{h}_N$ was chosen such that it would cost less for the control input to occur between $t = 10$ s and $t = 20$ s, and $t = 49$ s and $t = 59$ s. Fig. 6.22 and Fig. 6.23 show that no control inputs occur within these time intervals. A reason for this could be that even though it is cheaper to provide control inputs within these two intervals, doing so may not minimize the cost function in (5.9b) while satisfying the constraints. In other words, to minimize the cost function, it is more important to minimize the terms in the cost function which relate to attitude and angular velocity, than it is to move the control inputs to inside the predefined intervals. The control inputs occur outside $t = 10$ s and $t = 20$ s, and $t = 49$ s and $t = 59$ s, which is acceptable behaviour, since there are no terms in the moving L_0-optimal control problem that prevents the control torques from occurring outside the cheapest time intervals. Thus, this result does not indicate that the moving L_0-controller does not work, but it illustrates that the control inputs are only moved to the predefined interval if it contributes to minimizing the objective function while satisfying the constraints.

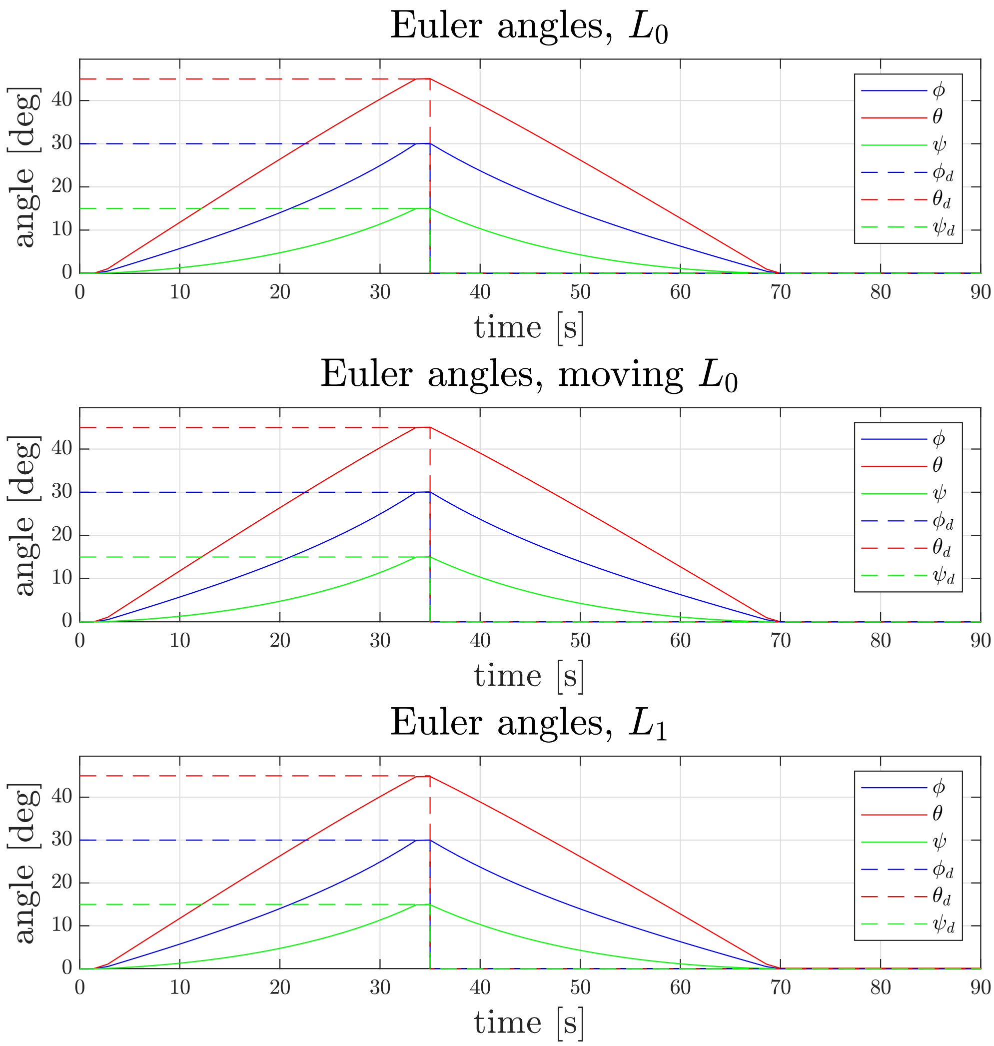


Figure 6.20: Euler angles for path-following using an extra quaternion in the cost function, from $(0°, 0°, 0°)$ through $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$.

The saturation limits for the reaction wheel torque are $\tau_{\text{limit}} = \pm 3 \cdot 10^{-3}$ N·m, given in Table 5.2. Fig. 6.23 shows that, for all three controllers, the reaction wheel related to $\boldsymbol{\tau}_{w,2}$ saturates at the time instant where the second control torque occur, at about $t = 34$ s. The first control input, which occurs around $t = 2$ s, pushes the spacecraft towards the desired intermediate attitude. The second control input, which occurs around $t = 34$ s, pushes the spacecraft towards the final attitude, but in doing so it also has to oppose the motion of the spacecraft resulting from the first control input. At the end of the optimization, the last control input stops the rotation of the spacecraft. Therefore, it makes sense that the second control input, which occurs around $t = 34$ s, is larger than the other two, and could explain why it saturates.

The path-following maneuver which uses an extra quaternion yielded a con-

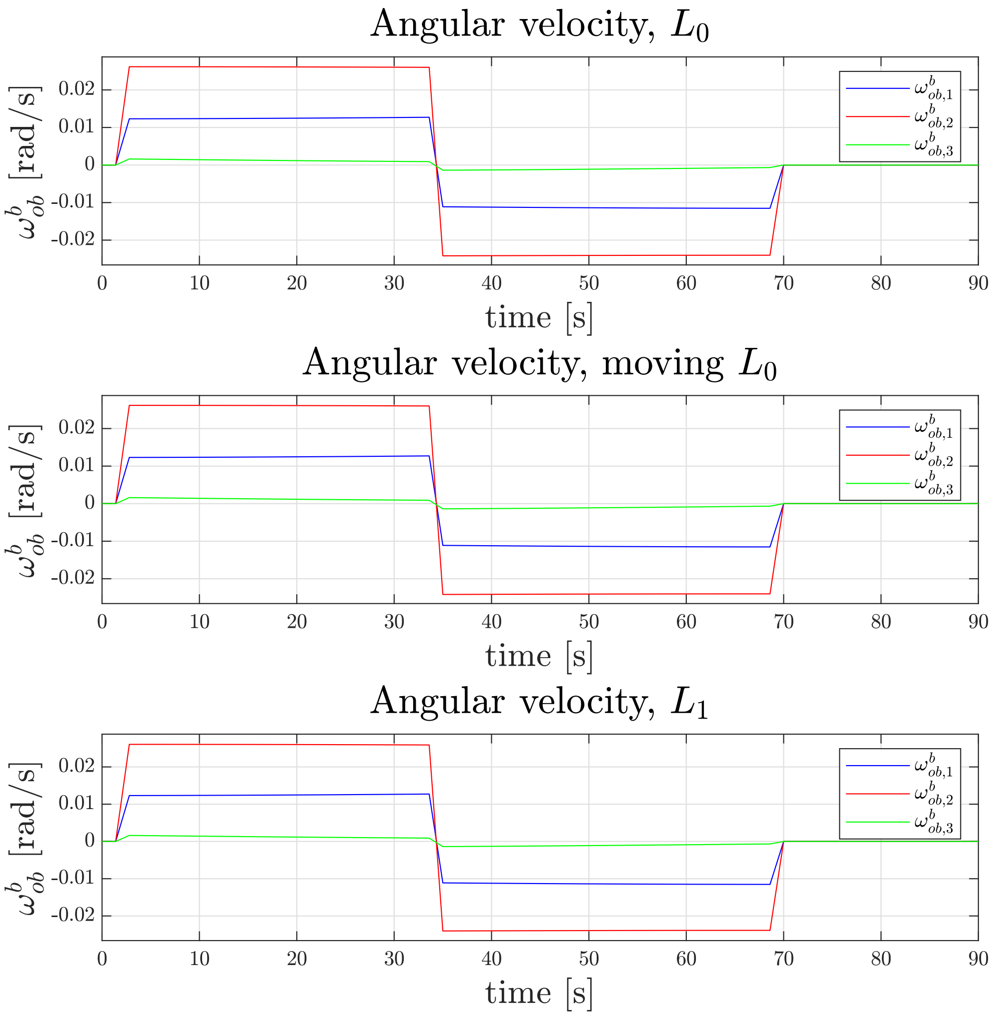


Figure 6.21: Angular velocity, $\boldsymbol{\omega}_{ob}^b$, for path-following using an extra quaternion in the cost function, from $(0°, 0°, 0°)$ through $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$.

trol signal which was more sparse than the control signal resulting from the path-following maneuver which uses multiple optimizations. The difference in sparsity can be seen by comparing Table 6.8 to Table 6.6. These results make sense because in the case for the path-following maneuver which uses an extra quaternion, the optimization of the maneuver is performed over the whole interval. For this maneuver, one control torque is required to push the spacecraft towards the intermediate orientation, one torque is required to turn, and one torque is required to stop the spacecraft at the end of the optimization interval. In the case for path-following which uses multiple optimizations, two separate optimizations are performed for two separate maneuvers which are put together to form the total path. For each of these separate maneuvers, the angular velocity $\boldsymbol{\omega}_{ob}^b$ goes to zero at the end of the optimization, due to the

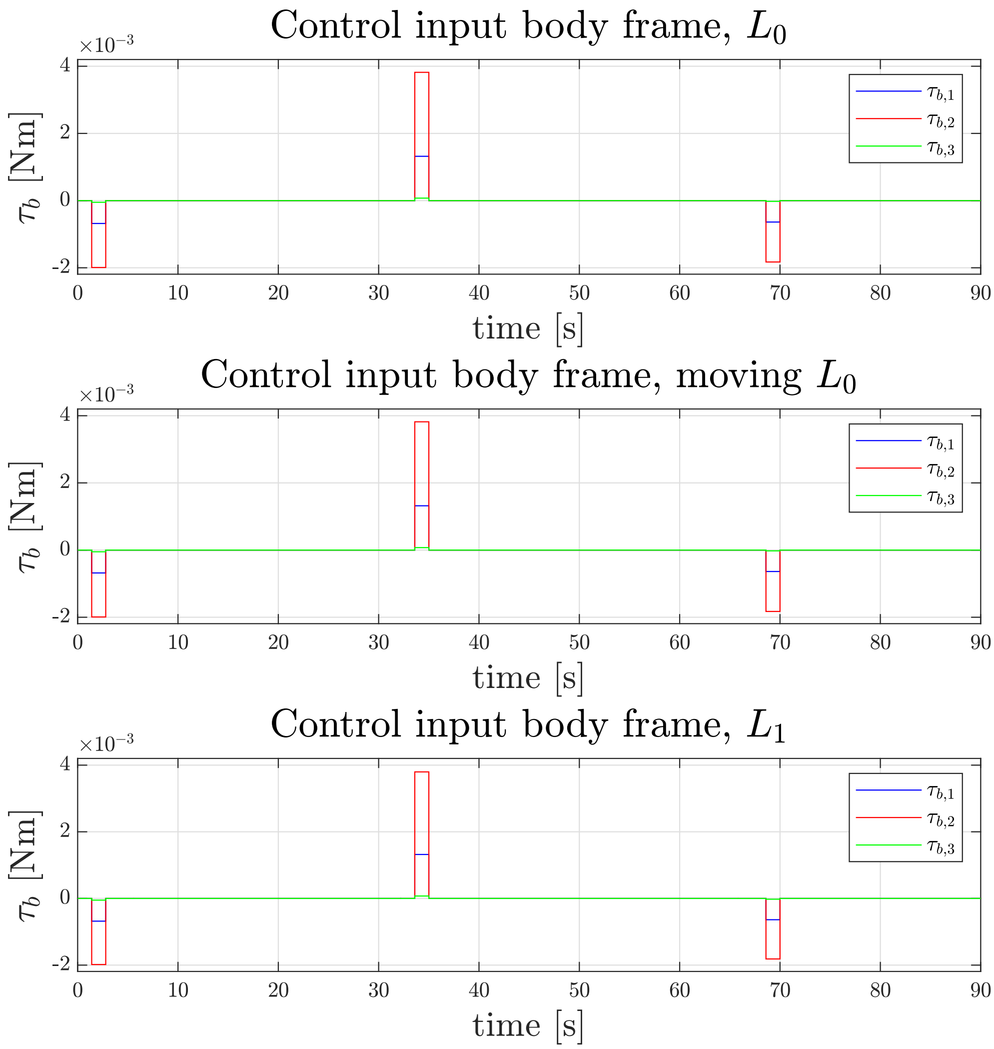


Figure 6.22: Control input body frame for path-following using an extra quaternion in the cost function, from $(0°, 0°, 0°)$ through $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$.

formulation of the cost functions in (5.3a), (5.5a), and (5.7a). Therefore, one torque is applied at the beginning of each maneuver to push the spacecraft towards the desired attitude, and a second torque is applied at the end of the optimization to stop the spacecraft, which yields a sparsity of 4 for the L_1- and L_0-control signals. A control signal having a sparsity of 4 is the sparsest possible control signal for a path-following maneuver split into two optimizations, due to the start and stop torques which has to be applied for each optimization. Similarly, a sparsity of 3 provides the sparsest possible control signal for the path-following which uses an intermediate quaternion, when the cost functions are given as in (5.9a) to (5.9c) and $\mathbf{h}_N$ is chosen such that it would cost less for the control input to occur between $t = 10$ s and $t = 20$ s, and $t = 49$ s and $t = 59$ s.

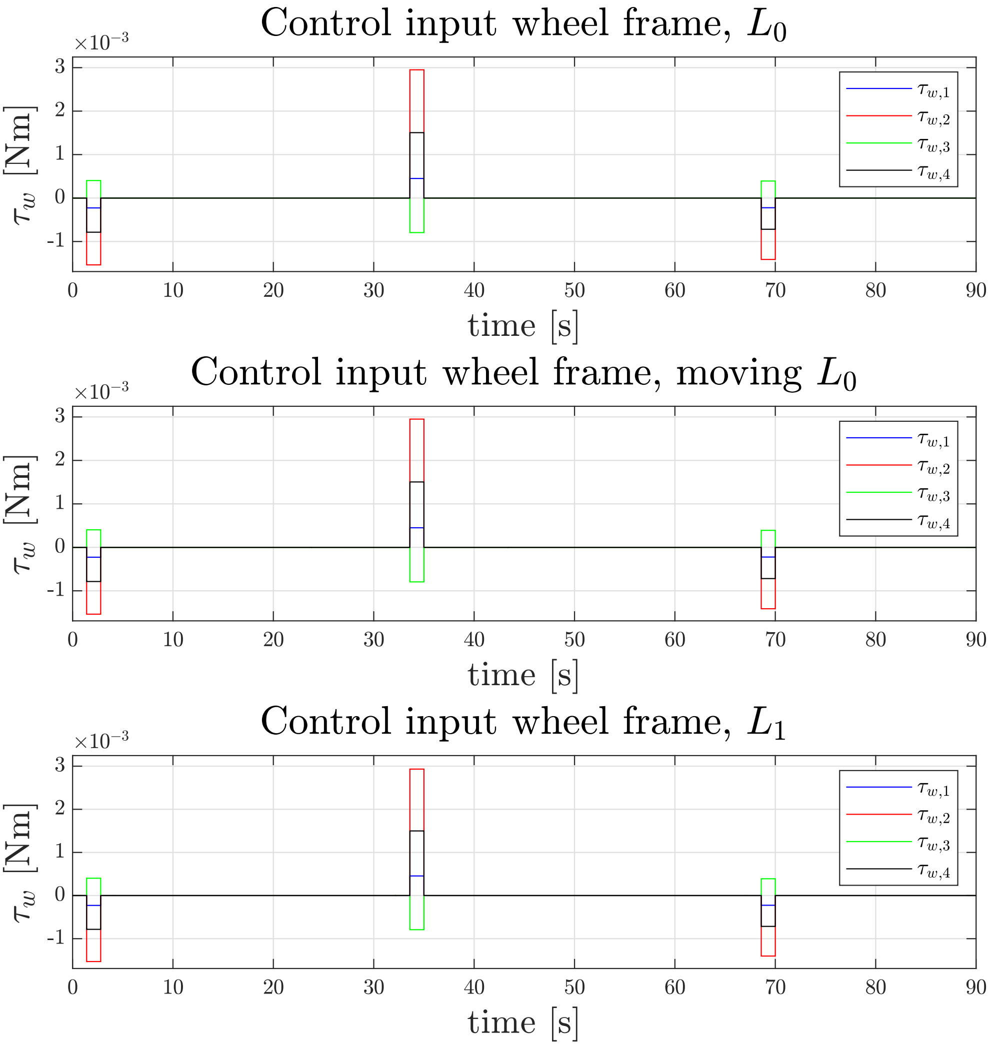


Figure 6.23: Control input wheel frame for path-following using an extra quaternion in the cost function, from $(0°, 0°, 0°)$ through $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$.

