TÉCNICO LISBOA

Modeling, simulation, analytic linearization and optimal control of a 6 tendon-driven biomimetic eye: a tool for studying human oculomotor control

Bernardo das Chagas e Silva Colaço Dias

Electrical and Computer Engineering

Supervisor(s): Prof. Alexandre Jose Malheiro Bernardino

Prof. Adrianus Johannes Van Opstal

Examination Committee

Chairperson: Prof. Joao Fernando Cardoso Silva Sequeira

Supervisor: Prof. Alexandre Jose Malheiro Bernardino

Member of the Committee: Prof. Pedro Tiago Martins Batista

December 2020

Resumo

O cérebro é um orgão fulcral nos organismos animais. No entanto, ainda não é inteiramente conhecido como é que controla todos os outros sub-sistemas do ser. Um destes sub-sistemas, o oculomotor, será o assunto central desta tese. O objectivo deste trabalho passa por estudar as propriedades do movimento ocular quando alteramos a nossa linha de visão. Foi desenvolvido um simulador e estratégias de controlo óptimo de forma a imitar o sacadas humanas (rápida deslocação ocular).No sistema real os músculos apenas podem exercer força numa direcção, ou seja, apenas puxam o olho, nunca empurram. Podemos ver esta situação, analogamente, como um elástico: se não estiver esticado, estará relaxado (slack). No entanto, até agora, o sistema não incorporou esta limitação, e como tal, tentou-se eliminar este fenómeno. Já que este sistema é não-linear, a fim de aplicar estratégias de controlo, o sistema teve de ser linearizado. Esta não foi uma tarefa trivial, dada a complexidade das equações de dinâmica e do cálculo das forças dependentes da direcção. Adicionalmente, a cinemática do olho é restrita a dois graus de liberdade orientacional (Lei de Listing). Sabendo que o olho é controlado por seis músculos independentes, esta restrição de dois graus de liberdade faz deste um problema extremamente interessante de investigar. Apesar de vários estudos terem modelado o sistema oculomotor, a maioria apenas o faz numa única dimensão e frequentemente apenas estudam a cinemática ou a dinâmica do sistema. Utiliza-se a relação entre a velocidade máxima, duração e amplitude conhecida como *Main Sequence* como métrica, a fim de avaliar a semelhança do modelo desenvolvido com o sistema real. Para completar, trabalhou-se a fim de alcançar resultados precisos acerca da cinemática e da dinâmica deste sistema, considerando todos os graus de liberdade do sistema. Sendo assim, todos os aspectos relacionados com a mudança da nossa linha de visão serão analisados: orientações do olho, perfis de velocidades, torção, e comandos de controlo óptimo.

Palavras-Chave: Linha de visão, Lei de Listing, Movimento sacádico, *Main Sequence*, Slack

Abstract

The brain is a pivotal organ in animal systems. However, it is not yet fully understood how it controls all the other subsystems that compose the full organism. One of these subsystems, the oculomotor system, will be the main focus of this thesis report. It is our objective to study the properties of eye movement when redirecting our line of sight. We expanded the existing simulator and optimal control strategies in order to mimic human saccades (rapid eye movements). However, in the real system muscles can only exert force in one direction, meaning they only pull the eye, never push. We can see this as some sort of rubber band: if it is not stretched, it goes slack. The previous developed system did not incorporate this limitation, and, therefore, here, we attempted to eliminate this phenomenon.

Since we are dealing with a non-linear system, in order to apply control strategies, the model had to be linearized. This is not a trivial task given the complexity of the dynamical equations and eye-direction dependent force computation. Furthermore, eye kinematics are known to be determined by two degrees of orientational freedom (Listings' Law). Given that it is controlled by six independent muscles, this constraint to two degrees of freedom makes it an interesting problem to investigate. Although many studies have modeled the oculomotor system, most of them only do it on one dimension, and often only study either the kinematics or dynamics. We also use the relation between peak velocity, duration and amplitude, known as *Main Sequence* as a metric for evaluating our system emulation of the human eye.

Furthermore, we worked towards having accurate results regarding the kinematics and dynamics of this system, considering all degrees of freedom. So, all aspects of shifting our gaze (eye orientation with respect to the world) will be analysed: eye orientations, velocity profiles, torsion and optimal motor commands.

Keywords: Gaze, Listing's law, Saccadic movement, *Main Sequence*, Slack

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Technology has been evolving at an astounding pace. More and more automation processes in modern society brings about a greater quality of life. As the ultimate automatic machine, humanoid robots have already been researched for several decades. Despite interesting developments and achievements in robot mechanisms and sensors, the most challenging aspect is their control. The natural human brain and body is the most complex system known, and more knowledge about its intricacies could help roboticists to improve the designs and functions of humanoid robots. The degree of complexity of each of its subsystems is nothing to frown upon. For example, how the brain controls a seemingly simple subsystem like the eyes has led neuroscientists to hypothesize that it performs any viewing task near optimally [1]: that is, at the highest speed and accuracy possible and with the lowest effort. Like other subsystems, the eye is controlled by a number of constraints, and expressing such constraints mathematically is not trivial, given the non-linear properties of the system and its underlying mechanics. So far it is still not settled which approach would best explain the observed behaviour of the human oculomotor system.

Conceptual breakthroughs have been made by studying this system in one and two dimensions (horizontal and vertical components) but not many studies have attempted to include the third dimension (cyclotorsion). Here, we will use those earlier studies as an inspirational background, aiming to analyse human eye coordination with all its subtle features in three dimensions (3D) in order to extend the current knowledge regarding the neural control of (3D) gaze, by applying and testing our insights on a humanoid robotic system.

This thesis follows up the progress already achieved in the ORIENT project on previous MSc theses [2–4]. These previous works were focused on a model which assumed the six muscles acted in threee antagonistic pairs. This model was based on a mechanical prototype of a biologically inspired eye (with three motors controlling three pairs of extraocular muscles)[4]. However, this robot has some undesired effects, like vibrations and hysteresis (dead zones).

To better understand the properties of the biomimetic prototype model, an eye simulator was de-

veloped. In [3] they were able to mimic the kinematics and dynamics of the human eye, minimizing accuracy error, duration and energy expenditure, using optimal control. In [2] it was verified, by using optimal feedback control, that by introducing signal dependent noise in the control signals and sensory measurements of the simulator, the energy cost was no longer needed. It is important to note that these results were achieved with a system linearized around the straight-ahead viewing direction, which is a crude approximation for the actual nonlinear behavior of the system at eccentric orientations.

For this thesis, our model was based on a more recently developed prototype (one with six independent motors), and, as such, we separated each individual muscle, giving the eye 3 extra degrees of freedom. Contrary to the model developed before, this new model is closer to the actual human eye anatomy, where muscles have different length, making it an asymmetrical model [5]. Another problem is that the eye muscles can not push, which means it can only exert force in one direction. This requires the muscles to be tense to move the eye. So we wish to eliminate the possibility of the muscles going slack (having a smaller length than the initial one). Additionally, in this work, the dynamics of the system can be linearized at any equilibrium point in the eye's visual range, allowing the control to be better suited for saccades not starting from the origin. Finally, we extended the dynamical analysis of our simulator to a coupled eye-head orienting system, which will allow future studies to also include combined eye-head movements in three dimensions.

1.2 Objectives

The objective of this thesis will be to identify,test and study the optimal control of the eye saccadic system (and its direct consequences), developed in the scope of the ORIENT project, a partnership between professor Alexandre Bernardino and professor Van Opstal's Donder Centre for Neuroscience at Radboud University Nijmegen, Netherlands. The earlier developed prototype robot was built in order to study the kinematic and dynamical properties of the human eye in three dimensions, by applying open [3] and closed loop [2] optimal control for rapid saccades.

Our main goals are to separately control all 6 extraocular muscles (EOM), similar to the actual system, model the head dynamics, linearize the system (so that we can have an approximation of the dynamics of the system at any equilibrium point), and implement a control strategy that provides accurate results even with the asymmetry of the extra-ocular muscles. The last objective of this thesis, is to eliminate slack of the muscles, in an attempt to be even closer to human behavior.

1.3 Thesis Outline

This thesis is divided in 6 chapters: Introduction (1), Background (2), Methods (3), Results (6), Conclusions (7), and the Appendix (8).

In the Background chapter, all the information necessary to be able to understand the work that was done is provided. The anatomy of the eye is explained so that it is understandable to the reader how the simulator was constructed. Also, rules that the eye is known to follow and that help us defend our results

are shown, along with a description of the kinematics and dynamics studied in humans and monkeys. Some mathematical approaches used in the overall implementation are also presented.

The Methods chapter describes how we approached the challenge, what problems were faced, how they were solved and justifications for why such strategies were chosen. First, the simulator is explained, then the way we approximate this non-linear system to a linear one is shown and finally we present the implemented optimal control and cost functions.

In Results, the validity of the simulator and the linearization are presented. Also, the implemented control strategies are documented and their respective results are shown.

Finally, in Conclusions we evaluate the work, draw some inferences from this work and make recommendations for future studies.

Chapter 2

Background

2.1 Human eye

The human eye is a complex optic and mechanical system: it has six extraocular muscles that control its movements and therefore the gaze (orientation of its line of sight). These muscles are, for the right eye: Lateral Rectus (LR) which rotates the eye rightward, the Superior Rectus (SR), which rotates the eye up and anticlockwise, and the Superior Oblique (SO) whose function is to rotate the eyes down and anticlockwise. For each of these muscles there is a corresponding muscle with which it forms a pair, respectively: the Medial Rectus (MR), in charge of rotating the eye leftward, the Inferior Rectus (IR) which rotates the eye down and clockwise and the Inferior Oblique (IO) that rotates it up and clockwise. All muscles are relevant for the task at hand and must be considered when developing an accurate robotic implementation of the human visual-motor system. Each of these muscles will be driven by an individual motor. For example, a pure upward movement will require simultaneous activation of the SR and IO muscles (see figure 2.2), in order to compensate their torsional actions. Additional complexities arise when it is considered that the pulling directions of the muscles depend on the eye's orientation, and that this effect depends on the actual path taken by the muscle over the globe.

2.1.1 Saccades

The sensory input to the eye is organized in a special way in the retina. It harbours a network of light-sensitive cells (Cones and Rods). The Rods are distributed more or less homogeneously across the retina and are specialized in low-light vision and motion detection. Cones are color sensitive, have an extremely high density in the centre (around 1 degree visual angle) called the fovea, allowing for high-resolution vision [6]. Their density decreases approximately as $1/r^2$ with distance r from the fovea, making the peripheral retina imprecise for target identification and localization.

The saccadic eye movement system moves the fovea as quickly and accurately as possible between targets to make sure that the desired orientation of the eye is achieved, and targets may be visually analysed with high spatial resolution.

