

POLITECNICO DI TORINO

Corso di Laurea in Ingegneria Aerospaziale



Master's Degree Thesis

**Boundary layer first order sliding mode
for attitude control of a CMG-based
spacecraft**

Supervisors

Prof. Elisa CAPELLO

Prof. Satoshi SATOH

Candidate

Marta PUPPI

October 2024

Abstract

Regardless of the satellite mission, its attitude control is crucial. Proper orientation in space is fundamental, necessitating the development of an effective control law to achieve the desired attitude. This thesis evaluates and compares classical and adaptive sliding mode control (SMC) algorithms for this purpose. The study focuses on a satellite equipped with four Control Moment Gyroscopes (CMGs) arranged in a pyramidal configuration. The CMGs serve as actuators to adjust the satellite attitude and achieve the desired orientation. Simulations using MATLAB are conducted to compare classical and adaptive Boundary Layer First Order Sliding Mode Control (BLFOSMC), a novel control algorithm that integrates adaptive control techniques with SMC. Subsequently, the MATLAB algorithm is translated into C++ to perform testing on the hardware testbed located in Satoh's laboratory at Osaka University.

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Professor Capello and Professor Satoh, for their invaluable guidance and encouragement throughout my research journey. I am also very grateful to Dr. Mancini for his help and insightful contributions to my work.

I am particularly thankful to my mother, father, grandmother and boyfriend, whose love and encouragement have kept me motivated during this challenging adventure.

Lastly, I extend my appreciation to my friends who supported me throughout my university years. Your companionship and encouragement have made this journey much more enjoyable.

Table of Contents

List of Tables	IV
List of Figures	V
1 Introduction	1
2 Mathematical models and control strategies	3
2.1 Quaternions	3
2.2 Dynamics equation	5
2.3 Kinematics equation	10
2.4 State equation	11
2.5 Error system	12
2.6 Plant system	14
2.7 Actuators	17
2.7.1 Control moment gyroscopes	17
2.7.2 CMG system dynamics	19
2.8 Sliding mode control	22
2.8.1 Description	22
2.8.2 Implementation	24
3 Simulations results and experimental results	27
3.1 Numerical simulations	28
3.2 Experimental results and comparison	36
3.2.1 Triangular angular velocity reference	37
3.2.2 Constant reference	63
3.2.3 Observations	89
4 Conclusions	90
Bibliography	92

List of Tables

3.1	Traditional BLFOSM parameters	28
3.2	Adaptive BLFOSM parameters	28
3.3	Traditional BLFOSM parameters for experimental results	36
3.4	Adaptive BLFOSM parameters for experimental results	36

List of Figures

2.1	Gimbal lock [12]	18
2.2	CMG system in pyramidal configuration [12]	18
2.3	Sliding surface [2]	22
2.4	Reaching phase and sliding phase [2]	22
2.5	Boundary layer [2]	23
2.6	Sign function vs saturation function [2]	25
3.1	Testbed platform	27
3.2	Comparison: quaternion q_1	29
3.3	Comparison: quaternion q_2	29
3.4	Comparison: quaternion q_3	30
3.5	Comparison: quaternion q_4	30
3.6	Comparison: angular velocity ω_1	31
3.7	Comparison: angular velocity ω_2	31
3.8	Comparison: angular velocity ω_3	32
3.9	Comparison: sliding variable σ_1	32
3.10	Comparison: sliding variable σ_2	33
3.11	Comparison: sliding variable σ_3	33
3.12	Comparison: slope λ	34
3.13	Numerical simulation - Traditional BLFOSM - $k = 0.01$ - Triangular angular velocity: angular velocity	37
3.14	Numerical simulation - Traditional BLFOSM - $k = 0.01$ - Triangular angular velocity: quaternion	38
3.15	Numerical simulation - Traditional BLFOSM - $k = 0.01$ - Triangular angular velocity: sliding variable	38
3.16	Numerical simulation - Traditional BLFOSM - $k = 0.01$ - Triangular angular velocity: control input	39
3.17	Numerical simulation - Adaptive BLFOSM - $k = 0.01$ - Triangular angular velocity: angular velocity	40
3.18	Numerical simulation - Adaptive BLFOSM - $k = 0.01$ - Triangular angular velocity: quaternion	40

3.19	Numerical simulation - Adaptive BLFOSM - $k = 0.01$ - Triangular angular velocity: sliding variable	41
3.20	Numerical simulation - Adaptive BLFOSM - $k = 0.01$ - Triangular angular velocity: slope	41
3.21	Numerical simulation - Adaptive BLFOSM - $k = 0.01$ - Triangular angular velocity: control input	42
3.22	Numerical simulation - Traditional BLFOSM - $k = 0.5$ - Triangular angular velocity: angular velocity	43
3.23	Numerical simulation - Traditional BLFOSM - $k = 0.5$ - Triangular angular velocity: quaternion	43
3.24	Numerical simulation - Traditional BLFOSM - $k = 0.5$ - Triangular angular velocity: sliding variable	44
3.25	Numerical simulation - Traditional BLFOSM - $k = 0.5$ - Triangular angular velocity: control input	44
3.26	Numerical simulation - Adaptive BLFOSM - $k = 0.5$ - Triangular angular velocity: angular velocity	46
3.27	Numerical simulation - Adaptive BLFOSM - $k = 0.5$ - Triangular angular velocity: quaternion	46
3.28	Numerical simulation - Adaptive BLFOSM - $k = 0.5$ - Triangular angular velocity: sliding variable	47
3.29	Numerical simulation - Adaptive BLFOSM - $k = 0.5$ - Triangular angular velocity: slope	47
3.30	Numerical simulation - Adaptive BLFOSM - $k = 0.5$ - Triangular angular velocity: control input	48
3.31	Experimental testing - Traditional BLFOSM - $k = 0.01$ - Triangular angular velocity: angular velocity	49
3.32	Experimental testing - Traditional BLFOSM - $k = 0.01$ - Triangular angular velocity: quaternion	50
3.33	Experimental testing - Traditional BLFOSM - $k = 0.01$ - Triangular angular velocity: sliding variable	50
3.34	Experimental testing - Traditional BLFOSM - $k = 0.01$ - Triangular angular velocity: control input	51
3.35	Experimental testing - Adaptive BLFOSM - $k = 0.01$ - Triangular angular velocity: angular velocity	52
3.36	Experimental testing - Adaptive BLFOSM - $k = 0.01$ - Triangular angular velocity: quaternion	53
3.37	Experimental testing - Adaptive BLFOSM - $k = 0.01$ - Triangular angular velocity: sliding variable	53
3.38	Experimental testing - Adaptive BLFOSM - $k = 0.01$ - Triangular angular velocity: slope	54

3.39	Experimental testing - Adaptive BLFOSM - $k = 0.01$ - Triangular angular velocity: control input	54
3.40	Experimental testing - Traditional BLFOSM - $k = 0.5$ - Triangular angular velocity: angular velocity	56
3.41	Experimental testing - Traditional BLFOSM - $k = 0.5$ - Triangular angular velocity: quaternion	57
3.42	Experimental testing - Traditional BLFOSM - $k = 0.5$ - Triangular angular velocity: sliding variable	57
3.43	Experimental testing - Traditional BLFOSM - $k = 0.5$ - Triangular angular velocity: control input	58
3.44	Experimental testing - Adaptive BLFOSM - $k = 0.5$ - Triangular angular velocity: angular velocity	59
3.45	Experimental testing - Adaptive BLFOSM - $k = 0.5$ - Triangular angular velocity: quaternion	60
3.46	Experimental testing - Adaptive BLFOSM - $k = 0.5$ - Triangular angular velocity: sliding variable	60
3.47	Experimental testing - Adaptive BLFOSM - $k = 0.5$ - Triangular angular velocity: slope	61
3.48	Experimental testing - Adaptive BLFOSM - $k = 0.5$ - Triangular angular velocity: control input	61
3.49	Numerical simulation - Traditional BLFOSM - $k = 0.01$ - Constant reference: angular velocity	63
3.50	Numerical simulation - Traditional BLFOSM - $k = 0.01$ - Constant reference: quaternion	64
3.51	Numerical simulation - Traditional BLFOSM - $k = 0.01$ - Constant reference: sliding variable	64
3.52	Numerical simulation - Traditional BLFOSM - $k = 0.01$ - Constant reference: control input	65
3.53	Numerical simulation - Adaptive BLFOSM - $k = 0.01$ - Constant reference: angular velocity	66
3.54	Numerical simulation - Adaptive BLFOSM - $k = 0.01$ - Constant reference: quaternion	66
3.55	Numerical simulation - Adaptive BLFOSM - $k = 0.01$ - Constant reference: sliding variable	67
3.56	Numerical simulation - Adaptive BLFOSM - $k = 0.01$ - Constant reference: slope	67
3.57	Numerical simulation - Adaptive BLFOSM - $k = 0.01$ - Constant reference: control input	68
3.58	Numerical simulation - Traditional BLFOSM - $k = 0.5$ - Constant reference: angular velocity	69

3.59	Numerical simulation - Traditional BLFOSM - $k = 0.5$ - Constant reference: quaternion	69
3.60	Numerical simulation - Traditional BLFOSM - $k = 0.5$ - Constant reference: sliding variable	70
3.61	Numerical simulation - Traditional BLFOSM - $k = 0.5$ - Constant reference: control input	70
3.62	Numerical simulation - Adaptive BLFOSM - $k = 0.5$ - Constant reference: angular velocity	72
3.63	Numerical simulation - Adaptive BLFOSM - $k = 0.5$ - Constant reference: quaternion	72
3.64	Numerical simulation - Adaptive BLFOSM - $k = 0.5$ - Constant reference: sliding variable	73
3.65	Numerical simulation - Adaptive BLFOSM - $k = 0.5$ - Constant reference: slope	73
3.66	Numerical simulation - Adaptive BLFOSM - $k = 0.5$ - Constant reference: control input	74
3.67	Experimental testing - Traditional BLFOSM - $k = 0.01$ - Constant reference: angular velocity	75
3.68	Experimental testing - Traditional BLFOSM - $k = 0.01$ - Constant reference: quaternion	76
3.69	Experimental testing - Traditional BLFOSM - $k = 0.01$ - Constant reference: sliding variable	76
3.70	Experimental testing - Traditional BLFOSM - $k = 0.01$ - Constant reference: control input	77
3.71	Experimental testing - Adaptive BLFOSM - $k = 0.01$ - Constant reference: angular velocity	78
3.72	Experimental testing - Adaptive BLFOSM - $k = 0.01$ - Constant reference: quaternion	79
3.73	Experimental testing - Adaptive BLFOSM - $k = 0.01$ - Constant reference: sliding variable	79
3.74	Experimental testing - Adaptive BLFOSM - $k = 0.01$ - Constant reference: slope	80
3.75	Experimental testing - Adaptive BLFOSM - $k = 0.01$ - Constant reference: control input	80
3.76	Experimental testing - Traditional BLFOSM - $k = 0.5$ - Constant reference: angular velocity	82
3.77	Experimental testing - Traditional BLFOSM - $k = 0.5$ - Constant reference: quaternion	83
3.78	Experimental testing - Traditional BLFOSM - $k = 0.5$ - Constant reference: sliding variable	83

3.79	Experimental testing - Traditional BLFOSM - $k = 0.5$ - Constant reference: control input	84
3.80	Experimental testing - Adaptive BLFOSM - $k = 0.5$ - Constant reference: angular velocity	85
3.81	Experimental testing - Adaptive BLFOSM - $k = 0.5$ - Constant reference: quaternion	86
3.82	Experimental testing - Adaptive BLFOSM - $k = 0.5$ - Constant reference: sliding variable	86
3.83	Experimental testing - Adaptive BLFOSM - $k = 0.5$ - Constant reference: slope	87
3.84	Experimental testing - Adaptive BLFOSM - $k = 0.5$ - Constant reference: control input	87

Chapter 1

Introduction

Proper attitude control of a satellite is a crucial aspect for the success of any mission. As explained by Yost and Weston in the NASA technical report [1], the Guidance, Navigation and Control (GNC) subsystem comprises the elements utilized for position determination and those designated for the Attitude Determination and Control System (ADCS). The ADCS relies on sensors that measure attitude and spin rate, including star trackers, sun sensors, horizon sensors, magnetometers, and gyroscopes. This system is essential for vehicle control during trajectory correction maneuvers, employing accelerometers to conclude these maneuvers when the desired velocity change has been achieved. The ADCS includes actuators, such as magnetic torquers, reaction wheels, and thrusters to adjust the spacecraft orientation and apply necessary velocity changes.

The increasing complexity of systems has made it essential to optimize control techniques. As mentioned by Mancini in [2], sliding mode control (SMC) is a non-linear method of variable structure control (VSC), initially proposed by Utkin in his 1977 publication [3]. As reported by Lee and Utkin in [4], the sliding mode control approach is a robust method for designing controllers for complex high-order nonlinear dynamic systems under uncertainty, characterized by low sensitivity to parameter variations and disturbances. However, its implementation may be hindered by chattering, undesirable oscillations caused by unmodeled dynamics or discrete time effects, which can decrease control accuracy and increase wear on mechanical components. Lee and Utkin proposed various strategies for mitigating chattering. As discussed by Levant in [5] higher order sliding modes can completely eliminate chattering. They generalize the basic sliding mode concept by acting on the higher-order time derivatives of the system deviation from the constraint, unlike standard sliding modes, which only influence the first deviation derivative. Mancini and Capello [6] designed continuous twisting controller with adaptive gain, which is an higher order SMC, to suppress the flexibility of the satellite movable appendages. Capello et al. [7] combined simplex and super-twisting sliding mode

controller.

This work aims to analyze and compare different SMC algorithms to enhance the effectiveness of satellite attitude control for a spacecraft equipped with four control moment gyroscopes disposed in a pyramidal configuration. Higashiyama et al. [8] examined spacecraft attitude control using a pyramid-type variable-speed control moment gyro (VSCMG) system. Facchino et al. [9] presented an attitude controller for the CMG system based on model predictive control (MPC), which directly computes the gimbal rates to prevent singularity issues that may arise from the steering law. Nair et al. [10] introduced a robust sliding mode attitude controller based on quaternions for a spacecraft equipped with single gimbal control moment gyros (SGCMGs), designed for rest-to-rest maneuvers considering uncertainties.

This research continues the work of D'Ortona and Sfasciamuro in [11] and [12], utilizing the adaptive boundary layer first-order sliding mode controller developed by Mancini [2]. The results obtained in [11] and [12] are obtained using super twisting sliding mode control and adaptive continuous twisting control, both of which are second-order control structures. In particular, the adaptive continuous twisting control allows for the adjustment of control parameters in real time. In contrast, the goal of this work is to evaluate the performance of traditional versus adaptive boundary layer first-order sliding mode control, emphasizing the improvements and effectiveness of the adaptive approach. As mentioned by Mancini in [2], the concept of boundary-layer sliding mode control aims to reduce chattering by smoothing the discontinuity of the control law within the boundary layer, a small region adjacent to the sliding surface [13].

This project presents numerical simulation results obtained through Matlab and experimental results from testing on a testbed that simulates the satellite.

The structure of the thesis is as follows: Chapter 2 offers an overview of the mathematical tools used to derive the equations implemented in the codes and describes the sliding mode approach; Chapter 3 presents the numerical and experimental results; Chapter 4 summarizes the results and discusses their implications.

Chapter 2

Mathematical models and control strategies

2.1 Quaternions

As discussed by Yang [14], determining the attitude of a spacecraft involves calculating the distance between its body frame and the desired reference frame, which can be expressed as a single rotation using quaternions.

Quaternions are an extension of complex numbers to the three-dimensional space. Formally, a quaternion q can be defined as in equation 2.1

$$q = q_1i + q_2j + q_3k + q_4 \quad (2.1)$$

where $q_1, q_2, q_3,$ and q_4 are real numbers and $i, j,$ and k are imaginary units.

A unit quaternion is a quaternion with norm equal to one, particularly useful to handle rotations in 3D space.

The imaginary units $i, j,$ and k are independent from each other and satisfy the following properties:

$$i^2 = j^2 = k^2 = -1 \quad (2.2)$$

$$\begin{aligned} ij &= k \\ jk &= i \\ ki &= j \end{aligned} \quad (2.3)$$

Let us define the quaternion p as $p = p_1i + p_2j + p_3k + p_4$, which can be written in

matrix form as in equation 2.4

$$p = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} \quad (2.4)$$

where p_1 , p_2 , and p_3 constitute the vector part and p_4 represents the scalar part.

Then, let us recall the quaternion $q = q_1i + q_2j + q_3k + q_4$, which in matrix form becomes as in 2.5.

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad (2.5)$$

The multiplication of the two quaternions can be represented using different notations, one of which is shown in equation 2.6.

$$pq = (p_1q_4 + p_2q_3 - p_3q_2 + p_4q_1) + (-p_1q_3 + p_2q_4 + p_3q_1 + p_4q_2)j + (p_1q_2 - p_2q_1 + p_3q_4 + p_4q_3)k + p_1q_1 - p_2q_2 - p_3q_3 + p_4q_4 \quad (2.6)$$

Quaternion multiplication can be expressed in compact matrix notation as in equation 2.7

$$pq = r = \begin{bmatrix} r_v \\ r_4 \end{bmatrix} = \begin{bmatrix} p_4q_v + q_4p_v + p_v \times q_v \\ p_4q_4 - p_v^T q_v \end{bmatrix} \quad (2.7)$$

and in extended matrix representation as in 2.8

$$pq = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} = \begin{bmatrix} q_4 & q_3 & -q_2 & q_1 \\ -q_3 & q_4 & q_1 & q_2 \\ q_2 & -q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} = \begin{bmatrix} p_4 & -p_3 & p_2 & p_1 \\ p_3 & p_4 & -p_1 & p_2 \\ -p_2 & p_1 & p_4 & p_3 \\ -p_1 & -p_2 & -p_3 & p_4 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad (2.8)$$

Let us introduce the conjugate quaternion q^\dagger , defined in equation 2.9.

$$q^\dagger = \begin{bmatrix} -q_v \\ \frac{q_1^2 + q_2^2 + q_3^2 + q_4^2}{q_4} \end{bmatrix} = \begin{bmatrix} -q_v \\ \frac{\|q\|^2}{q_4} \end{bmatrix} \quad (2.9)$$

The property of conjugate quaternions is shown in equation 2.10.

$$q^\dagger q = qq^\dagger = 1 \quad (2.10)$$

The demonstration of this property is found in 2.11.

$$\begin{aligned}
 qq^\dagger &= \begin{bmatrix} q_4 q_v^\dagger + q_4^\dagger q_v + q_v \times q_v^\dagger & \\ & q_4 q_4^\dagger - q_v^T q_v^\dagger \end{bmatrix} = \begin{bmatrix} q_4 \left(-\frac{q_v}{\|q\|^2} \right) + \frac{q_4}{\|q\|^2} q_v + q_v \times \left(-\frac{q_v}{\|q\|^2} \right) & \\ & q_4 \frac{q_4}{\|q\|^2} - q_v^T \left(-\frac{q_v}{\|q\|^2} \right) \end{bmatrix} = \\
 &= \begin{bmatrix} -\frac{(q_v \times q_v)}{\|q\|^2} & \\ \frac{q_4^2 + q_v^T q_v}{\|q\|^2} & \end{bmatrix} = \begin{bmatrix} 0 & \\ \frac{\|q\|^2}{\|q\|^2} & \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}
 \end{aligned} \tag{2.11}$$

Let us define some quaternion properties. p , q , and r are quaternions and a , b are real numbers.

- $pq \neq qp$
- $(pq)r = p(qr)$
- $(ap)q = a(pq)$
- $(ap + bq)r = a(pr) + b(qr)$
- $p(aq + br) = a(pq) + b(pr)$

2.2 Dynamics equation

Let us introduce the inertial frame C_I and the body frame C_B .

For any attitude in three-dimensional space, there exist a rotational axis $\hat{\alpha}$ and a rotational angle ϕ , such that C_I coincides with C_B by rotating C_I by ϕ around $\hat{\alpha}$. $\hat{\alpha} \in \mathbb{R}^3$ is a unit quaternion, $\|\hat{\alpha}\|^2 = 1$, and $\phi \in \mathbb{R}$.

Euler parameters can be written as in equation 2.12

$$\begin{aligned}
 q_v &= \hat{\alpha} \sin \left(\frac{\phi}{2} \right) \\
 q_4 &= \cos \left(\frac{\phi}{2} \right)
 \end{aligned} \tag{2.12}$$

where $q_v \in \mathbb{R}^3$ and $q_4 \in \mathbb{R}$.

The mathematical demonstration proving that Euler parameters are always a unit quaternion, thereby ensuring that the norm equals 1, is presented in equation 2.13.

$$\begin{aligned}
 \|q\|^2 &= [q_v^T \ q_4] \begin{bmatrix} q_v \\ q_4 \end{bmatrix} = q_v^T q_v + q_4^2 = \sin\left(\frac{\phi}{2}\right) \hat{\alpha}^T \sin\left(\frac{\phi}{2}\right) \hat{\alpha} + \cos^2\left(\frac{\phi}{2}\right) = \\
 &= \sin^2\left(\frac{\phi}{2}\right) \hat{\alpha}^T \hat{\alpha} + \cos^2\left(\frac{\phi}{2}\right) = \sin^2\left(\frac{\phi}{2}\right) \|\hat{\alpha}\|^2 + \cos^2\left(\frac{\phi}{2}\right) = \\
 &= \sin^2\left(\frac{\phi}{2}\right) + \cos^2\left(\frac{\phi}{2}\right) = 1
 \end{aligned} \tag{2.13}$$

C_{IB} is the coordinate transformation from C_B to C_I and is given as in equation 2.14

$$C_{IB} = \cos \phi I_3 + (1 - \cos \phi) \hat{\alpha} \hat{\alpha}^T + \sin \phi \hat{\alpha}^x \tag{2.14}$$

where $I_3 \in \mathbb{R}^{3 \times 3}$ is the identity matrix.

The notation $()^x$ indicates the cross product operator, which transforms a vector into a matrix, as shown in equation 2.15, $()^x : \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 3}$.

$$a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \rightarrow a^x = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{2.15}$$

For any two vectors $a, b \in \mathbb{R}^3$, the relation $a \times b = a^x b$ holds.

Moreover, for any vector $a \in \mathbb{R}^3$, the transpose of the cross product operator satisfies $(a^x)^T = -a^x$.

If q is an Euler parameter, then for any vector $p \in \mathbb{R}^3$, the equation 2.16 holds

$$qpq^\dagger = \cos(\phi)p + (1 - \cos(\phi))(\hat{\alpha}^T p)\hat{\alpha} + \sin(\phi)\hat{\alpha}^x p \tag{2.16}$$

which can be written in compact form as in equation 2.17.

$$\begin{bmatrix} q_v \\ q_4 \end{bmatrix} \begin{bmatrix} p \\ p \end{bmatrix} \begin{bmatrix} q_v^\dagger \\ q_4^\dagger \end{bmatrix} = C_{IB} p \tag{2.17}$$

This equation is known as Rodrigues's formula, in which the Euler parameter q acts as a rotational operator in three-dimensional space.

Let us consider the vector $p \in \mathbb{R}^3$. C_I is the inertial frame and C_B the body frame.

Vector p may be represented in the inertial frame as ${}^I p$ or in the body frame as ${}^B p$.

$${}^I p = p|_{C_I} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} \quad (2.18)$$

$${}^B p = p|_{C_B} = \begin{bmatrix} p_{B1} \\ p_{B2} \\ p_{B3} \end{bmatrix} \quad (2.19)$$

By defining q_{IB} as the Euler parameter representation of C_B with respect to C_I , it is possible to write equations 2.20 and 2.21.

$${}^I p = q_{IB} {}^B p q_{IB}^\dagger \quad (2.20)$$

$${}^B p = q_{BI} {}^I p q_{BI}^\dagger \quad (2.21)$$

Equation 2.20 is used to transform the frame from C_B to C_I using the Euler parameter q_{IB} , while equation 2.21 is used to transform the frame from C_I to C_B using the Euler parameter q_{BI} .

It is possible to introduce equation 2.22.

$$q_{IB} = q_{BI}^\dagger \quad (2.22)$$

To demonstrate equation 2.22, let us consider the case where the rotational angle ϕ is zero, so we can write

$$q_{II} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (2.23)$$

and

$$q_{BB} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (2.24)$$

Hence, it is possible to write:

$$q_{II} = q_{IB} q_{BI} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (2.25)$$

Multiplying by q_{IB}^\dagger from the left, expression 2.26 is obtained.

$$q_{IB}^\dagger = q_{IB}^\dagger q_{IB} q_{BI} \quad (2.26)$$

According to the rules of conjugate quaternions, $q_{IB}^\dagger q_{IB} = 1$, thus equation 2.27 is derived.

$$q_{IB}^\dagger = q_{BI} \quad (2.27)$$

In a similar manner, by multiplying from the right by q_{BI}^\dagger , the equation 2.28 is obtained.

$$q_{BI}^\dagger = q_{IB} \quad (2.28)$$

In the context of dynamics, the attitude of the body changes according to the applied external torque.

Let us introduce the following notation:

- ${}^I h$ is the angular momentum vector described in C_I frame
- ${}^I J$ is the inertial matrix described in C_I frame
- ${}^I \tau$ is the external torque vector described in C_I frame

The dynamics equation expressed in the inertial frame C_I is written in equation 2.29.

$$\frac{d{}^I h}{dt} = {}^I \tau \quad (2.29)$$

The angular momentum vector can be expressed as in equation 2.30

$${}^I h = {}^I J {}^I \omega \quad (2.30)$$

where ${}^I \omega$ is the angular velocity.

Since ${}^I J$ expressed in the inertial frame changes as the satellite attitude varies, it is more convenient to write the equation in the body frame, where the inertial matrix described in the body frame ${}^B J$ remains constant.

To formulate the dynamics equation in the body frame, it is necessary to apply the coordinate transformations shown in equations 2.31 and 2.32.

$${}^B \omega = C_{IB}^T {}^I \omega \quad (2.31)$$

$${}^B J = C_{IB} {}^I J C_{IB}^T \quad (2.32)$$

Note that $C_{IB}^{-1} = C_{IB}^T$ because C_{IB} is a rotational matrix, which is orthogonal.

In the body frame C_B , it is possible to write ${}^I\tau$ as in equation 2.33.

$$\begin{aligned} {}^I\tau &= \frac{d}{dt} ({}^I J {}^I\omega) = \frac{d}{dt} (C_{IB} {}^B J C_{IB}^T {}^I\omega) = \\ &= \dot{C}_{IB} {}^B J {}^B\omega + C_{IB} {}^B J \dot{C}_{IB}^T {}^I\omega + C_{IB} {}^B J C_{IB}^T {}^I\dot{\omega} \end{aligned} \quad (2.33)$$

In the body frame, ${}^B J$ is a constant matrix, therefore ${}^B \dot{J} = 0$.