The response in the spacecraft's states and the control signals are identical for all three controllers, as shown in Fig. 6.20, Fig. 6.21, Fig. 6.22, and Fig. 6.23. As discussed before, a control signal having the sparsity of 3 is the sparsest possible control signal for a path-following maneuver which uses an intermediate quaternion in the cost function. These findings suggest that all controllers have been able to find optimal control signals for the path-following maneuver which uses an intermediate quaternion in the cost function from $(0°, 0°, 0°)$ through $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$, since all the control signals have a sparsity of 3 and the spacecraft reaches its desired states. A possible explanation to why the three optimal solutions are identical could be that the initial guesses for each of the optimization procedures provide a starting point close to an identical local minimum.

6.5 Multiple-axis maneuver with known perturbations

This section presents and discusses the results from the experiment dealing with a multiple-axis maneuver with known perturbations. The experiment is explained in detail in Section 5.3.4. First, the results from the experiment are presented. Then, Section 6.5.1 discusses the results.

The computation times for the different control scenarios, the sparsity for each of the control signals, and the number of iterations used to find the optimal solution for each of the three controllers are shown in Table 6.9.

The change in attitude over the simulation time can be seen in Fig. 6.24. Fig. 6.25 shows the angular velocity $\boldsymbol{\omega}_{ob}^{b}$. The optimal control input in the body-frame and the wheel-frame can be seen in Fig. 6.26 and Fig. 6.27, respectively. The angular velocity of the reaction wheels $\boldsymbol{\omega}_{bw}^{w}$ is shown in Fig. A.5 in Appendix A.

Controller	**CPU time NLP, [s]**	**CPU time IPOPT, [s]**	**Sparsity**	**Iterations**
L_0-optimal control	3.153	0.165	2	18
Moving L_0-optimal control	113.364	7.430	2	678
L_1-optimal control	6.768	0.391	2	39

Table 6.9: Computation time, sparsity and iterations for a multiple-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$ with known perturbations.

6.5.1 Discussion

All three controllers are able to steer the spacecraft to the desired orientation, stop the spacecraft's rotation, and keep it there, even though there are known perturbations present, as shown in Fig. 6.24 and Fig. 6.25. These results indicate that the controllers are able to find optimal control signals when known perturbations are applied to the spacecraft. If all perturbations are known, they can be modelled and added to the spacecraft model. Then, the controllers will be able to steer the spacecraft to the desired attitude regardless of the perturbations. Even though the perturbations are known and included in the spacecraft model for this experiment, it is not realistic that all perturbations affecting a real-life spacecraft can be modelled perfectly.

The L_0-, moving L_0- and L_1-control signals are equally sparse, as shown in Table 6.9. From Fig. 6.26 and Fig. 6.27 it is clear that the L_0- and L_1-control signals are identical, whereas the moving L_0-control signal differs from the other two controllers. The similarity between the L_0- and L_1-control signals once again highlights why the L_1-controller could be used as an approximation to the L_0-controller, as suggested in Nagahara et al. (2015). A possible explanation to the difference between the control signals is due to the different designs

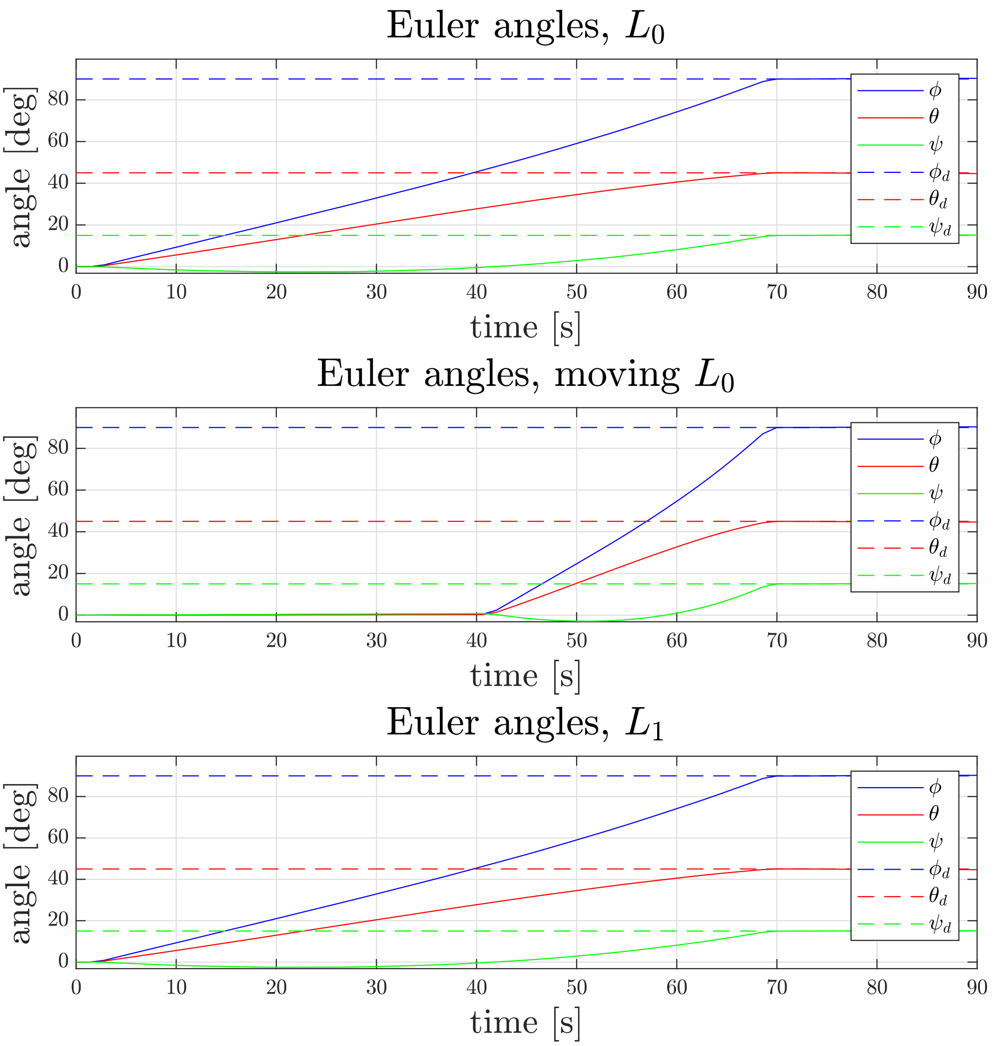
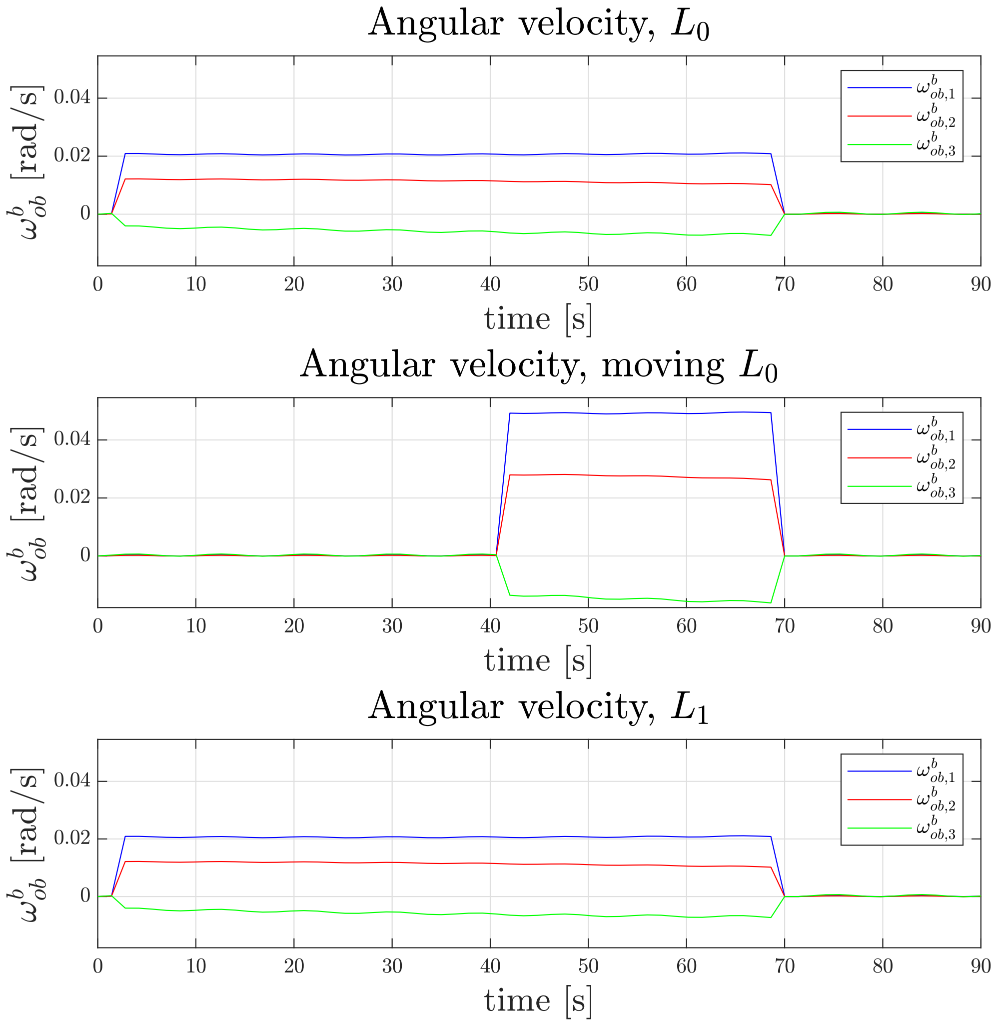


Figure 6.24: Euler angles for a multiple-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$ with known perturbations.

of the controllers. For the moving L_0-controller, it is cheaper to apply control input in the interval from $t = 28$ s to $t = 42$ s, defined by the vector $\mathbf{h}_N$. Only one of the two control inputs occur within this interval, and might once again indicate that although it is cheaper to apply control input between $t = 28$ s and $t = 42$ s, doing so might not minimize the cost function. In other words, even though the term related to control input is minimized between $t = 28$ s and $t = 42$ s, the terms related to attitude and angular velocity may not be minimized if control input is applied within this interval.

The computation time and the number of iterations used by the L_0-controller to find the optimal solution are smaller when known perturbations are applied, which becomes clear comparing Table 6.9 and Table 6.4. These results show that, for a multiple-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$, the



Angular velocity, L_0

Angular velocity, moving L_0

Angular velocity, L_1

Figure 6.25: Angular velocity, ω_{ob}^{b}, for a multiple-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$ with known perturbations.

L_0-controller finds the optimal solution faster when known perturbations are applied to the spacecraft. An explanation for this could be that the presence of disturbances makes it easier for the optimization algorithm to find an optimal solution. The known perturbations are included in the system model through (4.20b), and changes the total spacecraft system dynamics. The system dynamics where the known perturbations are included may yield an L_0-optimal control problem which is easier to solve, compared to when the known perturbations are not included in the model. Another explanation could be that the optimization algorithm gets stuck in a local minimum when the perturbations are not present, and therefore stops searching for a better solution.

Fig. 6.25 shows that the angular velocity ω_{ob}^{b}, i.e., the rotation of the spacecraft relative to its orbit, exhibits oscillating behaviour. The oscillating behaviour

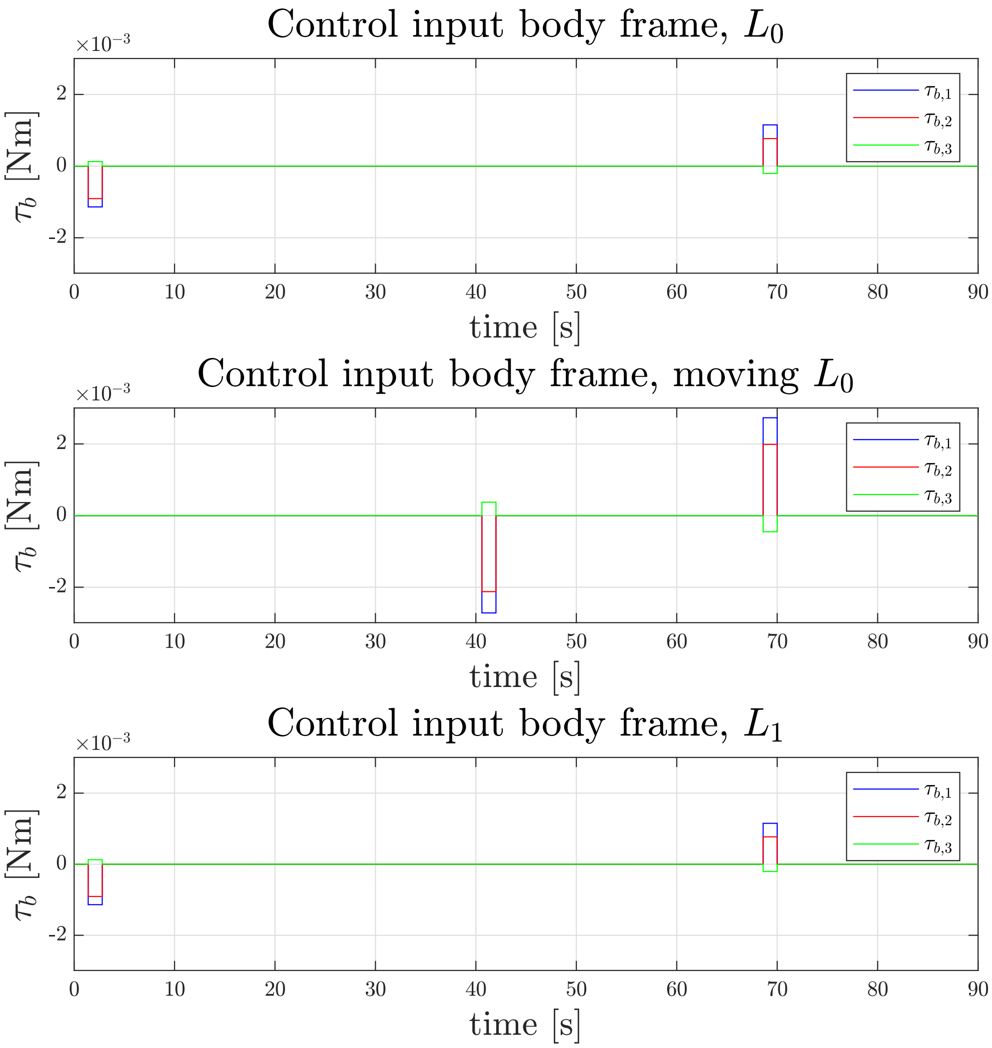


Figure 6.26: Control input body frame for a multiple-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$ with known perturbations.

occurs due to the sine wave present in the perturbations. Because of the perturbations, $\boldsymbol{\omega}^b_{ob}$ oscillates around zero when the spacecraft has reached its desired orientation, but the results in Fig. 6.24, shows that the spacecraft stays at the desired position regardless of the oscillations. Fig. 6.26 and Fig. 6.27 show that, despite the perturbations and oscillating behaviour in $\boldsymbol{\omega}^b_{ob}$, no additional control input is applied after the desired orientation is reached.