It has been shown that during this movement, the eye cannot perceive clear images [7], and that

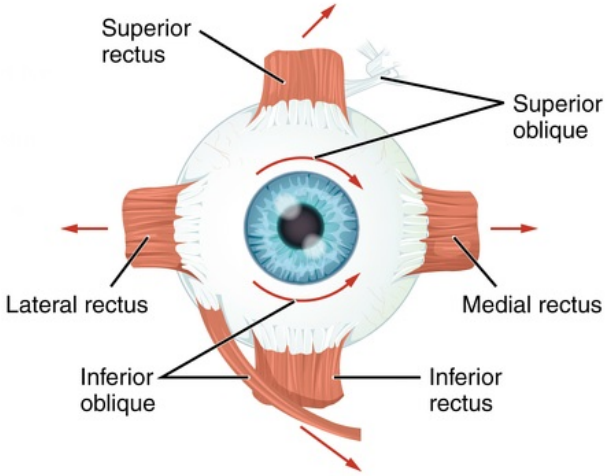


Figure 2.1: Anterior view of the right eye.

therefore the oculomotor system, has to move the eye at high speeds, to keep movement duration to a minimum, thus maximizing visual ”reward”. Peak velocities of large saccades, in monkeys reach up to $\approx$ 1200 deg/s [8], while in humans, they reach $\approx$ 500-700 deg/s [9].

Since we can only see sharp and clearly within 1 or 2 degrees around our fixation point , we cannot identify an object just by making one saccade. When scanning an image, we constantly change our fixation point, with saccades. By doing so, the visual system gathers incremental information about the visual environment. The fixation time is the interval between each saccade, and its about 200 to 300 ms on average, while the actual saccade duration varies between 20 to 100 ms, depending on its amplitude [10]. Typically, humans make about 3-4 saccades per second to scan the environment.

Besides saccades, the eye can also make smooth pursuit, which allows it to follow a moving object. The eye then makes a smooth trajectory that aims to hold the moving object on the fovea, by keeping its retinal slip to a minimum. The brain can achieve this by generating an eye movement with a similar velocity as the object. Although smooth pursuit is an interesting topic in itself, which involves nontrivial predictive and cognitive neural mechanisms, it will not be considered further in the current project. Instead, the focus of this work is on the saccadic gaze (orientation of the line of sight after an eye displacement) movement.

2.1.2 Donders' Law

Because the six extraocular muscles are organized in three nearly orthogonal antagonistic pairs, and given that the eye cannot translate in the socket, it has in theory three degrees of rotational freedom. However, pointing our gaze in a specific direction requires only two degrees of freedom. It was recognized, by Franciscus Cornelius Donders and peers, already in 1876, that for any given eye orientation, the amount of cyclotorsion is always the same, regardless the path followed by the eye to reach that orientation. This experimental fact has been called Donders' law [11].

More precisely, this law states that, for any single direction, the eye will always have the same unique orientation in three dimensions. Let's consider this example: if the target is in a downward and left direction, the eyes' orientation will be the same no matter the path they take: whether they go down first and then left or left first and then down, or direct along some oblique or curved path. There are different explanations for the origin of this law: either it is imposed entirely by the ocular muscle mechanics, or it is the result of a neural strategy that aims to optimize certain perceptual and/or motor performance criteria [12].

2.1.3 Listing's Law

Listing's law for the eye may be regarded as a specification of Donders' law. While Donders stated that the orientation of the eye for a specific gaze direction was fixed, Listing went one step further, by realizing that any orientation of a rigid body can be achieved by a hypothetical single-axis rotation from some fixed origin, so-called primary position.

Listing's law then states that the torsional component (x component of the 3D rotation axis, see figure 2.2 for conventions) of the rotation axes for all eye orientations is zero.

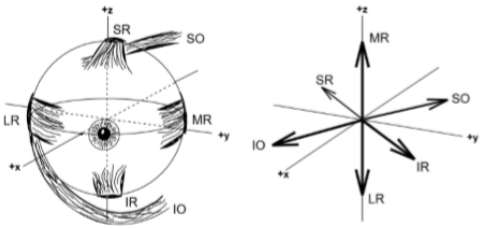


Figure 2.2: Six extraocular muscles, and their approximate pulling directions from the eye's primary position (the +x direction, [1,0,0]) in a right-handed Cartesian head-centred reference frame. The arrows correspond to the unit vectors of each muscle's rotation axis. E.g., the IR muscle pulls the eye in the +y (i.e., downward) and +x (i.e. clockwise cyclotorsion) direction. Also, a vertical upward movement ([0, -1, 0], requires the joint action of the SR and IO muscles.

As a result, all eye orientation axes are confined to a plane - the so called Listing's plane (LP) - for which the torsional component is zero: in figure 2.2, LP corresponds to the head-fixed yz plane.

However, it should be realized that this rule only applies when the head is stationary and upright, with the eyes looking at infinity. Indeed, with the head moving, or for fixating nearby targets, or even when the head is tilted due to gravity, Listing's law is invalid (although Donders' law still applies in most cases). This is a strong indication that (1) the eye has indeed 3 degrees of freedom and (2) that Listing's law is not only due to eye mechanics but results from a neural control principle.

The 3D Kinematics are studied by analysing data with respect to the Listing's laws. In order to do this, it is required to extract the instantaneous 3D orientation of the eye during movement, for which we use rotation matrices and quaternions. In the stationary head case, standard deviation of the torsional component of the eye's 3D orientation axis should fluctuate between 0.6 and 0.8 degrees (in primates). [13].

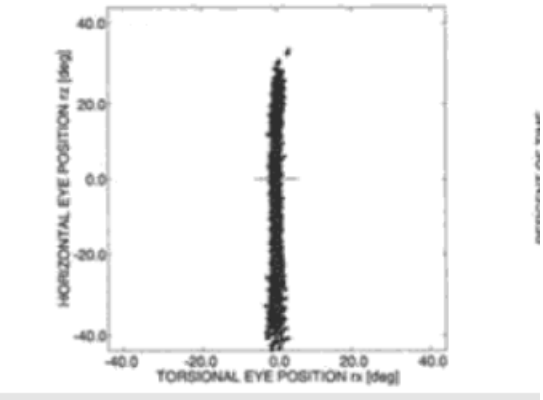


Figure 2.3: Side view (torsion-horizontal) of Listing's plane for 3500 monkey eye orientations. The standard deviation of the torsional component is about 0.6 degrees. From [13].

2.1.4 Main Sequence

The main sequence is a well-known relationship between duration, velocity and amplitude of head-fixed saccadic movements. It has been shown that for a given amplitude, there is an optimal eye trajectory, with an optimal duration and a generic velocity profile [1]. The duration of a saccade increases linearly with amplitude, and with increasing amplitude the peak velocity increases as well, until it saturates (figure 2.4 (A)).

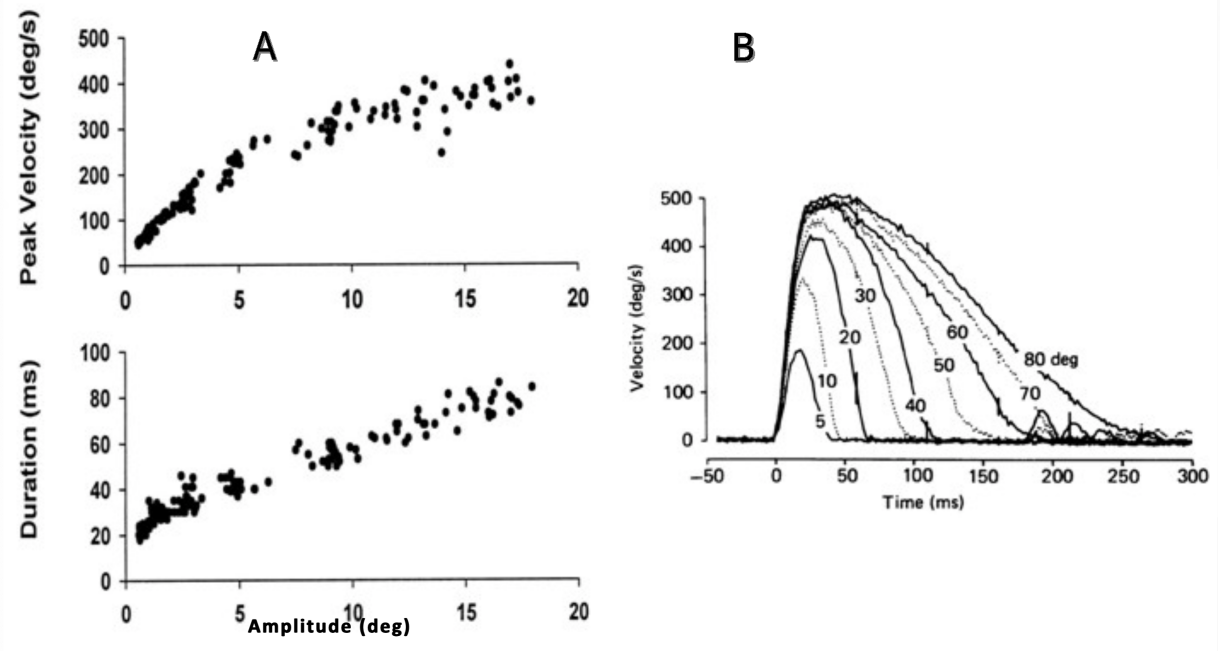


Figure 2.4: (A) Relation between amplitude of the saccade and its peak velocity and duration. As observed peak velocity increases with saccade magnitude until it saturates around 450 deg/s. The relation between duration and amplitude is also depicted here, with duration increasing almost linearly with magnitude. From [14]. (B) Increasing skewness of velocity profiles of different saccades with increasing amplitudes. From [15].

The acceleration phase of saccades is reached around the same time for all amplitude (around 25 ms) [16]. It thus follows that the increase in saccade duration is mainly due to the longer deceleration

phase of the saccadic eye movement. As such, saccades of small amplitudes have a nearly symmetric velocity profile whereas bigger amplitude saccades have a positively skewed velocity profile and this skewness increases with saccade duration as seen in figure 2.4 (B).

If we assume a linear system, the peak velocity should increase linearly with amplitude and movement duration and skewness should be constant. Therefore, the main sequence properties unveil a non-linearity in the saccadic system. This behaviour is thought to have evolved as a strategy that optimizes the trade-off between accuracy and speed, in the presence of internal noise/uncertainty in the system [1].

2.2 Rotation Representations

It is possible to depict a rigid body's orientation through different representations. The most commonly used representations, that are deemed relevant for this work, are presented in this section.

2.2.1 Rotation matrices

A rotation matrix (**R**) is a 3x3 matrix that is used to perform a rotation of a vector in three dimensional Euclidean space. This type of matrices obey the following set of rules :

$$det(\mathbf{R}) = 1$$
$$\mathbf{R}^T = \mathbf{R}^{-1}$$

They belong to $SO(3)$ (3D rotation group) which is a Lie group and for which we can define a Lie Algebra (tangent space of a Lie group at the identity, see section 2.2 for more on this).

In order to rotate a vector (**x**) we merely need to multiply it by a rotation matrix that represents the desired movement, giving us the rotated vector **y**.

$$\mathbf{y} = \mathbf{Rx} \tag{2.1}$$

It is also possible to rotate a vector several times. This is called a composed rotation and can be achieved (for finite rotations) by the multiplication of rotation matrices. Note, however, that this operation is non-commutative, which means that the order of the multiplication matters (Infinitesimal rotations, however, commute).