Substituting the kinematics equation expressed in the rotational matrix form $\dot{C}_{IB} = C_{IB} {}^B\omega^x$ in equation 2.33, equation 2.34 is obtained.

$${}^I\tau = C_{IB} {}^B\omega^x {}^B J {}^B\omega + C_{IB} {}^B J {}^B\omega^{xT} C_{IB}^T {}^I\omega + C_{IB} {}^B J C_{IB}^T \frac{d}{dt} (C_{IB} {}^B\omega) \quad (2.34)$$

In equation 2.34, the term $C_{IB} {}^B J {}^B\omega^{xT} C_{IB}^T {}^I\omega = 0$, as proven in equation 2.35, considering that ${}^I\omega = C_{IB} {}^B\omega$

$${}^B\omega^{xT} C_{IB}^T {}^I\omega = {}^B\omega^{xT} C_{IB}^T C_{IB} {}^B\omega = {}^B\omega^{xT} I_3 {}^B\omega = {}^B\omega^{xT} {}^B\omega = -{}^B\omega^x {}^B\omega = 0 \quad (2.35)$$

where ${}^B\omega^{xT} = -{}^B\omega^x$ holds since ${}^B\omega^x$ is a skew symmetric matrix. Hence, equation 2.34 can be rewritten as equation 2.36.

$${}^I\tau = C_{IB} {}^B\omega^x {}^B J {}^B\omega + C_{IB} {}^B J C_{IB}^T \frac{d}{dt} (C_{IB} {}^B\omega) \quad (2.36)$$

The time derivative that appears in the last term of equation 2.36 can be manipulated as in equation 2.37.

$$\frac{d}{dt} (C_{IB} {}^B\omega) = \dot{C}_{IB} {}^B\omega + C_{IB} {}^B\dot{\omega} = C_{IB} {}^B\omega^x {}^B\omega + C_{IB} {}^B\dot{\omega} = C_{IB} {}^B\dot{\omega} \quad (2.37)$$

therefore, equation 2.36 can be rewritten as 2.38.

$${}^I\tau = C_{IB} {}^B\omega^x {}^B J {}^B\omega + C_{IB} {}^B J C_{IB}^T C_{IB} {}^B\dot{\omega} = C_{IB} {}^B\omega^x {}^B J {}^B\omega + C_{IB} {}^B J {}^B\dot{\omega} \quad (2.38)$$

Equation 2.38 presents the left hand side expressed in the inertial frame and the left hand side in body frame. Knowing that ${}^I\tau = C_{IB} {}^B\tau$, it is possible to write equation 2.39.

$$C_{IB} {}^B\tau = C_{IB} {}^B\omega^x {}^B J {}^B\omega + C_{IB} {}^B J {}^B\dot{\omega} \quad (2.39)$$

Multiplying by C_{IB}^T from the left, the dynamics equation described in the body frame is obtained, as in equation 2.40

$${}^B\tau = {}^B\omega^x {}^B J {}^B\omega + {}^B J {}^B\dot{\omega} \quad (2.40)$$

where the inertial matrix described in the inertial frame ${}^B J$ is constant.

2.3 Kinematics equation

In the context of kinematics, the attitude of a body changes according to its angular velocity.

Let us consider the inertial fixed frame C_I and the body frame C_B . Assuming that Δt is a sufficiently small time interval during which a rotation of a sufficiently small angle $\Delta\psi$ occurs around the rotational axis ${}^B\hat{\alpha}$, the attitude of the satellite slightly changes from C_B to C'_B during the time interval Δt .

It is possible to introduce the Euler parameter $q_{BB'}$, which can be expressed in quaternion form, as in equation 2.41, or in vector form, as in equation 2.42.

$$q_{BB'} = {}^B\hat{\alpha} \sin\left(\frac{\Delta\psi}{2}\right) + \cos\left(\frac{\Delta\psi}{2}\right) \quad (2.41)$$

$$q_{BB'} = \begin{bmatrix} {}^B\hat{\alpha} \sin\left(\frac{\Delta\psi}{2}\right) \\ \cos\left(\frac{\Delta\psi}{2}\right) \end{bmatrix} \quad (2.42)$$

By using the "subscript chain rule", it is possible to write equation 2.43.

$$q_{IB'} = q_{IB}q_{BB'} = q_{IB} \left({}^B\hat{\alpha} \sin\left(\frac{\Delta\psi}{2}\right) + \cos\left(\frac{\Delta\psi}{2}\right) \right) \quad (2.43)$$

Let us now introduce the time derivative of the quaternion, \dot{q}_{IB} , as shown in equation 2.44.

$$\dot{q}_{IB} = \lim_{\Delta t \rightarrow 0} \frac{q_{IB'} - q_{IB}}{\Delta t} = \lim_{\Delta t \rightarrow 0} q_{IB} \left(\frac{{}^B\hat{\alpha} \sin\left(\frac{\Delta\psi}{2}\right) + \cos\left(\frac{\Delta\psi}{2}\right) - 1}{\Delta t} \right) \quad (2.44)$$

Since $|\Delta\psi|$ and Δt are sufficiently small, it is possible to utilize the approximations $\sin\left(\frac{\Delta\psi}{2}\right) \approx \frac{\Delta\psi}{2}$ and $\cos\left(\frac{\Delta\psi}{2}\right) \approx 1$.

therefore, substituting these approximations in equation 2.44, equation 2.45 is obtained

$$\begin{aligned} \dot{q}_{IB} &= \lim_{\Delta t \rightarrow 0} q_{IB} \left(\frac{{}^B\hat{\alpha} \left(\frac{\Delta\psi}{2}\right) + 1 - 1}{\Delta t} \right) = \lim_{\Delta t \rightarrow 0} \frac{q_{IB}}{2} {}^B\hat{\alpha} \frac{\Delta\psi}{\Delta t} = \\ &= \frac{1}{2} q_{IB} {}^B\hat{\alpha} \dot{\psi} \end{aligned} \quad (2.45)$$

where ${}^B\hat{\alpha}\dot{\psi} = {}^B\omega_{IB}$ is the angular velocity vector with respect to C_I described in the C_B frame, $\dot{\psi}$ is the amplitude, and ${}^B\hat{\alpha}$ is the unit vector.

The kinematics equation with angular velocity described in the body frame is written in equation 2.46.

$$\dot{q}_{IB} = \frac{1}{2}q_{IB} {}^B\omega_{IB} \quad (2.46)$$

Introducing the notation $\bar{V}(q) = q_v$ and $s(q) = q_4$, where $\bar{V}(q)$ is the operator that extracts the vector part of the quaternion q and $s(q)$ is the operator that extracts the scalar part of the quaternion q , it is possible to obtain the kinematics equation in vector and matrix form, as in equation 2.47, by explicitly calculating the quaternion multiplication

$$\dot{q}_{IB} = \begin{bmatrix} \bar{V}(\dot{q}_{IB}) \\ s(\dot{q}_{IB}) \end{bmatrix} = \frac{1}{2} \left[s(q_{IB})I_3 + \bar{V}(q_{IB})^x - \bar{V}(q_{IB})^T \right] {}^B\omega_{IB} \quad (2.47)$$

where $I_3 \in \mathbb{R}^{3 \times 3}$ is the identity matrix, ${}^B\omega_{IB} \in \mathbb{R}^{3 \times 1}$, $\bar{V}(q_{IB})^x \in \mathbb{R}^{3 \times 3}$ and $-\bar{V}(q_{IB})^T \in \mathbb{R}^{1 \times 3}$.

Equation 2.48 presents the kinematics equation expressed in element-wise form.

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \begin{bmatrix} {}^B\omega_1 \\ {}^B\omega_2 \\ {}^B\omega_3 \end{bmatrix} \quad (2.48)$$

Another matrix form used to express the kinematics equation is provided in equation 2.49.

$$\dot{q}_{IB} = \frac{1}{2} \begin{bmatrix} -{}^B\omega_{IB}^x & {}^B\omega_{IB} \\ -{}^B\omega_{IB}^T & 0 \end{bmatrix} q_{IB} \quad (2.49)$$

2.4 State equation

The motion in three dimensional space can be described combining the dynamics and kinematics equations.

The state variable is a column vector containing the quaternion and the angular velocity vector, as shown in 2.50.

$$x = \begin{bmatrix} q_{IB} \\ \omega_{IB} \end{bmatrix} \quad (2.50)$$

Then, it is necessary to calculate the state equation 2.51.

$$\dot{x} = \begin{bmatrix} \dot{q}_{IB} \\ \dot{\omega}_{IB} \end{bmatrix} \quad (2.51)$$

The dynamics and kinematics equation must be calculated simultaneously.

In this case, the state equation is nonlinear, as shown in 2.52.

$$\dot{x} = \begin{bmatrix} \dot{q}_{IB} \\ \dot{\omega}_{IB} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} q_{IB} {}^B \omega_{IB} \\ - {}^B J^{-1} {}^B \omega {}^B J {}^B \omega \end{bmatrix} + \begin{bmatrix} 0 \\ {}^B J^{-1} \end{bmatrix} {}^B \tau \quad (2.52)$$

where ${}^B \tau$ is the control input.

2.5 Error system

Let us define the body frame C_B , the fixed inertial frame C_I , and the virtual desired frame C_D . ${}^B \omega$ is the angular velocity vector of the actual spacecraft described in the body frame C_B , whereas ${}^D \omega_d$ is the desired angular velocity vector of the actual spacecraft described in the desired frame C_D . Let us also introduce the Euler parameters used to transform between frames: $q = q_{IB}$ is used to transform from the fixed inertial frame C_I to the body frame C_B , $q_d = q_{ID}$ is used to transform from the fixed inertial frame C_I to the virtual desired frame C_D , q_{BD} is used to transform from the body frame C_B to the virtual desired frame C_D , and q_{DB} is used to transform from the virtual desired frame C_D to the body frame C_B .

Now, let us define the error angular velocity ω_e and the error quaternion, as in equations 2.53 and 2.54, respectively.

$$\omega_e \triangleq {}^B \omega - {}^B \omega_d \quad (2.53)$$

$$q_e \triangleq q_{DB} \quad (2.54)$$

One simple case of feedback controller, which is negative, can be found in equation 2.55.

$$u = -k_1 \omega_e - k_2 \bar{v}_e \quad (2.55)$$

The error velocity vector q_e can be manipulated as in equation 2.56.

$$q_e = q_{DB} = q_{DI} q_{IB} = q_{IB}^\dagger q_{IB} = q_d^\dagger q \quad (2.56)$$

The desired angular velocity represented in the body frame can be expressed as in equation 2.57.

$${}^B \omega_d = q_{BD} {}^D \omega_d q_{BD}^\dagger = q_e^\dagger {}^D \omega_d q_e \quad (2.57)$$

Substituting equation 2.57 in the definition of error angular velocity, equation 2.58 is obtained.

$$\omega_e = {}^B \omega - q_e^\dagger {}^D \omega_d q_e \quad (2.58)$$

In the original system the desired attitude is time varying, but in the error system the desired value is constant and hence it is easier to design the controller.

Error kinematics equation

Equation 2.59 shows the time derivative of the error quaternion.

$$\dot{q}_e = \dot{q}_d^\dagger q + q_d^\dagger \dot{q} \quad (2.59)$$

The desired attitude should be designed to follow the kinematic equation, which is in the form $\dot{q} = \frac{1}{2}q\omega$, as shown in equation 2.60.

$$\dot{q}_d = \frac{1}{2}q_d {}^D\omega_d \quad (2.60)$$

By substituting equation 2.60 into the expression for time derivative of the error quaternion 2.59, equation 2.61 is obtained, using the following properties: $(q_d^\dagger {}^D\omega_d)^\dagger = {}^D\omega_d^\dagger q_d^\dagger$ and ${}^D\omega_d^\dagger = -{}^D\omega_d$

$$\begin{aligned} \dot{q}_w &= \left(\frac{1}{2}q_d {}^D\omega_d\right)^\dagger q + q_d^\dagger \left(\frac{1}{2}q {}^B\omega\right) = \frac{1}{2}{}^D\omega_d^\dagger q_d^\dagger q + \frac{1}{2}q_d^\dagger q {}^B\omega = \\ &= -\frac{1}{2}{}^D\omega_d q_d^\dagger q + \frac{1}{2}q_d^\dagger q {}^B\omega \end{aligned} \quad (2.61)$$

where ${}^D\omega_d$ is described in the desired frame and ${}^B\omega$ in the body frame.

Using equation 2.57, multiplying by q_e from the left and by q_e^\dagger from the right, knowing that $q_e q_e^\dagger = 1$, equation 2.62 is obtained.

$${}^D\omega_d = q_e {}^B\omega_d q_e^\dagger \quad (2.62)$$

Substituting equation 2.62 in 2.61, using the definition of error quaternion $q_e = q_d^\dagger q$, and remembering that $q_e^\dagger q_e = 1$, it is possible to obtain the error kinematic equation 2.63.

$$\begin{aligned} \dot{q}_e &= -\frac{1}{2}q_e {}^B\omega_d q_e^\dagger q_d^\dagger q + \frac{1}{2}q_d^\dagger q {}^B\omega = -\frac{1}{2}q_e {}^B\omega_d q_e^\dagger q_e + \frac{1}{2}q_e {}^B\omega = \\ &= -\frac{1}{2}q_e {}^B\omega_d + \frac{1}{2}q_e {}^B\omega = \frac{1}{2}q_e ({}^B\omega - {}^B\omega_d) = \frac{1}{2}q_e \omega_e \end{aligned} \quad (2.63)$$

where $\omega_e = ({}^B\omega - {}^B\omega_d)$ is taken from the definition of the angular velocity error.

The error kinematic equation 2.64 follows the same structure as the kinematic equation 2.46.

$$\dot{q}_e = \frac{1}{2}q_e \omega_e \quad (2.64)$$

Error dynamics equation

The dynamics equation 2.40 can be rewritten as in equation 2.65, isolating ${}^B\dot{\omega}$ on the left side and considering that the control input u coincides with the external torque τ .

$${}^B\dot{\omega} = -{}^B J^{-1} {}^B\omega^x {}^B J {}^B\omega + {}^B J^{-1} u \quad (2.65)$$

From the definition of error angular velocity in equation 2.53, equation 2.66 is obtained.

$${}^B\omega = \omega_e + {}^B\omega_d \quad (2.66)$$

By substituting equation 2.66 in 2.65, the error dynamics equation is derived as in 2.67.

$$\dot{\omega}_e = -{}^B\dot{\omega}_d - {}^B J^{-1}(\omega_e + {}^B\omega_d)^x {}^B J(\omega_e + {}^B\omega_d) + {}^B J^{-1}u \quad (2.67)$$

The state of the control system is defined as in equation 2.68

$$x = \begin{bmatrix} q_e \\ \omega_e \end{bmatrix} \quad (2.68)$$

and its time derivative, \dot{x} , is the error system, shown in equation 2.69.

$$\dot{x} = \begin{bmatrix} \dot{q}_e \\ \dot{\omega}_e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.69)$$

2.6 Plant system

Based on the error kinematic equation presented in 2.64, the corresponding matrix representation is provided in 2.70.

$$\begin{aligned} \dot{q}_e &= \frac{1}{2}q_e\omega_e = \frac{1}{2} \begin{bmatrix} q_{e4}\omega_{ev} + \omega_{e4}q_{ev} + q_{ev} \times \omega_{ev} \\ q_{e4}\omega_{e4} - q_{ev}^T\omega_{ev} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_{e4}\omega_{ev} + q_{ev} \times \omega_{ev} \\ -q_{ev}^T\omega_{ev} \end{bmatrix} = \\ &= \frac{1}{2} \begin{bmatrix} q_{e4}\omega_e + q_{ev} \times \omega_e \\ -q_{ev}^T\omega_e \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_{e4}\omega_e + q_{ev}^x\omega_e \\ -q_{ev}^T\omega_e \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -\omega_e^x q_{ev} + \omega_e q_{e4} \\ -\omega_e^T q_{ev} \end{bmatrix} = \\ &= \frac{1}{2} \begin{bmatrix} -\omega_e^x & \omega_e \\ -\omega_e^T & 0 \end{bmatrix} \begin{bmatrix} q_{ev} \\ q_{e4} \end{bmatrix} \end{aligned} \quad (2.70)$$

By multiplying the error dynamics equation 2.67 by J from the left, 2.71 is obtained.

$$J\dot{\omega}_e = -(\omega_e + \omega_d)^x J(\omega_e + \omega_d) - J\dot{\omega}_d + u \quad (2.71)$$

The objective is to track the desired attitude trajectory q_d and the desired angular velocity ω_d .

Let us introduce the PD (proportional derivative) controller u , as expressed in 2.72

$$u = -k_p q_{ev} - k_d \omega_e \quad (2.72)$$

where $k_p, k_d \in \mathbb{R}^{3 \times 3}$ are positive defined matrices.

The Lyapunov function candidate \bar{V} is defined in equation 2.73.

$$\bar{V} = \frac{a_1}{2} q_{ev}^T q_{ev} + \frac{a_2}{2} (1 - q_{e4})^2 + \frac{a_3}{2} \omega_e^T J \omega_e \quad (2.73)$$

The Lyapunov function candidate 2.73 is zero only if the state of the control error system is equal to zero, as in 2.74

$$\begin{bmatrix} q_e \\ \omega_e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.74)$$

otherwise $\bar{V} > 0$, since all the terms are squared and therefore cannot be negative.

Let us calculate the time derivative of the Lyapunov function, as in equation 2.75.

$$\begin{aligned} \dot{\bar{V}} &= a_1 q_{ev}^T \dot{q}_{ev} + a_2 (1 - q_{e4}) \dot{q}_{e4} + a_3 \omega_e^T J \dot{\omega}_e = \\ &= a_1 q_{ev}^T \left(-\frac{1}{2} \omega_e^x q_{ev} + \frac{1}{2} q_{e4} \omega_e \right) + a_2 (1 - q_{e4}) \frac{1}{2} \omega_e^T q_{ev} + \\ &\quad + a_3 \omega_e^T \left(-(\omega_e + \omega_d)^x J (\omega_e + \omega_d) - J \dot{\omega}_d + u \right) = \\ &= \frac{a_2}{2} q_{ev}^T \omega_e q_{ev} + \frac{a_1}{2} q_{e4} q_{ev}^T \omega_e + \frac{a_2}{2} \omega_e^T q_{ev} - \frac{a_2}{2} q_{e4} \omega_e^T q_{ev} + \\ &\quad - a_3 \omega_e^T (\omega_e^T + \omega_d)^x J (\omega_e + \omega_d) - a_3 \dot{\omega}_e^T J \omega_d + a_3 \omega_d^T u \end{aligned} \quad (2.75)$$

By setting $a_1 = a_2$, some terms cancel and equation 2.75 simplifies to the form in 2.76.

$$\dot{\bar{V}} = \frac{a_1}{2} \omega_e^T q_{ev} - a_3 \omega_e^T (\omega_e^T + \omega_d)^x J (\omega_e + \omega_d) - a_3 \dot{\omega}_e^T J \omega_d + a_3 \omega_d^T u \quad (2.76)$$

By choosing the controller $u = -\frac{a_1}{2a_3} q_{ev} + (\omega_e + \omega_d)^x J (\omega_e + \omega_d) + J \dot{\omega}_d - k_d \omega_e$, equation 2.76 transforms into the form shown in equation 2.77.

$$\dot{\bar{V}} = -\dot{\omega}_e^T k_d \omega_e \quad (2.77)$$

It is possible to prove that equation 2.77 is Lyapunov stable, since

$$\dot{\bar{V}} = -\omega_e^T k_d \omega_e \leq 0 \quad (2.78)$$

Expression 2.78 shows that the target point is Lyapunov stable, but not asymptotically stable.

$$\lim_{t \rightarrow \infty} \omega_e(t) = 0 \iff \lim_{t \rightarrow \infty} \omega(t) = \omega_d(t) \quad (2.79)$$

Now, let us prove the asymptotic stability. Substituting $\omega_e = 0$, 2.80 is obtained.

$$\lim_{t \rightarrow \infty} \dot{q}_e = 0 \quad (2.80)$$

so it converges to a constant.

In equation 2.81, the time derivative of the error angular velocity is calculated substituting the expression of u from 2.72.

$$\dot{\omega}_e(t) = -\frac{a_1}{2a_3} J^{-1} q_{ev} - J^{-1} k_d \omega_e \quad (2.81)$$

By taking the limit on both sides of equation 2.81, equation 2.82 can be expressed.

$$\lim_{t \rightarrow \infty} \dot{\omega}_e(t) = \lim_{t \rightarrow \infty} \left(-\frac{a_1}{2a_3} J^{-1} q_{ev} - J^{-1} k_d \omega_e \right) \quad (2.82)$$

Since $\bar{V} = \frac{1}{2} \omega_e^T J \omega_e$ and $\dot{\bar{V}} \leq 0$, ω_e is bounded.

Since $\lim_{t \rightarrow \infty} \omega_e = 0$, also $\lim_{t \rightarrow \infty} \dot{\omega}_e = 0$. Therefore 2.82 is reformulated as in equation 2.83.

$$0 = \lim_{t \rightarrow \infty} \left(-\frac{a_1}{2a_3} J^{-1} q_{ev} \right) = -\frac{a_1}{2a_3} J^{-1} \lim_{t \rightarrow \infty} q_{ev} \quad (2.83)$$

By multiplying both sides by J , 2.84 is obtained.

$$\lim_{t \rightarrow \infty} q_{ev} = 0 \quad (2.84)$$

Since q_e is a unit quaternion $\|q_e\|^2 = 1$, then

$$\lim_{t \rightarrow \infty} q_{e4}^2 = 1 \quad (2.85)$$

$$\lim_{t \rightarrow \infty} \begin{bmatrix} q_{ev} \\ \omega_{ev} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \pm 1 \\ 0 \\ 0 \end{bmatrix} \quad (2.86)$$

q and $-q$ represent the same attitude.

2.7 Actuators

The sensors on board the spacecraft measure its attitude. In an active attitude control process the difference between the attitude and the desired attitude, called error signal, is calculated and taken as input by the controller, which generates the command to impart to the actuators to produce an adequate torque to correct the attitude and make it coincide with the desired value.

As detailed in [15], there are various types of actuators, which can be classified into two categories based on the torques they generate. Thrusters and magnetic torquers generate torques external to the spacecraft that modify the spacecraft angular momentum and are classified as reaction-type actuators. Reaction wheels, momentum wheels and control moment gyroscopes generate torques internal to the spacecraft that do not modify the overall momentum of the spacecraft and are classified as momentum exchange actuators.

2.7.1 Control moment gyroscopes

A reaction wheel consists of a brushless motor attached to high-inertia flywheel free to spin along a fixed axis. The applications of reaction wheels are based on the principle of conservation of the angular momentum of the spacecraft. In fact, considering no external torques acting on the spacecraft, the variation of the angular momentum of the wheel, obtained by accelerating it in one direction about its spin-axis, causes an equal and opposite variation in the angular momentum of the platform of the spacecraft. Reaction wheels are widely employed in applications that require high pointing accuracy.

Momentum wheels are reaction wheels with constant or near-constant operating speed. The main drawback of wheels is wheel saturation, which occurs when the wheel reaches its maximum angular velocity and needs to be desaturated, removing momentum from it through the application of mass expulsion devices.

Control moment gyroscopes are momentum wheels that spin at constant high speed, with the spin axis gimballed. Tilting the wheel spin axis by applying a torque to the gimbal results in a gyroscopic reaction torque perpendicular to the gimbal axis. The main advantages of control moment gyroscopes are the rapid slew capability, the effectiveness and efficiency provided by the ability to generate high torque, and the high accuracy determined by the generation of continuous torque due to the change of the CMG angular momentum. Control moment gyros may suffer from gimbal lock, which consists of the loss of one degree of freedom. It occurs when the axes of two of the three gimbals are driven in a parallel configuration,

locking the system into rotation in a degenerate two-dimensional space.

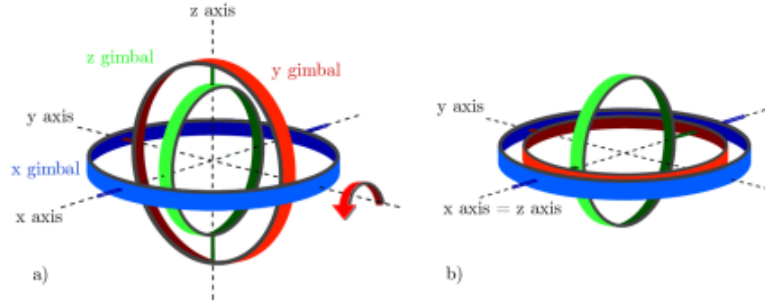


Figure 2.1: Gimbal lock [12]

To prevent the occurrence of gimbal lock, a singular direction avoidance steering law can be employed, as discussed in section 2.7.2. While a system of three CMGs is sufficient to ensure the attitude control of a spacecraft in three-dimensional space, four CMGs are typically used for redundancy. In this case, a system comprising four CMGs arranged in a pyramidal configuration is considered, as shown in Figure 2.2.

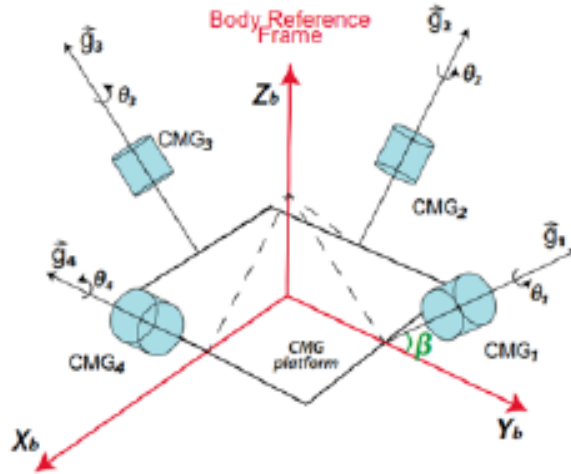


Figure 2.2: CMG system in pyramidal configuration [12]

In Figure 2.2, β is the skew angle, θ is the gimbal angle of each CMG, and g is the gimbal vector of each CMG. In this case, the skew angle β has the same value for all the four CMGs and is time-constant.

2.7.2 CMG system dynamics

Let us define the the dynamics equation of the spacecraft attitude control system as presented in equation 2.87

$${}^B\tau = {}^B\dot{h} + {}^B\omega \times {}^Bh \quad (2.87)$$

where prescript ${}^B()$ indicates that the used reference coordinate system is body fixed, ${}^B\tau$ is the external torque originated from external disturbances, and ${}^Bh \in \mathbb{R}^3$ is the angular momentum vector.

In general, the angular momentum vector can be expressed as in equation 2.88, and the external torque can be expressed by Euler's equation, as in 2.89.

$${}^Bh = {}^B J^B\omega \quad (2.88)$$

$${}^B\tau = {}^B J^B\dot{\omega} + {}^B\omega \times {}^B J^B\omega \quad (2.89)$$

In a GMC system, the angular momentum vector is given as in equation 2.90

$${}^Bh = {}^B J^B\omega + {}^B h_{CMG} \quad (2.90)$$

where the angular momentum is given by the sum of the contribution of the total CMGs system angular momentum vector h_{CMG} and the contribution of the spacecraft angular momentum except for the CMGs, ${}^B J^B\omega$.

The total CMGs system total angular momentum is given in equation 2.91

$${}^B h_{CMG} = {}^B h_w \sum_{i=1}^4 \bar{h}_i(\theta_i) \quad (2.91)$$

where $h_w = J_w\omega_w$ is the amplitude of the angular momentum, which is a constant scalar, and $\bar{h}_i(\theta_i)$ gives the direction of the i -th component of the sum ($i = 1, 2, 3, 4$ indicates the four CMGs).