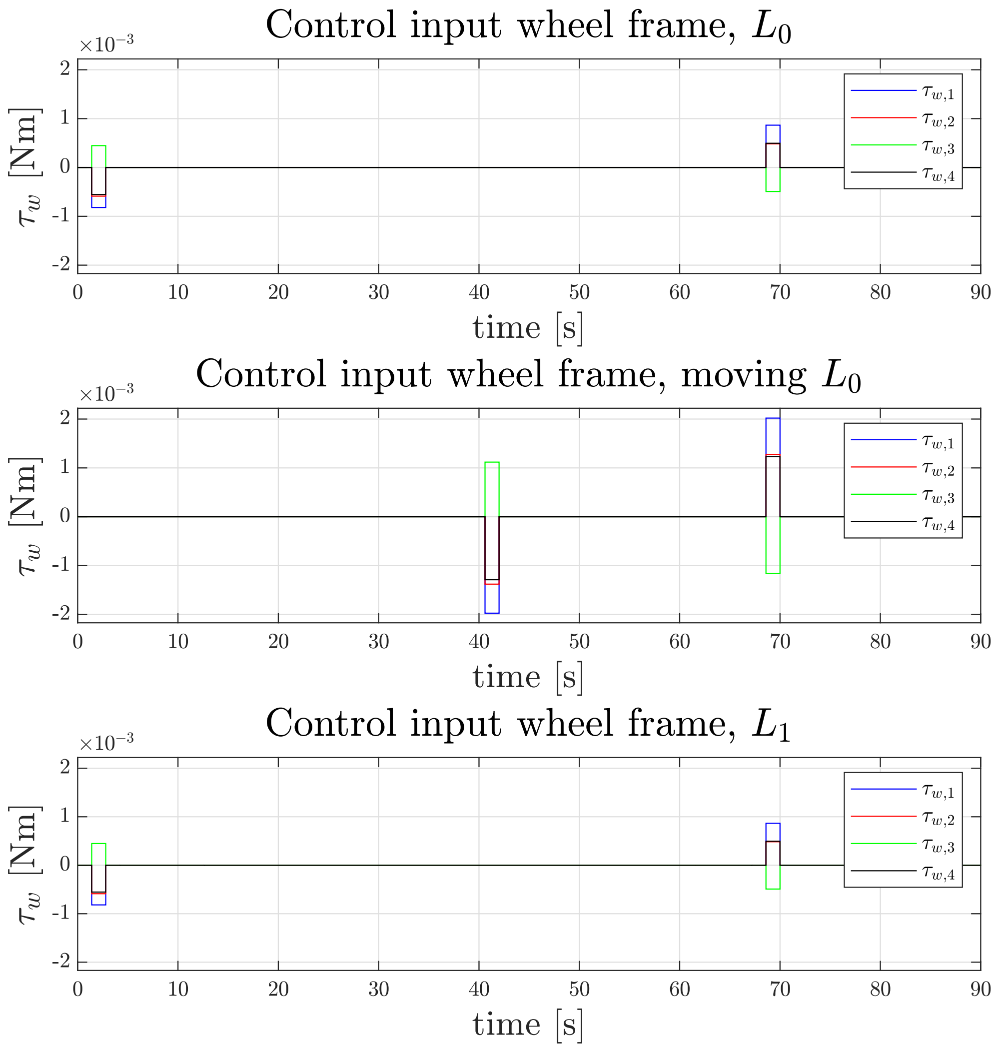


Figure 6.27: Control input wheel frame for a multiple-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$ with known perturbations.

Chapter 7

Conclusions and Future Work

This chapter presents the conclusions based on the work carried out in this thesis and states some suggestions for future work. First, a brief recap of the research objectives stated in Section 1.3 is presented. Then, Section 7.1 presents the conclusions, and Section 7.2 presents some suggestions for future work.

The main goal of this thesis has been to explore the use of maximum hands-off control, also called L_0-optimal control, for the spacecraft attitude control problem. Moreover, the use of the moving maximum hands-off controller in spacecraft attitude control has been explored. The moving maximum hands-off controller is a new concept within control, and it is also referred to as the moving L_0-optimal controller. Additionally, a comparison of the L_1-optimal controller, the maximum hands-off controller, and the moving maximum hands-off controller was conducted when the controllers are used to solve the spacecraft attitude control problem.

7.1 Conclusions

The work carried out in this thesis has shown that the maximum hands-off controller is able to steer the spacecraft to the desired attitude. It was also shown that the controller ensures that the spacecraft reaches the desired final states, i.e., $\boldsymbol{\omega}_{ob}^{b}$ becomes zero when the spacecraft arrives at the desired attitude. Therefore, through the work presented in this thesis, it is confirmed that the maximum hands-off controller works for the spacecraft attitude control problem.

Another important finding, which was discussed in Section 6.3, is that the L_1-optimal control signal was less sparse than the maximum hands-off control signal for a multiple-axis maneuver from $(0°, 0°, 0°)$ to $(75°, 50°, 15°)$. This result is particularly interesting as it confirmed what was suggested in Feng et al. (2016), i.e., that L_1-optimal solutions may provide suboptimal solutions to the L_0-optimal control problem. This is different from what was suggested in the work by Nagahara et al. (2015), where L_0-optimal control is approximated by L_1-optimal control. The difference between the sparsities, as defined in definition 5.1.1, of the two optimal control signals illustrates that using the L_1-norm to approximate the L_0-norm does not always yield the L_0-optimal solution.

Section 6.3 also discussed the results for the multiple-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$. The findings for this maneuver also suggested that the L_1-optimal solution does not always equal the L_0-optimal solution. On the other hand, they suggested that the L_1-optimal control problem could, in some cases, be a rather acceptable approximation to the L_0-optimal control problem.

The moving maximum hands-off controller was developed, designed, and used to solve the spacecraft attitude control problem. This study has shown that the moving maximum hands-off controller is able to steer the spacecraft to a predefined attitude, and it also ensures that the spacecraft reaches the desired final states. In general, the findings presented in Chapter 6 suggest that the moving maximum hands-off controller works as intended, which means that the controller produces a control signal that can be moved to a predefined interval specified by the vector $\mathbf{h}_N$. The optimization procedure aims to satisfy the constraints and reach the final state values while minimizing the cost function. If it is not possible to reach this goal by applying control input within the cheap interval specified by $\mathbf{h}_N$, some or all of the control input will occur outside this interval. Therefore, control inputs may occur outside the interval defined by $\mathbf{h}_N$.

7.2 Future Work

The results presented in Chapter 6 were found through experiments conducted in a simulation environment. Therefore, a suggestion for future work is to perform experimental testing of the controllers in order to obtain a better indication of how the controllers would perform for a real-life spacecraft.

As discussed in Section 6.1.1, the initial guesses for IPOPT are of significant importance when trying to find the optimal solutions. Future studies are encouraged to obtain a better understanding of how the initial guesses affect the performance of the optimization in order to provide the most optimal initial guesses for the solver.

The positive constants k_1, k_2, and k_3 are given in Table 5.1 and discussed in Section 5.5.3. The constants determine the cost on the different terms in the cost functions for the controllers in (5.3a), (5.5a), and (5.7a), and they were chosen to be identical for the three controllers to provide equal conditions for all of them. It is proposed that further research should investigate more in-depth how the values of k_1, k_2, and k_3 affect the outcome of the optimization. Additionally, an interesting extension to the work presented in this thesis, would be to provide different values for k_1, k_2, and k_3 for the three controllers in order to explore whether this would increase the performance of the controllers.

The findings in this study have shown that the control torques produced by the moving maximum hands-off controller do not necessarily occur within the time interval defined to be the cheapest by the vector $\mathbf{h}_N$. A suggestion for future work is, therefore, to constrain the control input to zero on the time intervals found to be optimal by the numerical solver to try to force the torques into the interval specified by $\mathbf{h}_N$.

Bibliography

Andersson, J. A., Gillis, J., Horn, G., Rawlings, J. B. and Diehl, M. (2019), ‘Casadi: a software framework for nonlinear optimization and optimal control’, *Mathematical Programming Computation* **11**(1), 1–36.

Andersson, J., Gillis, J. and Diehl, M. (2016), ‘User documentation for casadi v3.0.0’.
URL: *http://casadi.sourceforge.net/v3.0.0/users_guide/html/node8.html*

Argyriou, A., Foygel, R. and Srebro, N. (2012), ‘Sparse prediction with the k-support norm’, *arXiv preprint arXiv:1204.5043* .

Athans, M. (1963), ‘Minimum-fuel feedback control systems: second-order case’, *IEEE Transactions on Applications and Industry* **82**(65), 8–17.

Athans, M. and Falb, P. L. (2013), *Optimal control: an introduction to the theory and its applications*, Courier Corporation.

Banerjee, A., Amrr, S. M. and Nabi, M. (2019), ‘A pseudospectral method based robust-optimal attitude control strategy for spacecraft’, *Advances in Space Research* **64**(9), 1688–1700.

Beard, R. W. and McLain, T. W. (2012), *Small unmanned aircraft: Theory and practice*, Princeton university press.

Beck, A. and Eldar, Y. C. (2013), ‘Sparsity constrained nonlinear optimization: Optimality conditions and algorithms’, *SIAM Journal on Optimization* **23**(3), 1480–1509.

Bienstock, D. (1996), ‘Computational study of a family of mixed-integer quadratic programming problems’, *Mathematical programming* **74**(2), 121–140.

Biggs, J. D. and Colley, L. (2016), ‘Geometric attitude motion planning for spacecraft with pointing and actuator constraints’, *Journal of Guidance, Control, and Dynamics* **39**(7), 1672–1677.

Bock, H. G. and Plitt, K.-J. (1984), 'A multiple shooting algorithm for direct solution of optimal control problems', *IFAC Proceedings Volumes* **17**(2), 1603–1608.

Candes, E. J. and Tao, T. (2005), 'Decoding by linear programming', *IEEE transactions on information theory* **51**(12), 4203–4215.

Candes, E. J., Wakin, M. B. and Boyd, S. P. (2008), 'Enhancing sparsity by reweighted ℓ_1 minimization', *Journal of Fourier analysis and applications* **14**(5-6), 877–905.

CasADi (2018), 'On the importance of nlp scaling'.
URL: *https://web.casadi.org/blog/nlp-scaling/*

Chan, A. B., Vasconcelos, N. and Lanckriet, G. R. (2007), Direct convex relaxations of sparse svm, *in* 'Proceedings of the 24th international conference on Machine learning', pp. 145–153.

Chan, C. C. (2007), 'The state of the art of electric, hybrid, and fuel cell vehicles', *Proceedings of the IEEE* **95**(4), 704–718.

Chatterjee, D., Nagahara, M., Quevedo, D. E. and Rao, K. M. (2016), 'Characterization of maximum hands-off control', *Systems & Control Letters* **94**, 31–36.

Chaturvedi, N. A., Sanyal, A. K. and McClamroch, N. H. (2011), 'Rigid-body attitude control', *IEEE control systems magazine* **31**(3), 30–51.

Chobotov, V. A. (1991), 'Spacecraft attitude dynamics and control', *NASA Sti/recon Technical Report A* **92**, 40900.

Chou, J. C. (1992), 'Quaternion kinematic and dynamic differential equations', *IEEE Transactions on robotics and automation* **8**(1), 53–64.

Crassidis, J. L. and Markley, F. L. (1996), 'Sliding mode control using modified rodrigues parameters', *Journal of Guidance, Control, and Dynamics* **19**(6), 1381–1383.

Diehl, M., Bock, H. G., Diedam, H. and Wieber, P.-B. (2006), Fast direct multiple shooting algorithms for optimal robot control, *in* 'Fast motions in biomechanics and robotics', Springer, pp. 65–93.

Dong, B. and Zhang, Y. (2013), 'An efficient algorithm for $\updownarrow 0$ minimization in wavelet frame based image restoration', *Journal of Scientific Computing* **54**(2), 350–368.

Donoho, D. L. (2006), 'Compressed sensing', *IEEE Transactions on information theory* **52**(4), 1289–1306.

Egeland, O. and Gravdahl, J. T. (2003), *Modeling and simulation for automatic control*, corr., 2. print edn, Marine Cybernetics AS, Trondheim. OCLC: 255655256.

Feng, M., Mitchell, J. E., Pang, J.-S., Shen, X. and Wächter, A. (2016), 'Complementarity formulations of l0-norm optimization problems', *Industrial Engineering and Management Sciences. Technical Report. Northwestern University, Evanston, IL, USA* .

Fletcher, R. and Leyffer, S. (2004), 'Solving mathematical programs with complementarity constraints as nonlinear programs', *Optimization Methods and Software* **19**(1), 15–40.

Fossen, T. I. (2021), *Handbook of Marine Craft Hydrodynamics and Motion Control, Draft Manuscript*, John Wiley & Sons, Ltd.

Ge, D., Jiang, X. and Ye, Y. (2011), 'A note on the complexity of l p minimization', *Mathematical programming* **129**(2), 285–299.

Gondzio, J. (2012), 'Interior point methods 25 years later', *European Journal of Operational Research* **218**(3), 587–601.

Gros, S. and Diehl, M. (2019), 'Numerical optimal control (draft)'.

Grøtte, M. E., Birkeland, R., Honorè-Livermore, E., Bakken, S., Garrett, J. L., Prentice, E., Sigernes, F., Orlandić, M., Gravdahl, J. T. and Johansen, T. A. (2021), 'Ocean Color Hyperspectral Remote Sensing with High Resolution and Low Latency — the HYPSO-1 CubeSat Mission'.

Grøtte, M. E., Gravdahl, J. T., Johansen, T. A., Larsen, J. A., Vidal, E. M. and Surma, E. (2020), 'Spacecraft attitude and angular rate tracking using reaction wheels and magnetorquers', *IFAC-PapersOnLine* **53**(2), 14819–14826. 21st IFAC World Congress.

Hughes, P. C. (2004), *Spacecraft Attitude Dynamics*, Courier Corporation.

Huynh, D. Q. (2009), 'Metrics for 3d rotations: Comparison and analysis', *Journal of Mathematical Imaging and Vision* **35**(2), 155–164.

HYPSO (2021), 'The hypso mission'.
URL: `https://www.hypso.space/mission/`

Ismail, Z. and Varatharajoo, R. (2010), 'A study of reaction wheel configurations for a 3-axis satellite attitude control', *Advances in Space Research* **45**(6), 750–759.

Joshi, S., Kelkar, A. and Wen, J.-Y. (1995), ‘Robust attitude stabilization of spacecraft using nonlinear quaternion feedback’, *IEEE Transactions on Automatic control* **40**(10), 1800–1803.

Khalil, H. K. (2002), *Nonlinear systems*, Prentice Hall.

Khmelnitsky, E. (2000), ‘On an optimal control problem of train operation’, *IEEE transactions on automatic control* **45**(7), 1257–1266.

Kim, S.-J., Koh, K., Boyd, S. and Gorinevsky, D. (2009), ‘\ell_1 trend filtering’, *SIAM review* **51**(2), 339–360.

Kirchhoff, R., Thele, M., Finkbohner, M., Rigley, P. and Settgast, W. (2010), ‘Start-stop system distributed in-car intelligence’, *ATZextra worldwide* **15**(11), 52–55.

Kristiansen, B. A., Gravdahl, J. T. and Johansen, T. A. (2021), ‘Energy optimal attitude control for a solar-powered spacecraft’, *European Journal of Control* . accepted for publication in the 2021 European Control Conference Special Issue.

Kristiansen, B. A., Grøtte, M. E. and Gravdahl, J. T. (2020), ‘Quaternion-based generalized super-twisting algorithm for spacecraft attitude control’, *IFAC-PapersOnLine* **53**(2), 14811–14818. 21st IFAC World Congress.

Kristiansen, R., Nicklasson, P. J. and Gravdahl, J. T. (2008), ‘Satellite attitude control by quaternion-based backstepping’, *IEEE Transactions on Control Systems Technology* **17**(1), 227–232.

Krogstad, T. R. and Gravdahl, J. T. (2006), 6-dof mutual synchronization of formation flying spacecraft, *in* ‘Proceedings of the 45th IEEE Conference on Decision and Control’, IEEE, pp. 5706–5711.