2.2.2 Quaternions

Like rotation matrices, quaternions are a way to represent orientations. This type of representation have four components: one scalar (q_0), and a vector that entails complex numbers, representing the axis of rotation. In general, a quaternion is represented as:

$$q = q_0 + q_x i + q_y j + q_z k \tag{2.2}$$

where **q** = (q_x, q_y, q_z) is a vector of real numbers and **I** = (i, j, k) is a complex vector that satisfies the following condition:

$$i^2 = j^2 = k^2 = ijk = -1 \tag{2.3}$$

Quaternions, when used to describe a rotation of magnitude θ, about an axis **n** = (n_x, n_y, n_z) can be represented as

$$q = \cos(\frac{\theta}{2}) + (n_x i + n_y j + n_z k)\sin(\frac{\theta}{2}) \tag{2.4}$$

From this, one can state that quaternions that depict rotations have a unit norm (unit quaternions). This norm can be defined as

$$|q| = \sqrt{q_0^2 + q_x^2 + q_y^2 + q_z^2} = 1 \tag{2.5}$$

A relevant formula would be

$$\dot{q} = \omega \frac{q}{2} \tag{2.6}$$

where ω is the angular velocity represented by a scalar with zero scalar part. This equation allows to link the derivative of the quaternion (that expresses orientation) and its angular velocity. It is possible to transform a quaternion into a rotation matrix according to [17] as

$$\mathbf{R} = \begin{bmatrix} 1 - 2(q_y^2 + q_z^2) & 2(q_x q_y - q_0 q_z) & 2(q_x q_z + q_0 q_y) \\ 2(q_x q_y + q_0 q_z) & 1 - 2(q_x^2 + q_z^2) & 2(q_y q_z - q_0 q_x) \\ 2(q_x q_z - q_0 q_y) & 2(q_y q_z + q_0 q_x) & 1 - 2(q_x^2 + q_y^2) \end{bmatrix} \tag{2.7}$$

2.2.3 Rotation Vectors

One common notation among neuroscientists to depict orientations are rotation vectors. This type of rotation representation is intertwined with quaternions, since rotation vectors can be obtained by their equivalent unit quaternion. As seen before, the scalar part of a quaternion is redundant because it's information can be extrated from the vector part of said quaternion. Hence, rotation vectors are a more compact way to describe orientations. A rotation vector **r** can be mathematically related to a quaternion **q** by

$$\mathbf{r} = \frac{\mathbf{q}}{q_0} = \mathbf{n}\tan(\frac{\theta}{2}) \tag{2.8}$$

where **n** is the axis of rotation and θ is the magnitude of this same rotation. Rotation vector units are given in half-radians. To convert to degrees, we have:

$$\theta_{x,y,z} = \frac{360}{\pi}\arctan(r_{x,y,z}) \tag{2.9}$$

In this work compound rotations are made and computed resorting to rotation matrices, but results are shown in rotation vectors. More details on the transformation rules for quaternions and rotation vectors can be found in [3].

2.2.4 Angle-axis representation

This representation allows to depict the 3D orientation of a rigid body by a three-dimensional vector, composed of an angle θ and a unit vector $\hat{\mathbf{n}}$ which represents the rotation axis in a chosen reference frame. The angle represents the magnitude of rotation around the axis. Also, using Rodrigues' rotational formula (equation 2.10), this representation allows a transformation that rotates 3-dimensional vectors.

$$v_{rot} = v\cos(\theta) + (\hat{\mathbf{n}} \times v)\sin(\theta) + \hat{\mathbf{n}}(\hat{\mathbf{n}} \cdot v)(1 - \cos(\theta)) \tag{2.10}$$

Where $\mathbf{v}$ and $\mathbf{v_{rot}}$ are vectors $\in \mathbb{R}^3$. With angle-axis, any rotation, or sequence of rotations can be taken as a pure rotation about a fixed axis, for a rigid body in 3-dimensional space. This notation only needs three scalar values to represent a rotation.

$$\mathbf{w} = \theta\hat{\mathbf{n}} = \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \tag{2.11}$$

The composition of a sequence of two rotations given by $\mathbf{w}_1 = \theta_1\hat{\mathbf{n}_1}$ and $\mathbf{w}_2 = \theta_2\hat{\mathbf{n}_2}$, where the second is applied before the first one is given by $\mathbf{w}_3 = \theta_3\hat{\mathbf{n}_3}$, according to [18].

$$\theta_3 = 2\arccos(\cos\frac{\theta_1}{2}\cos\frac{\theta_2}{2} - \sin\frac{\theta_1}{2}\sin\frac{\theta_2}{2}\hat{\mathbf{n}}_1\hat{\mathbf{n}}_2) \tag{2.12}$$

$$\hat{\mathbf{n}}_3 = \frac{\sin\frac{\theta_1}{2}\cos\frac{\theta_2}{2}\hat{\mathbf{n}}_1 + \cos\frac{\theta_1}{2}\sin\frac{\theta_2}{2}\hat{\mathbf{n}}_2 + \sin\frac{\theta_1}{2}\sin\frac{\theta_2}{2}\hat{\mathbf{n}}_1 \times \hat{\mathbf{n}}_2}{\sin\frac{\theta_3}{2}} \tag{2.13}$$

2.2.5 Exponential map

Exponential mapping can be used to map angle-axis ($\mathbb{R}^3$) to rotation matrices. This gives us a compact way to depict rotations ($\mathbb{SO}^3$) as such:

$$\mathbf{R} = \exp(\mathbf{w}^\wedge) \tag{2.14}$$

where $\mathbf{w}^\wedge$ is the skew-symmetric matrix obtained from the angle-axis representation $\mathbf{w}$ (which is called the "hat operator") :

$$\mathbf{w}^\wedge = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix} \tag{2.15}$$

It is also possible to perform the inverse operation, i.e. obtain the original $3D$ vector from its respective skew-symmetric matrix. It's called the "vee operation", written as follows

$$\mathbf{w}^{\vee} = \begin{bmatrix} 0 & w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix}^{\vee} = \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \tag{2.16}$$

From equation 2.14 it also follows that:

$$\mathbf{w}^{\wedge} = \log(\mathbf{R}) \tag{2.17}$$

Note that the notation on this report is the one used in [19], where the hat and *vee* operators are introduced. More details on this notation are provided in the section below (2.3).

2.3 Variable Notation

In this document, the Multibody Dynamics Notation from Eindhoven University [19] was adopted, since we want to express different variables in different reference frames. This notation simplifies this task. Consider that orientations can be expressed in angle-axis for the eye in the head reference frame as:

$${}^{h}\mathbf{r}_{e} = \begin{bmatrix} {}^{h}r_{e}^{x} \\ {}^{h}r_{e}^{y} \\ {}^{h}r_{e}^{z} \end{bmatrix} \tag{2.18}$$

and for the head in the world frame

$${}^{w}\mathbf{r}_{h} = \begin{bmatrix} {}^{w}r_{h}^{x} \\ {}^{w}r_{h}^{y} \\ {}^{w}r_{h}^{z} \end{bmatrix} \tag{2.19}$$

The top-left index represents the reference frame in which the variable is represented and the bottom-right index indicates the object to which the variable is related.

The angular velocity of the eye with respect to the head, expressed in the head reference frame (also called the right trivialized velocity of the eye with respect to the head) is

$${}^{h}\boldsymbol{\omega}_{h,e} = \begin{bmatrix} {}^{h}\omega_{h,e}^{x} \\ {}^{h}\omega_{h,e}^{y} \\ {}^{h}\omega_{h,e}^{z} \end{bmatrix} \tag{2.20}$$

and the angular velocity of the head with respect to the world, expressed in the world reference frame is

$$ {}^{w}\boldsymbol{\omega}_{w,h} = \begin{bmatrix} {}^{w}\omega^{x}_{w,h} \\ {}^{w}\omega^{y}_{w,h} \\ {}^{w}\omega^{z}_{w,h} \end{bmatrix} \tag{2.21} $$

We can also define the left trivialized velocity for the eye with respect to the head ${}^{e}\boldsymbol{w}_{h,e}$, which is in the body frame. The right trivialized would be in the inertial frame.

Assuming that the angular velocity is approximately constant during a brief sampling interval Δt, it is possible to approximate the rotation matrix that expresses the orientation of the eye at time $t+1$ with respect to the eye orientation at time t, using the exponential map

$$ {}^{e(t)}\mathbf{R}_{e(t+1)} = \exp(\Delta t\, {}^{e}\boldsymbol{\omega}^{\wedge}_{h,e}) \tag{2.22} $$

Conversely, from the rotation matrix, we can also obtain the skew-symmetric matrix

$$ \Delta t\, {}^{e}\boldsymbol{\omega}^{\wedge}_{h,e} = \log({}^{e(t)}\mathbf{R}_{e(t+1)}) \tag{2.23} $$

To finalize the introduction to the notation used in this report, it's important to note that the indices on direction vectors and forces (which tell us in which frame the variable is written) appear on the bottom left. For instance, ${}_{e}\mathbf{v}_{i}$ represents the direction of a force applied on the eye, with respect to the eye and ${}_{e}\mathbf{f}_{i}$ represents a muscle force applied on the eye, with respect to the eye.

2.4 Lie Groups

A Lie group $\mathcal{G}$ is a differentiable (or smooth) manifold, where its elements satisfy a set of axioms that rule the group.

A smooth manifold is a topological space locally resembling Euclidean space near each point. This could be visualized as a curved, smooth surface without spikes. This smoothness implies there is a unique tangent space at each point.

A group $(\mathcal{G}, \circ)$ is a set, $\mathcal{G}$, with a composition operation, $\circ$, that, for elements $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \in \mathcal{G}$, satisfies the following axioms:

$$ \text{Closure under } '\circ' : \quad \mathcal{X} \circ \mathcal{Y} \in \mathcal{G} \tag{2.24} $$

$$ \text{Identity } \mathcal{E} : \quad \mathcal{E} \circ \mathcal{X} = \mathcal{X} \circ \mathcal{E} = \mathcal{X} \tag{2.25} $$

$$ \text{Inverse } \mathcal{X}^{-1} : \quad \mathcal{X}^{-1} \circ \mathcal{X} = \mathcal{X} \circ \mathcal{X}^{-1} = \mathcal{E} \tag{2.26} $$

$$ \text{Associativity} : \quad (\mathcal{X} \circ \mathcal{Y}) \circ \mathcal{Z} = \mathcal{X} \circ (\mathcal{Y} \circ \mathcal{Z}) \tag{2.27} $$

In a Lie group, the composition operator and inversion are smooth functions.

A Lie Group entails the existence of a Lie algebra, which can be defined as the tangent space of the Lie group at the identity. The exponential and logarithmic map allow to transfer elements between the Lie group and the Lie algebra (the exponential maps an element of the tangent space to the group and the logarithm works the other way around). Consider two elements of group $\mathcal{G}$, g_1 and g_2. A tangent space at g_1 is represented by $T_{g1}\mathcal{G}$. A point $a \in T_{g1}\mathcal{G}$ can be mapped to the group, resulting in $g_2 = g_1 \exp(a^{\wedge})$.