In our study case, constant velocity CMGs are considered, so the time derivative of h_w is expressed as in 2.92.

$$\dot{h}_w = h_w \sum_{i=1}^4 \frac{\partial \bar{h}_i(\theta_i)}{\partial \theta_i} \dot{\theta}_i \quad (2.92)$$

It possible to write the time derivative of the total CMGs system angular momentum as in 2.93.

$${}^B \dot{h}_{CMG} = h_w \sum_{i=1}^4 \frac{\partial \bar{h}_i(\theta_i)}{\partial \theta_i} \dot{\theta}_i = h_w \left[\frac{\partial \bar{h}_1(\theta_1)}{\partial \theta_1}, \frac{\partial \bar{h}_2(\theta_2)}{\partial \theta_2}, \frac{\partial \bar{h}_3(\theta_3)}{\partial \theta_3}, \frac{\partial \bar{h}_4(\theta_4)}{\partial \theta_4} \right] = h_w A(\theta) \dot{\theta} \quad (2.93)$$

Substituting the ${}^B h$ for CMG system in the dynamics equation, equation 2.94 is obtained.

$${}^B \tau = {}^B J^B \dot{\omega} + {}^B \omega \times {}^B h + {}^B \dot{h}_{CMG} = {}^B \dot{\omega} + {}^B \omega \times {}^B J^B \omega + {}^B \omega \times {}^B h_{CMG} + {}^B \dot{h}_{CMG} \quad (2.94)$$

Now, let us define the control torque ${}^B u$, as in equation 2.95.

$${}^B u = -{}^B \dot{h}_{CMG} = -h_w A(\theta) \dot{\theta} \quad (2.95)$$

By substituting the expressions of ${}^B \tau$ and ${}^B u$ into the dynamics equation, the dynamics equation for the CMGs system is obtained as in 2.96.

$${}^B \tau + {}^B u = {}^B J^B \dot{\omega} + {}^B \omega \times {}^B h + {}^B \dot{h}_{CMG} = {}^B \dot{\omega} + {}^B \omega \times {}^B J^B \omega + {}^B \omega \times {}^B h_{CMG} \quad (2.96)$$

The steering law connects the control torque and the gimbal rate and is expressed as shown in equation 2.97.

$$\dot{\theta} = -\frac{1}{h_w} A(\theta) {}^B u \quad (2.97)$$

Two assumptions will be now introduced:

1. there is no external disturbance torque, which implies that ${}^B \tau = 0$
2. the total angular momentum is preserved at zero, which implies that ${}^B h_{CMG} + {}^B J^B \omega = 0$. Consequently, ${}^B h_{CMG} = -{}^B J^B \omega$

By applying these assumptions to the dynamics equation for the CMG system 2.96, equation 2.98 is obtained.

$${}^B u = {}^B J^B \dot{\omega} + {}^B \omega \times {}^B J^B \omega + {}^B \omega \times (-{}^B J^B \omega) = {}^B J^B \dot{\omega} \quad (2.98)$$

From equation 2.98, it is possible to obtain equation 2.99.

$${}^B \dot{\omega} = {}^B J^{-1} u = {}^B h_w J^{-1} A(\theta) \dot{\theta} \quad (2.99)$$

The process involves calculating the torque u using sliding mode control. Then, starting from the value of u , the gimbal rate $\dot{\theta}$ is determined using steering law. Subsequently, the gimbal rate $\dot{\theta}$ is applied to the CMGs, resulting in a change in the angular velocity of the satellite according to the dynamic law.

Now, let us calculate the time derivative of the sliding surface to obtain the equivalent control u_{eq} .

The sliding variable σ is defined in equation 2.100.

$$\sigma = \omega_e + \lambda q_{e,v} \quad (2.100)$$

The time derivative of the sliding variable is expressed as shown in equation 2.101.

$$\dot{\sigma} = \dot{\omega}_e + \lambda \dot{q}_{e,v} = \dot{\omega} - \dot{\omega}_d + \lambda \dot{q}_{e,v} = J^{-1}u_{eq} - \dot{\omega}_d + \lambda \dot{q}_{e,v} \quad (2.101)$$

By setting the time derivative of the sliding variable to zero, the equivalent control u_{eq} is obtained as in 2.102

$$u_{eq} = J\dot{\omega}_d - \lambda J\dot{q}_{e,v} \quad (2.102)$$

Steering law

As mentioned earlier, the steering law establishes a relationship between the control torque and the gimbal rate. The steering law derived earlier in equation 2.97 includes the matrix $A(\theta) \in \mathbb{R}^{3 \times 4}$, which is defined in equation 2.103

$$A(\theta) = \begin{bmatrix} -\cos \beta \cos \theta_1 & \sin \theta_2 & \cos \beta \cos \theta_3 & -\sin \theta_4 \\ -\sin \theta_1 & -\cos \beta \cos \theta_2 & \sin \theta_3 & \cos \beta \cos \theta_4 \\ \sin \beta \cos \theta_1 & \sin \beta \cos \theta_2 & \sin \beta \cos \theta_3 & \sin \beta \cos \theta_4 \end{bmatrix} \quad (2.103)$$

where β is the skew angle, and θ_i ($i = 1,2,3,4$) denotes the gimbal angle of the i -th control moment gyro.

Since the matrix $A(\theta)$ is not square, to simplify calculations, it can be substituted with its pseudo-inverse $A^T(AA^T)^{-1}$. Unfortunately, AA^T may become singular for certain values of the gimbal angles θ_i , leading to gimbal lock. To prevent this situation, a Singular Direction Avoidance (SDA) steering law can be employed. Equation 2.104 shows the SDA steering law utilized in this case

$$\dot{\theta} = -\frac{1}{h_\omega} A^T (AA^T + \lambda u_3 u_3)^B u \quad (2.104)$$

where λ is a parameter introduced to avoid singularities and u_3 is the eigenvector associated with the smallest eigenvalue of A .

2.8 Sliding mode control

2.8.1 Description

As described in [2], the sliding mode process consists of two main steps: the reaching phase and the sliding phase. During the reaching phase, the sliding variable, which is a function of the system output and its time derivative, is non-zero and the control input is continuous. During the sliding phase, instead, the sliding variable is null, and the control input is discontinuous.

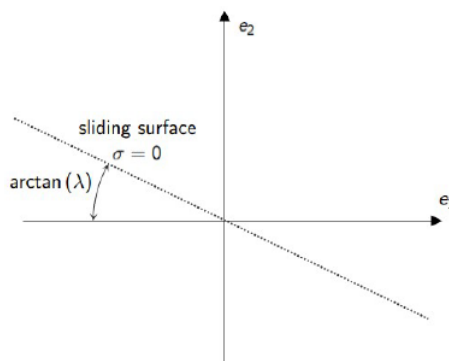


Figure 2.3: Sliding surface [2]

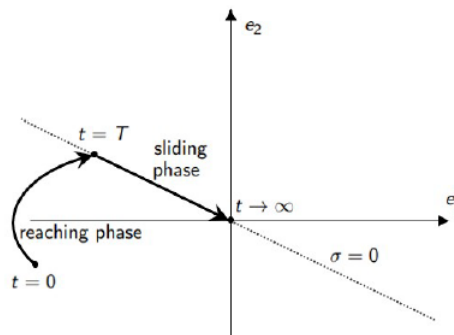


Figure 2.4: Reaching phase and sliding phase [2]

In Figure 2.4, it is possible to identify the initial state at time $t = 0$. At that moment, the reaching phase commences and continues until the sliding surface is attained. In Figure 2.3, the sliding surface is represented as a line passing through the origin of axes (e_1, e_2) , with a slope determined by the parameter lambda,

which conveys information regarding the rate of convergence. The sliding phase commences once the sliding surface, defined as the region of space where the sliding variable is zero, is attained. The sliding surface is designed to ensure the desired trajectory during the sliding phase, thereby causing the state errors to approach zero.

First order sliding mode (FOSM) control represents the initial generation of sliding mode control techniques. It involves defining the sliding variable as a function of the system output and its time derivative, along with implementing a discontinuous control law to guide the system trajectory. One of the primary drawbacks of the FOSM is chattering. Chattering consists of high frequency oscillations in the vicinity of the sliding surface, which can lead to reduced system efficiency and potential damage of the moving components.

Several methods have been developed to mitigate chattering, one of which is boundary layer sliding mode control. It involves smoothing the discontinuity of the control law within a small region adjacent to the sliding surface, the boundary layer, the thickness of which is determined by the parameter $\bar{\sigma}$, as shown in Figure 2.5.

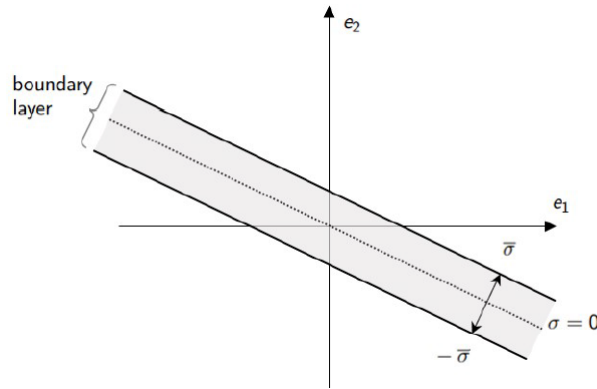


Figure 2.5: Boundary layer [2]

On one hand, the introduction of the boundary layer reduces the chattering phenomenon; on the other hand, it reduces tracking accuracy, since during the reaching phase the closed-loop system converges to the boundary layer adjacent to the sliding surface. Therefore, it is essential to strike a balance in the selection of $\bar{\sigma}$. On one side, selecting a value of $\bar{\sigma} = 0$ allows the tracking error to converge to zero for any value of lambda, but may result in chattering. On the other side, as the value of $\bar{\sigma}$ increases, the boundary layer widens, which mitigates chattering, but imposes constraints on the choice of λ , which must be sufficiently large to ensure tracking accuracy, namely $\lambda_{min} = \frac{\sigma}{E}$ (where E represents the maximum allowable deviation).

A more advanced type of controller involves the real-time adaptation of the sliding surface throughout the control process. Adaptive boundary layer first order sliding mode control is based on the conventional boundary layer first order sliding mode, allowing for the adaptation of the sliding surface. The mitigation of chattering is achieved through the implementation of the boundary layer approach. This algorithm is capable of adapting to a wide range of initial conditions while ensuring robustness, tracking accuracy and smooth response.

The real time adaptation of the parameter λ consists of two phases. In the first phase, the adaptive law rotates the sliding surface toward the initial condition, thereby reducing the reaching time and enhancing the robustness of the control. In the second phase, once the system has reached the sliding surface, the adaptive law adjusts λ to the value that ensures the desired behavior of the system near the equilibrium point, thereby improving the tracking accuracy.

2.8.2 Implementation

To implement the first order sliding mode control, it is essential to define the sliding variable. The sliding variable σ is defined in equation 2.105 as the sum of the error angular velocity and the product of the parameter λ and the vector component of the error quaternion

$$\sigma = \omega_e + \lambda q_{e,v} \quad (2.105)$$

where the error angular velocity is defined as the difference between the actual angular velocity and the desired angular velocity, namely $\omega_e = \omega - \omega_d$; the error quaternion is defined as the quaternion product of the conjugate desired quaternion and the actual quaternion, namely $q_e = q_d^\dagger q$; and the parameter λ defines the slope of the sliding surface.

Then, the control input u can be defined. The switching control is given in equation 2.106

$$u_{sw} = -{}^B J k \text{sign}(\sigma) \quad (2.106)$$

where ${}^B J$ is the inertial matrix, k is the control gain, the minus sign arises from the convention used to define the error angular velocity and the error quaternion, and the sign function is defined as in equation 2.107.

$$\text{sign}(\sigma_i) = \begin{cases} 1 & \text{if } \sigma_i > 0 \\ -1 & \text{if } \sigma_i < 0 \\ 0 & \text{if } \sigma_i = 0 \end{cases} \quad (2.107)$$

To improve the performance of the sliding mode controller, the equivalent control u_{eq} is included in the control law definition. The equivalent control u_{eq} is calculated

as in equation 2.102.

The control input for the first order sliding mode control is given by the sum the switching control and the equivalent control, as in equation 2.108.

$$u = u_{eq} + u_{sw} \quad (2.108)$$

To implement boundary layer first order sliding mode control, only the definition of the switching control is modified by substituting the sign function with the saturation function. This modification arises from the introduction of the boundary layer. The new expression for the switching control is presented in equation 2.109

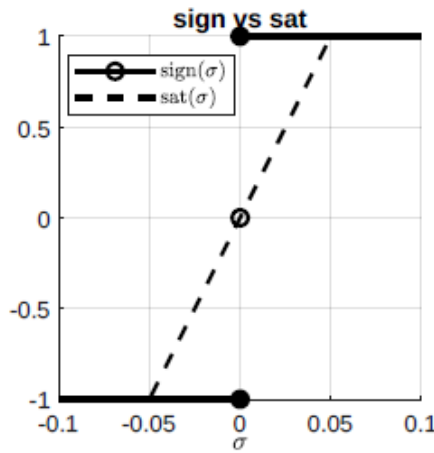


Figure 2.6: Sign function vs saturation function [2]

$$u_{sw} = -{}^B J k \text{sat}(\sigma) \quad (2.109)$$

where ${}^B J$ is the inertial matrix, k is the control gain, the minus sign is due to the convention utilized to define the error angular velocity and the error quaternion, and the saturation function is defined as in equation 2.110

$$\text{sat}(\sigma_i) = \begin{cases} \frac{\sigma_i}{\bar{\sigma}} & \text{if } |\sigma_i| \leq \bar{\sigma} \\ \text{sign}(\sigma_i) & \text{if } |\sigma_i| > \bar{\sigma} \end{cases} \quad (2.110)$$

where $\bar{\sigma} > 0$ is the thickness of the boundary layer.

As in the previous case, the control input is expressed as the sum of the switching control and the equivalent control, as indicated in equation 2.108. It is important to note that the switching control is defined differently from that in first order sliding mode control.

In implementing the adaptive boundary layer first order sliding mode control, it is necessary to recognize that the parameter λ is no longer a constant anymore and becomes an additional controller state with its own dynamics. By varying the value of the parameter λ , the sliding surface rotates in the phase plane. The equations employed to calculate the switching control u_{sw} and the equivalent control u_{eq} are the same as those used in the implementation of boundary layer first order sliding mode. To calculate the values of the parameter λ , it is necessary to time-integrate the values of the time derivative $\dot{\lambda}$, which are provided by the projection function $proj_{[\underline{\lambda}, \bar{\lambda}]}$. The projection function, shown in equation 2.111, is applied to constrain the parameter λ within the interval $[\underline{\lambda}, \bar{\lambda}]$.

$$\dot{\lambda} = proj_{[\underline{\lambda}, \bar{\lambda}]}(\lambda, h) = \begin{cases} \max\{0, h\} & \text{if } \lambda = \underline{\lambda} \\ h & \text{if } \underline{\lambda} < \lambda < \bar{\lambda} \\ \min\{0, h\} & \text{if } \lambda = \bar{\lambda} \end{cases} \quad (2.111)$$

The parameter h is given as in equation 2.112

$$h = G\zeta_{\bar{\sigma}}(\sigma)sign(q_{e,v}) - c(\lambda - \bar{\lambda}) \quad (2.112)$$

where c is a constant, G is a parameter which is given in equation 2.113, and $\zeta_{\bar{\sigma}}$ is a piecewise constant function parameterized by the value of the thickness of the boundary layer $\bar{\sigma}$, as described in equation 2.114.

$$G = \frac{c(\underline{\lambda} - \bar{\lambda})}{\bar{\sigma}} < 0 \quad (2.113)$$

$$\zeta_{\bar{\sigma}}(\sigma) = \begin{cases} \sigma & \text{if } |\sigma| \leq \bar{\sigma} \\ 0 & \text{if } |\sigma| < \bar{\sigma} \end{cases} \quad (2.114)$$

Chapter 3

Simulations results and experimental results

This chapter presents the results obtained through numerical simulations and experimental testing, and is divided into two sections. The first section focuses on comparing traditional and adaptive boundary layer first-order sliding mode control, simulated using Matlab. In the second section, experimental outcomes from testing conducted on the testbed in Satoh's laboratory at Osaka University are presented.

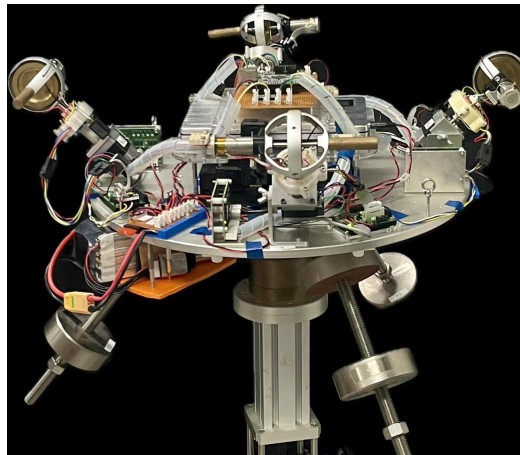


Figure 3.1: Testbed platform

The testbed allows the simulation of attitude maneuvers for a satellite equipped with a system of four CMGs arranged in pyramidal configuration. It consists of a metallic plate that houses the four CMGs. The testbed is designed to float and rotate without friction, achieved through the use of compressed air and a spherical bearing. The center of mass can be adjusted manually by moving some

counterweights located in the lower part of the testbed. Due to the physical characteristics of the device, attitude maneuvers can be performed exclusively about the z-axis.

Figure 3.1 illustrates the platform of testbed.

3.1 Numerical simulations

In the following section, the results obtained through Matlab simulation comparing traditional and adaptive boundary layer first order sliding mode control are presented.

The objective of this simulation is to demonstrate that the adaptive controller ensures faster convergence than the traditional approach. The initial conditions are generated using random unit quaternions, while the desired quaternion is set to $q_d = [0 \ 0 \ 0 \ 1]'$.

The simulation lasted for 300 seconds, with time steps of 25 milliseconds. The integrations were performed using the Euler method.

The following figures display the results obtained using twenty different initial quaternions.

Tables 3.1 and 3.2 summarize the parameters used in the simulation for traditional and adaptive approach, respectively.

λ	$\frac{5}{\sqrt{3}}$
k	0.01
$\bar{\sigma}$	0.005
q_d	$[0 \ 0 \ 0 \ 1]'$
q_0	random unit quaternion

Table 3.1: Traditional BLFOSM parameters

λ_{in}	$\frac{5}{\sqrt{3}}$
k	0.01
$\bar{\sigma}$	0.005
c	0.01
q_d	$[0 \ 0 \ 0 \ 1]'$
q_0	random unit quaternion