Lee, T., Leok, M. and McClamroch, N. H. (2008), Time optimal attitude control for a rigid body, *in* ‘2008 American Control Conference’, IEEE, pp. 5210–5215.

Liu, R. R. and Golovitcher, I. M. (2003), ‘Energy-efficient operation of rail vehicles’, *Transportation Research Part A: Policy and Practice* **37**(10), 917–932.

Mallat, S. G. and Zhang, Z. (1993), ‘Matching pursuits with time-frequency dictionaries’, *IEEE Transactions on signal processing* **41**(12), 3397–3415.

MATLAB (2020), *version 9.9.0.1495850 (R2020b)*, The MathWorks Inc., Natick, Massachusetts.

McDuffie, J. H. and Shtessel, Y. B. (1997), A de-coupled sliding mode controller and observer for satellite attitude control, *in* ‘Proceedings The Twenty-Ninth Southeastern Symposium on System Theory’, IEEE, pp. 92–97.

Meyer, G. (1971), *Design and global analysis of spacecraft attitude control systems*, National Aeronautics and Space Administration, Ames Research Center.

Nagahara, M. (2020), *Sparsity Methods for Systems and Control*, Now Publisher Inc., Kitakyushu.

Nagahara, M., Quevedo, D. E. and Nešić, D. (2013), Maximum hands-off control and l 1 optimality, *in* ‘52nd IEEE Conference on Decision and Control’, IEEE, pp. 3825–3830.

Nagahara, M., Quevedo, D. E. and Nešić, D. (2015), ‘Maximum hands-off control: a paradigm of control effort minimization’, *IEEE Transactions on Automatic Control* **61**(3), 735–747.

Ng, A. Y. (2004), Feature selection, l 1 vs. l 2 regularization, and rotational invariance, *in* ‘Proceedings of the twenty-first international conference on Machine learning’, p. 78.

Nocedal, J. and Wright, S. (2006), *Numerical optimization*, Springer Science & Business Media.

Oland, E. and Schlanbusch, R. (2009), Reaction wheel design for cubesats, *in* ‘2009 4th International Conference on Recent Advances in Space Technologies’, IEEE, pp. 778–783.

Park, Y. (2005), ‘Robust and optimal attitude stabilization of spacecraft with external disturbances’, *Aerospace Science and Technology* **9**(3), 253–259.

Pontryagin, L. S. (1962), ‘The mathematical theory of optimal processes’.

Pukdeboon, C. and Zinober, A. S. I. (2012), ‘Control lyapunov function optimal sliding mode controllers for attitude tracking of spacecraft’, *Journal of the Franklin Institute* **349**(2), 456–475.

Royden, H. L. and Fitzpatrick, P. (1988), *Real analysis*, Vol. 32, Macmillan New York.

Schaub, H. and Lappas, V. J. (2009), ‘Redundant reaction wheel torque distribution yielding instantaneous l2 power-optimal spacecraft attitude control’, *Journal of guidance, control, and dynamics* **32**(4), 1269–1276.

Sciavicco, L. and Siciliano, B. (2012), *Modelling and control of robot manipulators*, Springer Science & Business Media.

Shakouri, P., Ordys, A., Darnell, P. and Kavanagh, P. (2013), 'Fuel efficiency by coasting in the vehicle', *International Journal of Vehicular Technology* **2013**.

Shen, H. and Tsiotras, P. (1999), 'Time-optimal control of axisymmetric rigid spacecraft using two controls', *Journal of Guidance, Control, and Dynamics* **22**(5), 682–694.

Show, L.-L., Juang, J.-C., Lin, C.-T. and Jan, Y.-W. (2002), Spacecraft robust attitude tracking design: Pid control approach, *in* 'Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)', Vol. 2, IEEE, pp. 1360–1365.

Sola, J. (2017), 'Quaternion kinematics for the error-state kalman filter', *arXiv preprint arXiv:1711.02508* .

Vallado, D. A. (2001), *Fundamentals of astrodynamics and applications*, Vol. 12, Springer Science & Business Media.

Wächter, A. and Biegler, L. T. (2006), 'On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming', *Mathematical programming* **106**(1), 25–57.

Weerakoon, S. and Fernando, T. (2000), 'A variant of newton's method with accelerated third-order convergence', *Applied Mathematics Letters* **13**(8), 87–93.

Wen, J.-Y. and Kreutz-Delgado, K. (1991), 'The attitude control problem', *IEEE Transactions on Automatic control* **36**(10), 1148–1162.

Wertz, J. R. (2012), *Spacecraft attitude determination and control*, Vol. 73, Springer Science & Business Media.

Wu, C. and Han, X. (2019), 'Energy-optimal spacecraft attitude maneuver path-planning under complex constraints', *Acta Astronautica* **157**, 415–424.

Yin, P., Lou, Y., He, Q. and Xin, J. (2015), 'Minimization of 1-2 for compressed sensing', *SIAM Journal on Scientific Computing* **37**(1), A536–A563.

Yu, J., Eriksson, A., Chin, T.-J. and Suter, D. (2014), 'An adversarial optimization approach to efficient outlier removal', *Journal of mathematical imaging and vision* **48**(3), 451–466.

Yuan, G. and Ghanem, B. (2015), l0tv: A new method for image restoration in the presence of impulse noise, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 5369–5377.

Yuan, G. and Ghanem, B. (2016), ‘Sparsity constrained minimization via mathematical programming with equilibrium constraints’, *arXiv preprint arXiv:1608.04430* .

Zhang, T. (2010), ‘Analysis of multi-stage convex relaxation for sparse regularization.’, *Journal of Machine Learning Research* **11**(3).

Appendix A

Angular velocity of reaction wheels

A.1 Multiple-axis maneuver

A.1.1 Multiple-axis maneuver $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$

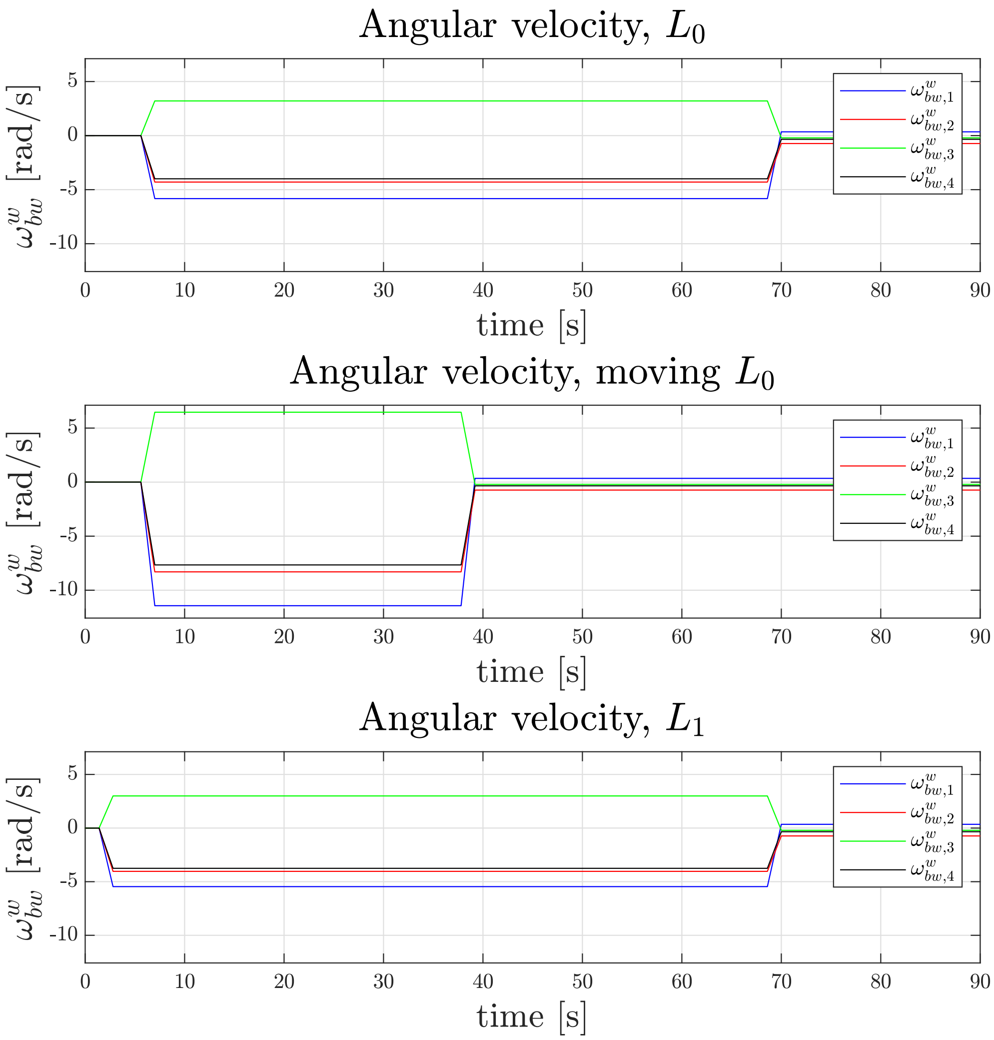


Figure A.1: Angular velocity, ω_{bw}^{w}, for a multiple-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$.

A.1.2 Multiple-axis maneuver $(0°, 0°, 0°)$ to $(75°, 50°, 15°)$

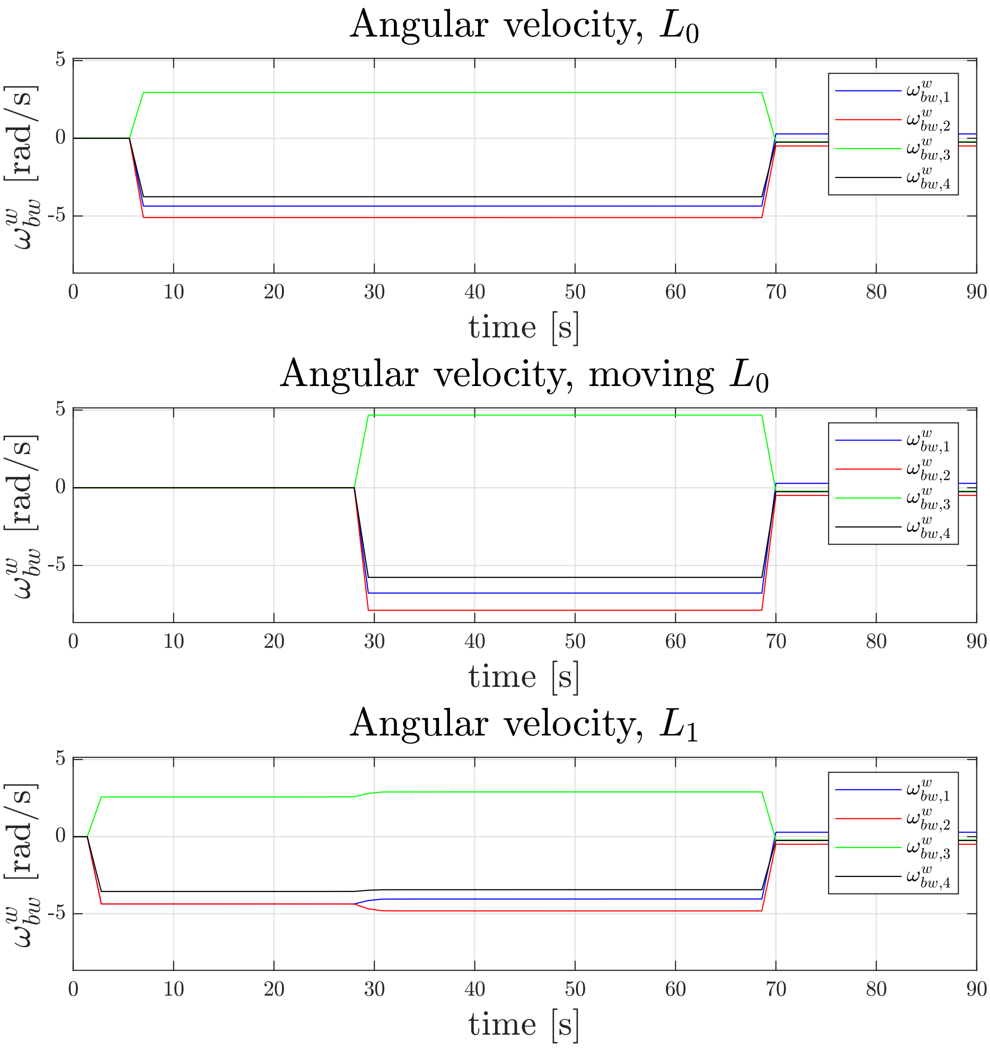


Figure A.2: Angular velocity, $\boldsymbol{\omega}_{bw}^{w}$, for a multiple-axis maneuver $(0°, 0°, 0°)$ to $(75°, 50°, 15°)$.