Lie groups also allow us to combine groups. Given two Lie groups $\mathcal{F}$ and $\mathcal{G}$ with operators $\bullet$ and $\circ$, respectively, the direct product of the groups ($\mathcal{F} \times \mathcal{G}$) is a Lie group as well. This resulting group consists of the ordered pairs (f, g) where $f \in \mathcal{F}$ and $g \in \mathcal{G}$, given by component-wise operation

$$(h_1, g_1)(h_2, g_2) = (h_1 \bullet h_2, g_1 \circ g2) \tag{2.28}$$

This topic is relevant in the task at hand since for instance, in the linearization, we will need to map points to a tangent plane, in which Lie Algebra will play an important role. Another use of this subject is the need for a combined group consisting of rotation matrices and vectors, as seen in section 3.4.

2.5 Neural Control

As mentioned before, we wish to mimic human oculomotor behaviour with a biomimetic robotic eye. In order to achieve it, we have to account for the mechanical constraints of the human eye. Furthermore, we also wish to emulate the dynamical properties. Some studies have suggested that the brain uses an optimal control strategy to command the eyes, since it has to deal with different types of constraints (internal noise, low spatial resolution in the retinal periphery and the need to prevent overshooting the goal) [20]. These studies have proposed that these constraints are best met with an accuracy/velocity trade-off, which explains the emergence of the nonlinear main sequence. The straightness of the trajectories also complies with this idea, since the eye follows the shortest path to the target. It has been suggested that the desired optimal trajectory is encoded in the midbrain Superior Colliculus [21].

Optimal control minimizes a cost function, that can penalize deviations of several factors, for example: accuracy, duration, energy, stability, or even force. Some studies claim that given the poor resolution of the retina at eccentric positions and the lack of vision during the trajectory, the most important minimization relationship is accuracy/duration. However, given that there is signal dependent noise related to the brain signals that command the saccade, sit has been suggested that the standard deviation of the noise is proportional to the mean of the control signal and therefore, for saccades with large amplitudes (consequentially, higher velocities) there will be a bigger variation in the end goal orientation. Harris and Wolpert [1] proposed that in order to minimize the mean absolute error of saccades over time, in the presence of signal-dependent noise, the brain should lower the saccade speed to have optimal average accuracy, i.e., to get the orientation of the eye on the target.

The ocular plant can be well approximated by a low-pass second-order system, like a mass-spring-

damper system. It has muscle elastic forces, acting like a spring and also it's fat tissues create damping in its orbit (dynamic damping) and some static friction. It has been observed that the system's transfer function resembles a second-order overdamped characteristic with one zero and two poles. If we assume the perfect saccade is depicted as a step output, then given the output and the transfer function, it follows that the input is a combination of three different components: a pulse, a slide and a step. Empirical data from neural recordings is in accordance with this, as seen in figure 2.5.

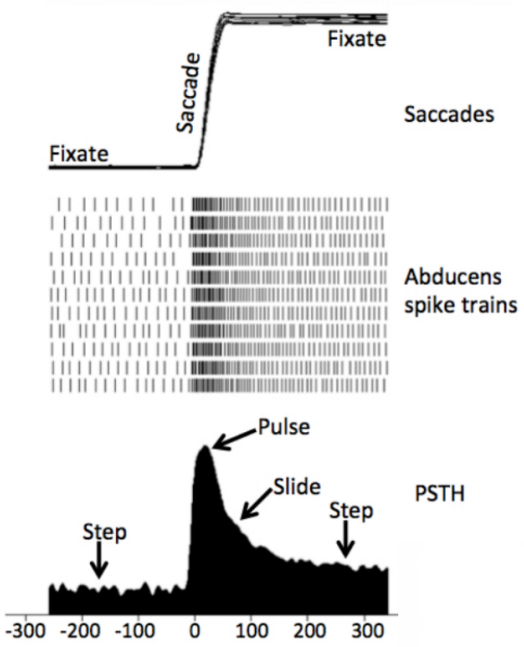


Figure 2.5: Depiction of the 3 stages of the oculomotor control signal and the corresponding stages of a saccade, as seen in [10]. Abducens is the oculomotor nucleus for the lateral rectus eye muscle, and each tick represents a single action potential of the recorded neuron. PSTH stands for peri-saccade time histogram (all time traces have been re-aligned with the saccade onset). The bottom axis represents time in ms.

Regarding the minimization of the cost function, it is relevant to mention the work of Shadmehr, [22], because we will be implementing his duration cost term. Shadmehr studied different functions to toll the duration of the saccade, and he concluded that the one that emulated the human eye better was an hyperbolic cost term. This is because other functions like a exponential discount, underestimate the reward value for longer periods of time. Hyperbolic cost, however, fits the empirical data much better, being for seconds or years. Although the parameters for this cost term are different for every species and individual (depends on the "impulsiveness" of each), the shape of the function remains the same, and so far, in this field, has been the best suggestion regarding the duration penalty.

2.6 State of the art

There are many models of the saccadic eye movement system, both for animals and humans, using computer simulators and mechanical models [1],[23],[24]. However, most of these studies have been

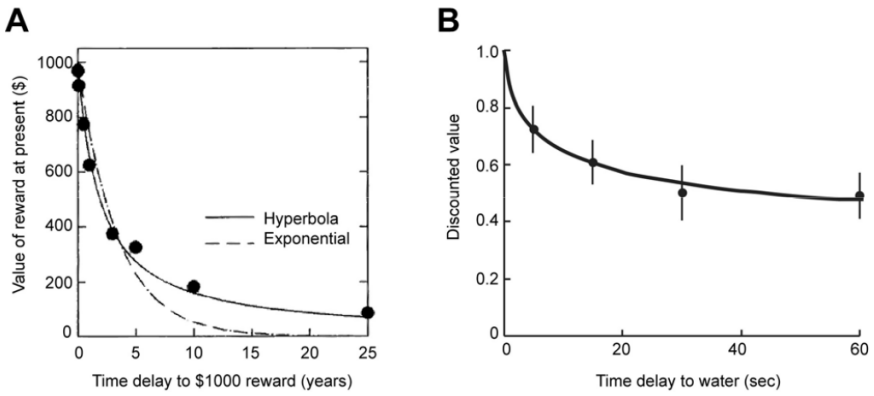


Figure 2.6: temporal discount functions comparison between exponential, hyperbolic (A) and real human data (B).

confined to one (typically horizontal) dimension (1D). Only few studies incorporated models in 3 dimensions [25–28].

A recent study on physical modeling of the eye [29], developed a robotic prototype of the eye with three degrees of rotational freedom that obeys Listing's law. The authors were able to build a tendon driven prototype (but only with 4 tendons, figure 2.7) focusing on showing that the compliance with Listing's law can be achieved using particular geometry (functionally incorporating the EOM pulley system, in order to mechanically fix Listing's plane) and placement of four extraocular muscles (figure 2.8). However, they did not study the dynamics nor the control. Also, it has been shown that the actual eye can move out of Listing's plane, so mechanically constraining it to follow Listing's law seems limiting, considering the objectives of this thesis.

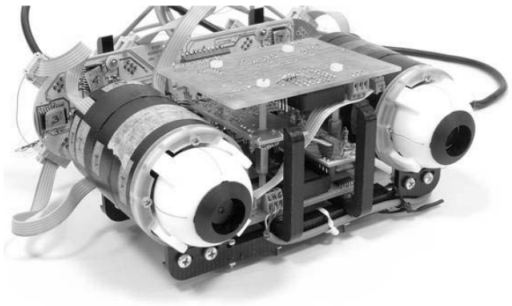


Figure 2.7: Mac-Eye prototype

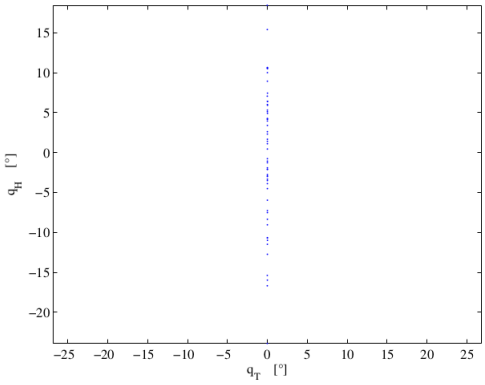


Figure 2.8: Mac-Eye's Listing's plane with no noise or perturbations

An anatomically accurate model of the oculomotor system was described in [28]. This work was based on the work from the founder of this field, DA Robinson in 1975 [30]. From MRI scans, they were able to build 3D non-linear geometric models of the EOM and investigate how they influence eye movement. They studied different computational models (as shown in figure 2.9) of the muscles in order to generate correct eye kinematics, fitting it to experimental data without emphasis on understanding the

underlying control or the Listing's law compliance.

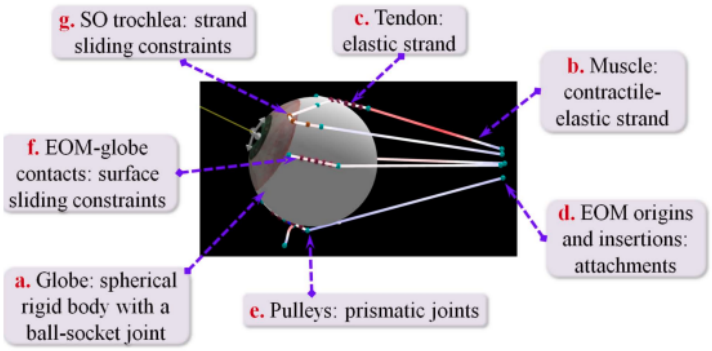


Figure 2.9: Bio-mechanical model implemented in [28] with mechanical constraints.

On the other hand, [3] considered an optimal control strategy on a simplified 3D model, with a cost function that included accuracy, movement duration, deviation from Listing's plane, total energy expenditure, and the total amount of force generated by the muscles. They showed that proper combinations of these cost terms could explain the asymmetric shapes of saccade velocity profiles, the saturation of peak eye velocity with saccade amplitude, the near-linear increase of saccade duration with amplitude, straight eye-movement trajectories, component coupling in which velocity components are matched in duration and shape, and Listing's law. Later, [2], demonstrated that the energy cost could be abandoned, if the model included signal-dependent noise in the controllers and the Listing constraint could be omitted when replaced by the force term.

In this project, it is aimed to extend this modeling strategy to a 3D robotic eye model that is asymmetrical, with six muscles set-up in a way that approaches the actual positions in the human eye. From this, we will study both the kinematics and dynamics.

Chapter 3

Modeling

3.1 Model Design

The first step to take in order to build an eye-head simulator is studying the real system. As already mentioned, this system is highly complex which makes it very difficult to pass from natural human behaviour and movement into mathematical equations capable of describing the kinematics and dynamics.

In this thesis we build an eye-head simulator but will concentrate our efforts in the eye saccadic control. Thus, we create a biologically inspired model for eye kinematics, dynamics and actuator, considering possible robotics implementation. Furthermore, we design a simplified head model, to be further developed in future work. The eye was designed as a ball joint, subject to the Newton's rigid body equations, and Euler's equation of motion with six tendon-driven actuators mimicking the human eye muscle kinematics. We also define the system parameters of inertia, stiffness and damping to replicate closely the time constants and overdamped characteristics of the human eye.