Table 3.2: Adaptive BLFOSM parameters

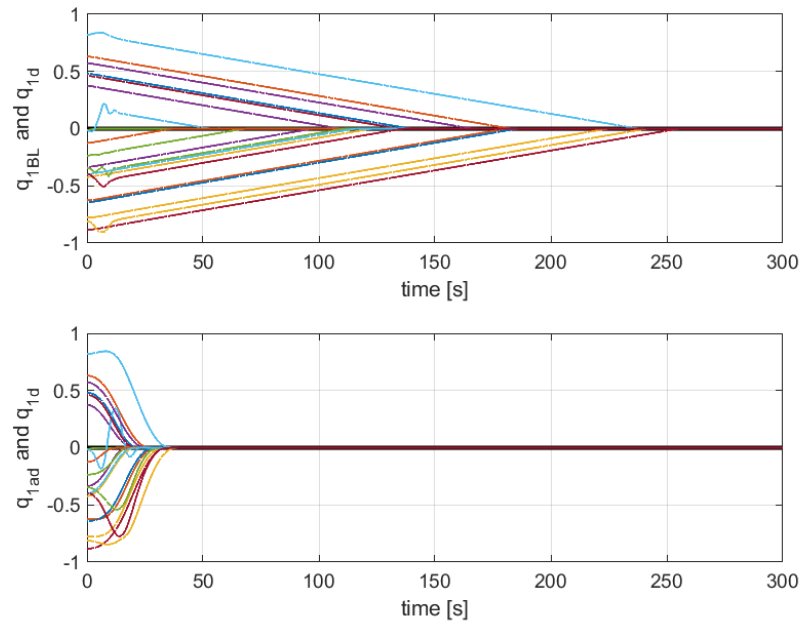


Figure 3.2: Comparison: quaternion q_1

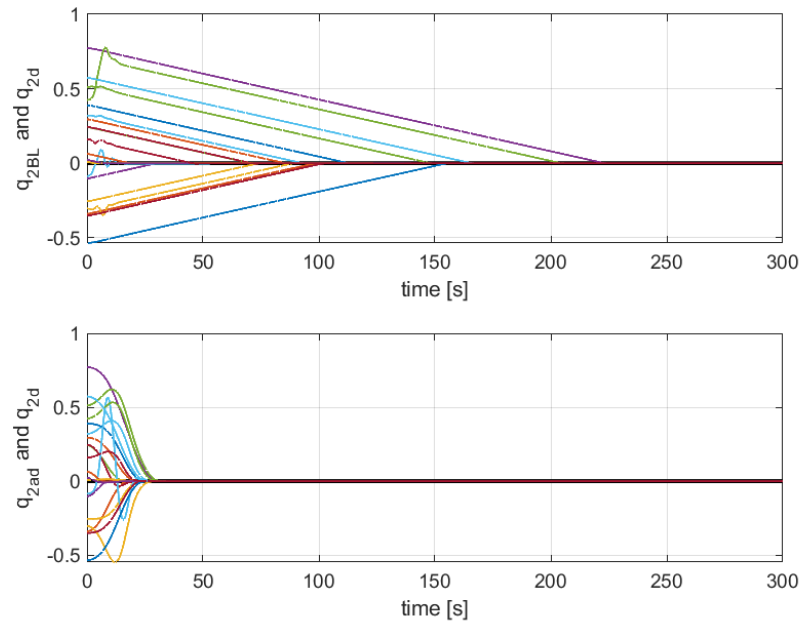


Figure 3.3: Comparison: quaternion q_2

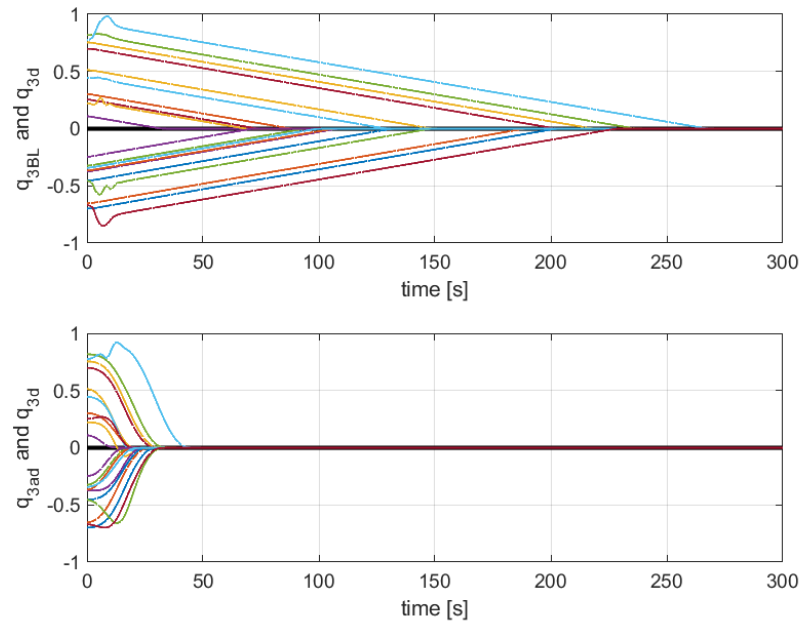


Figure 3.4: Comparison: quaternion q_3

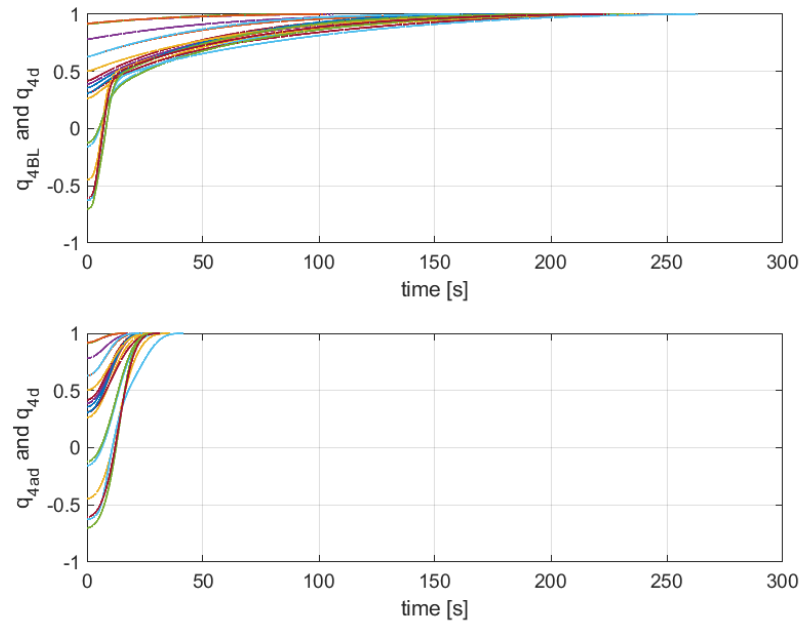


Figure 3.5: Comparison: quaternion q_4

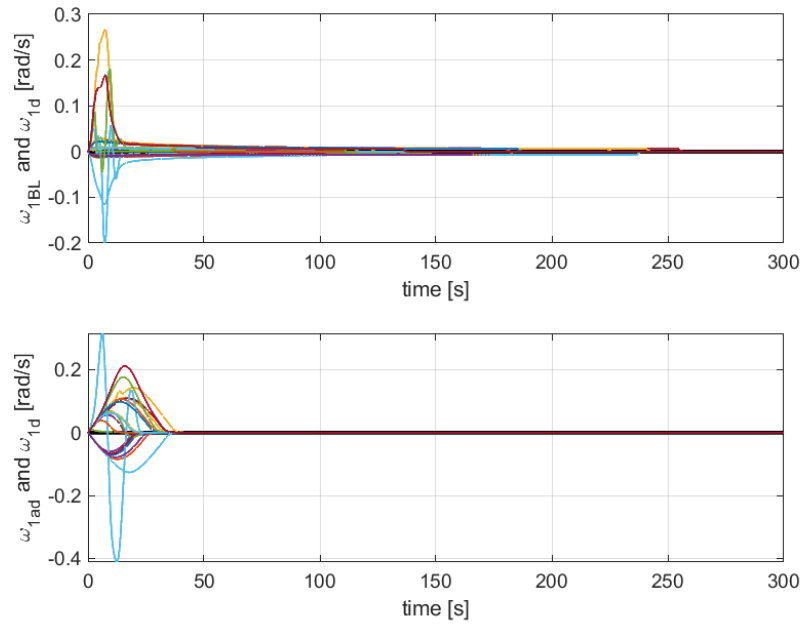


Figure 3.6: Comparison: angular velocity ω_1

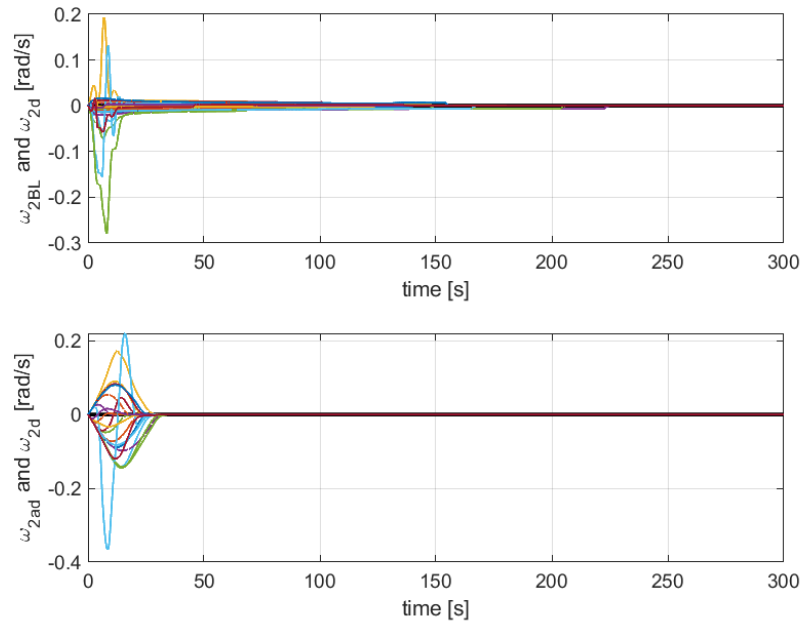


Figure 3.7: Comparison: angular velocity ω_2

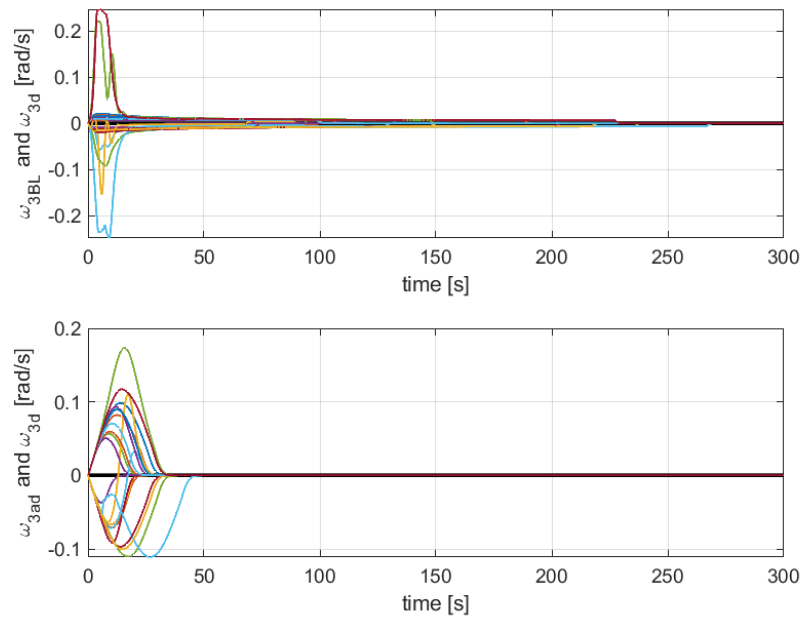


Figure 3.8: Comparison: angular velocity ω_3

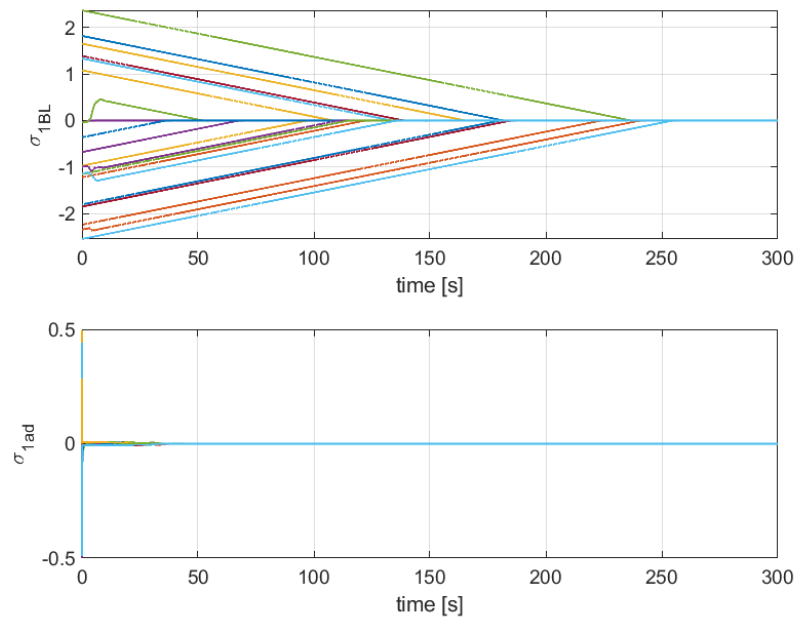


Figure 3.9: Comparison: sliding variable σ_1

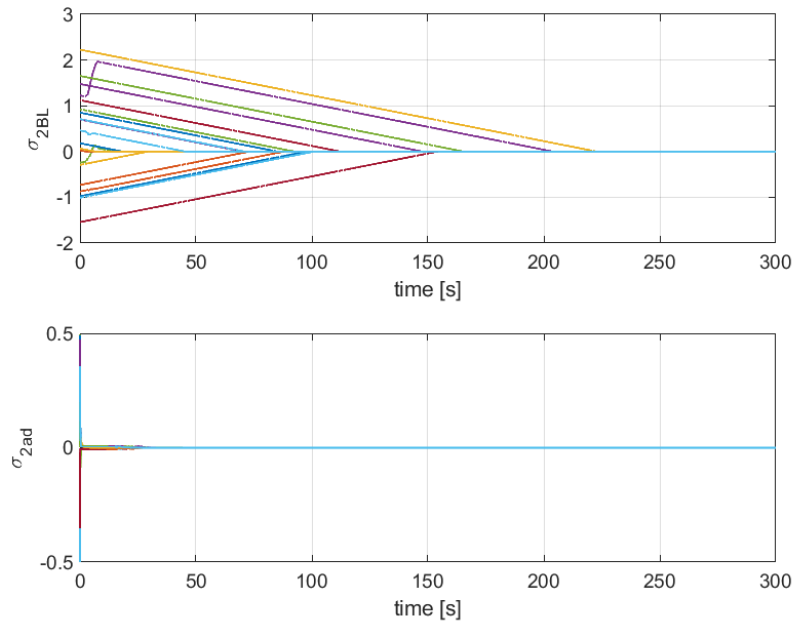


Figure 3.10: Comparison: sliding variable σ_2

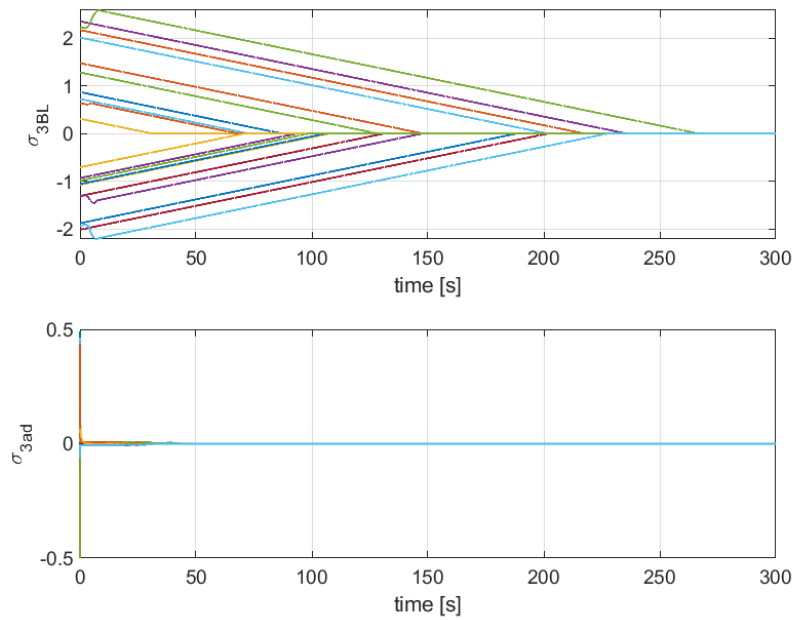


Figure 3.11: Comparison: sliding variable σ_3

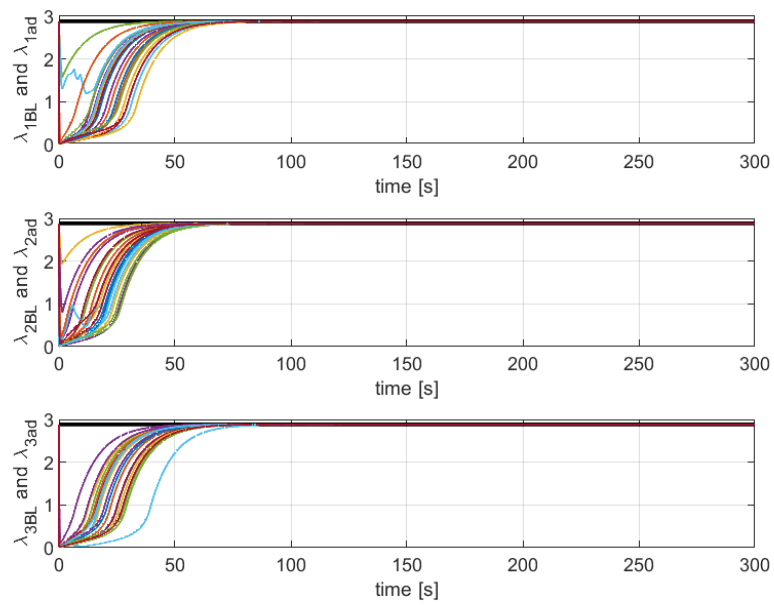


Figure 3.12: Comparison: slope λ

Figures 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 3.10, and 3.11 display the trend of the traditional boundary layer first-order sliding mode control in the upper graphs, while the lower graphs illustrate the trend of the adaptive approach.

In Figures 3.2, 3.3, 3.4, and 3.5, it is evident that the adaptive approach converges to the desired quaternion value significantly faster. Specifically, for all initial values, the adaptive BLFOSM takes less than 50 seconds to converge, while the traditional method takes over 250 seconds for some initial values. Even better results are noticeable in figures 3.9, 3.10, and 3.11, which illustrate the trend of the sliding surface. With the adaptive approach, the sliding variable converges to zero rapidly for any initial value, demonstrating that the adaptive approach is more effective in reducing the reaching time.

Figure 3.12 depicts the trend of the slope λ . In the traditional approach, the value is maintained constant, while in the adaptive approach, the value of lambda starts from the same initial value but varies, ultimately converging to the initial value.

3.2 Experimental results and comparison

In this section, both experimental and simulation results are presented. The simulations were conducted using Matlab, while the experimental results were obtained through testing on the testbed. The Matlab code was translated into C++ to control the testbed.

Simulations and experiments were conducted under two different conditions. The first set of results was obtained considering a triangular angular velocity reference, where the maximum value of angular velocity is achieved linearly over 10 seconds followed by a linear decrease to zero of the same duration. After that, the value is maintained zero. The second set of results was obtained by setting the desired quaternion to a constant value, with the desired angular velocity set to zero. The desired quaternion was set to $[0 \ 0 \ \cos \frac{\theta}{2} \ \sin \frac{\theta}{2}]'$, with $\theta = 60^\circ$, to ensure the testbed rotates solely about the z-axis to achieve the desired attitude.

The two sets of experimental tests and numerical simulations were conducted for two values of the control gain k to highlight the differing behaviors of the two approaches at varying control gain values.

The experimental simulations lasted for 30 seconds, with time steps of 25 milliseconds. The integrations were performed using the Euler method.

Tables 3.3 and 3.4 summarize the parameters used in the simulation for traditional and adaptive approach, respectively.

λ	$\frac{5}{\sqrt{3}}$
k	0.01 or 0.5
$\bar{\sigma}$	0.005
q_0	[0 0 0 1]'

Table 3.3: Traditional BLFOSM parameters for experimental results

λ_{in}	$\frac{5}{\sqrt{3}}$
k	0.01 or 0.5
$\bar{\sigma}$	0.005
c	0.1
q_0	[0 0 0 1]'

Table 3.4: Adaptive BLFOSM parameters for experimental results

3.2.1 Triangular angular velocity reference

The following subsection presents the results for the triangular angular velocity reference. Numerical results will be illustrated first, followed by the experimental results.

For each set of results, the trends of the following parameters will be shown: angular velocity, quaternion, sliding variable, control input, and, in the case of the adaptive approach, slope of the sliding variable.

Numerical - $k = 0.01$ - Traditional BLFOSM

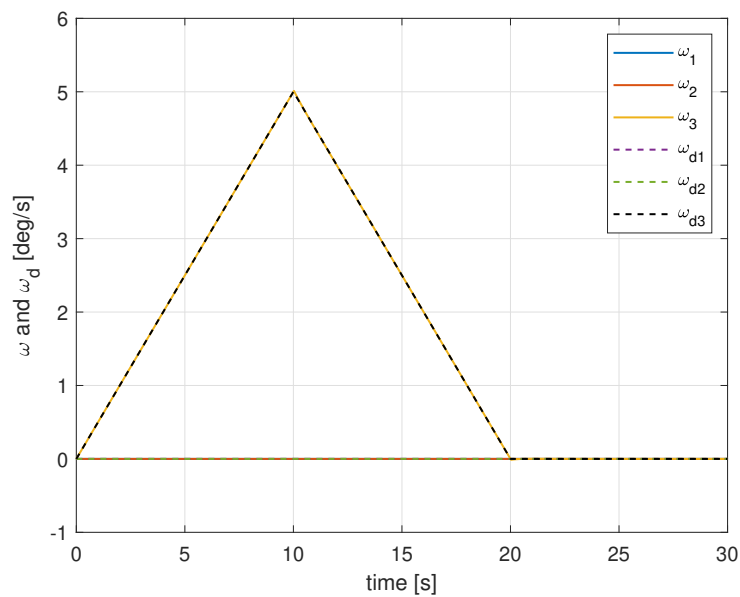


Figure 3.13: Numerical simulation - Traditional BLFOSM - $k = 0.01$ - Triangular angular velocity: angular velocity

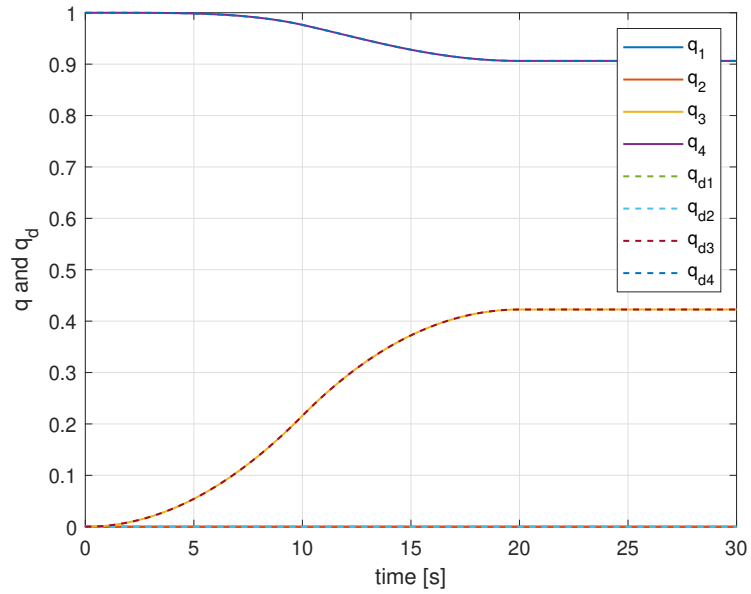


Figure 3.14: Numerical simulation - Traditional BLFOSM - $k = 0.01$ - Triangular angular velocity: quaternion

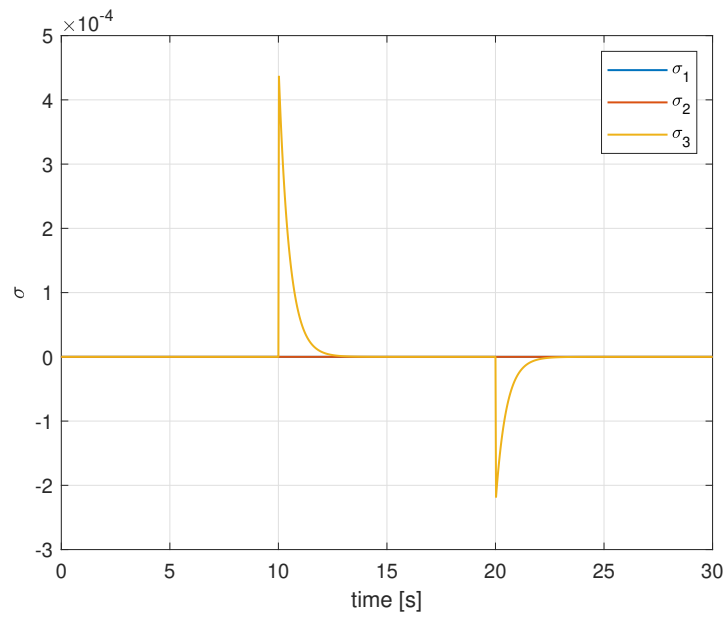


Figure 3.15: Numerical simulation - Traditional BLFOSM - $k = 0.01$ - Triangular angular velocity: sliding variable

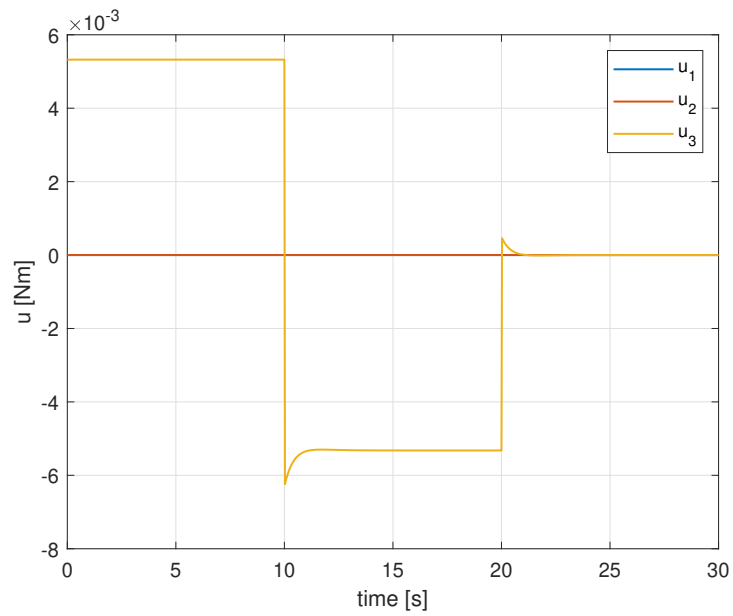


Figure 3.16: Numerical simulation - Traditional BLFOSM - $k = 0.01$ - Triangular angular velocity: control input

Figures 3.13 and 3.14 illustrate that the numerical simulation of the traditional approach with a control gain value of 0.01 successfully follows the desired angular velocity and quaternion. Figure 3.15 shows that the control law makes the sliding variable converge. Figure 3.16 demonstrates the absence of chattering in the control input.

Numerical - $k = 0.01$ - Adaptive BLFOSM

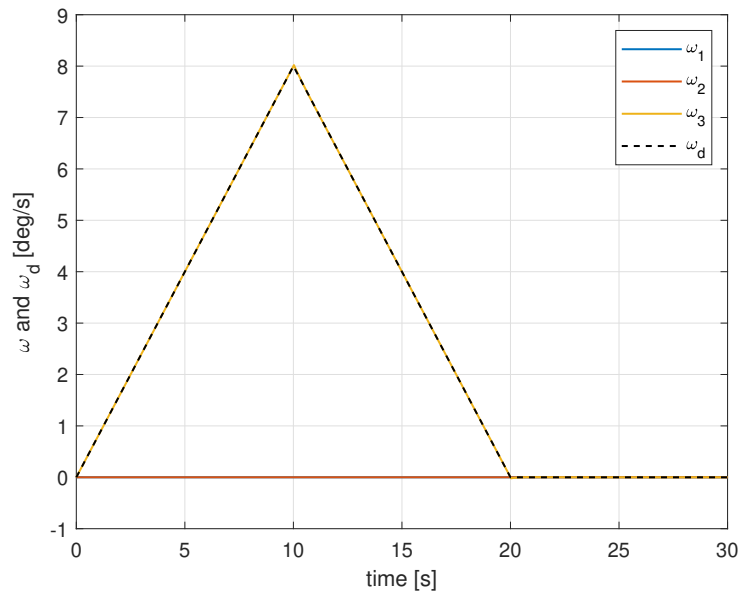


Figure 3.17: Numerical simulation - Adaptive BLFOSM - $k = 0.01$ - Triangular angular velocity: angular velocity

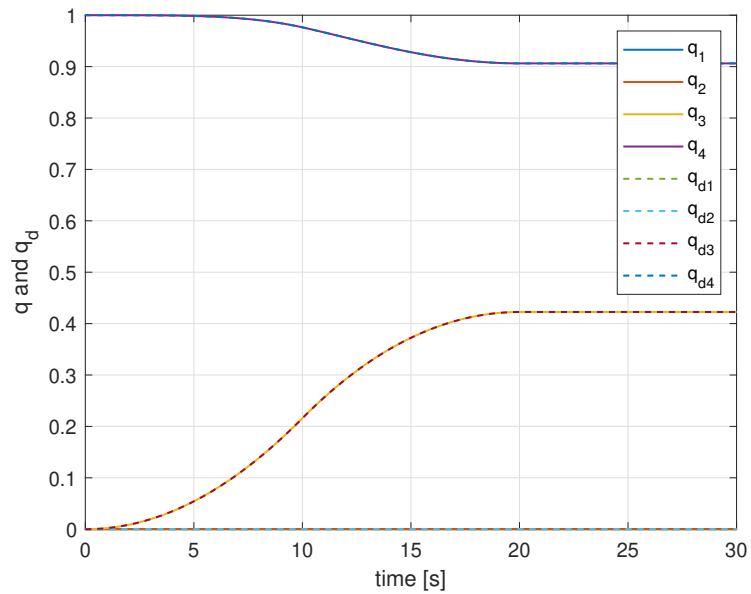


Figure 3.18: Numerical simulation - Adaptive BLFOSM - $k = 0.01$ - Triangular angular velocity: quaternion

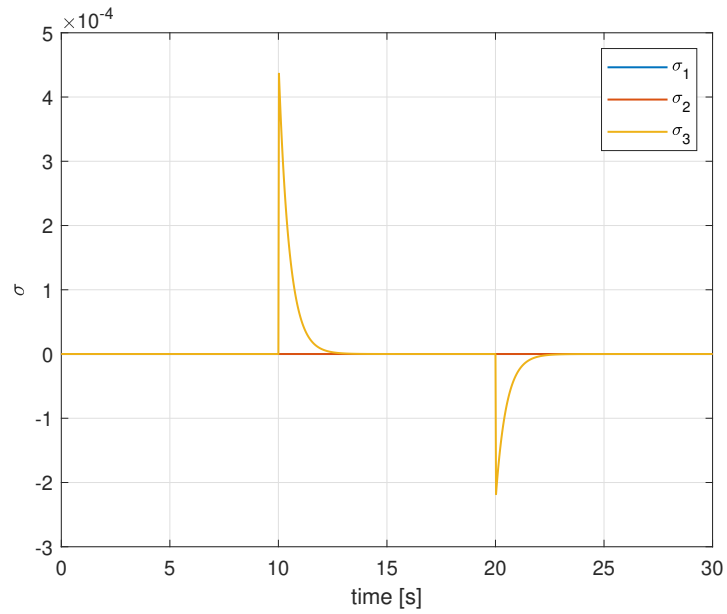


Figure 3.19: Numerical simulation - Adaptive BLFOSM - $k = 0.01$ - Triangular angular velocity: sliding variable

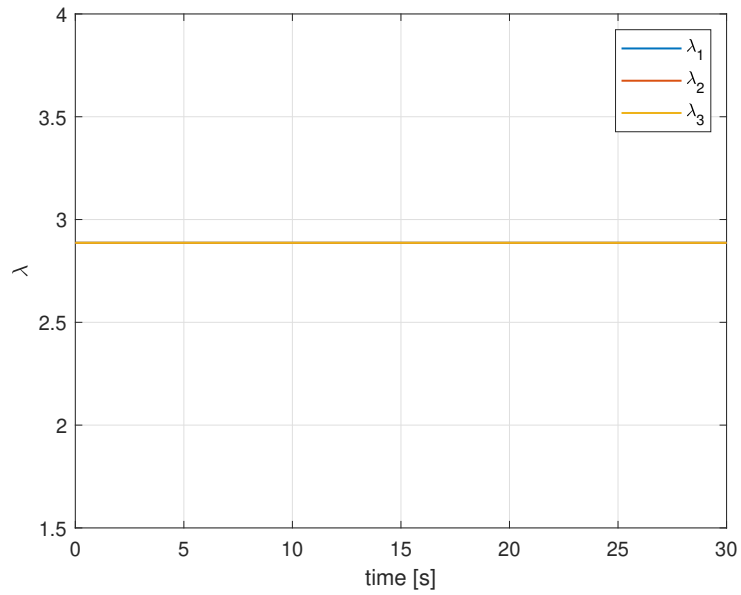


Figure 3.20: Numerical simulation - Adaptive BLFOSM - $k = 0.01$ - Triangular angular velocity: slope

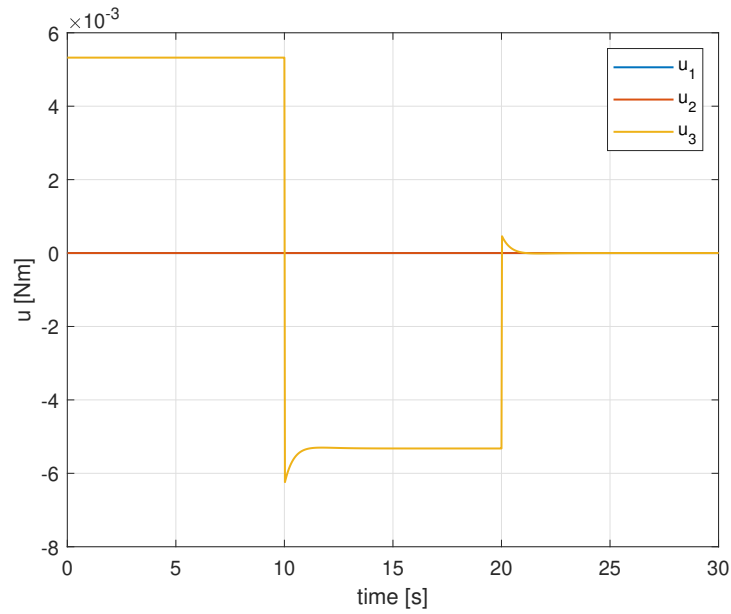


Figure 3.21: Numerical simulation - Adaptive BLFOSM - $k = 0.01$ - Triangular angular velocity: control input

Figures 3.17 and 3.18 illustrate that the numerical simulation of the adaptive approach with a control gain value of 0.01 successfully follows the desired angular velocity and quaternion. Figure 3.20 demonstrates that in this scenario the slope of the sliding variable remains constant throughout the simulation. For the triangular angular velocity reference, with a control gain value of 0.01, the adaptive approach numerically behaves identically to the traditional approach. Indeed, the results in Figures 3.13-3.16 correspond exactly to those in Figures 3.17-3.21.