A.2 Path-following maneuver

A.2.1 Path-following using multiple optimizations

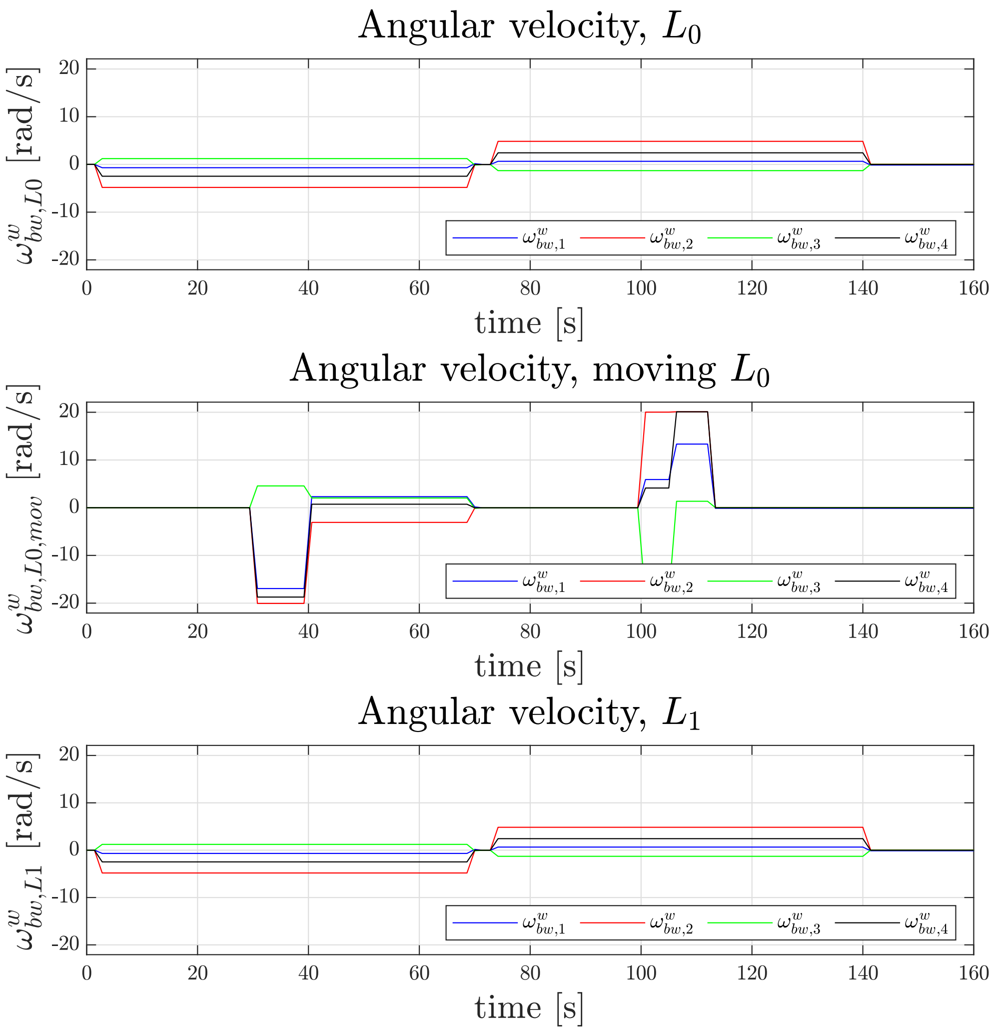
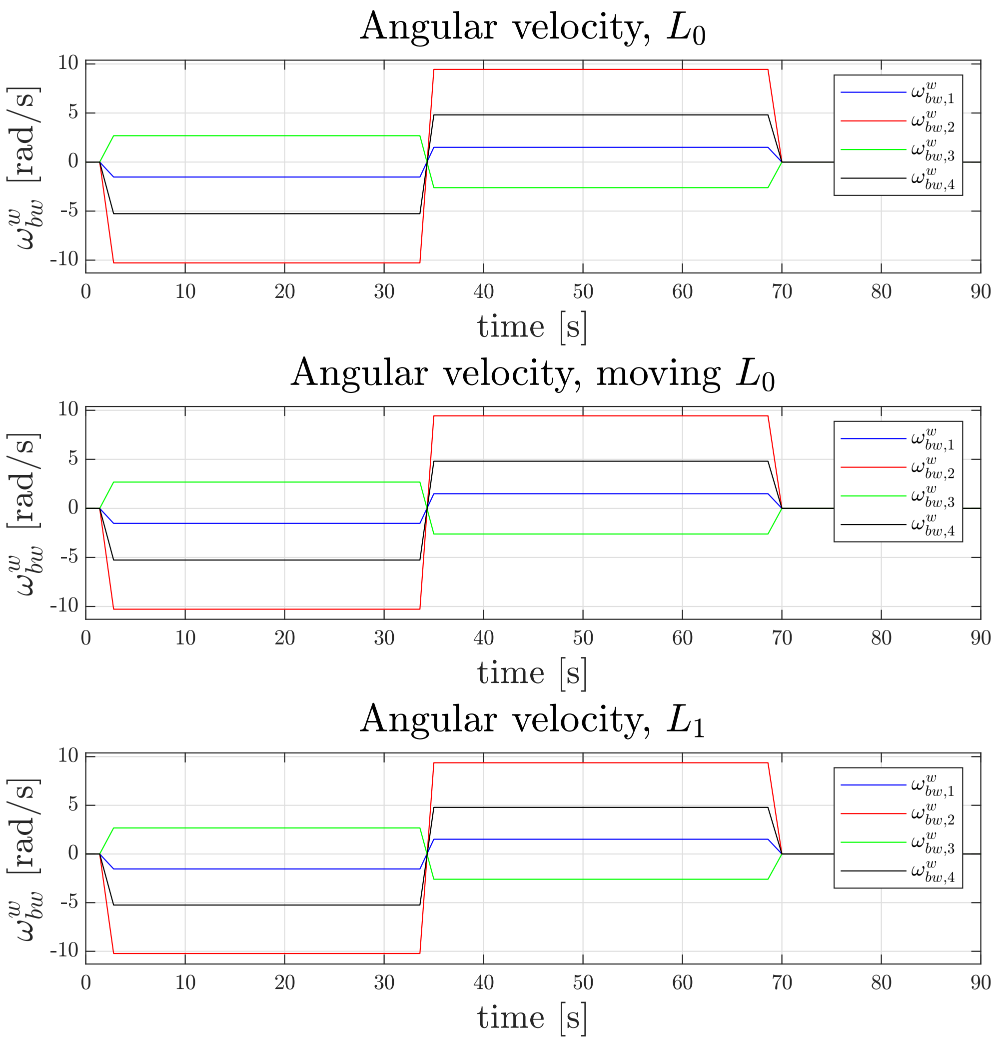


Figure A.3: Angular velocity, ω^w_{bw}, for path-following using multiple optimizations, from $(0°, 0°, 0°)$ through $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$.

A.2.2 Path-following using an extra quaternion in the cost function


Angular velocity, L_0

Angular velocity, moving L_0

Angular velocity, L_1

Figure A.4: Angular velocity, $\boldsymbol{\omega}_{bw}^{w}$, for path-following using an extra quaternion in the cost function, from $(0°, 0°, 0°)$ through $(30°, 45°, 15°)$ to $(0°, 0°, 0°)$.

A.3 Multiple-axis maneuver with known perturbations

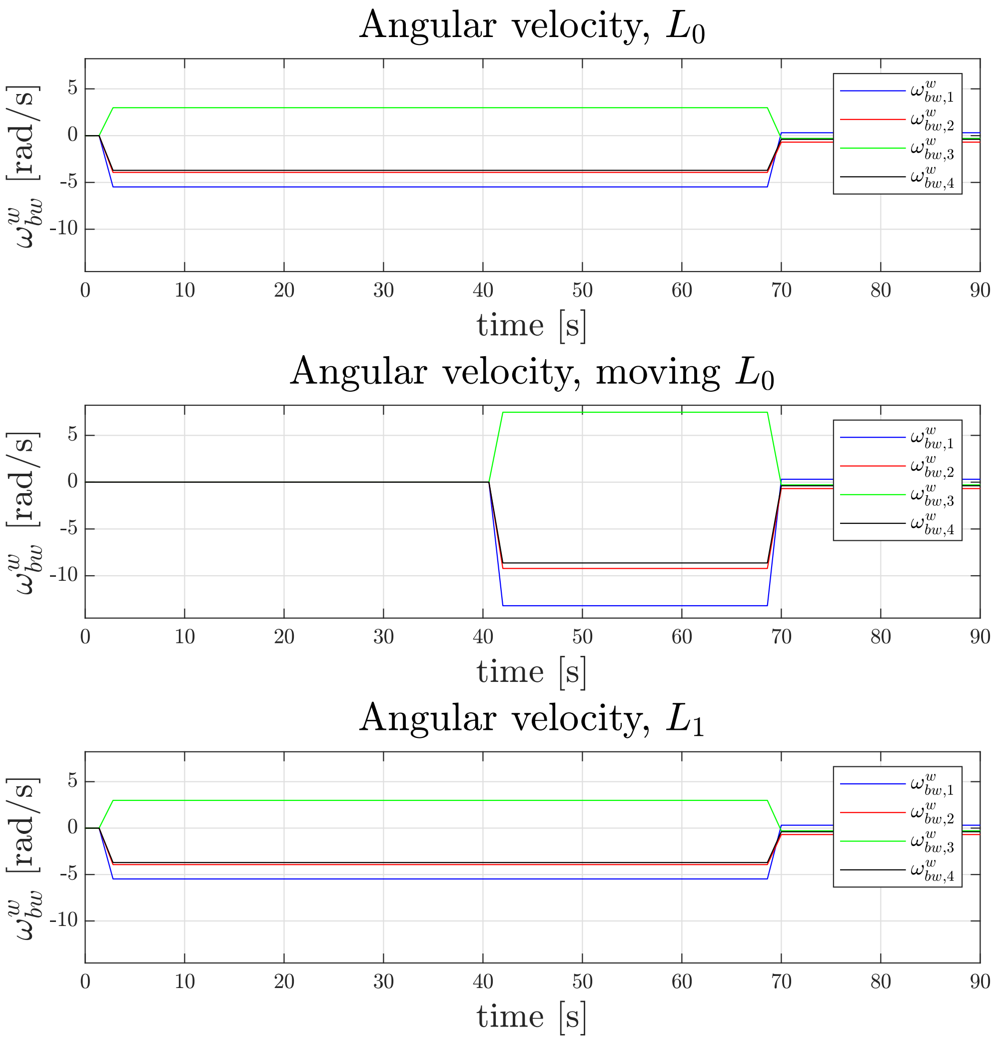


Figure A.5: Angular velocity, ω_{bw}^{w}, for a multiple-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$ with known perturbations.

Appendix B

Draft paper to be submitted for the American Control Conference 2022

Maximum Hands-Off Attitude Control*

Sigrid Kjønnø Schaanning[1], Bjørn Andreas Kristiansen[2], Jan Tommy Gravdahl[2]

Abstract— **In this paper, we explore the use of maximum hands-off control for attitude control of a spacecraft actuated by reaction wheels. The maximum hands-off, or L_0-optimal, controller aims to find the sparsest control signal among all admissible control signals. However, L_0-optimal problems are generally hard to solve as L_0-cost functions are discontinuous and non-convex. Previous research have investigated methods to approximate the L_0-norm in the cost function, for instance using an L_1-norm. We propose an approach to the maximum hands-off control problem for spacecraft attitude control involving an L_0-cost function relaxed through complementarity constraints. Then, the controller is applied to the spacecraft attitude control problem, and the sparsity of the maximum hands-off controller is compared to that of the L_1-optimal controller. The simulations based on a 6U CubeSat were conducted using CasADi as the primary optimization tool, and the L_1- and L_0-optimal control problems were discretized using direct multiple-shooting and solved using the IPOPT solver. In addition to these results, we propose a new paradigm of control, called *moving* maximum hands-off control, which lets the user specify in which time interval the control should occur, and then aims to find the sparsest control among all admissible controls based on this information. The moving maximum hands-off controller is shown to be as sparse as the maximum hands-off controller for some spacecraft maneuvers.**

I. INTRODUCTION

Attitude control of spacecraft is a field in which multiple studies have been conducted [1]–[3]. Several solutions have been suggested to solve the spacecraft attitude control problem. Some of the most regularly employed attitude control strategies include proportional-derivative (PD) control laws [1], [4]. Other approaches taken to solve the spacecraft attitude control problem are techniques based on quaternion feedback [2], [5]. Another approach to design the attitude controller is provided by the theory of optimal control [3]. Optimal control has been applied for spacecraft attitude control in multiple cases, and with the use of different cost functions. For instance, a time optimal attitude control problem has been studied, where the objective is to minimize the time it takes to rotate a rigid body to a desired attitude and angular velocity, while subject to control input constraints [6]. Other time optimal attitude control problems have been studied, one where the objective is to orient a spacecraft from an initial attitude and angular velocity to a final attitude and angular velocity in minimum time, using two independent control torques [7], and a second where the objective is the same as before only using reaction wheels [8]. A cost function based on angular velocity have been used to optimize the attitude motion planning of a spacecraft, with pointing and actuator constraints [9]. Maximum hands-off control has, to the authors' best knowledge, never been applied to spacecraft attitude control. A maximum hands-off controller is a type of optimal controller with control values which are most often zero, i.e., the control values are *sparse*, but still manage to achieve the control objectives [10]. A hands-off control holds the control values at exactly zero over a time interval, and the *maximum* hands-off control maximizes the time interval over which the control input is exactly zero [11].

The main contribution of this paper is the use of maximum hands-off control to solve the attitude control problem for a spacecraft actuated by reaction wheels. The maximum hands-off controller is tested through simulations based on the configuration of a 6U CubeSat. Another contribution is the presentation of an example, which shows that using the L_1-norm to approximate the L_0-norm does not always yield the L_0-optimal solution. We design and implement the moving maximum hands-off as an extension to the maximum hands-off controller. The moving maximum hands-off controller lets the user specify in which time interval the control inputs should occur and is, to the authors' best knowledge, a novel concept within control. Finally, this paper provides a comparison of the responses of the L_1-optimal controller, the maximum hands-off controller, and the moving maximum hands-off controller when solving the spacecraft attitude control problem.

The rest of this paper is organized as follows. Section II describes the coordinate frames used and presents the total spacecraft dynamics used for the simulations. Section III introduces the maximum hands-off controller. Section IV introduces the controller designs, including the design of the novel moving maximum hands-off controller, and Section V presents the simulation setup. The simulation results are presented in Section VI, whereas Section VII discusses the findings. Section VIII provides the conclusion.

II. SPACECRAFT MODEL

In this section, we present the model of a spacecraft orbiting the Earth, actuated by reaction wheels.

---

*The work is sponsored by the Research Council of Norway through the Centre of Excellence funding scheme, project number 223254, AMOS.

[1] Sigrid K. Schaanning is with the Department of Engineering Cybernetics at NTNU Norwegian University of Science and Technology, NO-7491 Trondheim, Norway. `sigrid.schaanning@gmail.com`

[2] Bjørn Andreas Kristiansen and Jan Tommy Gravdahl are with the Centre for Autonomous Marine Operations and Systems and the Department of Engineering Cybernetics at NTNU Norwegian University of Science and Technology, NO-7491 Trondheim, Norway. `E-mails: {bjorn.a.kristiansen, jan.tommy.gravdahl}@ntnu.no`

A. Coordinate frames

A reference frame, or a coordinate frame, is a choice of coordinate system given as $\{r\} = \{O_r, \mathbf{x}_r, \mathbf{y}_r, \mathbf{z}_r\}$, where O_r is the origin and $\mathbf{x}_r, \mathbf{y}_r, \mathbf{z}_r$ are the orthonormal unit vectors.

1) Earth-centered inertial (ECI) frame: The Earth-centered inertial frame, denoted $\{i\}$, is considered to be an inertial frame where Newton's laws are valid. The origin of $\{i\}$ is located at the Earth's center of mass, with the z-axis points through the North Pole, the x-axis points towards the vernal equinox and the y-axis completes the right-hand system [12].

2) Body frame: The body frame, denoted $\{b\}$, is a moving coordinate frame fixed to the spacecraft, with origin at the spacecraft's center of mass [12]. The body frame axes follow the spacecraft structure.

3) Orbit frame: The Vehicle Velocity, Local Horizontal (VVLH) frame, or the orbit frame, is denoted $\{o\}$ and has origin at the spacecraft's center of mass. The z-axis points in the direction of the Earth's center of mass, the x-axis points in the direction of the orbit velocity vector, while the y-axis completes the right-handed coordinate system. The unit vectors of the orbit frame are defined as [13]

$$\hat{\mathbf{z}}^o = -\frac{\mathbf{r}^i}{\|\mathbf{r}^i\|_2},\ \hat{\mathbf{x}}^o = \frac{\mathbf{v}^i}{\|\mathbf{v}^i\|_2},\ \hat{\mathbf{y}}^o = \frac{\hat{\mathbf{z}}^o \times \hat{\mathbf{x}}^o}{\|\hat{\mathbf{z}}^o \times \hat{\mathbf{x}}^o\|_2}, \tag{1}$$

where $\mathbf{r}^i$ and $\mathbf{v}^i$ are the distance between the spacecraft and the center of the Earth, and the inertial velocity of the spacecraft, respectively, in the ECI frame.

4) Wheel frame: The wheel frame is used to specify vectors directly related to the reaction wheels, and it is denoted by $\{w\}$ [13]. There are two vectors represented in the wheel frame, namely the torque applied to each wheel, $\boldsymbol{\tau}_u^w$, and the wheel angular velocity, $\boldsymbol{\omega}_{bw}^w$. The dimensions of $\boldsymbol{\tau}_u^w$ and $\boldsymbol{\omega}_{bw}^w$ equals the number of the reaction wheels on the satellite. Each channel of the vectors gives the torque applied or angular velocity about each wheel's axis of rotation. The matrix $\mathbf{A} \in \mathbb{R}^{3\times n}$ maps the wheel frame to the body frame as follows [13]

$$\boldsymbol{\tau}_u^b = \mathbf{A}\boldsymbol{\tau}_u^w \implies \boldsymbol{\tau}_u^w = \mathbf{A}^+\boldsymbol{\tau}_u^b, \tag{2}$$

where the matrix $\mathbf{A}^+$ is the Moore-Penrose pseudo-inverse of $\mathbf{A}$. Due to a fixed reaction wheel configuration, $\mathbf{A}$ represents a constant mapping between $\{w\}$ and $\{b\}$.

B. Attitude representation

Unit quaternions are used to describe the attitude of the spacecraft. The unit quaternion, $\mathbf{q}_b^o$, denotes the attitude of $\{b\}$ relative to $\{o\}$. The rotation matrix from $\{o\}$ to $\{b\}$, $\mathbf{R}_o^b$ is defined as

$$\begin{aligned} \mathbf{R}_b^o &= \mathbf{R}(\mathbf{q}_b^o) = \mathbf{I}_{3\times 3} + 2\eta_b^o\mathbf{S}(\boldsymbol{\epsilon}_b^o) + 2\mathbf{S}^2(\boldsymbol{\epsilon}_b^o), \\ \mathbf{R}_o^b &= (\mathbf{R}_b^o)^\top, \end{aligned} \tag{3}$$

where $\mathbf{q}_b^o = [\eta_b^o, \boldsymbol{\epsilon}_b^{o\top}]^\top \in \mathbb{R}^4$ satisfy the constraint $\eta^2 + \boldsymbol{\epsilon}_b^{o\top}\boldsymbol{\epsilon}_b^o = 1$, $\mathbf{I}_{3\times 3}$ is the 3×3-identity matrix, and $\mathbf{S}(\cdot)$ is a skew-symmetric matrix. The time derivative of $\mathbf{R}_o^b$ is given as $\dot{\mathbf{R}}_o^b = -\mathbf{S}(\boldsymbol{\omega}_{ob}^b)\mathbf{R}_o^b$ where the angular velocity of $\{b\}$ relative to $\{o\}$ is given by $\boldsymbol{\omega}_{ob}^b$.