3.1.1 Eye model

Several psychophysical tests have been performed on monkeys and humans in order to figure out how the eye behaves kinematically, i.e., how its position, velocity and acceleration evolve along the saccadic eye movement. Our objective is to create and develop a simulator able to closely emulate the behaviour of gaze during saccadic eye movements.

A noteworthy fact is that the model in this work is not the exact dimension of the human eye. Actually, our model is developed after a robotic prototype (figure 3.2) developed to study biomimetic eye movements. Our robot model is around three times bigger than a real eyeball. Furthermore, relative to the eye's inertia tensor, ours is around 1000 times higher than the approximate inertia tensor built to represent the human eye, calculated using its mass and dimensions. Also, the head inertia was scaled to match the eye's proportions. Although our artificial eye's parameters are different from the human eye, they are configured to roughly replicate its main dynamics characteristics in terms of time constants and damping. These are important features to develop controllers that can give us insights on human oculomotor control.

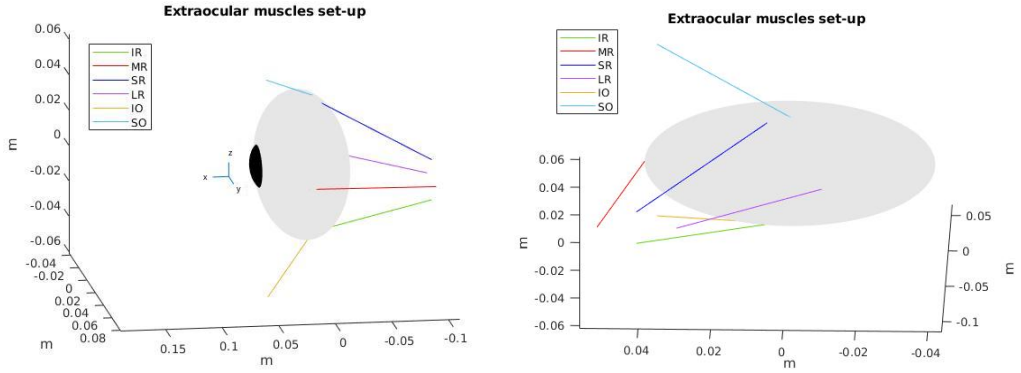


Figure 3.1: Insertions points on the right eye and head of the extraocular muscles (left and rear view). The head is not shown, but it would encompass the eye and EOM. It has the same center of rotation as the eye.

Figure 3.2: Prototype on which our eye model is based.

From table 8.1 it is possible to notice a slight difference in lengths between agonist and antagonists, with the biggest dissimilarity being between the horizontal muscles. We have a ratio of 0.84 between Medial Rectus and Lateral Rectus. Moreover, the cable pulling directions are also not symmetric about the eye, since both the insertion points on the head are skewed towards one side (figure 3.1, right side picture).

The eye and head are connected through six extraocular muscles. Thus, one point of each muscle is attached to the head and the other end to the eye (fig 3.1). The extraocular muscles are roughly arranged according to the anatomy of the human eye [5] (their geometric characteristics are shown in figure 3.1 and table 8.1). In reality, these muscles are not actually completely straight. It happens that in the human eye, the contact locus between the muscle and the eye is not really a point, but a curve. For now, we are simplifying this fact by considering it a mere point (the last point of contact). In the real system, the elastic component of the muscles make them go stiffer or relax by a certain amount, rotating the eye. Since it's very difficult to develop a prototype of a spring that changes stiffness, our robot artificial eye implements this by making the change in length of the muscle (making it exert a different amount of force), move the eye. In our model, the cables are wrapped around a spindle attached to a motor. The changes in length occur because of the rotation of the motors, winding the cable around the spindle. From the spindle, these cables are routed through a via point, which corresponds to the

Extraocular muscles setup						
	IR	SR	MR	LR	IO	SO
Insertion(eye)						
x(mm)	-2	-2	7.7	7.7	-11.2	-11.8
y(mm)	40	-40	0	0	38.3	-38.1
z(mm)	-1.4	0.8	39.3	-39.3	-1.4	-1.2
Insertion(head)						
x(mm)	-100.1	-100.1	-100.1	-100.1	45	45
y(mm)	7.8	-14.9	-3.5	-3.5	62	-62
z(mm)	40.7	40.7	51.6	29.6	37.5	37.5
Initial length(mm)	111.5	108.8	108.6	127.9	72.3	72.7

Table 3.1: Insertion points, on the eye (measured starting from its center) and the head in Cartesian coordinates taken from the prototype on which this model was based. The radius of the eye is approximately 40 mm.

insertion point on the head.

For the eye, in order to compute the torque, we need the EOM (extraocular muscles) starting point in the skull ,${}^{h}S_{0i}$, (the via point in figure 3.3) and the insertion points on the eye, ${}^{e}Q_{0i}$. The first one is in head frame and the second in the eye reference frame. The 0 subindex refers to the eye at resting position near the straight ahead gaze direction, when all angles are at zero rotation angle. These are represented in figure 3.1.

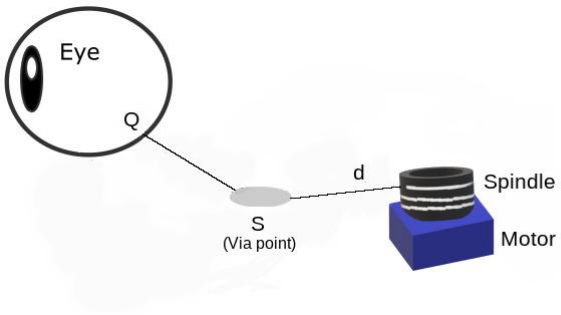


Figure 3.3: Diagram showing simplified version of the real prototype. d is a fixed value, representing the length between the insertion point in the head and the motor that makes the string extend.

Next, muscle length is computed. It is determined by the magnitude of vectorial difference between the fixed cranial insertion point of the muscles (in head reference frame) and the final contact point on the eye, which are in the eye reference frame. However, to compute the initial length, both points have to be expressed in the same reference frame. So we transform the insertion points on the eye to the head reference frame, as given by equation 3.1.

$$ {}^{h}\mathbf{Q}_i = {}^{h}\mathbf{R}_e \, {}^{e}\mathbf{Q}_{0i} \tag{3.1} $$

where ${}^{h}\mathbf{R}_e$ is the rotation matrix that converts the eye reference frame into the head reference frame and

${}^{e}\mathbf{Q}_{0i}$ is the insertion point on the eye for a specific muscle (insertion(eye) in table 8.1).

In order to compute the force applied by a specific muscle, one needs to know both its initial length and its length when contracting or relaxing. Initial length is determined by:

$$l_{0i} = d_i + \left\| {}^{h}\mathbf{S}_{0i} - {}^{h}\mathbf{Q}_i \right\| \tag{3.2}$$

d_i is the distance between the spindle and the via point of the muscle and S_{0i} is the muscle's insertion point on the head (Insertion (head) in table 8.1).

The actual length of each muscle, at any time, is calculated according to

$$l_i = d_i + \left\| {}^{h}\mathbf{S}_{0i} - {}^{h}\mathbf{Q}_i \right\| + r\theta_i \tag{3.3}$$

where r is the spindle radius and θ_i is the angular position of the motor with respect to the angular position at resting position (looking straight ahead).

We can compute the direction of the force, expressed in the eye reference frame, since we are interested in the force exerted on the eye

$${}_{e}\vec{\phi}_i = \frac{{}^{e}\mathbf{R}_h \, {}^{h}\mathbf{S}_{0i} - {}^{e}\mathbf{Q}_{0i}}{\left\| {}^{e}\mathbf{R}_h \, {}^{h}\mathbf{S}_{0i} - {}^{e}\mathbf{Q}_{0i} \right\|} \tag{3.4}$$

The force is then computed, by Hooke's law, as such:

$${}_{e}\mathbf{f}_i = \frac{k}{l_{0i}}(l_i - l_{0i}) \, {}_{e}\vec{\boldsymbol{\phi}}_{\boldsymbol{i}} \tag{3.5}$$

where k is the spring constant (Young's modulus) of the eye muscles and l_{0i} is the muscle length when θ_i is zero (unstretched length). Also, to make sure the initial length is always smaller than the current length, the difference between these values can not be lower than zero, making the muscle stay taut. As it happens, the initial resting orientation is not actually obtained with motor angles to zero, but instead with some pre-tension computed to make the eye look straight ahead.

The torques are then obtained by the cross product between the final contact point on the eye and the forces.

$${}_{e}\boldsymbol{\tau}_i = {}^{e}\mathbf{Q}_i \times {}_{e}\mathbf{f}_i \tag{3.6}$$

Total elastic torque applied by the muscles on the eye will be

$${}_{e}\boldsymbol{\tau}_k = \sum_{i=1}^{6} {}_{e}\boldsymbol{\tau}_i \tag{3.7}$$

In addition, the torque resulting from dynamic friction forces is

$${}_{e}\boldsymbol{\tau}_d = -\mathbf{D}_{eye} \, {}^{e}\boldsymbol{\omega}_{h,e} \tag{3.8}$$

where $\mathbf{D}_{eye}$ quantifies the eye's damping matrix and ${}^{e}\boldsymbol{\omega}_{h,e}$ is the angular velocity of the eye with respect to the head, depicted in the eye reference frame. From equations 3.7 and 3.8, one can write the resulting

torque as

$$ {}_e\boldsymbol{\tau}_{eye} = {}_e\boldsymbol{\tau}_k + {}_e\boldsymbol{\tau}_d \tag{3.9} $$

The input signals for the eye are the angles by which the motors rotate the spindles.

$$ \mathbf{u}_{eye} = \begin{bmatrix} \theta_{IR} \\ \theta_{MR} \\ \theta_{SR} \\ \theta_{LR} \\ \theta_{IO} \\ \theta_{SO} \end{bmatrix} \tag{3.10} $$

3.1.2 Head model

Since the focus of this work is the eye saccadic control, we simplify the head model as a sphere that rotates arbitrarily around the same point as the eye. This sphere is coupled in the neck through elastic springs and generalized friction. In order to rotate said sphere, generic torques can be directly applied.

The total torque applied on the head also depends on the coupling of the eye in the head and the head on the neck. There is also an external torque that has to be accounted for due to the neuronal signal that is sent from the brain to innervate the neck muscles.

According to Newton's third law, the eye's influence on the head is equal in magnitude, but with opposite direction, to the total eye torque, expressed in head frame:

$$ {}_h\boldsymbol{\tau}_{eye-head} = -({}_h\boldsymbol{\tau}_k + {}_h\boldsymbol{\tau}_d) \tag{3.11} $$

Regarding the damping torque resulting from the frictional forces between the head and the neck, it may be represented as

$$ {}_h\boldsymbol{\tau}_{dhead} = -\mathbf{D}_{head}({}^h\boldsymbol{\omega}_{w,h}) \tag{3.12} $$

Since the neck is fixed, it's angular velocity is not required to be included in the equation above.

The connection between head and neck is assumed to be represented by a linear torsional spring, and as such, we have

$$ {}_h\boldsymbol{\tau}_{khead} = (-\mathbf{K}_{head} \log ({}^w\mathbf{R}_h))^\vee \tag{3.13} $$

K_{neck} is the elasticity matrix for the neck.