Numerical - $k = 0.5$ - Traditional BLFOSM

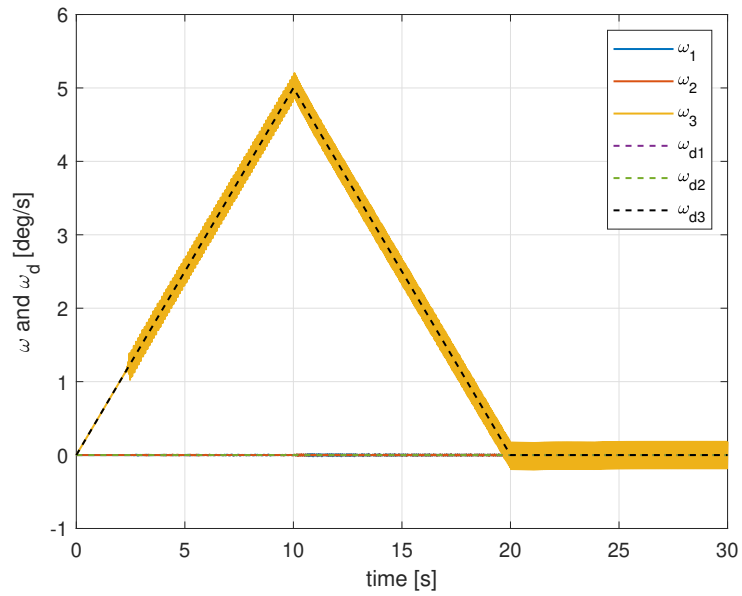


Figure 3.22: Numerical simulation - Traditional BLFOSM - $k = 0.5$ - Triangular angular velocity: angular velocity

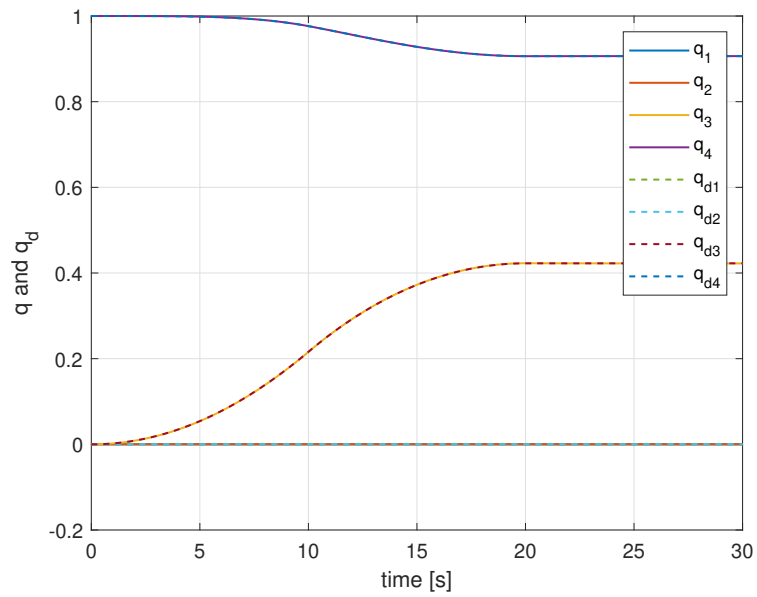


Figure 3.23: Numerical simulation - Traditional BLFOSM - $k = 0.5$ - Triangular angular velocity: quaternion

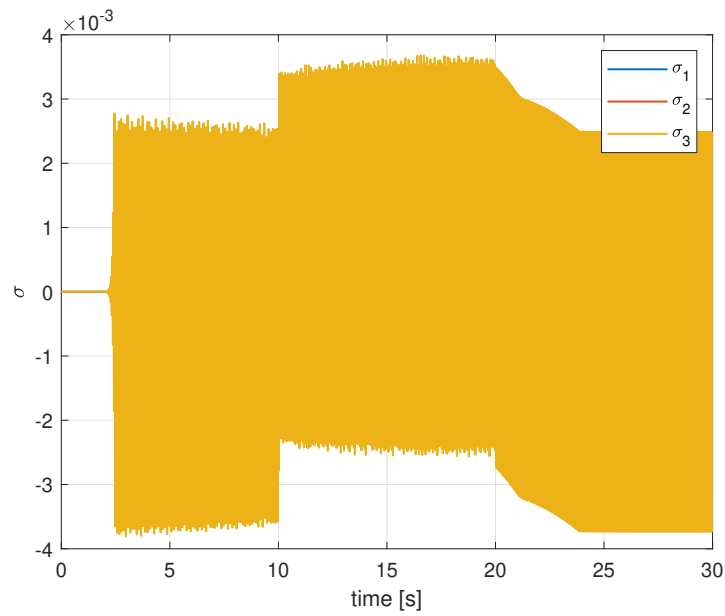


Figure 3.24: Numerical simulation - Traditional BLFOSM - $k = 0.5$ - Triangular angular velocity: sliding variable

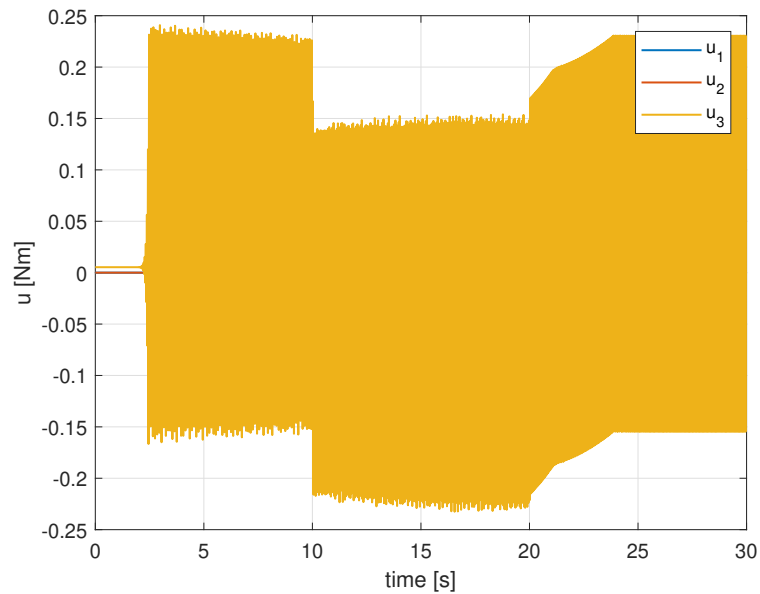


Figure 3.25: Numerical simulation - Traditional BLFOSM - $k = 0.5$ - Triangular angular velocity: control input

Figure 3.23 illustrates that the numerical simulation of the traditional approach with a control gain value of 0.5 successfully follows the desired quaternion. Figure

3.22 shows that the desired value of the z-component of the angular velocity is followed, but exhibits oscillations. Figures 3.24 and 3.25 display that also the sliding variable and the the control input present oscillations.

Numerical - $k = 0.5$ - Adaptive BLFOSM

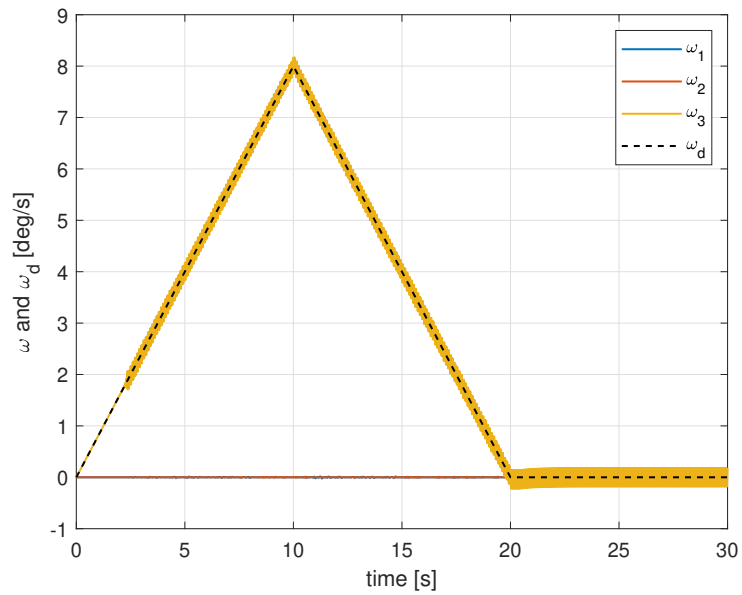


Figure 3.26: Numerical simulation - Adaptive BLFOSM - $k = 0.5$ - Triangular angular velocity: angular velocity

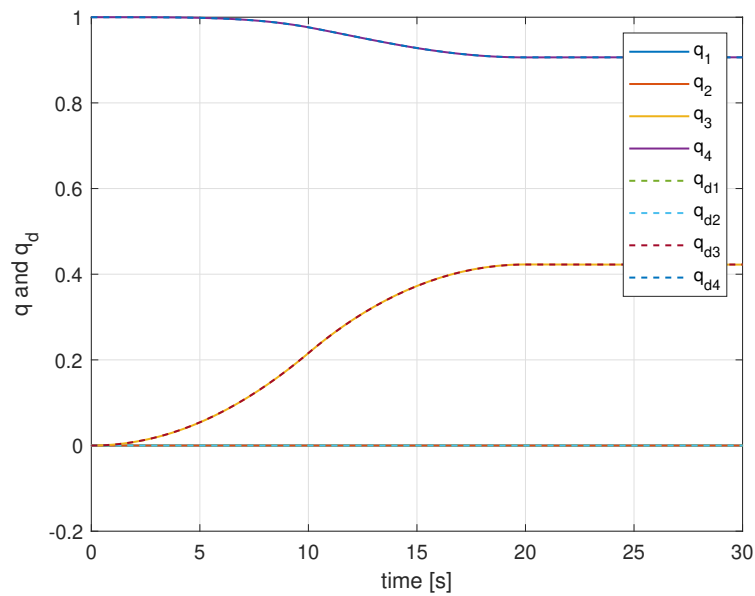


Figure 3.27: Numerical simulation - Adaptive BLFOSM - $k = 0.5$ - Triangular angular velocity: quaternion

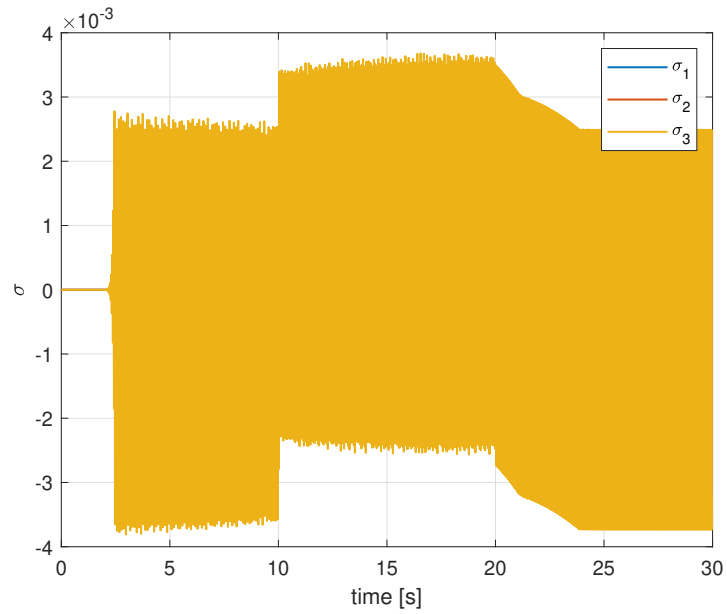


Figure 3.28: Numerical simulation - Adaptive BLFOSM - $k = 0.5$ - Triangular angular velocity: sliding variable

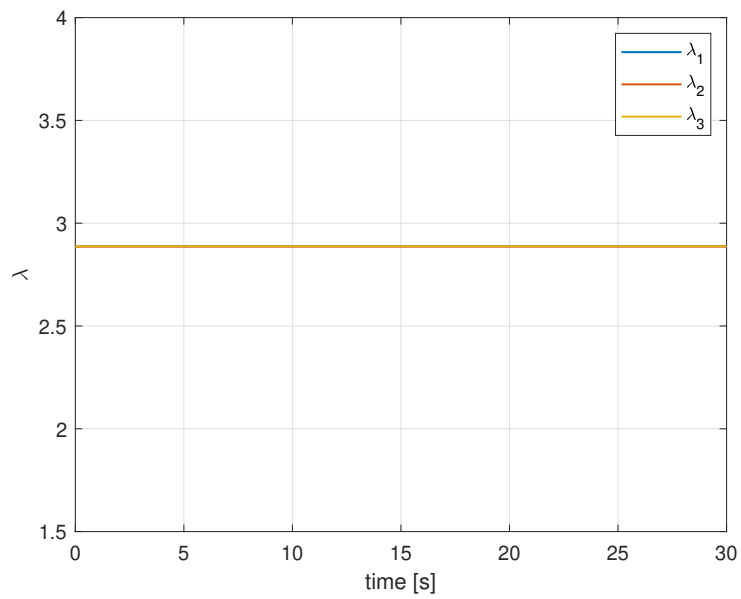


Figure 3.29: Numerical simulation - Adaptive BLFOSM - $k = 0.5$ - Triangular angular velocity: slope

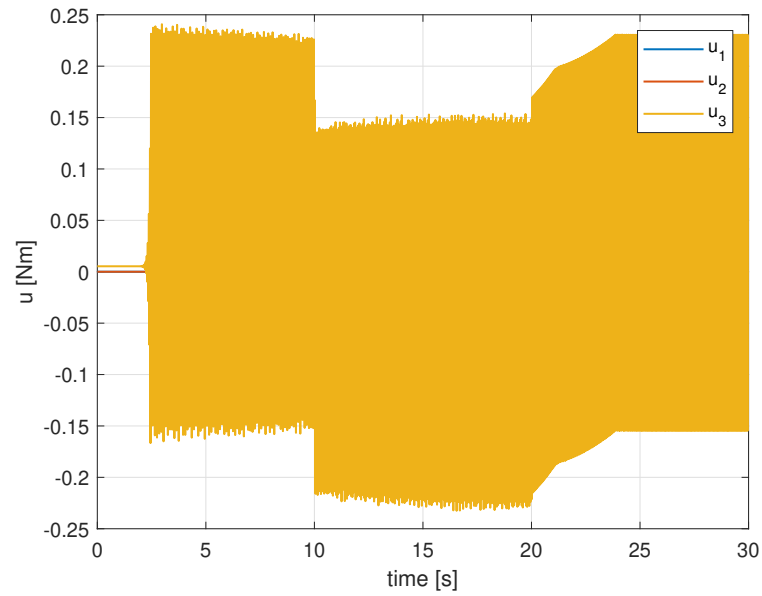


Figure 3.30: Numerical simulation - Adaptive BLFOSM - $k = 0.5$ - Triangular angular velocity: control input

As observed for $k = 0.01$, the adaptive controller (Figures 3.26-3.30) behaves identically to the traditional controller (Figures 3.22-3.25) for $k = 0.5$.

Experimental - $k = 0.01$ - Traditional BLFOSM

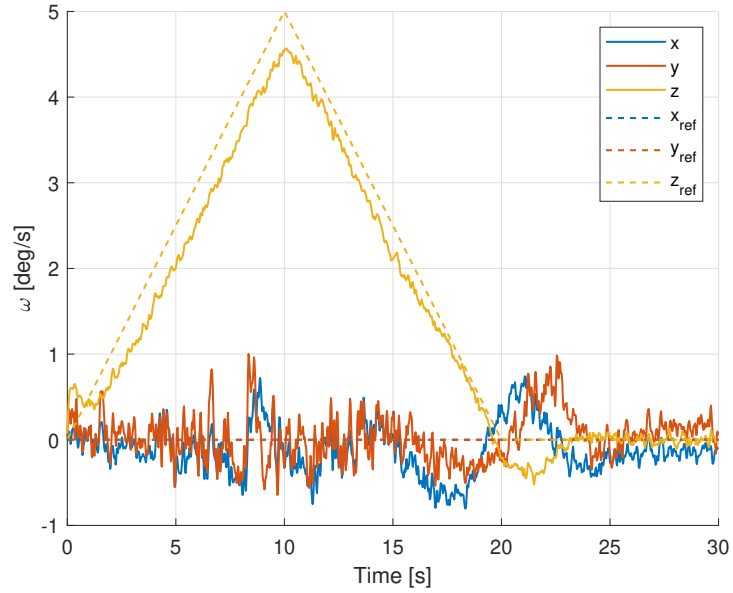


Figure 3.31: Experimental testing - Traditional BLFOSM - $k = 0.01$ - Triangular angular velocity: angular velocity

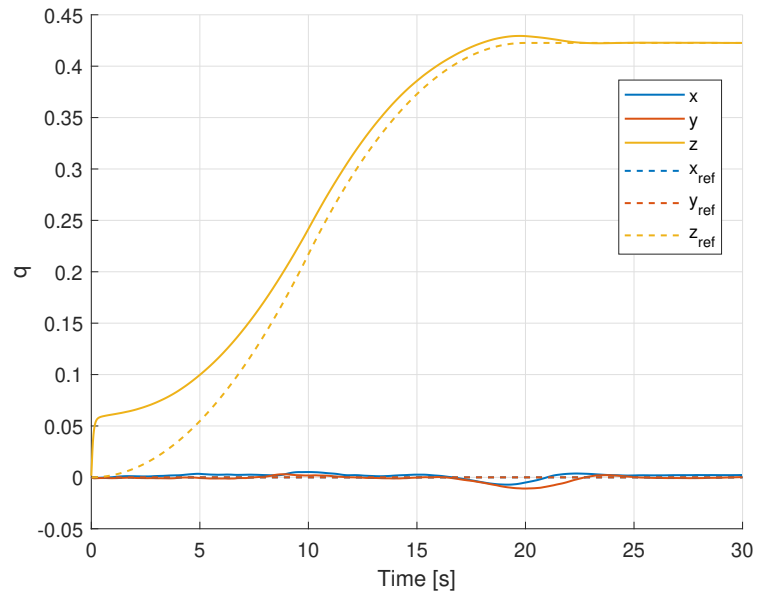


Figure 3.32: Experimental testing - Traditional BLFOSM - $k = 0.01$ - Triangular angular velocity: quaternion

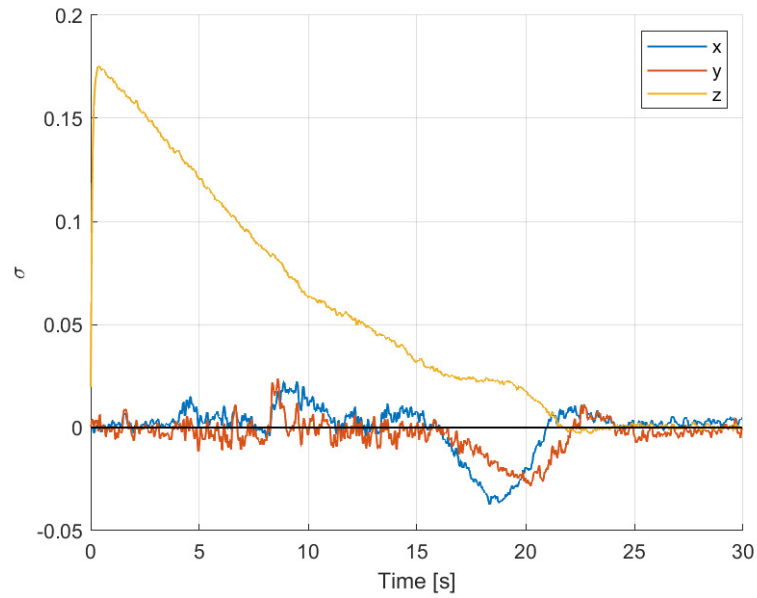


Figure 3.33: Experimental testing - Traditional BLFOSM - $k = 0.01$ - Triangular angular velocity: sliding variable

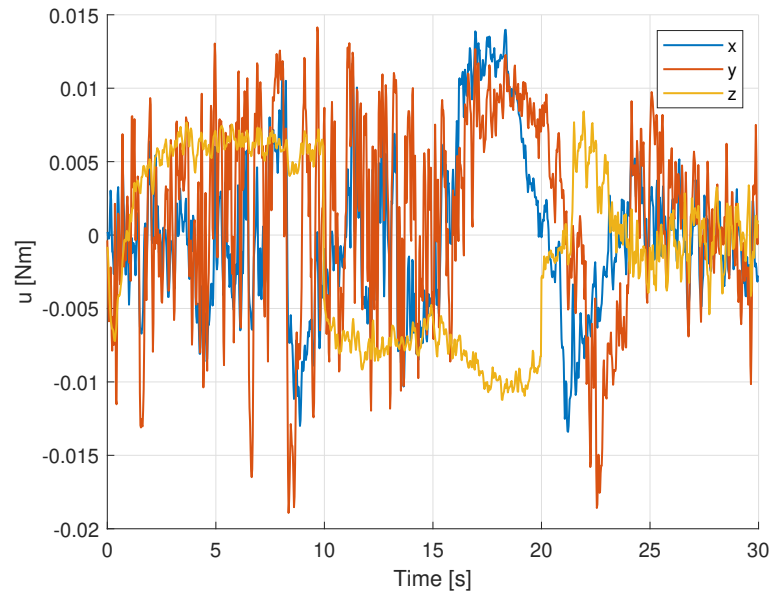


Figure 3.34: Experimental testing - Traditional BLFOSM - $k = 0.01$ - Triangular angular velocity: control input

Figures 3.31 and 3.32 illustrate that the experimental test of the traditional approach with a control gain value of 0.01 attempts to follow the desired angular velocity and quaternion for the first 23 seconds, successfully tracking them in the final 7 seconds of the experiment. Figure 3.33 shows that after about 22 seconds the sliding variable reaches and maintains zero.

Experimental - $k = 0.01$ - Adaptive BLFOSM

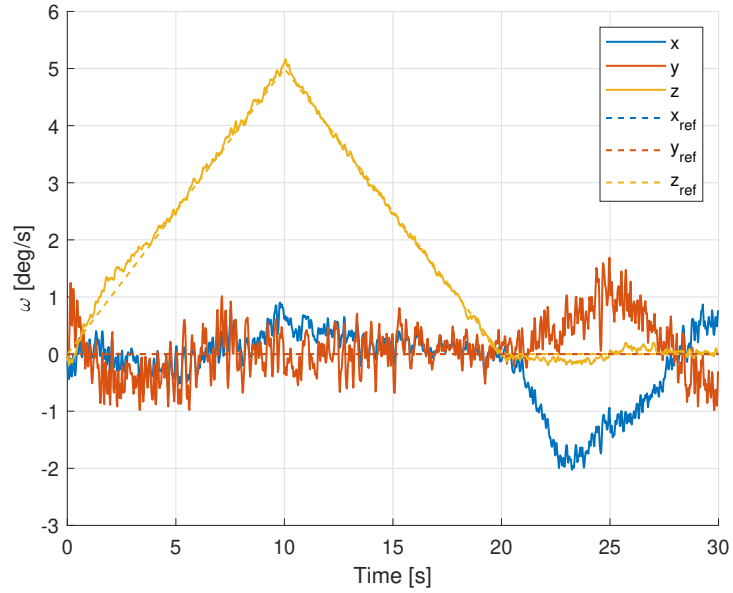


Figure 3.35: Experimental testing - Adaptive BLFOSM - $k = 0.01$ - Triangular angular velocity: angular velocity

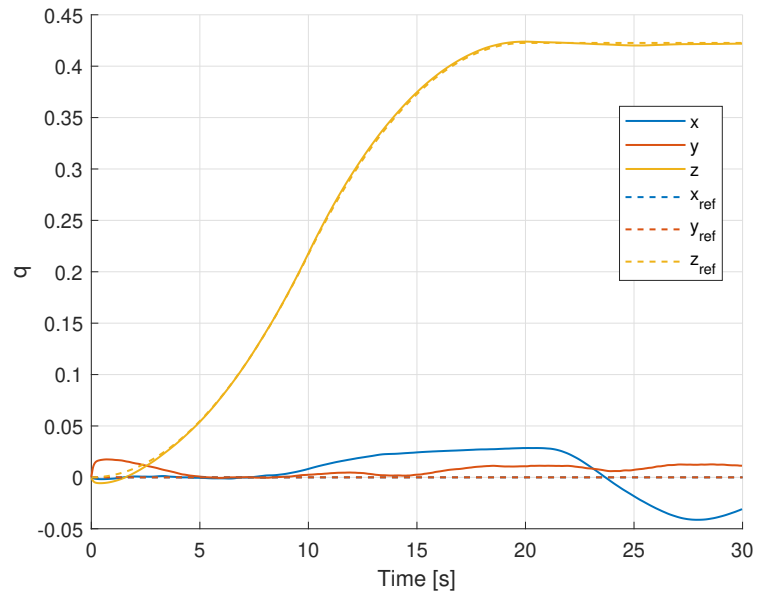


Figure 3.36: Experimental testing - Adaptive BLFOSM - $k = 0.01$ - Triangular angular velocity: quaternion

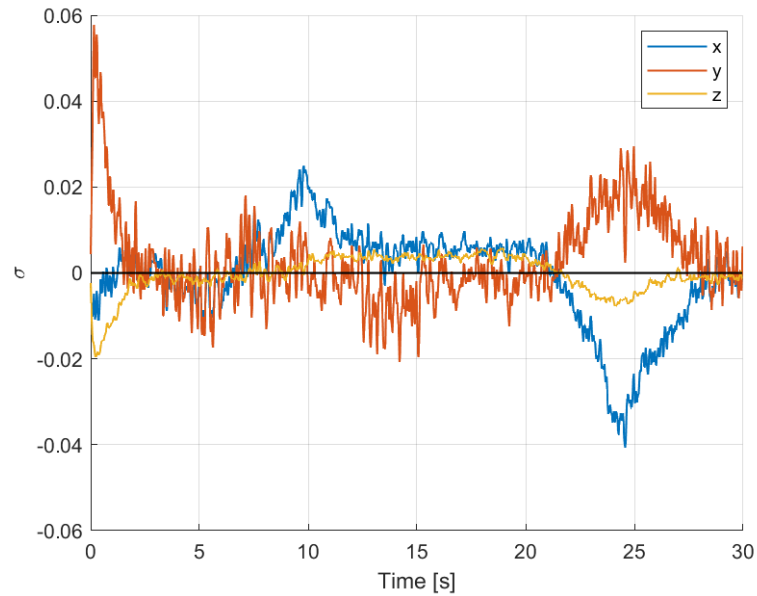


Figure 3.37: Experimental testing - Adaptive BLFOSM - $k = 0.01$ - Triangular angular velocity: sliding variable

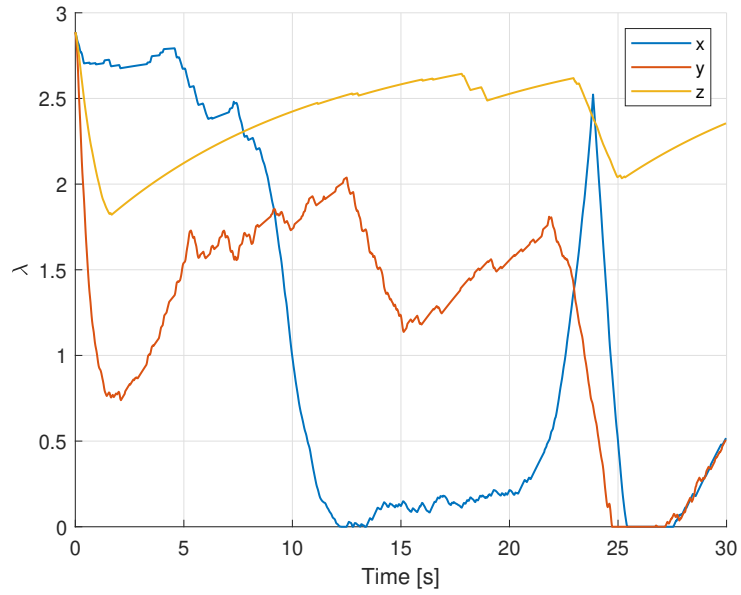


Figure 3.38: Experimental testing - Adaptive BLFOSM - $k = 0.01$ - Triangular angular velocity: slope

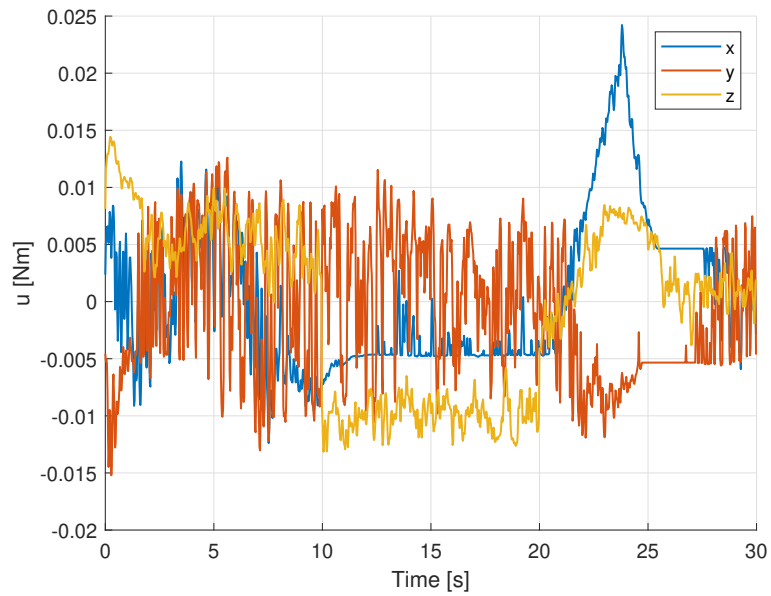


Figure 3.39: Experimental testing - Adaptive BLFOSM - $k = 0.01$ - Triangular angular velocity: control input

Figures 3.35 and 3.36 illustrate that the experimental test of the adaptive approach with a control gain value of 0.01 follows the desired angular velocity and quaternion quite well, demonstrating better performance than the traditional approach under the same conditions. Figure 3.37 shows that after approximately 4 seconds the sliding variable reaches zero and maintains this value, exhibiting some oscillations with a magnitude of 0.01.