The kinematic differential equation for the spacecraft's attitude $\mathbf{q}_b^o$, is given by [12]

$$\dot{\mathbf{q}}_b^o = \begin{bmatrix} \dot{\eta}_b^o \\ \dot{\boldsymbol{\epsilon}}_b^o \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -\boldsymbol{\epsilon}_b^{o\top} \\ \eta_b^o\mathbf{I}_{3\times 3} + \mathbf{S}(\boldsymbol{\epsilon}_b^o) \end{bmatrix}\boldsymbol{\omega}_{ob}^b = \frac{1}{2}\mathbf{T}(\mathbf{q}_b^o)\boldsymbol{\omega}_{ob}^b, \tag{4}$$

where $\mathbf{T}(\cdot)$ denotes the angular velocity transformation matrix.

C. Angular velocity

When analysing the attitude of a spacecraft orbiting the Earth, three different angular velocities are of interest, namely the angular velocity of $\{b\}$ relative to $\{o\}$, $\boldsymbol{\omega}_{ob}^b$, the angular velocity of $\{o\}$ relative to $\{i\}$, $\boldsymbol{\omega}_{io}^b$, and the angular velocity of $\{b\}$ relative to $\{i\}$, $\boldsymbol{\omega}_{ib}^b$. The angular velocities relate to one another as follows

$$\boldsymbol{\omega}_{ob}^b = \boldsymbol{\omega}_{ib}^b - \boldsymbol{\omega}_{io}^b = \boldsymbol{\omega}_{ib}^b - \mathbf{R}_o^b\boldsymbol{\omega}_{io}^o, \tag{5}$$

and the angular velocity $\boldsymbol{\omega}_{io}^o$ is defined as [14]

$$\boldsymbol{\omega}_{io}^o = \mathbf{R}_i^o\frac{\mathbf{S}(\mathbf{r}^i)\mathbf{v}^i}{(\mathbf{r}^i)^\top\mathbf{r}^i}. \tag{6}$$

Moreover, the inertial acceleration of the spacecraft is defined as [15]

$$\dot{\mathbf{v}}^i = -\frac{\mu}{\|\mathbf{r}^i\|_2^3}\mathbf{r}^i, \tag{7}$$

where μ is the standard gravitational parameter of the Earth and $\|\mathbf{r}^i\|_2$ denotes the 2-norm of $\mathbf{r}^i$. The rate of change for $\mathbf{r}^i$ is given as

$$\dot{\mathbf{r}}^i = \mathbf{v}^i. \tag{8}$$

D. Total system dynamics

The total spacecraft dynamics for a spacecraft orbiting the Earth are given as [2], [13], [14], [16]

$$\dot{\mathbf{q}}_b^o = \frac{1}{2}\mathbf{T}(\mathbf{q}_b^o)\boldsymbol{\omega}_{ob}^b \tag{9a}$$

$$\dot{\boldsymbol{\omega}}_{ib}^b = {\mathbf{J}_s}^{-1}(-\mathbf{A}\boldsymbol{\tau}_u^w - \mathbf{S}(\boldsymbol{\omega}_{ib}^b)\mathbf{H}_s^b) \tag{9b}$$

$$\dot{\boldsymbol{\omega}}_{ob}^b = \dot{\boldsymbol{\omega}}_{ib}^b + \mathbf{S}(\boldsymbol{\omega}_{ob}^b)\mathbf{R}_o^b\boldsymbol{\omega}_{io}^o + \mathbf{R}_o^b\mathbf{S}(\mathbf{R}_b^o\boldsymbol{\omega}_{ob}^b)\boldsymbol{\omega}_{io}^o \tag{9c}$$

$$\dot{\boldsymbol{\omega}}_{bw}^w = {\mathbf{J}_w}^{-1}\boldsymbol{\tau}_u^w - \mathbf{A}^\top\dot{\boldsymbol{\omega}}_{ib}^b, \tag{9d}$$

where $\mathbf{J} \in \mathbb{R}^{3\times 3}$ is the total system inertia of the spacecraft rigid body, defined as $\mathbf{J} = \mathbf{J}_s + \mathbf{A}\mathbf{J}_w\mathbf{A}^\top$, where $\mathbf{J}_s \in \mathbb{R}^{3\times 3}$ denotes the inertia of the spacecraft rigid body excluding the inertia about the spinning axes of the reaction wheels, and $\mathbf{J}_w \in \mathbb{R}^{n\times n}$ denotes the inertia matrix of the reaction wheels about the spinning axes, where n is the number of reaction wheels. The total angular momentum of the spacecraft is denoted by $\mathbf{H}_s^b$ [16]. Note that $\dot{\boldsymbol{\omega}}_{ib}^b = \frac{{}^b d}{dt}\boldsymbol{\omega}_{ib}^b$, $\dot{\boldsymbol{\omega}}_{ob}^b = \frac{{}^b d}{dt}\boldsymbol{\omega}_{ob}^b$, and $\dot{\boldsymbol{\omega}}_{bw}^w = \frac{{}^b d}{dt}\boldsymbol{\omega}_{bw}^w$.

III. MAXIMUM HANDS-OFF CONTROL

In this section, we present some mathematical preliminaries required to understand the concept of and theoretical derivation of maximum hands-off control.

A. Mathematical preliminaries

The content presented in this section is based on the review by [11], which provides a detailed review of the mathematics behind maximum hands-off control.

The L_1-norm of a vector $\mathbf{x} \in \mathbb{R}^n$ is defined as

$$\|\mathbf{x}\|_1 \triangleq \sum_{i=1}^{n} |x_i|. \tag{10}$$

The L_p-norm, with $p \in [1, \infty)$, for a vector of continuous-time signals $\mathbf{u}(t)$ over the time interval $[0, T)$ is defined as

$$\|\mathbf{u}\|_p \triangleq \left(\int_0^T \|\mathbf{u}(t)\|^p dt \right)^{\frac{1}{p}}. \tag{11}$$

The norm $\|\cdot\|$ inside the integral in (11) can be any norm p-norm for $p \in [1, \infty)$ [17]. If $p \in (0, 1)$, in (11), then $\|\cdot\|_p$ is not a norm as it fails to satisfy the triangle inequality [11].

The support of a function is the set of points where the function takes on nonzero values [18], and the support set of a function $u(t)$, is defined as [11]

$$\text{supp}(u(t)) \triangleq \{t \in [0, T] : u(t) \neq 0\}, \tag{12}$$

and by using (12), the L_0-norm for a vector of continuous-time signals $\mathbf{u}(t)$ can be defined by the length of the support of each of each signal $u(t)$ accordingly:

$$\|\mathbf{u}\|_0 \triangleq \mu(\text{supp}(u(t))), \tag{13}$$

where $\mu(\cdot)$ is the Lebesgue measure. The Lebesgue measure essentially "counts" the length of the support of the signal u, i.e, the Lebesgue measure sums up the parts where the control signal is nonzero.

B. Maximum Hands-off Control Problem Formulation

The maximum hands-off control is the control that maximizes the time interval over which the control input is exactly zero. To put it more precisely, the controller minimizes the Lebesgue measure of the support, i.e., the L_0-norm, to find the sparsest of the admissible controls [11]. The L_0-cost function is given as [11]

$$J_0(\mathbf{u}) \triangleq \sum_{i=1}^{m} \lambda_i \|u_i\|_0, \tag{14}$$

where m is the number of control inputs, $\mathbf{u}$ is the control input vector, λ_i are positive weights, and u_i denotes each element i in $\mathbf{u}$. The control that minimizes (14) is called the *maximum hands-off control*, or the *L_0-optimal control*, and it is the sparsest control among all admissible controls [11].

The L_0-cost function in (14) is discontinuous and non-convex [11]. Solving discontinous and non-convex optimization problems are generally hard [11], and solving the L_0-optimal control problem is NP-hard [19]. Several relaxation methods and reformulations have been suggested to (14), for instance replacing the L_0-norm by the L_1-norm [11]. The L_0-optimal control problem has also been reformulated using a set of complementarity constraints [19] accordingly

$$\begin{aligned}
\underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) + \boldsymbol{\gamma}^\top (\mathbf{1}_N - \boldsymbol{\xi}) && \text{(15a)}\\
\text{subject to} \quad & c_i(\mathbf{x}) = 0, \ i \in \mathcal{E} && \text{(15b)}\\
& c_i(\mathbf{x}) \leq 0, \ i \in \mathcal{I} && \text{(15c)}\\
& \boldsymbol{\xi} \leq \mathbf{1}_N && \text{(15d)}\\
& \boldsymbol{\xi} \circ \mathbf{x} \leq \epsilon \mathbf{1}_N && \text{(15e)}\\
& -\boldsymbol{\xi} \circ \mathbf{x} \leq \epsilon \mathbf{1}_N && \text{(15f)}\\
& \boldsymbol{\xi} \geq 0, && \text{(15g)}
\end{aligned}$$

where $\mathbf{1}_N$ is the N-vector of ones, N is the number of control intervals, and $\mathbf{1}_N - \boldsymbol{\xi}$ is the support vector of $\mathbf{x}$. The support $1 - \xi_j$ of the state x_j essentially plays the same role as the support, supp$(\cdot)$, in (12). The notation $\mathbf{a} \circ \mathbf{b}$ denotes the componentwise product between the vectors $\mathbf{a}$ and $\mathbf{b}$. $\mathcal{E}$ and $\mathcal{I}$ are two finite index sets, $\boldsymbol{\gamma} > 0$ is a positive vector, $f(\cdot)$ is the continuously differentiable objective function, and c_i the continuously differentiable constraint functions. $\epsilon > 0$ is a relaxation scalar. It is desirable to investigate the properties of the relaxed problem when ϵ approaches zero, because then the complementarity constraints would equal zero.

IV. CONTROL DESIGN

A. Control objectives

The control objective of the maximum hands-off controller is to find the sparsest control among all admissible control. We explore how the maximum hands-off controller works for attitude control of a spacecraft. Note that for the remaining parts of this study, the term *sparsity* is defined accordingly:

Definition 4.1 (Sparsity): The *sparsity* of a control signal refers to the number of control intervals for which the control signal takes on nonzero values.

Although the maximum hands-off controller yields the sparsest control, the control might not occur at the most favorable instants of time, which motivates the design the moving maximum hands-off controller. The term *moving* refers to the characteristic of the controller which lets the user move the sparse control according to a desired set of preferences, for instance environmental constraints.

B. Maximum hands-off controller

The maximum hands-off controller, or the L_0-optimal controller, aims to minimize the L_0-norm of the control input. The design of the maximum hands-off controller implemented in this paper is inspired by the relaxed formulation in (15) with $\boldsymbol{\gamma} = \mathbf{1}_N$, and is formulated as

$$\underset{\boldsymbol{\tau}_u^b, \boldsymbol{\xi}}{\text{minimize}} \quad k_1 f(\boldsymbol{\omega}_{ob}^b) + k_2 g(\mathbf{q}_b^o) + k_3(\mathbf{1}_N - \boldsymbol{\xi})\mathbf{1}_N^\top \tag{16a}$$

$$\text{subject to} \quad \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\tau}_u^w) \tag{16b}$$

$$\boldsymbol{\tau}_u^w \leq \boldsymbol{\tau}_{\text{limit}} \tag{16c}$$

$$\boldsymbol{\tau}_u^w \geq -\boldsymbol{\tau}_{\text{limit}} \tag{16d}$$

$$\mathbf{x}(0) = \mathbf{x}_0 \tag{16e}$$

$$\boldsymbol{\xi} \leq \mathbf{1}_N \tag{16f}$$

$$\boldsymbol{\xi} \circ \boldsymbol{\tau}_{u,1}^b \leq \epsilon \mathbf{1}_N \tag{16g}$$

$$-\boldsymbol{\xi} \circ \boldsymbol{\tau}_{u,1}^b \leq \epsilon \mathbf{1}_N \tag{16h}$$

$$\boldsymbol{\xi} \circ \boldsymbol{\tau}_{u,2}^b \leq \epsilon \mathbf{1}_N \tag{16i}$$

$$-\boldsymbol{\xi} \circ \boldsymbol{\tau}_{u,2}^b \leq \epsilon \mathbf{1}_N \tag{16j}$$

$$\boldsymbol{\xi} \circ \boldsymbol{\tau}_{u,3}^b \leq \epsilon \mathbf{1}_N \tag{16k}$$

$$-\boldsymbol{\xi} \circ \boldsymbol{\tau}_{u,3}^b \leq \epsilon \mathbf{1}_N \tag{16l}$$

$$\boldsymbol{\xi} \geq \mathbf{0}, \tag{16m}$$

where k_1, k_2 and k_3 are positive constants, $\mathbf{1}_N$ is the N-dimensional vector of all ones, where N is the number of control intervals, $\boldsymbol{\xi}$ is the complementarity vector to the control input $\boldsymbol{\tau}_u^b$, and $\mathbf{1}_N - \boldsymbol{\xi}$ is the support vector of $\boldsymbol{\tau}_u^b$. $\boldsymbol{\tau}_{u,1}^b$, $\boldsymbol{\tau}_{u,2}^b$, and $\boldsymbol{\tau}_{u,3}^b$ denotes the components of $\boldsymbol{\tau}_u^b$ about the x-, y- and z-axis in $\{b\}$, respectively. The state vector $\mathbf{x} = [\mathbf{q}_b^o, \boldsymbol{\omega}_{ib}^b, \boldsymbol{\omega}_{ob}^b, \boldsymbol{\omega}_{bw}^w, \mathbf{v}^i, \mathbf{r}^i]^\top$, $\dot{\mathbf{x}} = [\dot{\mathbf{q}}_b^o, \dot{\boldsymbol{\omega}}_{ib}^b, \dot{\boldsymbol{\omega}}_{ob}^b, \dot{\boldsymbol{\omega}}_{bw}^w, \dot{\mathbf{v}}^i, \dot{\mathbf{r}}^i]^\top$ denotes the system dynamics, and $\mathbf{x}_0$ the initial state values. The functions $f(\cdot)$ and $g(\cdot)$ are designed to steer $\boldsymbol{\omega}_{ob}^b$ and $\mathbf{q}_b^o$, respectively, to their desired final states. More specifically,

$$\begin{aligned} f(\boldsymbol{\omega}_{ob}^b) &= \sum_{i=1}^{n} (\omega_{ob,i}^b(T) - \omega_{ob,\text{ref},i}^b)^2 \\ g(\mathbf{q}_b^o) &= 1 - \left|(\mathbf{q}_b^o(T))^\top \mathbf{q}_{b,\text{ref}}^o\right|, \end{aligned} \tag{17}$$

where T denotes the final time, $\boldsymbol{\omega}_{ob,\text{ref}}^b$ and $\mathbf{q}_{b,\text{ref}}^o$ are the reference angular velocity and reference quaternion, n denotes the number of entries in $\boldsymbol{\omega}_{ob}^b(T)$ and $\boldsymbol{\omega}_{ob,\text{ref}}^b$, and $\omega_{ob,i}^b(T)$ and $\omega_{ob,\text{ref},i}^b$ denotes the ith component of $\boldsymbol{\omega}_{ob}^b(T)$ and $\boldsymbol{\omega}_{ob,\text{ref}}^b$, respectively. The function $f(\cdot)$ yields the sum of the squared elements of a vector, and is implemented using MATLAB's SUMSQR-function. The function $g(\cdot)$ is a pseudometric on the unit quaternion, but a metric on $SO(3)$ [20]. The absolute value, $\left|(\mathbf{q}_b^o(T))^\top \mathbf{q}_{b,\text{ref}}^o\right|$, is implemented using the smooth maximum function $\max(x_1, x_2) = \frac{1}{2} \cdot \left((x_1 + x_2) + \sqrt{(x_1 - x_2)^2 + \alpha}\right)$, where α is a parameter to be tuned [21].