Furthermore, we have the $\boldsymbol{\tau}_{ext}$, torque created by external forces, i.e., by brain signals.

So, the total torque experienced by the head is given by:

$$\boldsymbol{\tau}_{head} = {}_h\boldsymbol{\tau}_{eye-head} + {}_h\boldsymbol{\tau}_{dhead} + {}_h\boldsymbol{\tau}_{khead} + {}_h\boldsymbol{\tau}_{ext} \tag{3.14}$$

where

$${}_h\boldsymbol{\tau}_{ext} = \mathbf{u}_{head} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} \tag{3.15}$$

The value c_i describes the torque generated at each axis by the neural control.

3.2 Global State and State Equations

Before getting into the state equations, it is important to introduce the rotational dynamics considered for the system.

The dynamics of a rigid body (in this particular case, the eye), rotating in 3D on the object frame, with respect to another frame, according to [31], can be represented by

$${}^e\boldsymbol{\alpha}_{h,e} = {}_e\mathbf{I}_e{}^{-1}({}_e\boldsymbol{\tau}_{eye} - {}^e\boldsymbol{\omega}_{h,e} \times {}_e\mathbf{I}_e\,{}^e\boldsymbol{\omega}_{h,e}) \tag{3.16}$$

And for the head

$${}^h\boldsymbol{\alpha}_{w,h} = {}_h\mathbf{I}_h{}^{-1}({}_h\boldsymbol{\tau}_{head} - {}^h\boldsymbol{\omega}_{w,h} \times {}_h\mathbf{I}_h\,{}^h\boldsymbol{\omega}_{w,h}) \tag{3.17}$$

Note that ${}_e\mathbf{I}_e$ and ${}_h\mathbf{I}_h$ are the eye and head's inertia tensors, respectively, in their own reference frames.

3.2.1 Global State

Our state variables were chosen carefully, to lower the complexity of managing different reference frames, but at the same time, to allow the state equations to fully represent the system dynamics.

The selected set of global state variables is thus represented by the 4-tuple

$$\mathbf{x} = \left\{{}^w\mathbf{R}_e, {}^w\mathbf{R}_h, {}^e\boldsymbol{\omega}_{w,e}, {}^h\boldsymbol{\omega}_{w,h}\right\} \tag{3.18}$$

The rotation matrices determine the orientation difference compared to an initial three dimensional orientation. Using equation (2.28) we can join the groups of rotations and vectors together, to achieve a new group $\mathcal{S}$, which is an element of the Lie group defined as

$$\mathcal{S} := SO(3) \times SO(3) \times \mathbb{R}^3 \times \mathbb{R}^3 \tag{3.19}$$

From the state, it's possible to express the dynamics of the system according to four equations, one

for each state variable.

$$ {}^{w}\dot{\mathbf{R}}_{e} = {}^{w}\mathbf{R}_{e}\,{}^{e}\boldsymbol{\omega}^{\wedge}_{w,e} \tag{3.20} $$

$$ {}^{w}\dot{\mathbf{R}}_{h} = {}^{w}\mathbf{R}_{h}\,{}^{h}\boldsymbol{\omega}^{\wedge}_{w,h} \tag{3.21} $$

$$ {}^{e}\dot{\boldsymbol{\omega}}_{w,e} = {}_{e}I_{e}{}^{-1}({}_{e}\boldsymbol{\tau}_{eye} - {}^{e}\boldsymbol{\omega}_{w,e} \times {}_{e}I_{e}\,{}^{e}\boldsymbol{\omega}_{w,e}) \tag{3.22} $$

$$ {}^{h}\dot{\boldsymbol{\omega}}_{w,h} = {}_{h}I_{h}{}^{-1}({}_{h}\boldsymbol{\tau}_{head} - {}^{h}\boldsymbol{\omega}_{w,h} \times {}_{h}I_{h}\,{}^{h}\boldsymbol{\omega}_{w,h}) \tag{3.23} $$

Equations (3.20) and (3.21) are taken from [19] and equations (3.22) and (3.23) directly follow from (3.16) and (3.17), respectively. Being functions of the state variables, these equations can be written as

$$ \begin{aligned} {}^{w}\dot{\mathbf{R}}_{e} &= f_1({}^{w}\mathbf{R}_{e}, {}^{e}\boldsymbol{\omega}_{w,e}) \\ {}^{w}\dot{\mathbf{R}}_{h} &= f_2({}^{w}\mathbf{R}_{h}, {}^{h}\boldsymbol{\omega}_{w,h}) \\ {}^{e}\dot{\boldsymbol{\omega}}_{w,e} &= f_3({}^{w}\mathbf{R}_{e}, {}^{w}\mathbf{R}_{h}, {}^{e}\boldsymbol{\omega}_{w,e}, {}^{h}\boldsymbol{\omega}_{w,h}, u_{eye}) \\ {}^{h}\dot{\boldsymbol{\omega}}_{w,h} &= f_4({}^{w}\mathbf{R}_{e}, {}^{w}\mathbf{R}_{h}, {}^{e}\boldsymbol{\omega}_{w,e}, {}^{h}\boldsymbol{\omega}_{w,h}, u_{eye}, u_{head}) \end{aligned} \tag{3.24} $$

It is relevant to explain some less intuitive dependencies. The derivative of the eye's angular velocity depends on the head orientation and velocity since we need it to compute eye torques. To compute the eye's elastic torque, we require insertion points in the same reference so we can compute the initial length of each muscle (3.1). For damping torque on the eye, the angular velocity of the head is needed to compute the relative velocity between eye and head (3.8). Similarly, there is also influence of the eye in the head dynamics. The forces produced by the head on the eye are the same as the ones produced by the eye, applied on the head, with opposite direction.

3.3 Simulator

Since we already explained the modeling and the equations that rule the system, the only thing left to explain is the functioning of the simulator.

We define an initial state, comprised of initial orientations and velocities. Initial values are given to the commands for the eye and head, and simulation duration and sampling time are defined. With these values we can compute the number of sampling instants.

The initial state alone allows us to compute the initial length of the muscles (3.2), the force direction (3.4), the damping torque for the eye (3.8), and both the elastic and damping torques for the head (equations 3.12 and 3.13, respectively).

With eye commands, we are able to compute the current length of each muscle (3.3), and therefore the complete elastic torque (3.9). Given that we already have the eye's damping torque, it is possible to calculate the full eye torque (3.9). After this, eye acceleration (3.21) and the eye's influence on the head (3.11) are attainable.

Having the input commands for the head (3.15, the commands are seen as torques and applied

directly), it is then feasible to compute the head's full torque (3.14), and afterwards, its acceleration (3.22). Notice that, initially, both eye and head commands are considered null and that the update of these variables is a task for the controller.

Since the dynamics are computed, it is possible to integrate them over a small time interval, getting us the next state.

The angular velocities are integrated from the angular accelerations, assuming that they are approximately constant during the integration interval.

$$ {}^{h}\boldsymbol{\omega}_{w,h}(t+1) = {}^{h}\boldsymbol{\omega}_{w,h}(t) + \Delta t \, {}^{h}\boldsymbol{\alpha}_{w,h}(t) \tag{3.25} $$

$$ {}^{e}\boldsymbol{\omega}_{h,e}(t+1) = {}^{e}\boldsymbol{\omega}_{h,e}(t) + \Delta t \, {}^{e}\boldsymbol{\alpha}_{h,e}(t) \tag{3.26} $$

The time evolution of the reference frames can be approximated, considering constant angular velocity ω during an infinitesimal time step. We can define the orientation for the eye (in the head reference frame), head(in the world frame) and gaze(in the the world frame, as well) according to:

$$ {}^{h}\mathbf{R}_{e(t+1)} = {}^{h}\mathbf{R}_{e(t)} \, {}^{e(t)}\mathbf{R}_{e(t+1)} \tag{3.27} $$

$$ {}^{w}\mathbf{R}_{h(t+1)} = {}^{w}\mathbf{R}_{h(t)} \, {}^{h(t)}\mathbf{R}_{h(t+1)} \tag{3.28} $$

$$ {}^{w}\mathbf{R}_{e} = {}^{w}\mathbf{R}_{h} \, {}^{h}\mathbf{R}_{e} \tag{3.29} $$

The simulator saves all these values and goes back to the earlier computations, given the new state. It is possible to return the state, forces, torques or accelerations at any sampling instant.

Chapter 4

Linearization

In order to control the system, a linear approximation of the kinematics is convenient, since a highly non-linear system like the one modeled is not trivial to control. These non-linearities (e.g. force computation, rotation matrices) also proved hard to differentiate, needing a fair amount of mathematical research and understanding. The linearization performed was based on an analytical perturbation, in which an infinitesimal perturbation was applied to the system around an equilibrium point to get a response from the linearized system that approximates the non-linear response [32]. The mathematical results are non-trivial, and will be shown in Section 8.

4.1 Local State and Dynamics

In this work, a variation-based method was used, in order to linearize the non-linear dynamics of our robotic system. This kind of linearization represents dynamics of the variation as a time dependant linear system and it's globally valid [32]. Consider the dynamics of a state y that is given by the differential equation

$$\frac{\partial y}{\partial t} = g(y) \tag{4.1}$$

We can define an equilibrium state $\bar{y}$ as a point where $g(\bar{y}) = 0$. In our case, we represent an equilibrium point by

$$\bar{\mathbf{x}} = \left\{ {}^{w}\bar{\mathbf{R}}_{e}, {}^{w}\bar{\mathbf{R}}_{h}, {}^{e}\bar{\boldsymbol{\omega}}_{w,e}, {}^{h}\bar{\boldsymbol{\omega}}_{w,h} \right\} \tag{4.2}$$

which satisfies the condition

$$f(\bar{\mathbf{x}}, \bar{\mathbf{u}}) = 0 \tag{4.3}$$

where $\bar{\mathbf{u}}$ is the input command that makes the dynamic equation be zero. According to equations (3.20-3.23) this happens when

$$\boldsymbol{\tau}_{eye} = 0$$
$$\boldsymbol{\tau}_{head} = 0$$
$${}^{e}\bar{\boldsymbol{\omega}}_{w,e} = 0$$
$${}^{h}\bar{\boldsymbol{\omega}}_{w,h} = 0$$

Let us define a local state around the equilibrium point $\bar{\mathbf{x}}$ as

$$\tilde{\mathbf{x}} = \left\{ {}^{w}\tilde{\mathbf{R}}_{e}, {}^{w}\tilde{\mathbf{R}}_{h}, {}^{e}\tilde{\boldsymbol{\omega}}_{w,e}, {}^{h}\tilde{\boldsymbol{\omega}}_{w,h} \right\} \tag{4.4}$$

with

$${}^{w}\tilde{\mathbf{R}}_{e} = {}^{w}\bar{\mathbf{R}}_{e}^{T}\,{}^{w}\mathbf{R}_{e} \tag{4.5}$$
$${}^{w}\tilde{\mathbf{R}}_{h} = {}^{w}\bar{\mathbf{R}}_{h}^{T}\,{}^{w}\mathbf{R}_{h} \tag{4.6}$$
$${}^{e}\tilde{\boldsymbol{\omega}}_{w,e} = {}^{e}\boldsymbol{\omega}_{w,e} - {}^{e}\bar{\boldsymbol{\omega}}_{w,e} \tag{4.7}$$
$${}^{h}\tilde{\boldsymbol{\omega}}_{w,h} = {}^{h}\boldsymbol{\omega}_{w,h} - {}^{e}\bar{\boldsymbol{\omega}}_{w,h} \tag{4.8}$$

where $\tilde{\mathbf{R}}$ is the rotation between equilibrium orientation $\bar{\mathbf{R}}$ and the actual orientation $\mathbf{R}$. Using exponential notation, we can represent $\tilde{\mathbf{R}}$ with a rotational perturbation $\boldsymbol{\eta}$ as

$$\tilde{\mathbf{R}}(t) = \exp(\boldsymbol{\eta}^{\wedge}) \tag{4.9}$$

At this point, let's consider the following simplification in the notation

$${}^{x}\boldsymbol{\omega}_{w,x} \equiv \boldsymbol{\omega}_{x} \tag{4.10}$$
$${}^{x}\tilde{\boldsymbol{\omega}}_{w,x} \equiv \delta\boldsymbol{\omega}_{x} \tag{4.11}$$

Adapted from [32], an infinitesimal variation, with respect to a reference $\bar{\mathbf{R}}(t) \in SO(3)$ is given by

$$\delta\mathbf{R}(t) = \left.\frac{d}{d\epsilon}\right|_{\epsilon=0} \bar{\mathbf{R}}(t)\exp(\epsilon\boldsymbol{\eta}^{\wedge}) = \bar{\mathbf{R}}(t)\boldsymbol{\eta}^{\wedge}(t) \tag{4.12}$$

where ϵ is a small rotation around an axis .