Figure 3.38 presents the trend of the slope, which is subject to oscillations for all the three components.

Experimental - $k = 0.5$ - Traditional BLFOSM

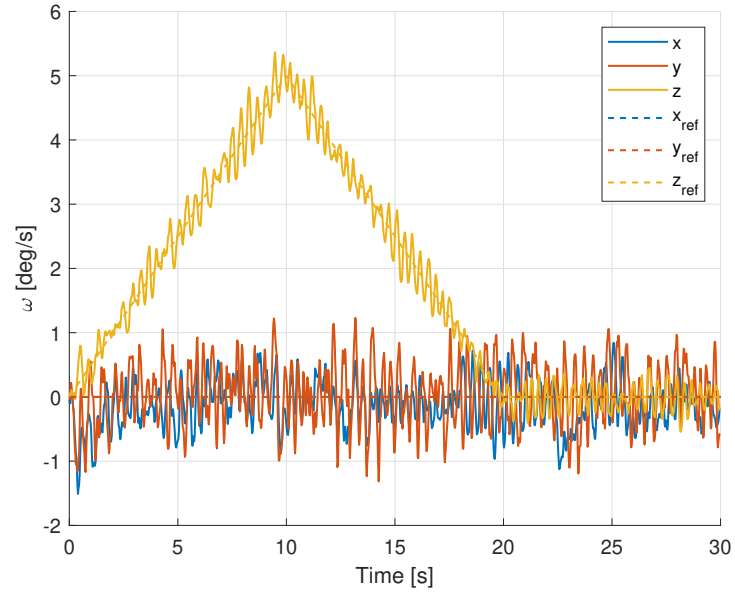


Figure 3.40: Experimental testing - Traditional BLFOSM - $k = 0.5$ - Triangular angular velocity: angular velocity

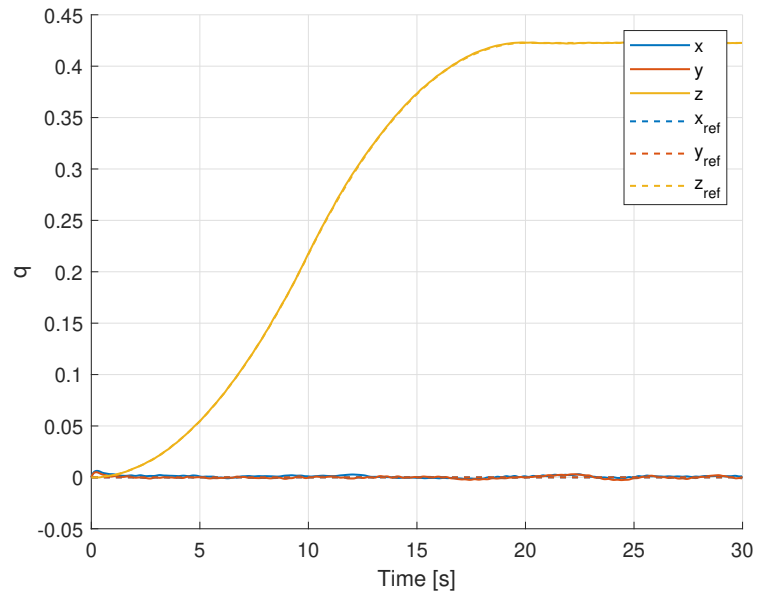


Figure 3.41: Experimental testing - Traditional BLFOSM - $k = 0.5$ - Triangular angular velocity: quaternion

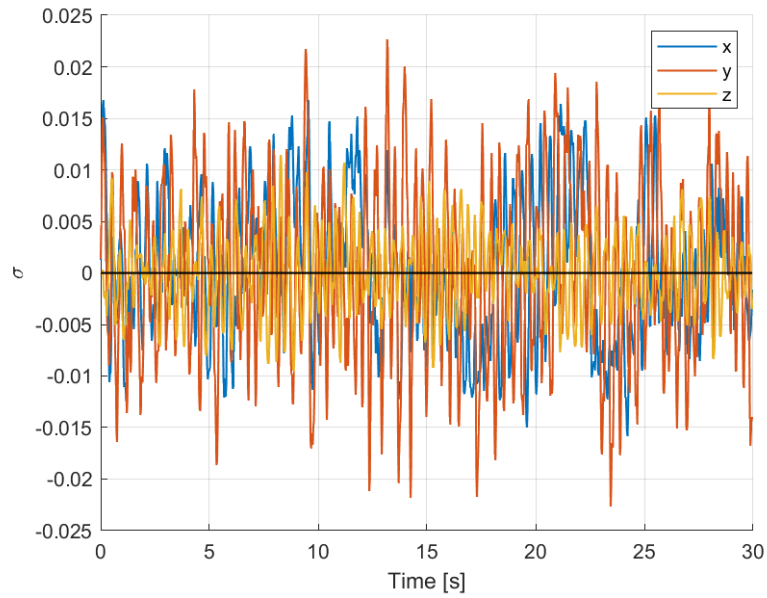


Figure 3.42: Experimental testing - Traditional BLFOSM - $k = 0.5$ - Triangular angular velocity: sliding variable

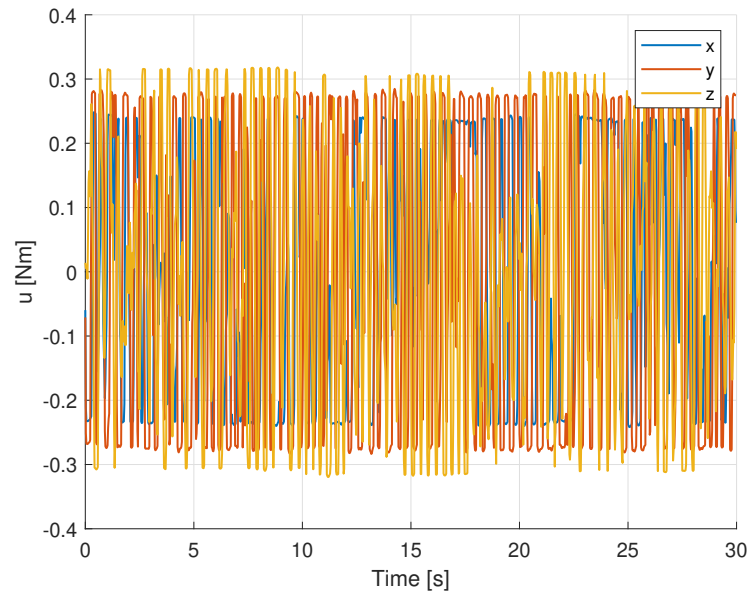


Figure 3.43: Experimental testing - Traditional BLFOSM - $k = 0.5$ - Triangular angular velocity: control input

Figures 3.40 and 3.41 illustrate that the experimental test of the traditional approach with a control gain value of 0.5 follows the desired angular velocity and quaternion. However, Figure 3.40 presents wider oscillations than the case with a control gain of 0.01 (Figure 3.31). Figure 3.42 shows that the sliding variable reaches zero almost immediately, but presents oscillations. Figure 3.43 shows that the control input exhibits oscillation that are one order of magnitude wider than in the case with a control gain of 0.01 (Figure 3.34).

Experimental - $k = 0.5$ - Adaptive BLFOSM

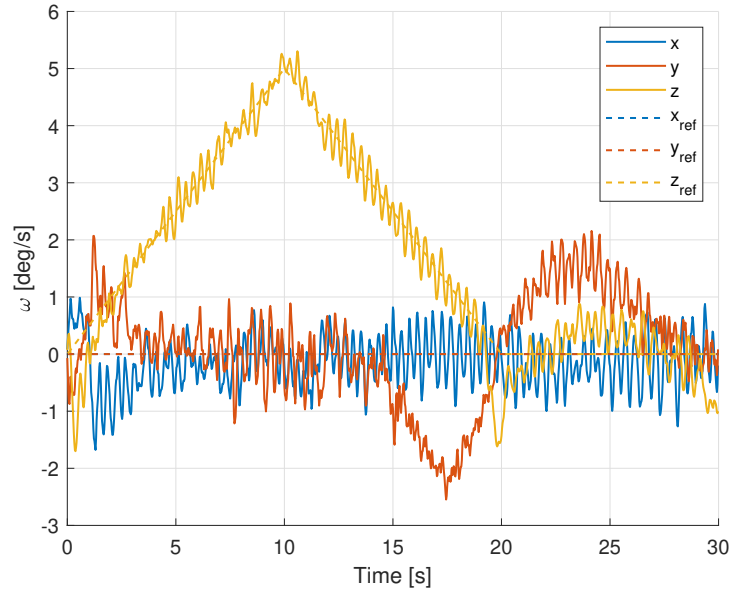


Figure 3.44: Experimental testing - Adaptive BLFOSM - $k = 0.5$ - Triangular angular velocity: angular velocity

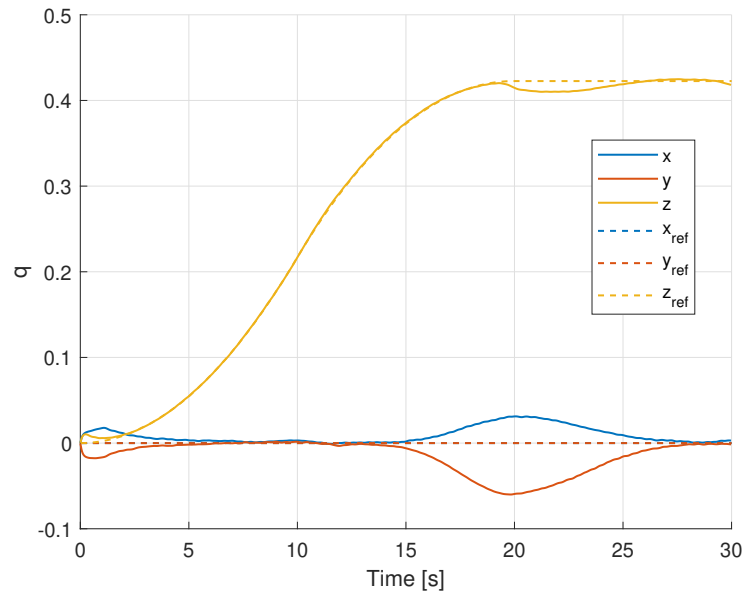


Figure 3.45: Experimental testing - Adaptive BLFOSM - $k = 0.5$ - Triangular angular velocity: quaternion

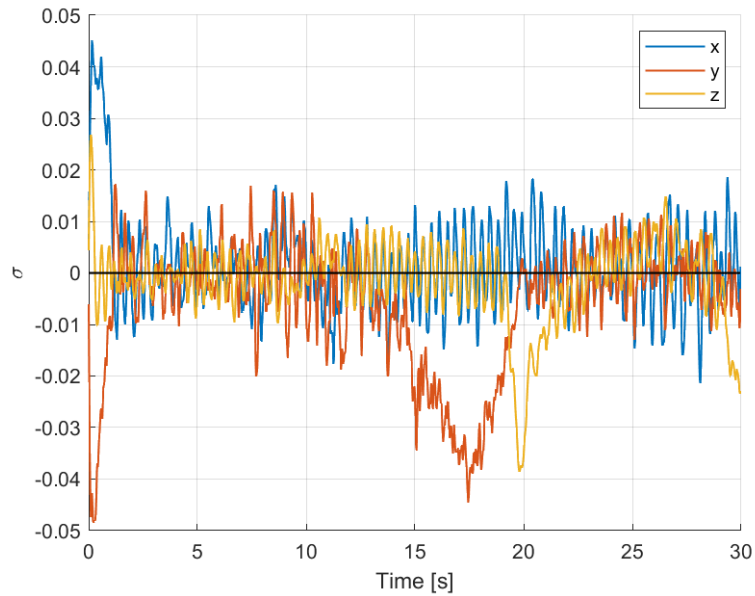


Figure 3.46: Experimental testing - Adaptive BLFOSM - $k = 0.5$ - Triangular angular velocity: sliding variable

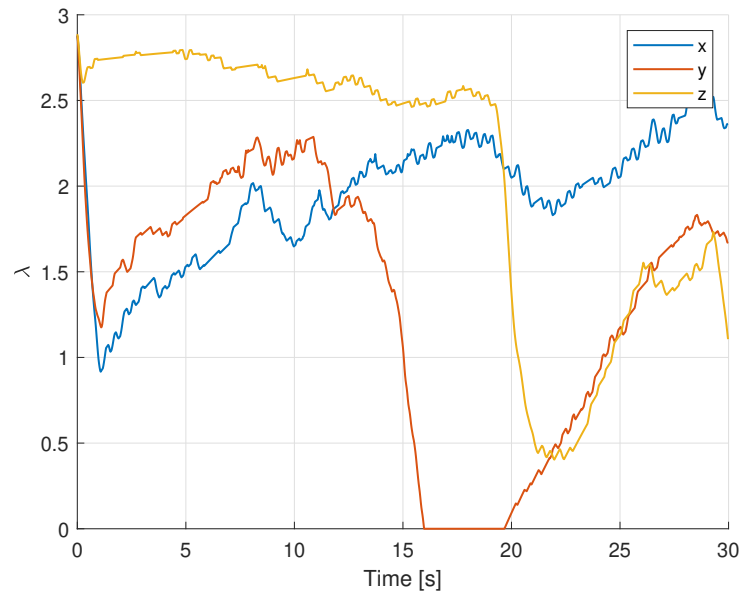


Figure 3.47: Experimental testing - Adaptive BLFOSM - $k = 0.5$ - Triangular angular velocity: slope

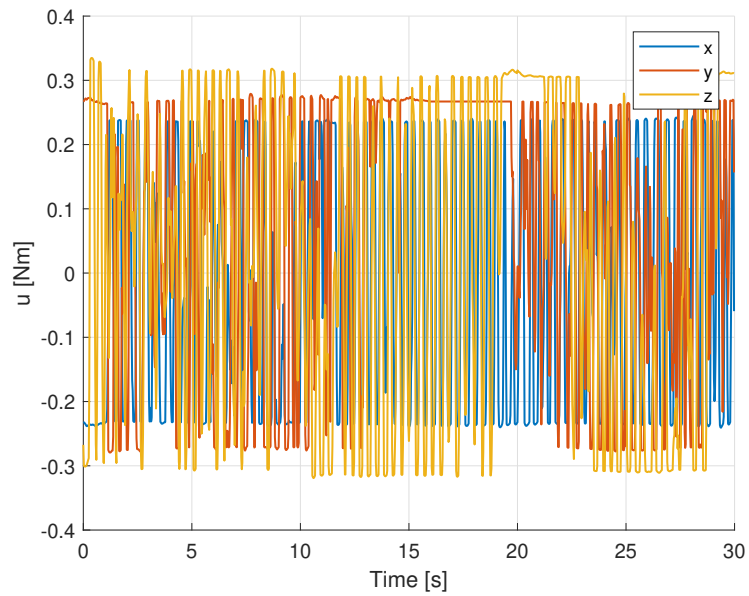


Figure 3.48: Experimental testing - Adaptive BLFOSM - $k = 0.5$ - Triangular angular velocity: control input

The experimental results of the adaptive approach with a control gain value of 0.5 (Figures 3.44-3.48) are similar to those obtained using the traditional approach with the same value of control gain (Figures 3.40-3.43), except for a peak occurring between approximately 20 and 25 seconds, likely caused by the delicate test environment.

3.2.2 Constant reference

The following subsection presents the results for the triangular angular velocity reference. Numerical results will be illustrated first, followed by the experimental results.

For each set of results, the trends of the following parameters will be shown: angular velocity, quaternion, sliding variable, control input, and, in the case of the adaptive approach, slope of the sliding variable.

Numerical - $k = 0.01$ - Traditional BLFOSM

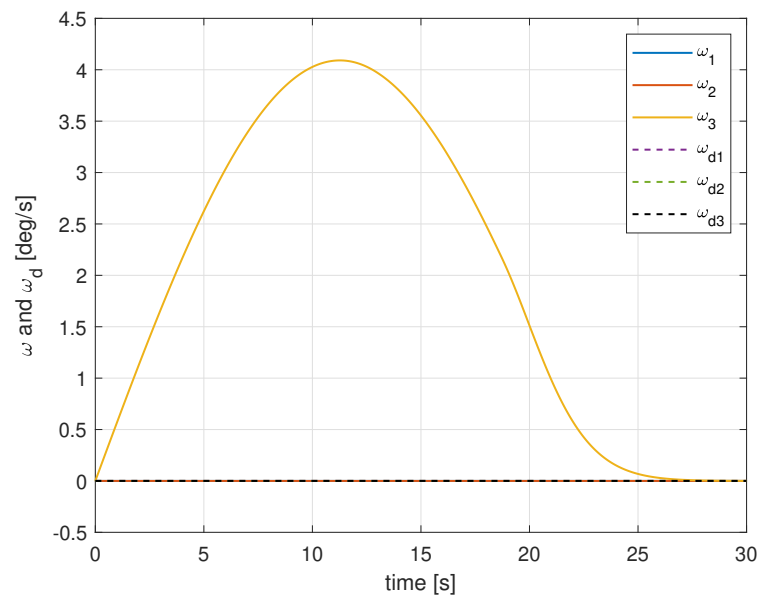


Figure 3.49: Numerical simulation - Traditional BLFOSM - $k = 0.01$ - Constant reference: angular velocity

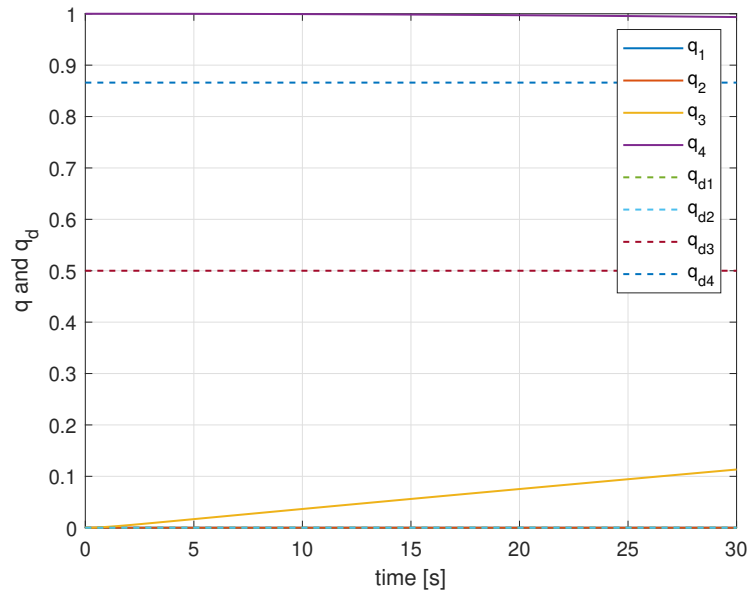


Figure 3.50: Numerical simulation - Traditional BLFOSM - $k = 0.01$ - Constant reference: quaternion

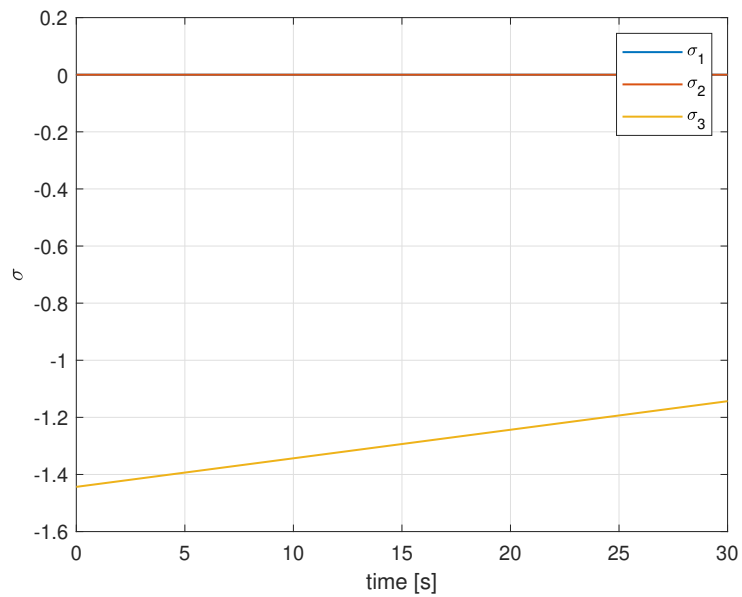


Figure 3.51: Numerical simulation - Traditional BLFOSM - $k = 0.01$ - Constant reference: sliding variable

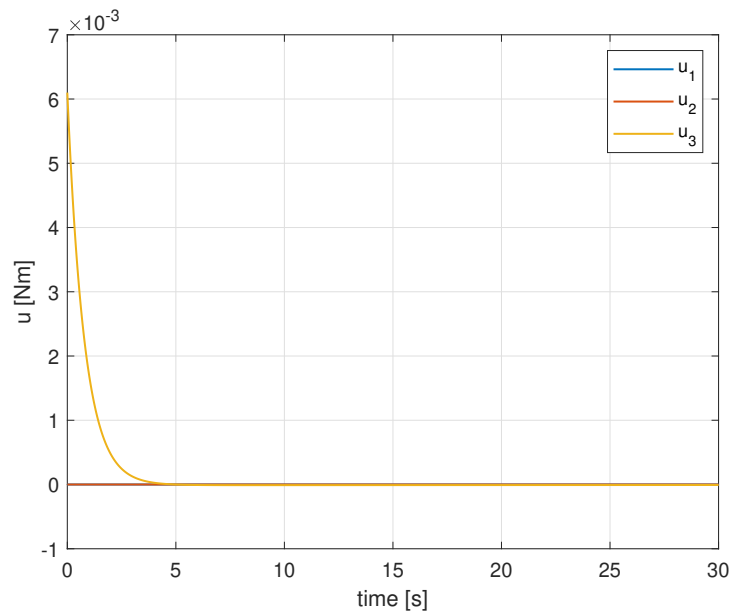


Figure 3.52: Numerical simulation - Traditional BLFOSM - $k = 0.01$ - Constant reference: control input

Figures 3.49 and 3.50 illustrate that the 30 seconds of the simulations are insufficient for the z-component of the angular velocity and for the quaternion components to reach the desired values using the traditional approach with a control gain value of 0.01. The sliding surface does not reach zero within the simulation time, as shown in Figure 3.51.

Numerical - $k = 0.01$ - Adaptive BLFOSM

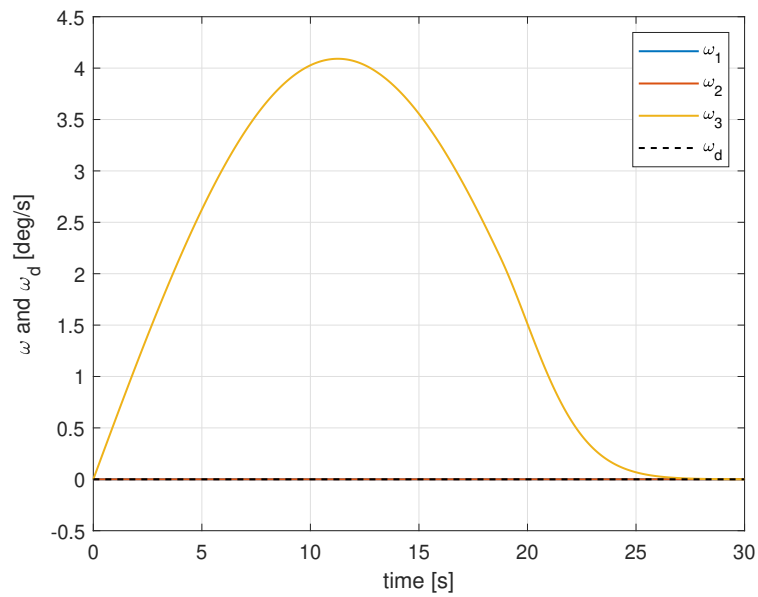


Figure 3.53: Numerical simulation - Adaptive BLFOSM - $k = 0.01$ - Constant reference: angular velocity

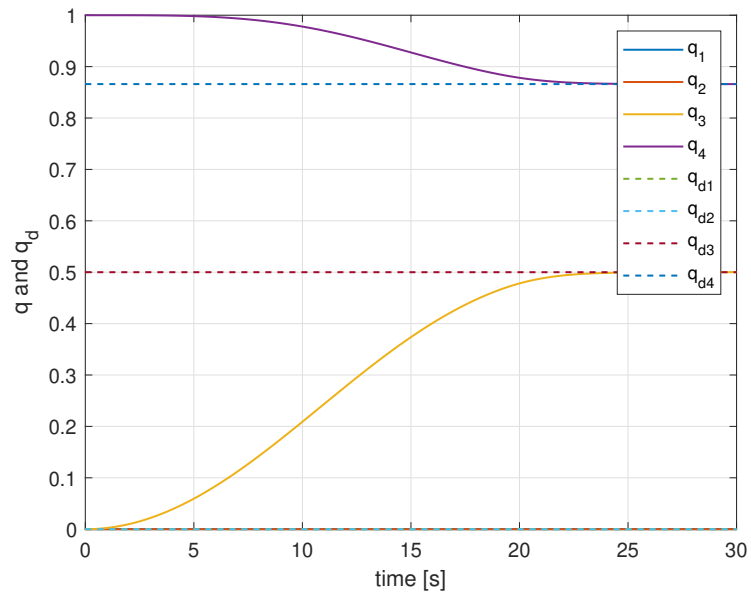


Figure 3.54: Numerical simulation - Adaptive BLFOSM - $k = 0.01$ - Constant reference: quaternion

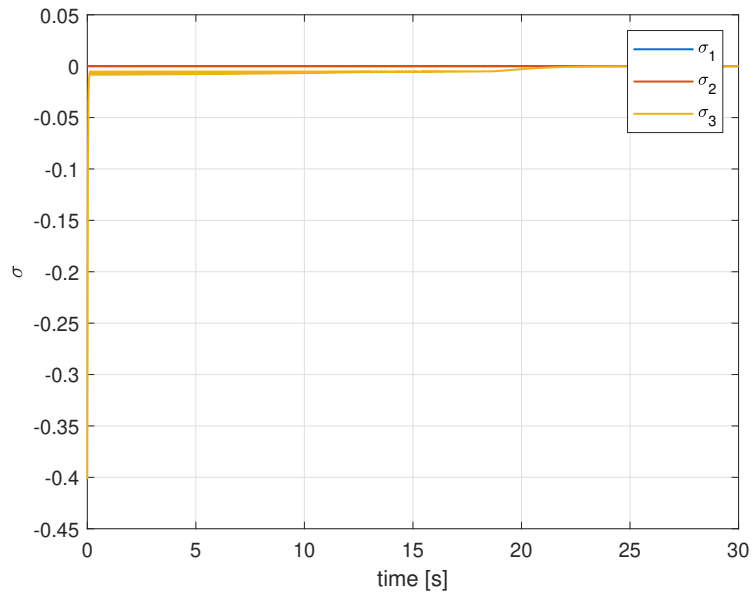


Figure 3.55: Numerical simulation - Adaptive BLFOSM - $k = 0.01$ - Constant reference: sliding variable

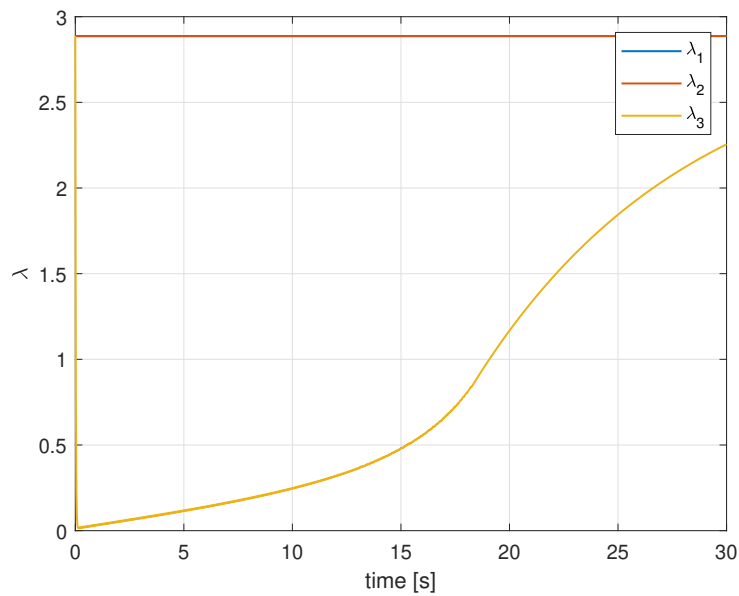


Figure 3.56: Numerical simulation - Adaptive BLFOSM - $k = 0.01$ - Constant reference: slope

Figures 3.53 and 3.54 illustrate that the z-component of the angular velocity and the quaternion components reach the desired values within the simulation

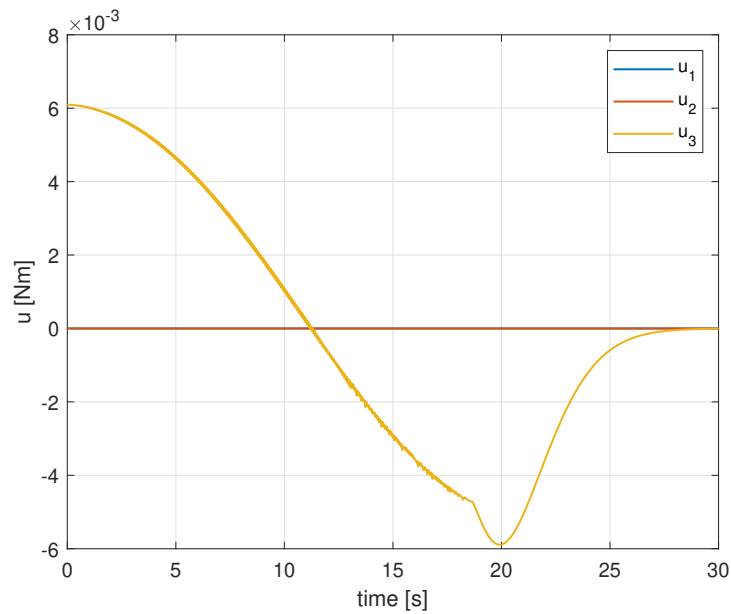


Figure 3.57: Numerical simulation - Adaptive BLFOSM - $k = 0.01$ - Constant reference: control input

time using the adaptive approach with a control gain value of 0.01. The sliding surface reaches and maintains zero after approximately 23 seconds, exhibiting slight oscillations between 0 and 19 seconds, as shown in Figure 3.55. Figure 3.56 depicts the trend of the slope of the sliding variable: the z-component of the slope does not converge to its initial value within the simulation time. Figure 3.57 shows that the z-component of the control input exhibits slight oscillations between 0 and 19 seconds.