During the design phase of the maximum hands-off controller, different ways to formulate the control problem were explored, before the final design choice was made. The formulation of the maximum hands-off control problem presented in (16) was chosen because it yielded better results for the spacecraft attitude control problem compared to the other designs that were tested.

C. *Moving maximum hands-off controller*

The design of the moving maximum hands-off controller is similar to that of the maximum hands-off controller in (16). The moving maximum hands-off optimal control problem is formulated as

$$\underset{\boldsymbol{\tau}_u^b, \boldsymbol{\xi}}{\text{minimize}} \quad k_1 f(\boldsymbol{\omega}_{ob}^b) + k_2 g(\mathbf{q}_b^o) + k_3(\mathbf{1}_N - \boldsymbol{\xi} \circ \mathbf{h}_N)\mathbf{1}_N^\top, \tag{18}$$

where the constraints on (18) are identical to (16b) to (16l). The vector $\mathbf{h}_N$ specifies where the control torques should occur. By default, the vector $\mathbf{h}_N$ is a N-dimensional vector of all ones. The user might change the values of $\mathbf{h}_N$ to values between 0 and 1, to indicate for which time intervals the control input should occur. For instance, if the value of $\mathbf{h}_N$ is set to 0.1 for $N = 10, \ldots, 20$, it could yield a more optimal solution if the control occur between these control intervals as sparsity comes at a lower cost in this interval than the rest.

D. *L_1-optimal controller*

The design of the L_1-optimal control problem is formulated as

$$\underset{s, \boldsymbol{\tau}_u^b}{\text{minimize}} \quad k_1 f(\boldsymbol{\omega}_{ob}^b) + k_2 g(\mathbf{q}_b^o) + k_3 \sum_{i=0}^{N} s_k \tag{19}$$

where the constraints on (19) equals (16b) to (16e) in addition to the constraint $-\mathbf{s}_N \leq \boldsymbol{\tau}_u^b \leq \mathbf{s}_N$. $\sum_{i=0}^{N} s_k$ denotes the L_1-norm of $\boldsymbol{\tau}_u^b$, i.e., $\|\boldsymbol{\tau}_u^b\|_1 = \sum_{i=0}^{n} s_i = \mathbf{1}^\top \mathbf{s}$, where $\mathbf{s} \in \mathbb{R}^n$ is a set of slack variables. The formulation of the L_1-norm in (19) is inspired by [22].

E. *PD-controller*

A PD controller was implemented to test if the satellite dynamics work as intended. The PD controller is defined in [1]

$$\boldsymbol{\tau}_u^b = \mathbf{K}_d \boldsymbol{\omega}_e^b + \mathbf{K}_p \boldsymbol{\epsilon}_e, \tag{20}$$

where $\boldsymbol{\omega}_e^b$ is the angular velocity error, defined as $\boldsymbol{\omega}_e^b = \boldsymbol{\omega}_{ob}^b - \boldsymbol{\omega}_d^b$, where $\boldsymbol{\omega}_d^b$. $\boldsymbol{\epsilon}_e$ is the vector part of the error quaternion, defined as $\mathbf{q}_e = \mathbf{q}_d^{-1} \otimes \mathbf{q}_b^o$, where $\mathbf{q}_d = [\eta_d, \boldsymbol{\epsilon}_b^\top]^\top$ is the desired attitude represented using quaternions. $\mathbf{K}_d$ and $\mathbf{K}_p$ are constant and positive definite controller gain matrices. The numerical values of the gain matrices were chosen while tuning the PD controller.

V. SIMULATION SETUP

The experiments were conducted using CasADi as the optimization tool [23]. The NLP-solver IPOPT was used to solve the optimization problems, using the solver's default options. The optimal control problems in (16), (18) and (19) were discretized using direct multiple-shooting, and the dynamics of the spacecraft were discretized and implemented using Runge-Kutta 4 integration. Initial guesses for the starting point were given to IPOPT at the beginning of the optimization. The output from the PD controller was given as the initial guesses for the L_1-optimal controller, and the output from the L_1-optimal controller was given as initial guesses for the maximum hands-off controller and the moving maximum hands-off controller. The initial guesses were applied to the states $\boldsymbol{q}_b^o$, $\boldsymbol{\omega}_{ib}^b$, and $\boldsymbol{\omega}_{bw}^w$. No initial guesses were applied to the control torque $\boldsymbol{\tau}_u^b$, i.e., the default initial guess used by CasADi were applied. CasADi's default initial guess is a vector of all zeros.

Some tests were conducted with different initial guesses to see how the controllers responded. The tests revealed that the initial guesses for IPOPT are of significant importance when trying to find the optimal solutions. Different initial guesses causes the optimization to iterate fast or slow towards an optimal solution, and affect the quality of the solution.

The experiments reported in this thesis were conducted using a 2 GHz Intel Core i7-9700T CPU computer running Windows. The simulation of the experiments were conducted using the parameters for a 6U M6P CubeSat as the spacecraft rigid body, and it is assumed to orbit in Low-Earth-Orbit (LEO). The CubeSat uses a M6P platform from NanoAvionics, and four reaction wheels are used to control the attitude of the CubeSat. This is part of the setup that is going to be used for the HYPSO-1 mission which motivates this work. Three of the reaction wheels are placed orthogonally along the three axis of the body frame. The fourth reaction wheel is placed such that it's torque yields equal components in each of the body axes. The torque distribution matrix $\mathbf{A}$ is given as [13]

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & \frac{1}{\sqrt{3}} \\ 0 & 1 & 0 & \frac{1}{\sqrt{3}} \\ 0 & 0 & 1 & \frac{1}{\sqrt{3}} \end{bmatrix}. \tag{21}$$

The total inertia matrix for the spacecraft rigid body and the inertia matrix of the reaction wheels are given as [13]

$$\mathbf{J} = \begin{bmatrix} 0.0775 & 0.0002 & -0.0002 \\ 0.0002 & 0.1067 & 0.0005 \\ -0.0002 & 0.0005 & 0.0389 \end{bmatrix} \text{kg}\cdot\text{m}^2,\ \mathbf{J}_w = J_w \mathbf{I}_{4\times 4}, \tag{22}$$

where $J_w = 2.1 \cdot 10^{-4}$ kg·m^2 is the inertia of a single reaction wheel. The inertia of all four reaction wheels are set to be identical. The controller gains, and the parameters required for the optimization are shown in Table I.

After the optimization procedure finishes, the system is propagated for an additional number of control intervals, denoted N_{prop}, to visualize what happens after the optimization finishes. The final state from the optimization serves as the initial state for the propagation, and the control input is set to zero for the whole propagation. The orbit of the spacecraft is initialized using the orbital parameters in Table I, which are transformed into ECI coordinates using the RANDV-function from [24].

Parameter	**Value**	**Unit**
k_1	$1 \cdot 10^6$	s^2
k_2	$1 \cdot 10^2$	-
k_3	$1 \cdot 10^1$	-
$\mathbf{K}_p$	$3 \cdot \mathbf{J}_s$	N·m
$\mathbf{K}_d$	$2.7 \cdot \mathbf{J}_s$	N·m·s
Simulation time (T)	70	s
Control intervals (N)	50	-
Step size (h)	1.4	s
Control intervals for propagation (N_{prop})	20	-
ϵ	$1 \cdot 10^{-8}$	-
τ_{limit}	$\pm 3 \cdot 10^{-3}$	N·m
Semi-major axis	6852.2	km
Eccentricity	0.002	-
Inclination	97	°
Right ascension of the ascending node	280	°
Argument of periapsis	0	°
True anomaly	0	°
Standard grav. parameter, Earth (μ)	$3.986 \cdot 10^{14}$	m^3/s^2

TABLE I

CONTROLLER GAINS, OPTIMIZATION CONSTANTS, AND ORBITAL PARAMETERS.

VI. RESULTS

Fig. 1 shows the change in the spacecraft's attitude over time, and reveals that all three controllers are able to steer the spacecraft to the desired orientation of (ϕ, θ, ψ) = $(45°, 0°, 0°)$ and keep the spacecraft at this attitude. The dotted lines ϕ_d, θ_d, and ψ_d denotes the angles of the desired attitude, whereas the solid lines ϕ, θ, and ψ corresponds to the actual states. The torque vector $\boldsymbol{\tau}_u^b$ can be seen in Fig. 3. The torque vector $\boldsymbol{\tau}_u^w$ can be seen in Fig. 3. The angular velocity $\boldsymbol{\omega}_{ob}^b$ is shown in Fig. 2 and reveals that the spacecraft stops rotating, i.e., $\{b\}$ stops rotating relative to $\{o\}$, when the desired orientation is reached. The angular velocity $\boldsymbol{\omega}_{bw}^w$ can be seen in Fig. 5, and illustrates the dynamical response of the reaction wheels. The computation times, the sparsity, and the number of iterations used to find the optimal solution for each of the three controllers are shown in Table II.

Fig. 6 shows the change in the spacecraft's attitude over time from a multiple-axis maneuver from (ϕ, θ, ψ) = $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$. The optimal control torques $\boldsymbol{\tau}_u^b$ and $\boldsymbol{\tau}_u^w$ are shown in Fig. 8 and Fig. 9, respectively. The angular velocity $\boldsymbol{\omega}_{ob}^b$ is shown in Fig. 7. The angular velocity $\boldsymbol{\omega}_{bw}^w$ is shown in Fig. 10. The computation times, the sparsity, and the number of iterations used to find the optimal solution for each of the three controllers are shown in Table III.

Controller	**CPU time NLP, [s]**	**CPU time IPOPT, [s]**	**Sparsity**	**Iterations**
L_0	3.235	0.253	2	25
Moving L_0	46.914	3.828	2	371
L_1	2.479	0.204	2	19

TABLE II

COMPUTATION TIME, SPARSITY AND NUMBER OF ITERATIONS SINGLE-AXIS MANEUVER FROM $(0°, 0°, 0°)$ TO $(45°, 0°, 0°)$.

VII. DISCUSSION

In this section, the maximum hands-off controller will be referred to as the L_0-optimal controller or the L_0-controller, whereas the moving maximum hands-off controller will

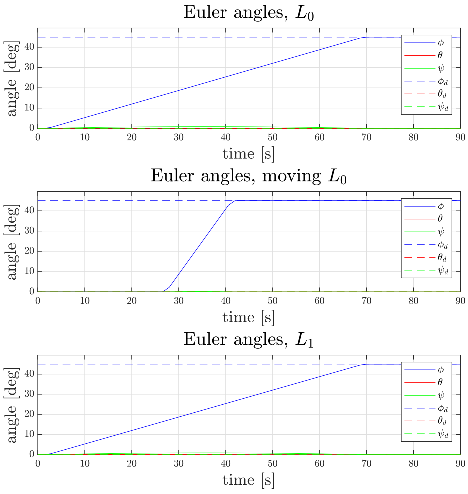


Fig. 1. Euler angles, single-axis maneuver from $(0°, 0°, 0°)$ to $(45°, 0°, 0°)$.

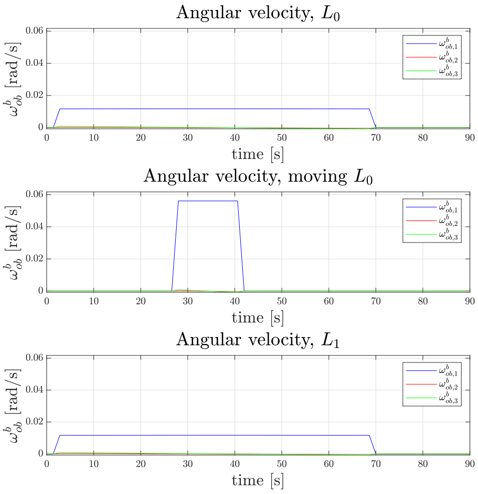


Fig. 2. Angular velocity, $\boldsymbol{\omega}^b_{ob}$, single-axis maneuver from $(0°, 0°, 0°)$ to $(45°, 0°, 0°)$.

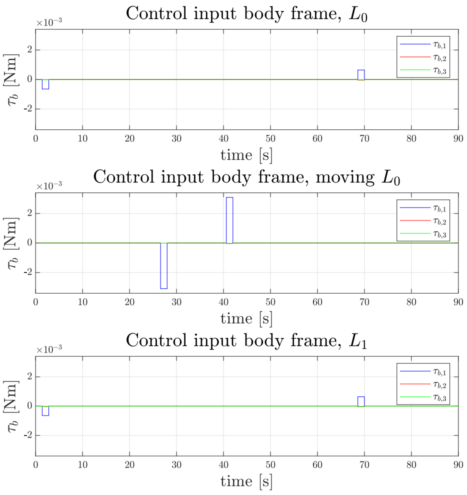


Fig. 3. Control input in $\{b\}$, single-axis maneuver from $(0°, 0°, 0°)$ to $(45°, 0°, 0°)$.

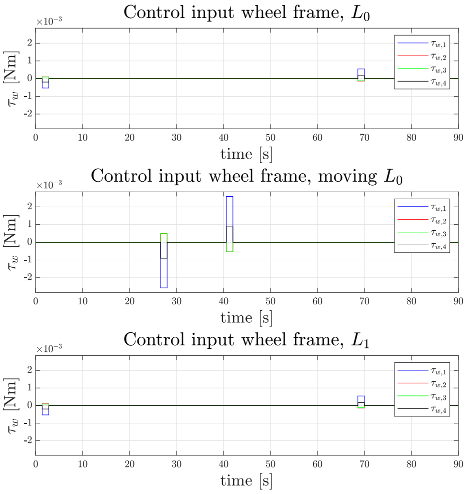


Fig. 4. Control input in $\{w\}$, single-axis maneuver from $(0°, 0°, 0°)$ to $(45°, 0°, 0°)$.

Controller	**CPU time NLP, [s]**	**CPU time IPOPT, [s]**	**Sparsity**	**Iterations**
L_0	66.216	6.106	2	497
Moving L_0	90.175	7.706	2	699
L_1	5.131	0.360	2	39

TABLE III

COMPUTATION TIME, SPARSITY AND NUMBER OF ITERATIONS MULTIPLE-AXIS MANEUVER FROM $(0°, 0°, 0°)$ TO $(90°, 45°, 15°)$.

be referred to as the moving L_0-optimal controller or the moving L_0-controller. The *sparsity* of a control signal refers to the number of nonzero control intervals in the signal, as defined in Definition 4.1.

A. *Single-axis*

Figs. 1 to 4, Fig. 2 show that the spacecraft's state space trajectories and the control signals are identical for the L_0-controller and the L_1-controller. These results are in agreement with the findings in [11], and suggests that the L_1-norm may be used as an approximation to the L_0-norm.

Figs. 1 to 4 show that the spacecraft's state space trajectories and the control signals resulting from the moving L_0-controller differs from the two other controllers. The differences can be explained by comparing the cost functions of the three controllers in (16a), (18), and (19). For the moving L_0-controller, the vector $\mathbf{h}_N$ was chosen such that it would cost less for the control inputs to occur between $t = 28$ s and $t = 42$ s. For the two other controllers, it is equally expensive for the control inputs to occur over the whole time interval. Fig. 3 and Fig. 4 show that the control inputs produced by the moving L_0-controller occur

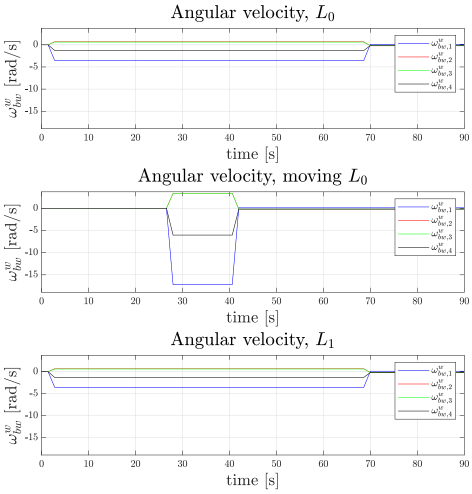


Fig. 5. Angular velocity, $\boldsymbol{\omega}_{bw}^{w}$, single-axis maneuver from $(0°, 0°, 0°)$ to $(45°, 0°, 0°)$.