From [33], the corresponding infinitesimal change in body angular velocities can be given as:

$$\delta\boldsymbol{\omega}(t) = \boldsymbol{\omega}^{\wedge}(t)\boldsymbol{\eta}(t) + \dot{\boldsymbol{\eta}}(t) \tag{4.13}$$

Let's also consider the following exponential coordinates

$$\boldsymbol{\eta}_e = \log({}^w\tilde{\mathbf{R}}_e)^\vee \tag{4.14}$$

$$\boldsymbol{\eta}_h = \log({}^w\tilde{\mathbf{R}}_h)^\vee \tag{4.15}$$

These two equalities allow for elements of our local state to be mapped into the Lie group of the global state and vice-versa.

Our local state can thus, be represented in local coordinates, as :

$$\xi = \begin{bmatrix} \boldsymbol{\eta_e} \\ \boldsymbol{\eta_h} \\ \delta\boldsymbol{\omega}_e \\ \delta\boldsymbol{\omega}_h \end{bmatrix} \tag{4.16}$$

This state is a vector in the tangent space, as represented in figure 4.1, of the Lie group $\mathcal{S}$ (containing our global state), considered at some point $x \in \mathcal{S}$, represented by

$$T_x\mathcal{S} := \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \tag{4.17}$$

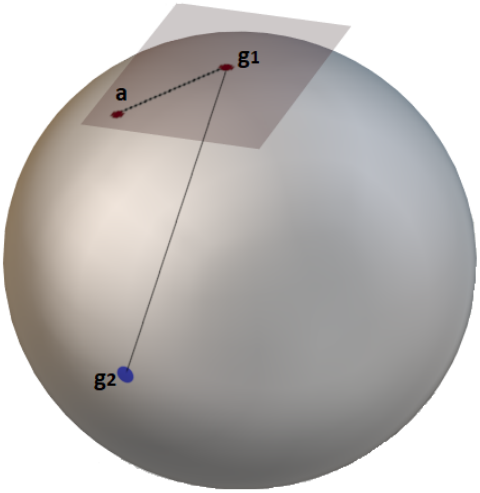


Figure 4.1: Lie group transfer between the group and the tangent plane. Here, g_1 corresponds to $\bar{\mathbf{x}}$. a symbolizes $\boldsymbol{\xi}$ and can be mapped from the tangent space to the group, using exponential mapping, resulting in $g_2 = \mathbf{x}$ (4.2).

From (4.13), we can write our first two local state equations as:

$$\dot{\boldsymbol{\eta}}_e = -\boldsymbol{\omega}_e^\wedge \boldsymbol{\eta}_e + \delta\boldsymbol{\omega}_e \tag{4.18}$$

$$\dot{\boldsymbol{\eta}}_h = -\boldsymbol{\omega}_h^\wedge \boldsymbol{\eta}_h + \delta\boldsymbol{\omega}_h \tag{4.19}$$

and the complete equations (in matrix form) in the following way:

$$\dot{\boldsymbol{\xi}} = \frac{d}{dt}\begin{bmatrix} \boldsymbol{\eta}_e \\ \boldsymbol{\eta}_h \\ \delta\boldsymbol{\omega}_e \\ \delta\boldsymbol{\omega}_h \end{bmatrix} = \begin{bmatrix} -\boldsymbol{\omega}_e^{\wedge}\boldsymbol{\eta}_e + \delta\boldsymbol{\omega}_e \\ -\boldsymbol{\omega}_h^{\wedge}\boldsymbol{\eta}_h + \delta\boldsymbol{\omega}_h \\ {}_eI_e^{-1}\left({}_e\boldsymbol{\tau}_e - \boldsymbol{\omega}_e \times {}_eI_e\boldsymbol{\omega}_e\right) \\ {}_hI_h^{-1}\left({}_h\boldsymbol{\tau}_h - \boldsymbol{\omega}_h \times {}_hI_h\boldsymbol{\omega}_h\right) \end{bmatrix} \tag{4.20}$$

Knowing that in equilibrium, angular velocities are zero, we can show $\dot{\boldsymbol{\xi}}$ as a function of the local state $\boldsymbol{\xi}$ and command inputs $\mathbf{u}$.

$$\dot{\boldsymbol{\xi}} = f(\boldsymbol{\xi}, \mathbf{u}) \tag{4.21}$$

4.2 Jacobians

In order to have a linear state space model, a Jacobian linearization is performed. This is computed around an equilibrium point. The objective is to get a linear approximation of the non-linear state, which should be valid for small perturbations.

Dynamics of a general variation ($\delta_{\mathbf{x}}(t) = f(x, u)$) may be defined as

$$\dot{\delta}_{\mathbf{x}}(t) = f\left(\bar{\mathbf{x}} + \delta_{\mathbf{x}}(t), \bar{\mathbf{u}} + \delta_{\mathbf{u}}(t)\right) \tag{4.22}$$

Doing a Taylor's expansion on the right side, we have

$$\dot{\delta}_{\mathbf{x}}(t) \approx f(\bar{\mathbf{x}}, \bar{\mathbf{u}}) + \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\substack{\mathbf{x}=\bar{\mathbf{x}} \\ \mathbf{u}=\bar{\mathbf{u}}}} \delta_{\mathbf{x}}(t) + \left.\frac{\partial f}{\partial \mathbf{u}}\right|_{\substack{\mathbf{x}=\bar{\mathbf{x}} \\ \mathbf{u}=\bar{\mathbf{u}}}} \delta_{\mathbf{u}}(t) + \epsilon(\left\| \delta_{\mathbf{x},\mathbf{u}}(t) - \delta_{\mathbf{x},\mathbf{u}}(t)|_{\substack{\mathbf{x}=\bar{\mathbf{x}} \\ \mathbf{u}=\bar{\mathbf{u}}}} \right\|^2) \tag{4.23}$$

If it is linearized around an equilibrium point ($f(\bar{\mathbf{x}}, \bar{\mathbf{u}}) = 0$), and neglecting higher order terms, we get

$$\dot{\delta}_{\mathbf{x}}(t) \approx \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\substack{\mathbf{x}=\bar{\mathbf{x}} \\ \mathbf{u}=\bar{\mathbf{u}}}} \delta_{\mathbf{x}}(t) + \left.\frac{\partial f}{\partial \mathbf{u}}\right|_{\substack{\mathbf{x}=\bar{\mathbf{x}} \\ \mathbf{u}=\bar{\mathbf{u}}}} \delta_{\mathbf{u}}(t) \tag{4.24}$$

Adapting (4.23) to our case where f depends on our local state $\boldsymbol{\xi}$ and input $\mathbf{u}$. The Taylor series around a fixed point $(\bar{\boldsymbol{\xi}}, \bar{\mathbf{u}})$, can be written as

$$f(\boldsymbol{\xi}, \mathbf{u}) = f(\bar{\boldsymbol{\xi}}, \bar{\mathbf{u}}) + \mathbf{J}(\bar{\boldsymbol{\xi}})(\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}) + \mathbf{G}(\bar{\mathbf{u}})(\mathbf{u} - \bar{\mathbf{u}}) \tag{4.25}$$

where $\mathbf{J}(\bar{\boldsymbol{\xi}})$ and $\mathbf{G}(\bar{\mathbf{u}})$ are the Jacobian matrices and the higher order terms of the Taylor Series expansion are neglected. Since $\bar{\boldsymbol{\xi}}$ is our state evaluated at an equilibrium point and at equilibrium, our local state is zero ($\bar{\xi} = 0$), we can write

$$f(\boldsymbol{\xi}, \mathbf{u}) = \mathbf{J}(\bar{\boldsymbol{\xi}})\boldsymbol{\xi} + \mathbf{G}(\bar{\mathbf{u}})(\mathbf{u} - \bar{\mathbf{u}}) \tag{4.26}$$

Instead of explicitly computing the partial derivatives of f with respect to each component of $\boldsymbol{\xi}$ and $\mathbf{u}$, an alternate way to compute the Jacobian matrix is to write f in the form of (4.26) and identify the term where the unit power of $\boldsymbol{\xi}$ appears to the right of a multiplication with a matrix. This matrix, evaluated at the equilibrium point, is the desired Jacobian ($\mathbf{J}$). The same logic applies to $\mathbf{G}$, but with a term where the unit power of $\mathbf{u}$ appears on the right.

From (4.24) we get a state in the form of

$$\dot{\boldsymbol{\xi}} = \mathbf{A}\boldsymbol{\xi} + \mathbf{B}\delta\mathbf{u} \tag{4.27}$$

where $\boldsymbol{\xi}$ is the local state, $\delta\mathbf{u} = \mathbf{u} - \bar{\mathbf{u}}$, is the control signal increment with respect to the equilibrium value and

$$\mathbf{A} = \begin{bmatrix} \frac{\partial f_1}{\partial \boldsymbol{\eta}_e} & \frac{\partial f_1}{\partial \boldsymbol{\eta}_h} & \frac{\partial f_1}{\partial \boldsymbol{\omega}_e} & \frac{\partial f_1}{\partial \boldsymbol{\omega}_h} \\ \frac{\partial f_2}{\partial \boldsymbol{\eta}_e} & \frac{\partial f_2}{\partial \boldsymbol{\eta}_h} & \frac{\partial f_2}{\partial \boldsymbol{\omega}_e} & \frac{\partial f_2}{\partial \boldsymbol{\omega}_h} \\ \frac{\partial f_3}{\partial \boldsymbol{\eta}_e} & \frac{\partial f_3}{\partial \boldsymbol{\eta}_h} & \frac{\partial f_3}{\partial \boldsymbol{\omega}_e} & \frac{\partial f_3}{\partial \boldsymbol{\omega}_h} \\ \frac{\partial f_4}{\partial \boldsymbol{\eta}_e} & \frac{\partial f_4}{\partial \boldsymbol{\eta}_h} & \frac{\partial f_4}{\partial \boldsymbol{\omega}_e} & \frac{\partial f_4}{\partial \boldsymbol{\omega}_h} \end{bmatrix} \tag{4.28}$$

$$B = \begin{bmatrix} \frac{\partial f_1}{\partial \mathbf{u}} \\ \frac{\partial f_2}{\partial \mathbf{u}} \\ \frac{\partial f_3}{\partial \mathbf{u}} \\ \frac{\partial f_4}{\partial \mathbf{u}} \end{bmatrix} \tag{4.29}$$

The computation of A (or jacobian) matrix is far from trivial. The differentiation of each state equation required a significant amount of effort, since the mathematical equalities used are not obvious (most of them had to be derived) and there are some non-linearities ruling the dynamics of the system. Below, the result of each differentiation is showed (where $\mathbf{I}_{3\times3}$ represents the identity matrix). The full differentiation steps are presented in chapter 8.