Numerical - $k = 0.5$ - Traditional BLFOSM

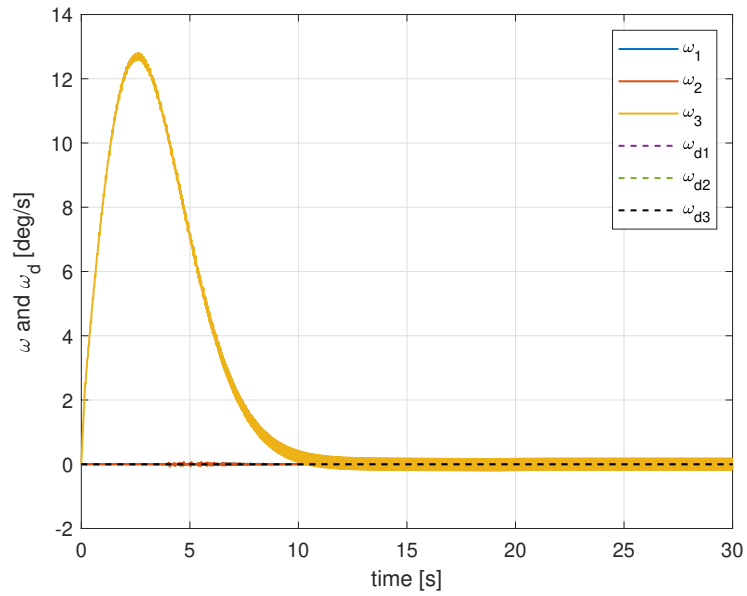


Figure 3.58: Numerical simulation - Traditional BLFOSM - $k = 0.5$ - Constant reference: angular velocity

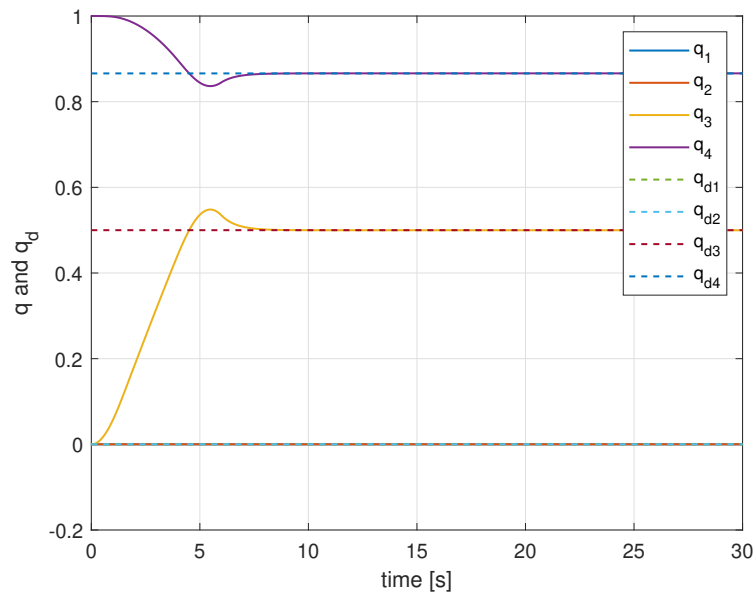


Figure 3.59: Numerical simulation - Traditional BLFOSM - $k = 0.5$ - Constant reference: quaternion

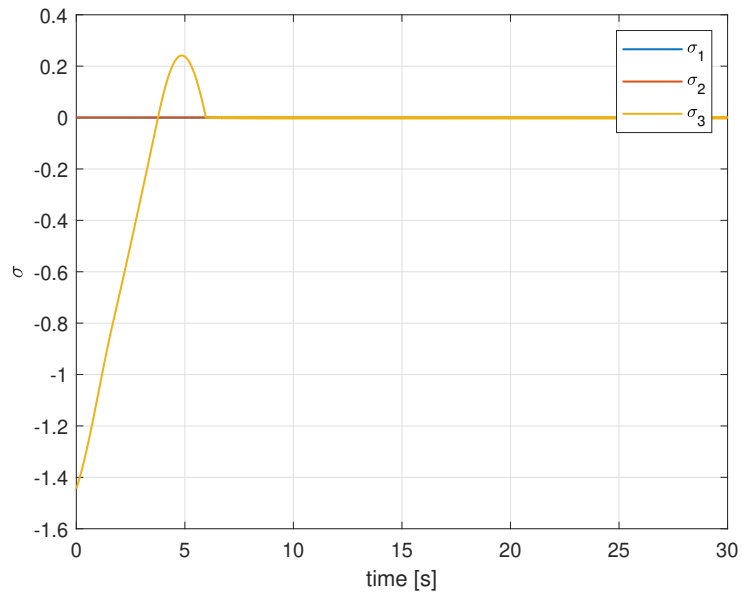


Figure 3.60: Numerical simulation - Traditional BLFOSM - $k = 0.5$ - Constant reference: sliding variable

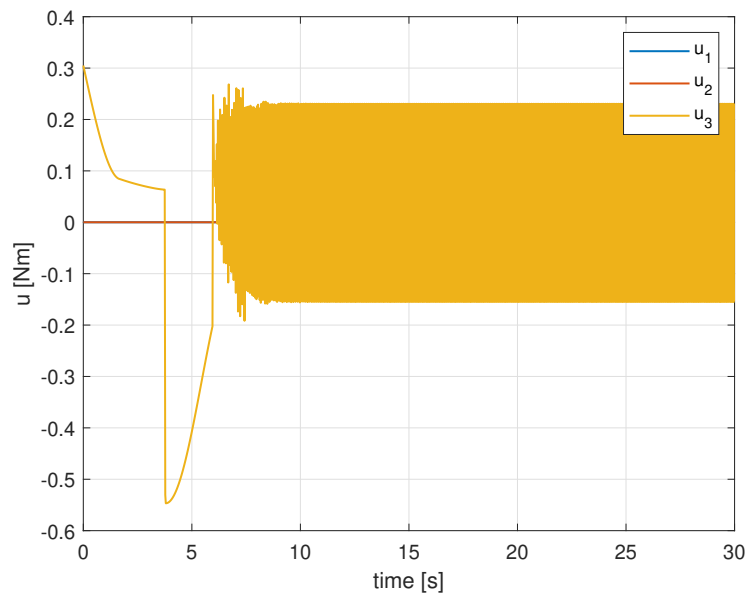


Figure 3.61: Numerical simulation - Traditional BLFOSM - $k = 0.5$ - Constant reference: control input

Figures 3.58 and 3.59 illustrate that the z-component of the angular velocity and the quaternion components reach the desired values within the simulation time using

the traditional approach with a control gain value of 0.5 . The sliding surface reaches and maintains zero after approximately 7 seconds, exhibiting slight oscillations between 7 and 30 seconds, as shown in Figure 3.60. Figure 3.61 shows that the z-component of the control input exhibits oscillations between approximately 7 and 30 seconds.

Numerical - $k = 0.5$ - Adaptive BLFOSM

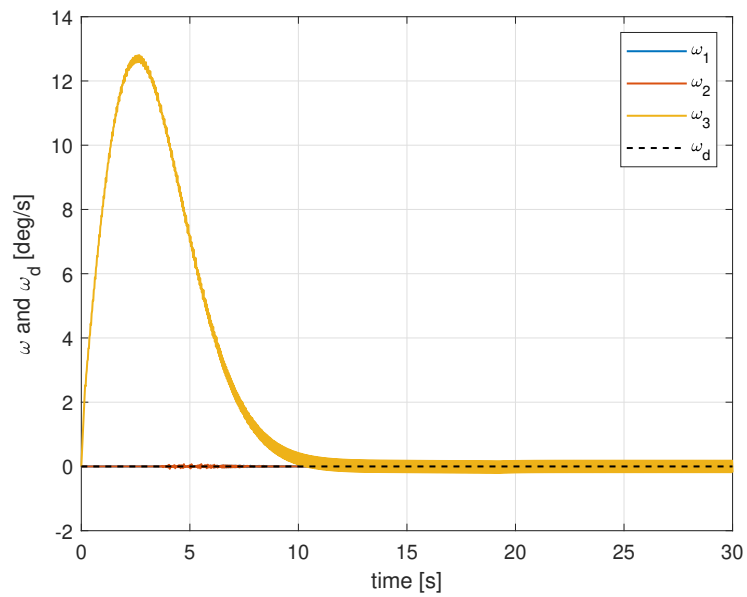


Figure 3.62: Numerical simulation - Adaptive BLFOSM - $k = 0.5$ - Constant reference: angular velocity

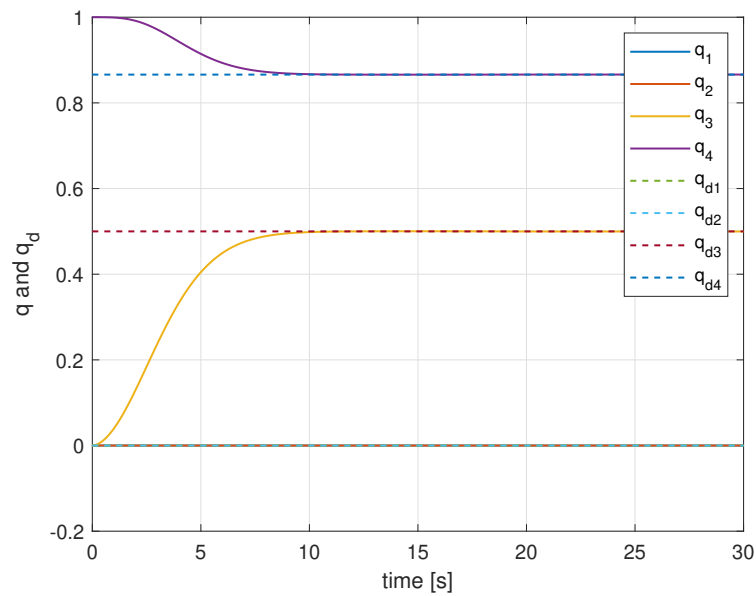


Figure 3.63: Numerical simulation - Adaptive BLFOSM - $k = 0.5$ - Constant reference: quaternion

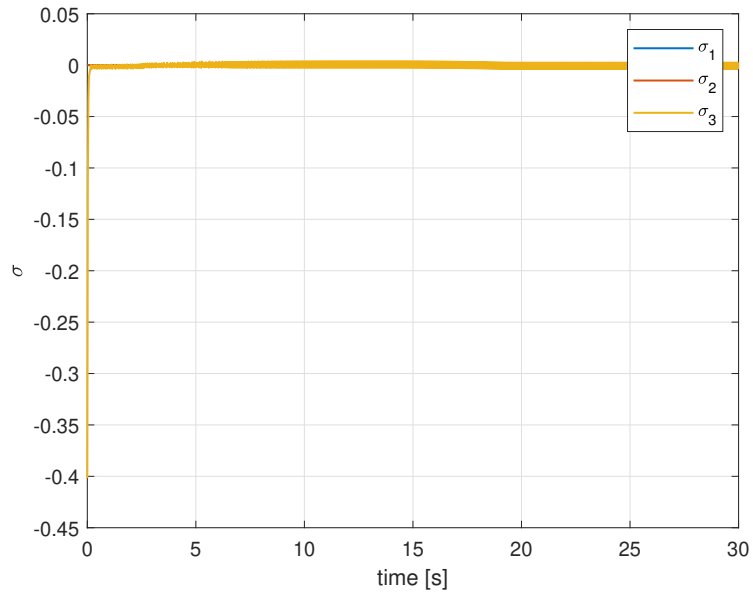


Figure 3.64: Numerical simulation - Adaptive BLFOSM - $k = 0.5$ - Constant reference: sliding variable

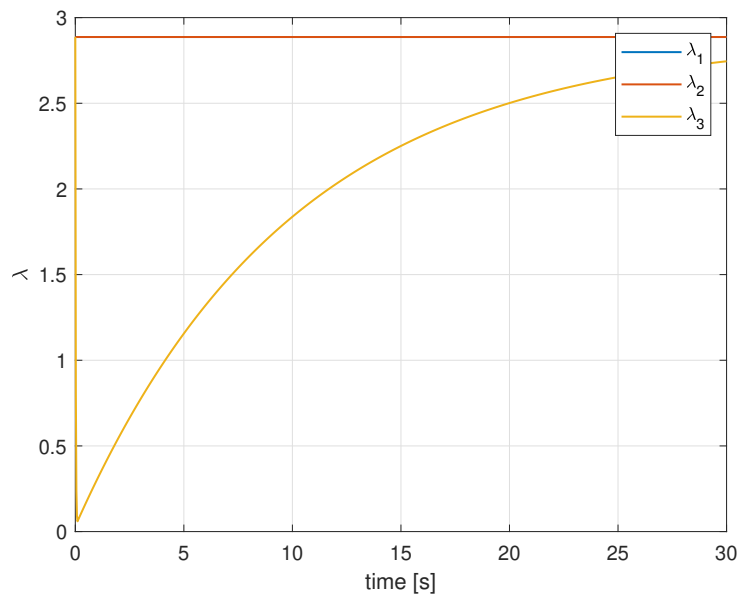


Figure 3.65: Numerical simulation - Adaptive BLFOSM - $k = 0.5$ - Constant reference: slope

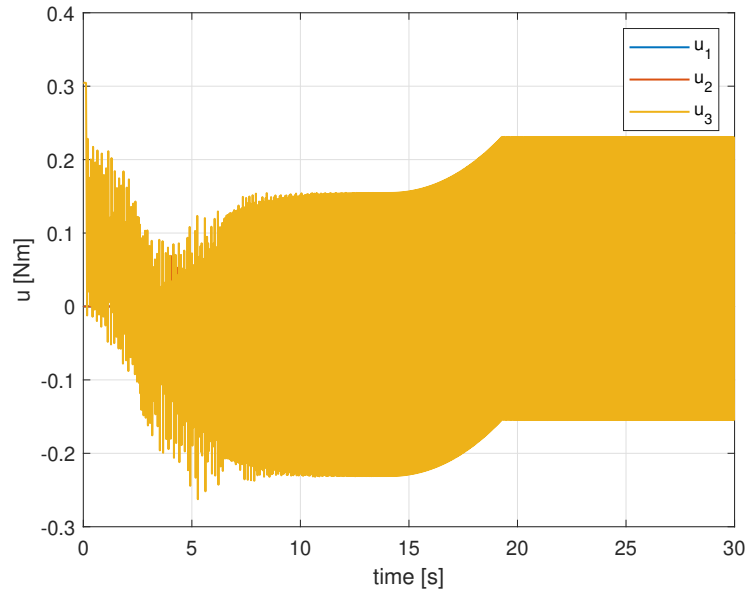


Figure 3.66: Numerical simulation - Adaptive BLFOSM - $k = 0.5$ - Constant reference: control input

Figures 3.62 and 3.63 illustrate that the z-component of the angular velocity and the quaternion components reach the desired values within the simulation time using the adaptive approach with a control gain value of 0.5 . The sliding surface reaches and maintains zero extremely fast, in less than 1 second, exhibiting slight oscillations between 0 and 30 seconds, as shown in Figure 3.64. Figure 3.66 shows that the z-component of the control input exhibits oscillations, between approximately 0 and 30 seconds, of the same magnitude as the traditional approach with the same value of control gain (Figure 3.61).

Experimental - $k = 0.01$ - Traditional BLFOSM

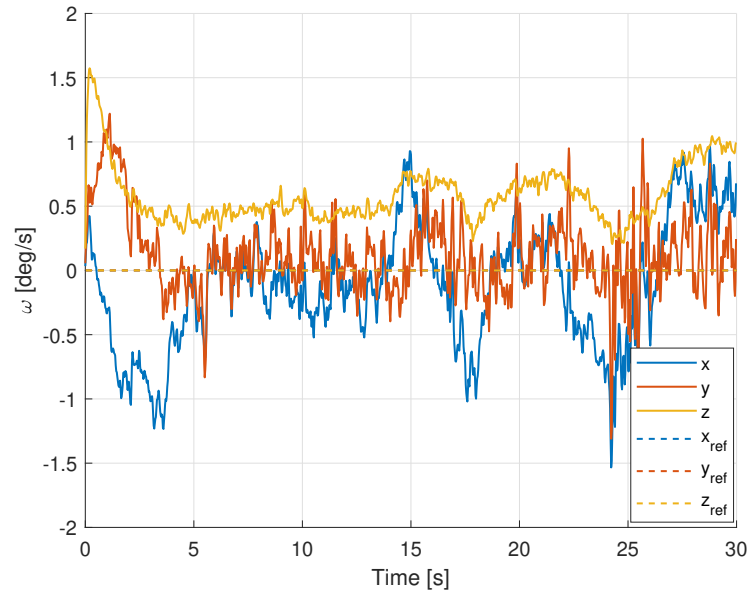


Figure 3.67: Experimental testing - Traditional BLFOSM - $k = 0.01$ - Constant reference: angular velocity

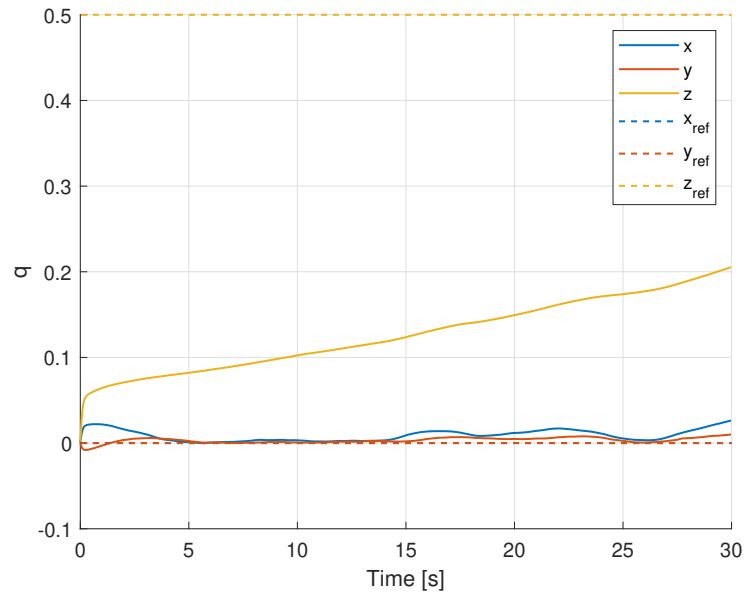


Figure 3.68: Experimental testing - Traditional BLFOSM - $k = 0.01$ - Constant reference: quaternion

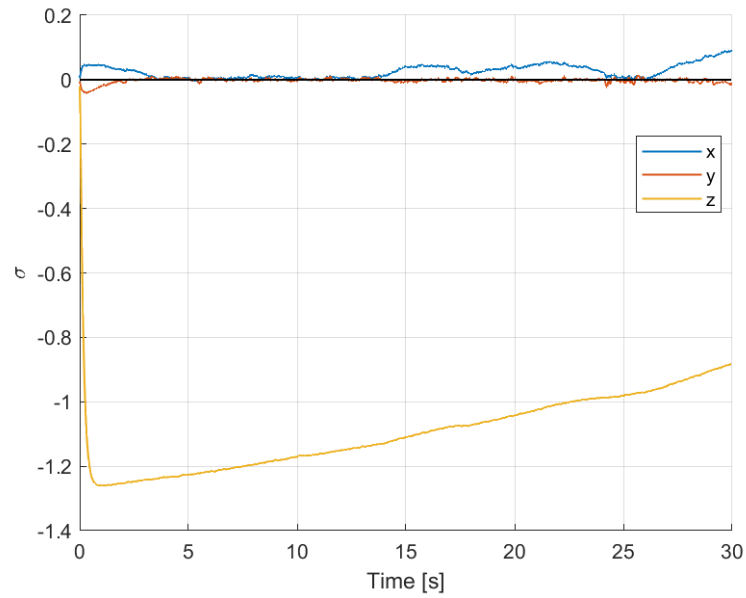


Figure 3.69: Experimental testing - Traditional BLFOSM - $k = 0.01$ - Constant reference: sliding variable

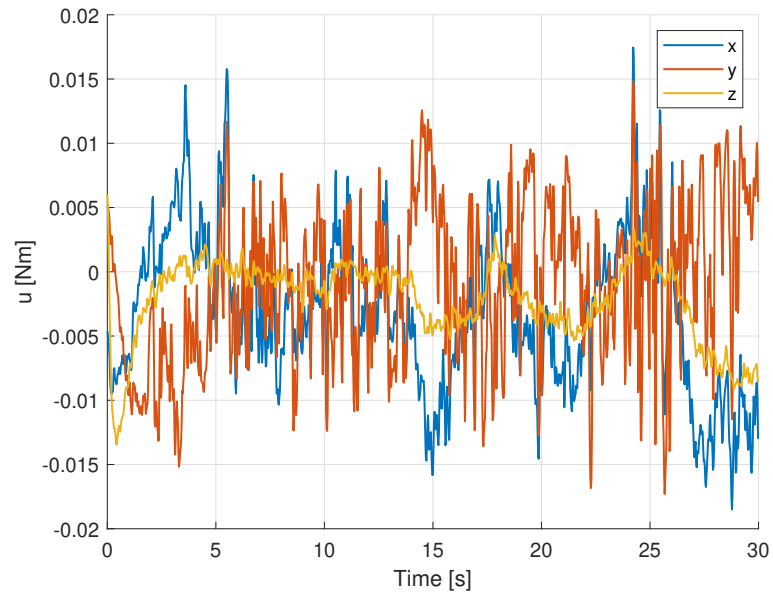


Figure 3.70: Experimental testing - Traditional BLFOSM - $k = 0.01$ - Constant reference: control input

Figures 3.67 and 3.68 illustrate that the 30 seconds of the simulations are insufficient for the z-component of the angular velocity and for the quaternion components to reach the desired values using the traditional approach with a control gain value of 0.01. Even the sliding surface does not converge to zero within the acquisition time, as shown in Figure 3.69.