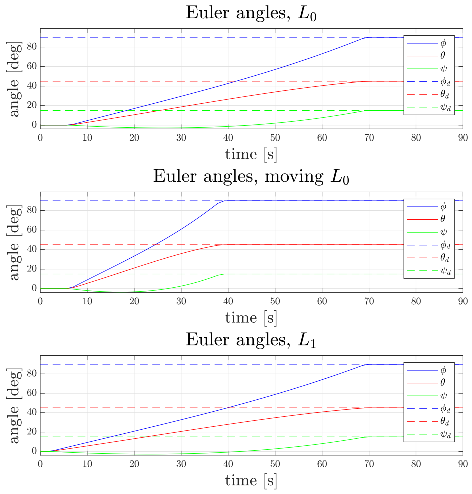


Fig. 6. Euler angles, multiple-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$.

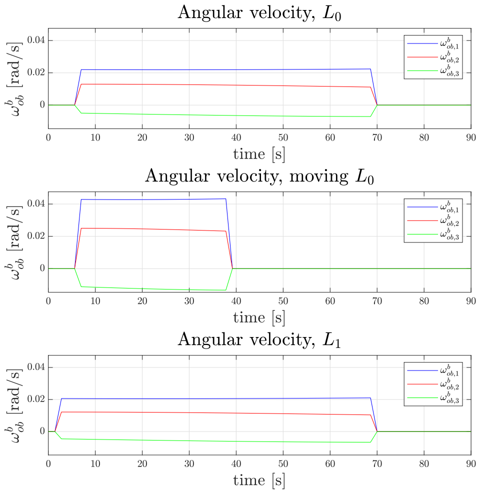


Fig. 7. Angular velocity, $\boldsymbol{\omega}_{ob}^{b}$, multiple-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$.

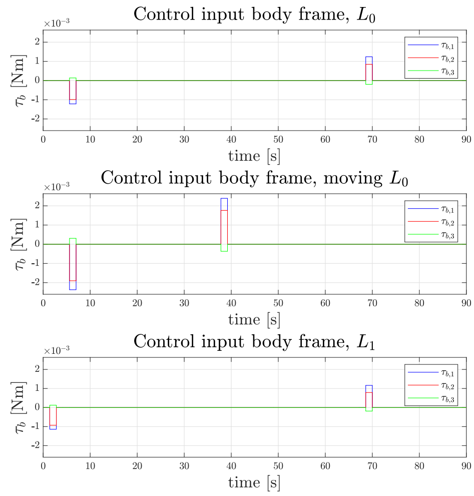
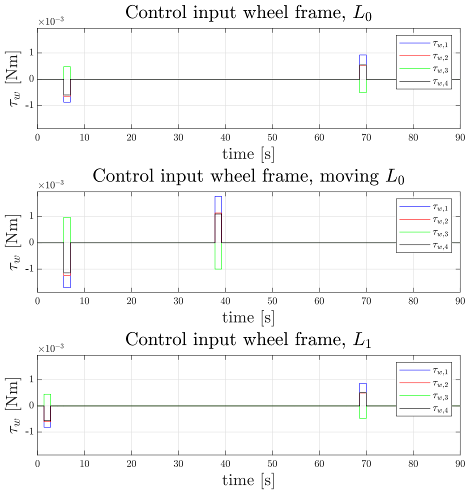


Fig. 8. Control input in $\{b\}$, multiple-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$.

at $t = 28$ s and $t = 42$ s, and Fig. 1 and Fig. 2 show that the spacecraft's states changes within this interval.

Fig. 3 and Fig. 4 show that the optimal control signal computed by the moving L_0-controller has larger amplitude than the control signals produced by the two other controllers. A reasonable explanation for this may be that since the moving L_0-control torques occur closer in time, the torque applied at each of the two time instants has to be larger in order to steer the spacecraft to the desired orientation within a smaller time interval. To reach the desired position in a short time interval, the control torques applied to the spacecraft has to be larger than if the time interval was longer. If the time interval was larger, the control inputs could be smaller as the spacecraft would have more time to rotate towards the desired orientation after the initial control input has been applied.

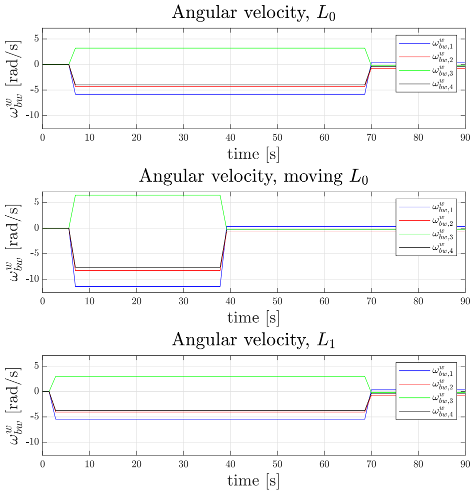
Fig. 3 and Fig. 4 show that the control inputs produced by the moving L_0-controller occur at $t = 28$ s and $t = 42$ s, which means that they occur exactly at the boundaries of the time interval specified by $\mathbf{h}_N$. The saturation limits for the control torque are $\tau_{\text{limit}} = \pm 3 \cdot 10^{-3}$ N·m, and Fig. 4 shows that the control torques produced by the moving L_0-controller are close to the saturation limits. If the control torques had occurred at other time instants in the interval specified by $\mathbf{h}_N$, they would occur closer in time and the torques would therefore have larger values. The control torques are already close to the saturation limits when they occur at $t = 28$ s and $t = 42$ s, and if the torque values were to increase the reaction wheels may saturate. If the reaction wheels saturate, an additional control torque may be required to perform the spacecraft maneuver, and an additional control torque would yield a less sparse control signal. It is cheaper for the control input to occur between $t = 28$ s and $t = 42$ s, but if the control torques are too close in time they may saturate. Then, the controller would have to apply an additional control torque which would result in


Control input wheel frame, L_0

Control input wheel frame, moving L_0

Control input wheel frame, L_1

Fig. 9. Control input in $\{w\}$, multiple-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$.


Angular velocity, L_0

Angular velocity, moving L_0

Angular velocity, L_1

Fig. 10. Angular velocity, $\boldsymbol{\omega}_{bw}^{w}$, multiple-axis maneuver from $(0°, 0°, 0°)$ to $(90°, 45°, 15°)$.

a less sparse control signal. Therefore, it makes sense that the control torques occur at the borders of the time interval specified by $\mathbf{h}_N$, i.e., at $t = 28$ s and $t = 42$ s.

Table II shows that all three controllers yield optimal control signals which have the same sparsity. This finding confirms that all three controllers are able find the sparsest solution. For a spacecraft single-axis maneuver, it may not be possible to find a control signal which are sparser than 2, as one control torque has to push the spacecraft towards the desired attitude and one control input has to stop the spacecraft rotation. If the spacecraft was operating inside the Earth's atmosphere it might be possible to obtain a sparser control signal, as it may be possible to apply control torque to the spacecraft at only one instant. This control torque would push the spacecraft towards its desired attitude, and instead of applying a second control torque to stop the spacecraft, it could be possible to rely on the air resistance to slow the spacecraft down. Relying on the air resistance to slow the spacecraft down would require a rather long time horizon, since the air resistance does not affect the spacecraft much in orbit. Air resistance is less important as the altitude of the spacecraft increases, and therefore a control torque would often be needed to stop the spacecraft from rotating when the spacecraft is operating at a higher altitude. Hence, for the conditions provided in this paper, a control signal in which torques occur at two different time instants provides the sparsest optimal control signal for the spacecraft single-axis maneuver.

When trying to maneuver the spacecraft an angle $\phi = 45°$ about the x-axis, one might expect an optimal controller to yield control torque simply about the body-frame x-axis. Fig. 3 shows that control torques are applied about all three axes, although the control torque about the x-axis is the most prominent. The reason why the optimal control algorithms yield torque about all three axes is because the spacecraft rotates relative to it's orbit at the same time as it orbits the Earth. If the spacecraft is not operating in orbit, which would be the case if the spacecraft operated in deep-space, then the orbit-frame $\{o\}$ would not be well-defined. Then, the angular velocity of the spacecraft would be defined as $\boldsymbol{\omega_{ob}^{b}} = \boldsymbol{\omega_{ib}^{b}}$. A spacecraft operating in LEO, such as the one used in this work, would rotate relative to its orbit, which results in the angular velocity in (9c). Because of the spacecraft's rotation around the Earth, there will be rotation about the y- and z-axis throughout the optimization horizon, in addition to the control effort made about the x-axis. Therefore, torque from the reaction wheels has to be applied to compensate for the drift about the y- and z-axis.

The total inertia matrix for the spacecraft rigid body, $\mathbf{J}$, in (22) also contributes to rotation about multiple axes. Due to the nonlinearity of the spacecraft dynamics, the terms are coupled, which results in torque about all three axes, even though motion is only needed about one axis. The total system inertia matrix is not diagonal and given in (22). If a diagonal $\mathbf{J}$-matrix were used instead of the one in (22), the states would be less coupled, which is clear from (9b). Less coupling of the dynamics would yield less torque about the y- and z-axis, when a maneuver is performed about the x-axis.

B. Multiple-axis maneuver

Fig. 8 and Fig. 9 show that the first control torque from the moving L_0-controller occurs after about $t = 5$ s, and the second control torque occurs close to $t = 40$ s. The second control torque occurs within the interval specified by $\mathbf{h}_N$. The vector $\mathbf{h}_N$ was chosen such that it would cost less for the control inputs to occur between $t = 28$ s and $t = 42$ s. One might have expected all control inputs to occur within this time interval. However, one control input occurs outside this interval. The reason for this is that there are no constraints on where the control input should not occur; it only costs less between $t = 28$ s and $t = 42$ s. The optimization procedure aims to satisfy the constraints and reach the final state values while minimizing the cost function. If it is not possible to reach this goal by applying control input within the cheap interval specified by $\mathbf{h}_N$, some or all of the control input will occur outside this interval. Therefore, control inputs may occur outside the interval defined by $\mathbf{h}_N$.

The L_1-optimal control signal and L_0-optimal control signal are not identical, which can be seen in Fig. 8 and Fig. 9. The first control input produced by the L_1-controller occurs before the first control input produced by the L_0-controller, whereas the two last control inputs occur simultaneously. The amplitude of the control torques from the L_1-controller are smaller than those from the L_0-controller. A possible explanation for the difference in amplitudes is that since there are more time between the two L_1-control inputs, the spacecraft will have more time to rotate to the desired orientation, and thus less torque would need to be applied. Therefore, it makes sense that the control torque produced by the L_1-controller are smaller the the control torque produced by the L_0-controller. These results suggest that the L_1-optimal solution does not always equal the L_0-optimal solution. On the other hand, they suggest that the L_1-optimal control problem could be an acceptable approximation to the L_0-optimal control problem.

VIII. CONCLUSION

The main goal of this paper has been to explore the use of maximum hands-off control, also called L_0-optimal control, for the spacecraft attitude control problem. Our work has shown that the maximum hands-off controller is able to steer the spacecraft to the desired attitude and the desired final states. Thus, our work confirms that the maximum hands-off controller works for the spacecraft attitude control problem. The use of the moving maximum hands-off controller has also been explored for the spacecraft attitude control problem. Our findings suggest that the controller works as intended, which means that the controller produces a control signal that can be moved to a predefined interval specified by the vector $\mathbf{h}_N$. While the cost in at certain intervals might be lowered with the choice of $\mathbf{h}_N$, control torque may still occur outside this interval as the optimization procedure aims to satisfy the constraints while minimizing the cost function.

REFERENCES

[1] J.-Y. Wen and K. Kreutz-Delgado, "The attitude control problem," *IEEE Transactions on Automatic control*, vol. 36, no. 10, pp. 1148–1162, 1991.

[2] 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, 2008.

[3] G. Meyer, *Design and global analysis of spacecraft attitude control systems*. National Aeronautics and Space Administration, Ames Research Center, 1971.

[4] L.-L. Show, J.-C. Juang, C.-T. Lin, and Y.-W. Jan, "Spacecraft robust attitude tracking design: Pid control approach," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*, vol. 2. IEEE, 2002, pp. 1360–1365.

[5] S. Joshi, A. Kelkar, and J.-Y. Wen, "Robust attitude stabilization of spacecraft using nonlinear quaternion feedback," *IEEE Transactions on Automatic control*, vol. 40, no. 10, pp. 1800–1803, 1995.

[6] T. Lee, M. Leok, and N. H. McClamroch, "Time optimal attitude control for a rigid body," in *2008 American Control Conference*. IEEE, 2008, pp. 5210–5215.

[7] H. Shen and P. Tsiotras, "Time-optimal control of axisymmetric rigid spacecraft using two controls," *Journal of Guidance, Control, and Dynamics*, vol. 22, no. 5, pp. 682–694, 1999.

[8] T. Zhang, "Analysis of multi-stage convex relaxation for sparse regularization." *Journal of Machine Learning Research*, vol. 11, no. 3, 2010.

[9] J. D. Biggs and L. Colley, "Geometric attitude motion planning for spacecraft with pointing and actuator constraints," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 7, pp. 1672–1677, 2016.

[10] D. Chatterjee, M. Nagahara, D. E. Quevedo, and K. M. Rao, "Characterization of maximum hands-off control," *Systems & Control Letters*, vol. 94, pp. 31–36, 2016.

[11] M. Nagahara, D. E. Quevedo, and D. Nešić, "Maximum hands-off control: a paradigm of control effort minimization," *IEEE Transactions on Automatic Control*, vol. 61, no. 3, pp. 735–747, 2015.

[12] O. Egeland and J. T. Gravdahl, *Modeling and simulation for automatic control*, corr., 2. print ed. Trondheim: Marine Cybernetics AS, 2003, oCLC: 255655256.

[13] B. A. Kristiansen, M. E. Grøtte, and J. T. Gravdahl, "Quaternion-based generalized super-twisting algorithm for spacecraft attitude control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 14 811–14 818, 2020, 21st IFAC World Congress.

[14] E. Oland and R. Schlanbusch, "Reaction wheel design for cubesats," in *2009 4th International Conference on Recent Advances in Space Technologies*. IEEE, 2009, pp. 778–783.

[15] M. E. Grøtte, J. T. Gravdahl, T. A. Johansen, J. A. Larsen, E. M. Vidal, and E. Surma, "Spacecraft attitude and angular rate tracking using reaction wheels and magnetorquers," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 14 819–14 826, 2020, 21st IFAC World Congress.

[16] T. R. Krogstad and J. T. Gravdahl, "6-dof mutual synchronization of formation flying spacecraft," in *Proceedings of the 45th IEEE Conference on Decision and Control*. IEEE, 2006, pp. 5706–5711.

[17] H. K. Khalil, "Nonlinear systems," 2002.

[18] H. L. Royden and P. Fitzpatrick, *Real analysis*. Macmillan New York, 1988, vol. 32.

[19] M. Feng, J. E. Mitchell, J.-S. Pang, X. Shen, and A. Wächter, "Complementarity formulations of l0-norm optimization problems," *Industrial Engineering and Management Sciences. Technical Report. Northwestern University, Evanston, IL, USA*, 2016.

[20] D. Q. Huynh, "Metrics for 3d rotations: Comparison and analysis," *Journal of Mathematical Imaging and Vision*, vol. 35, no. 2, pp. 155–164, 2009.

[21] B. A. Kristiansen, J. T. Gravdahl, and T. A. Johansen, "Energy optimal attitude control for a solar-powered spacecraft," *European Journal of Control*, 2021, accepted for publication in the 2021 European Control Conference Special Issue.

[22] S. Gros and M. Diehl, "Numerical optimal control (draft)," 2019.

[23] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[24] D. A. Vallado, *Fundamentals of astrodynamics and applications*. Springer Science & Business Media, 2001, vol. 12.

Sigrid Kjønnø Schaanning

Maximum Hands-Off Control for Attitude Control of a Spacecraft

NTNU
Norwegian University of
Science and Technology