Experimental - $k = 0.01$ - Adaptive BLFOSM

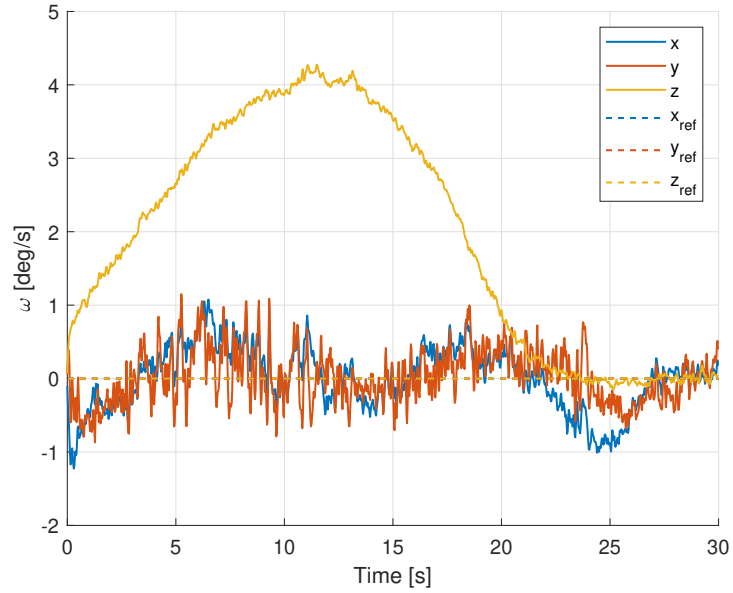


Figure 3.71: Experimental testing - Adaptive BLFOSM - $k = 0.01$ - Constant reference: angular velocity

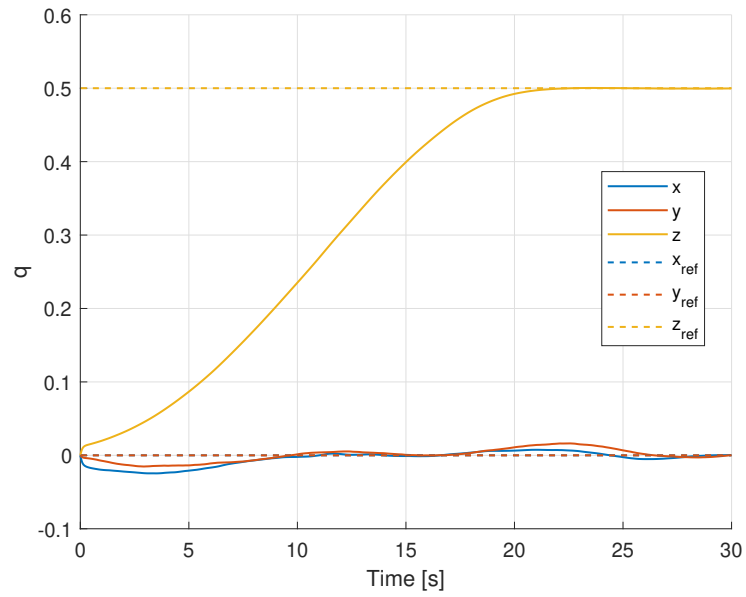


Figure 3.72: Experimental testing - Adaptive BLFOSM - $k = 0.01$ - Constant reference: quaternion

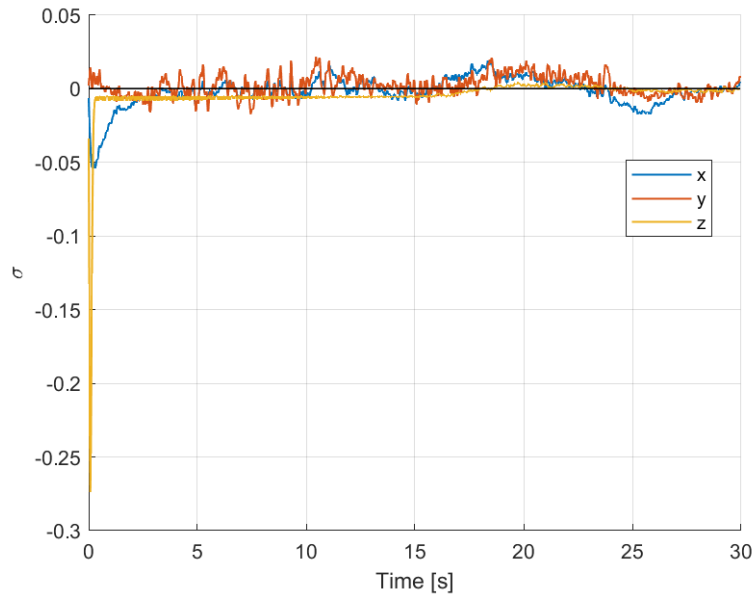


Figure 3.73: Experimental testing - Adaptive BLFOSM - $k = 0.01$ - Constant reference: sliding variable

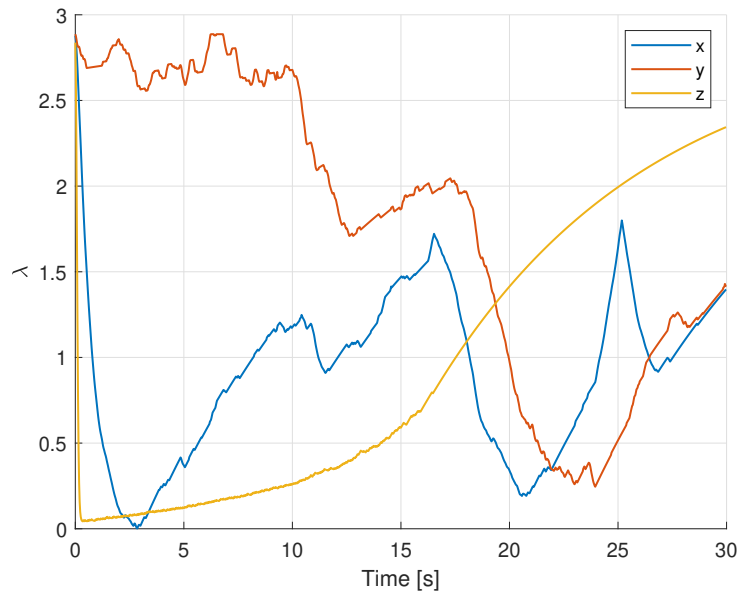


Figure 3.74: Experimental testing - Adaptive BLFOSM - $k = 0.01$ - Constant reference: slope

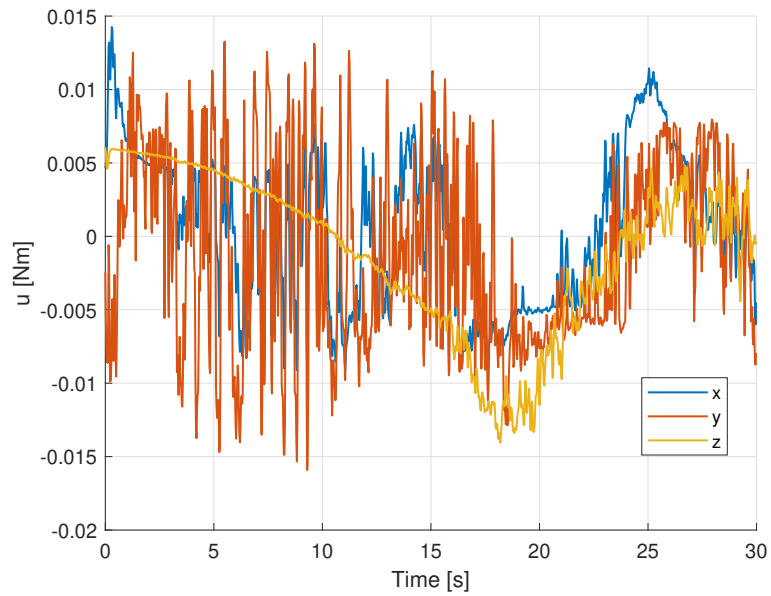


Figure 3.75: Experimental testing - Adaptive BLFOSM - $k = 0.01$ - Constant reference: control input

Figures 3.71 and 3.72 illustrate that the z-component of the angular velocity and the quaternion components reach the desired values within the simulation time using the adaptive approach with a control gain value of 0.01. The sliding surface reaches and maintains zero after approximately 18 seconds, as shown in Figure 3.73. Figure 3.74 depicts the trend of the slope of the sliding variable: the z-component of the slope does not converge to its initial value within the simulation time.

Experimental - $k = 0.5$ - Traditional BLFOSM

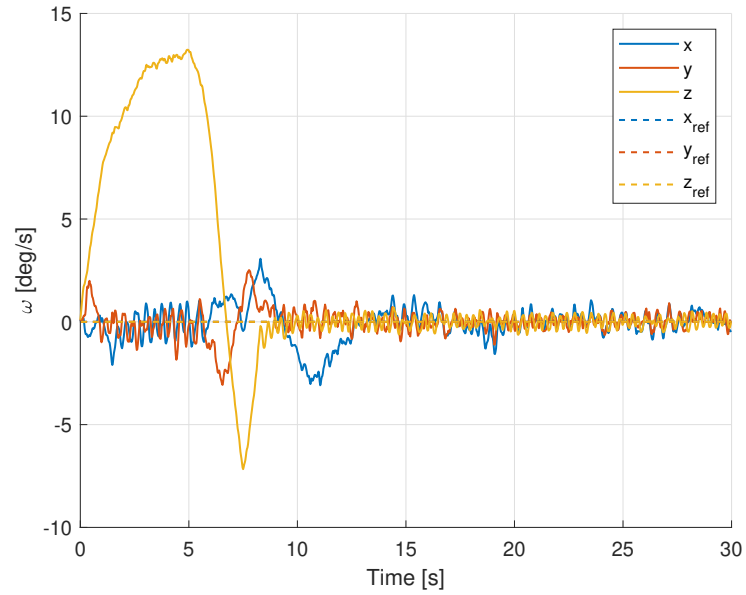


Figure 3.76: Experimental testing - Traditional BLFOSM - $k = 0.5$ - Constant reference: angular velocity

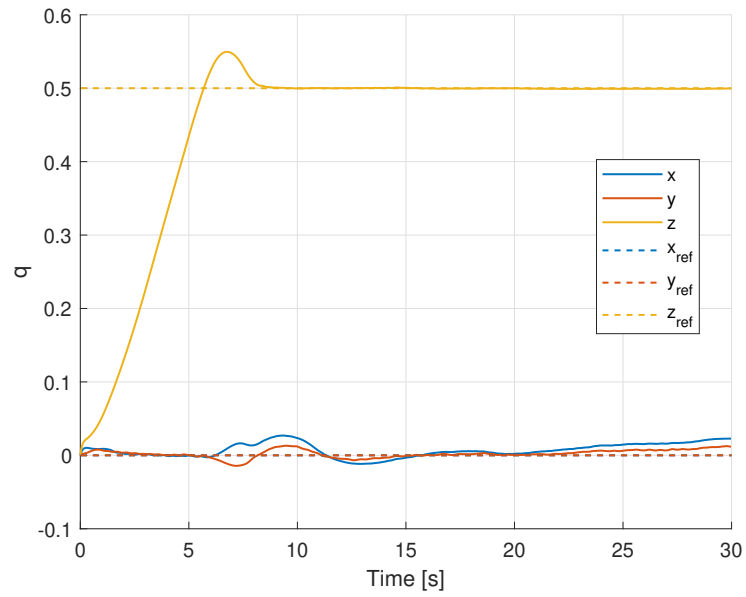


Figure 3.77: Experimental testing - Traditional BLFOSM - $k = 0.5$ - Constant reference: quaternion

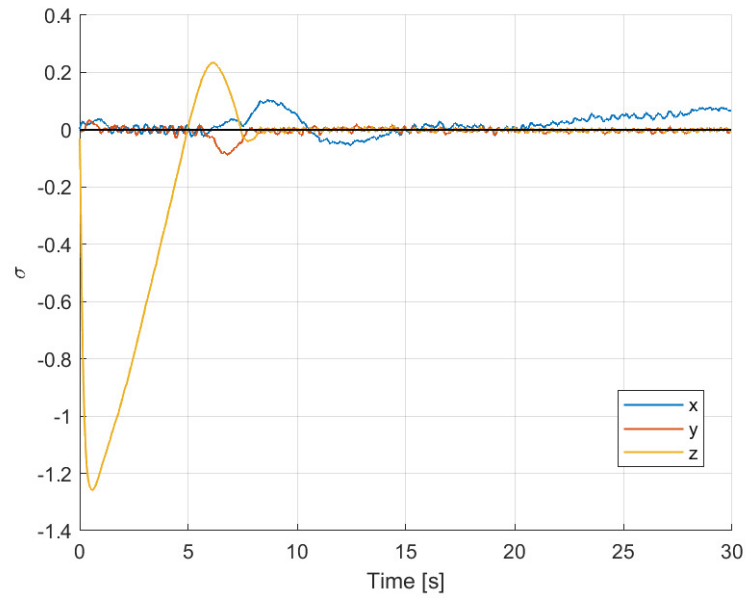


Figure 3.78: Experimental testing - Traditional BLFOSM - $k = 0.5$ - Constant reference: sliding variable

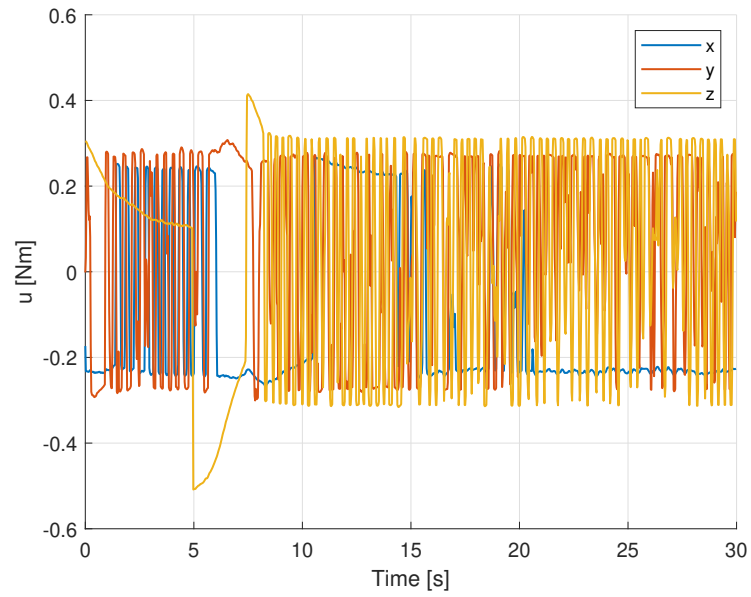


Figure 3.79: Experimental testing - Traditional BLFOSM - $k = 0.5$ - Constant reference: control input

Figures 3.76 and 3.77 illustrate that the z-component of the angular velocity and the quaternion components reach the desired values within the simulation time using the traditional approach with a control gain value of 0.5. The sliding surface reaches and maintains zero after approximately 8 seconds, as shown in Figure 3.78. Figure 3.79 shows that the z-component of the control input exhibits oscillations one order of magnitude wider than with a control gain of 0.01. (Figure 3.70).

Experimental - $k = 0.5$ - Adaptive BLFOSM

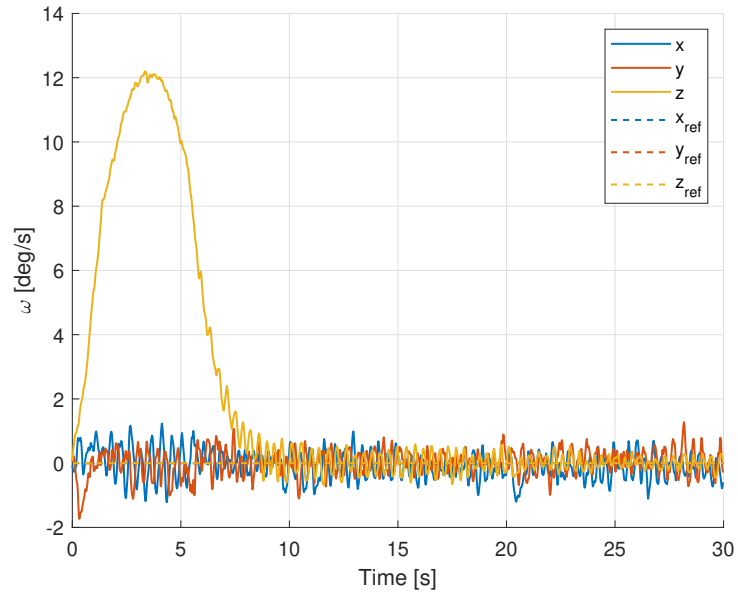


Figure 3.80: Experimental testing - Adaptive BLFOSM - $k = 0.5$ - Constant reference: angular velocity

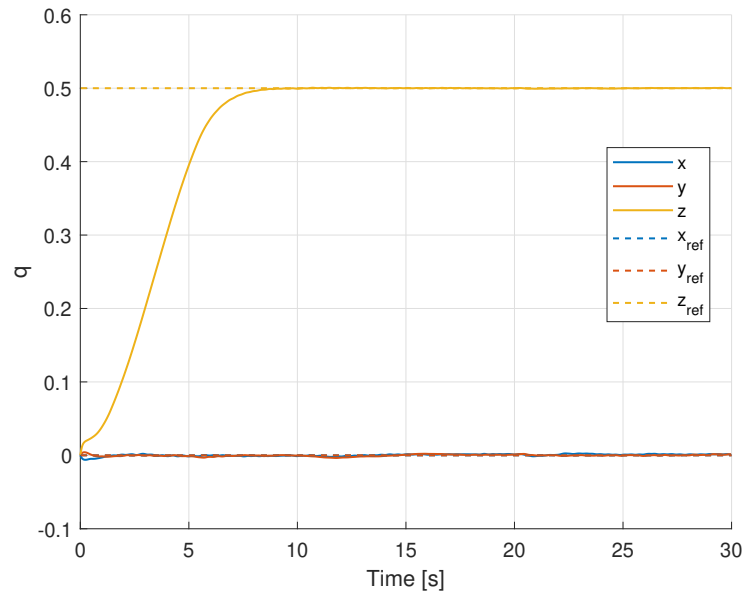


Figure 3.81: Experimental testing - Adaptive BLFOSM - $k = 0.5$ - Constant reference: quaternion

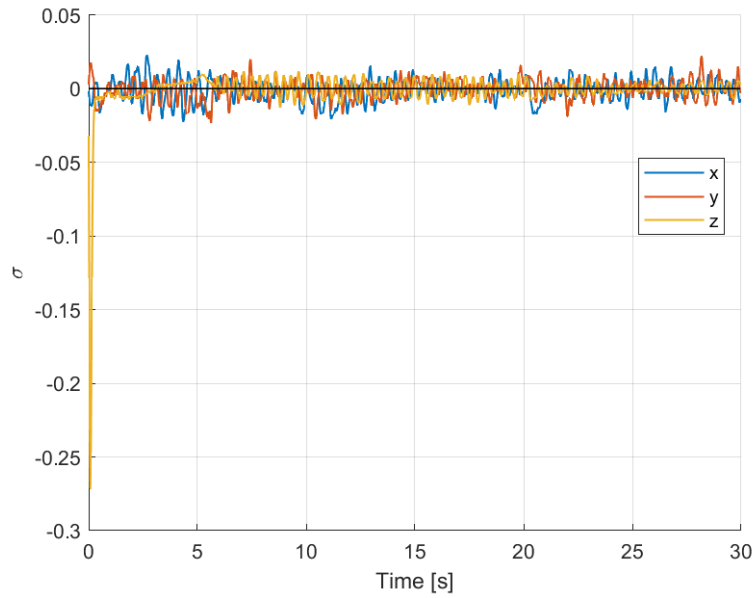


Figure 3.82: Experimental testing - Adaptive BLFOSM - $k = 0.5$ - Constant reference: sliding variable

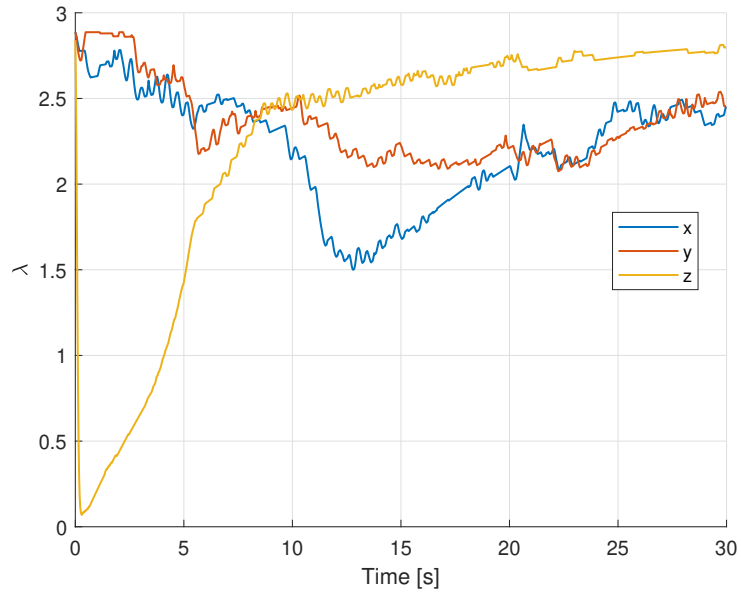


Figure 3.83: Experimental testing - Adaptive BLFOSM - $k = 0.5$ - Constant reference: slope

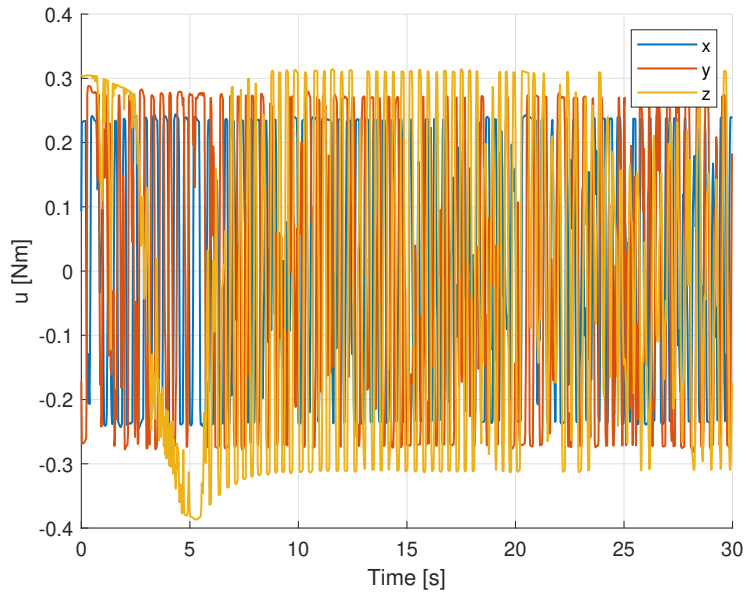


Figure 3.84: Experimental testing - Adaptive BLFOSM - $k = 0.5$ - Constant reference: control input

Figures 3.80 and 3.81 illustrate that the z-component of the angular velocity and the quaternion components reach the desired values within the simulation time using the adaptive approach with a control gain value of 0.5 . The z-component of the sliding surface reaches and maintains zero extremely fast, in less than 1 second, as shown in Figure 3.82. Figure 3.84 shows that the control input exhibits oscillations of the same magnitude as the traditional approach with the same value of control gain (Figure 3.79), and of one order of magnitude wider than the adaptive approach with a control gain of 0.01 (Figure 3.75).

3.2.3 Observations

In the triangular angular velocity reference scenario, the performance of the two approaches shows similar behavior and convergence times. Both methods clearly enhance performance by reducing the convergence time of the sliding surface when using a higher control gain k , although this leads to increased oscillations.

Let us consider the case of constant reference, where the initial attitude differs from the desired attitude, resulting in a non-zero initial error. In this scenario, the enhanced performance in terms of shorter reaching time for the adaptive approach is more evident.

Figures 3.68 and 3.69 show that the 30 seconds of simulation are insufficient for the traditional approach with a control gain of 0.01 to make the Euler parameter and the sliding variable converge to the desired values. Figures 3.72 and 3.73, in contrast, demonstrate that the adaptive approach with $k = 0.01$ is able to achieve convergence of the Euler parameter and the sliding variable to the desired values in a shorter time, well within the 30 seconds of the simulation.

Let us now consider figures 3.77 and 3.78, which depict the trend of the Euler parameters and the sliding variable for the traditional approach with a control gain of $k = 0.5$. It is evident that a higher control gain, facilitates faster convergence for this approach. Specifically, with this control gain, the z-component of the Euler parameter and the sliding variable converge to the desired values in less than 10 seconds. However, the performance of the adaptive approach at a control gain of 0.5 is even better. Figure 3.81 shows that the adaptive approach takes roughly the same time as the traditional approach to make the z-component of the Euler parameter converge to the desired value. In contrast, Figure 3.82 illustrates that the z-component of the sliding variable converges to zero in less than 5 seconds, making the adaptive approach more effective by significantly shortening the reaching time.

To sum up, comparing traditional and adaptive boundary layer first-order sliding mode control for the same value of the control gain k in the constant reference scenario, it is evident that the z-component of the sliding variable converges to zero faster for the the adaptive approach, therefore reducing the reaching time.

Increasing the value of the control gain parameter k results in faster convergence for both approaches in every scenario, although this results in wider oscillations.

Chapter 4

Conclusions

In this thesis, traditional and adaptive boundary layer first-order sliding mode control were analyzed and compared.

Initially, the mathematical equations implemented in the codes were derived for a spacecraft equipped with four control moment gyroscopes arranged in pyramidal configuration.

Subsequently, traditional and adaptive boundary layer first-order sliding mode control were introduced and described.

Following this, the results obtained from mathematical simulation using Matlab and experimental testing conducted on the testbed were presented. The numerical results compared the performance of traditional and adaptive approach in the situation where a desired attitude needed to be achieved starting from a randomly selected initial attitude represented by twenty different random unit quaternions. The experimental tests were performed in two distinct scenarios, each with two different values of the control gain k . The first scenario involved a triangular distribution for the z-component of angular velocity, while the second scenario focused on achieving a desired attitude by rotating the testbed solely about the z-axis.

Both the numerical and experimental results demonstrated that the adaptive approach enhances the performance, shortening the reaching time. Additionally, the experimental results indicated that a higher value of the control gain accelerates convergence for both approaches, although resulting in wider oscillations.

However, the experimental tests were conducted considering rotation exclusively about the z-axis. In the future, it would be interesting to conduct experiments allowing rotations about the three axes.

Bibliography

- [1] Bruce Yost and Sasha Weston. *State-of-the-Art: Small Spacecraft Technology*. Tech. rep. NASA, Feb. 2024 (cit. on p. 1).
- [2] Mauro Mancini. «Adaptive Variable Structure Control System for Attitude Spacecraft Applications». Doctoral Dissertation. Politecnico di Torino, 2023 (cit. on pp. 1, 2, 22, 23, 25).
- [3] Vadim I. Utkin. «Variable structure systems with sliding modes». In: *IEEE Transactions on Automatic Control* 22.2 (1977), pp. 212–222 (cit. on p. 1).
- [4] Hoon Lee and Vadim I. Utkin. «Chattering suppression methods in sliding mode control systems». In: *Annual Reviews in Control* 31.2 (2007), pp. 179–188 (cit. on p. 1).
- [5] Arie Levant. «Introduction to High-Order Sliding Modes». In: *School of Mathematical Sciences*. Tel-Aviv University, Israel, 2003 (cit. on p. 1).
- [6] Mauro Mancini and Elisa Capello. «Adaptive Sliding Mode-based Control System for Flexible Spacecraft». In: *2021 American Control Conference (ACC)*. 2021, pp. 2968–2973 (cit. on p. 1).
- [7] Elisa Capello, Elisabetta Punta, and Giorgio Bartolini. «Simplex Sliding Mode Control Strategies for Spacecraft Rendezvous Maneuvers». In: *IFAC-PapersOnLine* 50.1 (2017). 20th IFAC World Congress, pp. 8496–8501 (cit. on p. 1).
- [8] Daiki Higashiyama, Yasuhiro Shoji, Satoshi Satoh, Ichiro Jikuya, and Katsuhiko Yamada. «Attitude control for spacecraft using pyramid-type variable-speed control moment gyros». In: *Acta Astronautica* 173 (2020), pp. 252–265 (cit. on p. 2).
- [9] Matteo Facchino, Atsushi Totsuka, Elisa Capello, Satoshi Satoh, Giorgio Guglieri, and Katsuhiko Yamada. «Design and Validation of an MPC Controller for CMG-based Testbed». In: *Optimization and Engineering* 24.1 (2023), pp. 185–221 (cit. on p. 2).

- [10] S Durga Nair, P. S Lal Priya, and Arjun Narayanan. «Quaternion Based Sliding Mode Attitude Controller for a Spacecraft with Control Moment Gyros». In: *2018 15th International Workshop on Variable Structure Systems (VSS)*. 2018, pp. 245–250 (cit. on p. 2).
- [11] Antonio D’Ortona. «Design of Sliding Mode Techniques for a CMG-based Testbed Attitude Control System». Master Degree Thesis. Politecnico di Torino, 2021 (cit. on p. 2).
- [12] Domenico Edoardo Sfasciamuro. «Design and implementation of second-order sliding mode controllers for a CMG-based experimental setup.» Master Degree Thesis. Politecnico di Torino, 2022/2023 (cit. on pp. 2, 18).
- [13] Jean-Jacques E. Slotine. «Sliding controller design for non-linear systems». In: *International Journal of Control* 40.2 (1984), pp. 421–434 (cit. on p. 2).
- [14] Yaguang Yang. «Spacecraft attitude determination and control: Quaternion based method». In: *Annual Reviews in Control* 36.2 (2012), pp. 198–219 (cit. on p. 3).
- [15] Elisa Capello. *Lecture notes: Dinamica e controllo di Veicoli Spaziali*. Politecnico di Torino, 2024 (cit. on p. 17).