Given that the first two state equations are rather simple, their differentiation is trivial. Since the control will be focused on the eye, a special interest is taken in the differentiation of the state equation that rule its dynamics, f_3. The rest of the computations (the trivial ones, along with the redundant ones) are only shown in chapter 8.

So, if we expand (3.22) we have

$$\begin{aligned} {}^{e}\dot{\boldsymbol{\omega}}_{w,e} &= {}_{e}\mathbf{I}_{e}{}^{-1}({}_{e}\boldsymbol{\tau}_k + {}_{e}\boldsymbol{\tau}d - {}^{e}\boldsymbol{\omega}_{w,e} \times {}_{e}\mathbf{I}_{e}\,{}^{e}\boldsymbol{\omega}_{w,e}) && (4.30) \\ &= {}_{e}\mathbf{I}_{e}{}^{-1}(\sum_{i=1}^{6}({}^{e}Q_i \times \frac{k}{l_{0i}}(l_i - l_{0i})\,{}_{e}\vec{\phi}_i) - \mathbf{D}_{eye}\,{}^{e}\boldsymbol{\omega}_{h,e} - {}^{e}\boldsymbol{\omega}_{w,e} \times {}_{e}\mathbf{I}_{e}\,{}^{e}\boldsymbol{\omega}_{w,e}) && (4.31) \end{aligned}$$

The most complex derivative is the one of the direction vector $\vec{\phi}_i$ w.r.t our first two state variables $\boldsymbol{\eta}_e$

and $\boldsymbol{\eta}_h$. Thus, we will explore this derivative thoroughly. For simplicity, let's ignore the subscript i in the intermediate steps.

$$ {}_e\vec{\boldsymbol{\phi}} = \frac{({}^e\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q})}{\left\|{}^e\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q}\right\|} = ({}^e\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q})\frac{1}{\left\|{}^e\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q}\right\|} \tag{4.32} $$

$$ \frac{\partial\,{}_e\vec{\boldsymbol{\phi}}}{\partial\boldsymbol{\eta}_e} = \frac{\partial}{\partial\boldsymbol{\eta}_e}({}^e\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q})\frac{1}{\left\|{}^e\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q}\right\|} + ({}^e\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q})\frac{\partial}{\partial\boldsymbol{\eta}_e}\frac{1}{\left\|{}^e\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q}\right\|} \tag{4.33} $$

Let us focus on the first term of the previous equation, $\frac{\partial}{\partial\boldsymbol{\eta}_e}({}^e\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q})$, and remember equation (3.29). This way, we can write that first term as:

$$ \frac{\partial}{\partial\boldsymbol{\eta}_e}({}^e\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q}) = \frac{\partial}{\partial\boldsymbol{\eta}_e}({}^e\mathbf{R}_w\,{}^w\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q}) \tag{4.34} $$

Our perturbation, according to a Taylor Series expansion can be seen as

$$ \tilde{\mathbf{R}}(t) = \exp(\boldsymbol{\eta}^\wedge) = \mathbf{I} + \boldsymbol{\eta}^\wedge + \mathcal{O}(\left\|\tilde{\mathbf{x}} - \bar{\tilde{\mathbf{x}}}\right\|^2) \tag{4.35} $$

We wish to apply the perturbation on the eye frame, and as such, we need to apply it on the rotation matrix of the eye in the world frame.

$$ \frac{\partial}{\partial\boldsymbol{\eta}_e}(({}^h\mathbf{R}_w\,{}^w\bar{\mathbf{R}}_e\tilde{\mathbf{R}}_e)^T\,{}^h\mathbf{S} - {}^e\mathbf{Q}) = \frac{\partial}{\partial\boldsymbol{\eta}_e}(\tilde{\mathbf{R}}_e^T\,{}^e\bar{\mathbf{R}}_w\,{}^w\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q}) \tag{4.36} $$

Remember that the transpose of a skew-symmetric matrix is it's negative. Thus, from equation (4.35), we can replace $\tilde{\mathbf{R}}_e$ with

$$ \frac{\partial}{\partial\boldsymbol{\eta}_e}({}^e\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q}) = \frac{\partial}{\partial\boldsymbol{\eta}_e}(\mathbf{I}_{3\times3} - \boldsymbol{\eta}_e^\wedge + \mathcal{O}({\tilde{\eta}_e}^2))\,{}^e\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q} \tag{4.37} $$

We can dispose of the terms that don't depend on $\boldsymbol{\eta}_e$ (since their derivatives would be 0), simplify the ones that do depend on this variable and neglect the higher order terms since we aim to get linear expressions, leaving us with:

$$ \frac{\partial}{\partial\boldsymbol{\eta}_e}({}^e\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q}) = \frac{\partial}{\partial\boldsymbol{\eta}_e} - \boldsymbol{\eta}_e^\wedge\,{}^e\mathbf{R}_h\,{}^h\mathbf{S} = ({}^e\mathbf{R}_h\,{}^h\mathbf{S})^\wedge \tag{4.38} $$

Let us now focus on the second term from Eq.(4.33). So, let us first find the derivative of the norm and, in the end, use it to find its inverse.

$$ \begin{aligned} \frac{\partial}{\partial\boldsymbol{\eta}_e}\left\|{}^e\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q}\right\| &= \frac{\partial}{\partial\boldsymbol{\eta}_e}(({}^e\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q})^T({}^e\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q}))^{\frac{1}{2}} \\ = \frac{1}{2}(({}^e\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q})^T({}^e\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q}))^{-\frac{1}{2}}&\frac{\partial}{\partial\boldsymbol{\eta}_e}(({}^e\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q})^T({}^e\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q})) \\ &= \frac{1}{2}\frac{1}{\left\|{}^e\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q}\right\|}\frac{\partial}{\partial\boldsymbol{\eta}_e}(({}^e\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q})^T({}^e\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q})) \end{aligned} \tag{4.39} $$

Let's take the last term of the previous equation.

$$\begin{aligned}&\frac{\partial}{\partial \boldsymbol{\eta}_e}(({}^e\mathbf{R}_h\,{}^h\mathbf{S}-{}^e\mathbf{Q})^T({}^e\mathbf{R}_h\,{}^h\mathbf{S}-{}^e\mathbf{Q}))=\\&\frac{\partial}{\partial \boldsymbol{\eta}_e}(-{}^e\mathbf{Q}^T\,{}^e\mathbf{R}_h\,{}^h\mathbf{S}+{}^e\mathbf{Q}^T\,{}^e\mathbf{Q}+{}^h\mathbf{S}^T\,{}^h\mathbf{R}_e\,{}^e\mathbf{R}_h\,{}^h\mathbf{S}-{}^h\mathbf{S}^T\,{}^h\mathbf{R}_e\,{}^e\mathbf{Q})\end{aligned} \tag{4.40}$$

Again, let's consider only the terms that depend on ${}^w\mathbf{R}_e$, because it's where our perturbation is applied. As such, the other terms' derivative will be zero, including the higher order terms.

Let us also expand our perturbation variable, according to equation 4.35.

$$\begin{aligned}&\frac{\partial}{\partial \boldsymbol{\eta}_e}(-{}^e\mathbf{Q}^T\,{}^e\mathbf{R}_h\,{}^h\mathbf{S}-{}^h\mathbf{S}^T\,{}^h\mathbf{R}_e\,{}^e\mathbf{Q})\\=&\frac{\partial}{\partial \boldsymbol{\eta}_e}(-{}^e\mathbf{Q}^T({}^h\mathbf{R}_w\,{}^w\mathbf{R}_e\tilde{\mathbf{R}}_e)^T\,{}^h\mathbf{S}-{}^h\mathbf{S}^T\,{}^h\mathbf{R}_w\,{}^w\mathbf{R}_e\tilde{\mathbf{R}}_e\,{}^e\mathbf{Q})\\=&\frac{\partial}{\partial \boldsymbol{\eta}_e}({}^e\mathbf{Q}^T(-\mathbf{I}_{3\times3}+\boldsymbol{\eta}_e^\wedge-\mathcal{O}(\tilde{\boldsymbol{\eta}_e}^2))\,{}^e\mathbf{R}_h\,{}^h\mathbf{S}-{}^h\mathbf{S}^T\,{}^h\mathbf{R}_e(\mathbf{I}_{3\times3}+\boldsymbol{\eta}_e^\wedge+\mathcal{O}(\tilde{\boldsymbol{\eta}_e}^2))\,{}^e\mathbf{Q})\\=&\frac{\partial}{\partial \boldsymbol{\eta}_e}({}^e\mathbf{Q}^T\boldsymbol{\eta}_e^\wedge\,{}^e\mathbf{R}_h\,{}^h\mathbf{S}-{}^h\mathbf{S}^T\,{}^h\mathbf{R}_e\boldsymbol{\eta}_e^\wedge\,{}^e\mathbf{Q})\\=&-\frac{\partial}{\partial \boldsymbol{\eta}_e}({}^e\mathbf{Q}^T({}^e\mathbf{R}_h\,{}^h\mathbf{S})^\wedge\boldsymbol{\eta}_e+{}^h\mathbf{S}^T\,{}^h\mathbf{R}_e\,{}^e\mathbf{Q}^\wedge\boldsymbol{\eta}_e)\end{aligned} \tag{4.41}$$

According to equations (4.26, 4.40 and 4.41), we get

$$\frac{\partial}{\partial \boldsymbol{\eta}_e}(({}^e\mathbf{R}_h\,{}^h\mathbf{S}-{}^e\mathbf{Q})^T({}^e\mathbf{R}_h\,{}^h\mathbf{S}-{}^e\mathbf{Q}))=-{}^e\mathbf{Q}^T({}^e\mathbf{R}_h\,{}^h\mathbf{S})^\wedge-{}^h\mathbf{S}^T\,{}^h\mathbf{R}_e\,{}^e\mathbf{Q}^\wedge \tag{4.42}$$

The terms from equation (4.42) are actually the same term written in a different form. Therefore, we have

$$\frac{\partial}{\partial \boldsymbol{\eta}_e}({}^e\mathbf{R}_h\,{}^h\mathbf{S}-{}^e\mathbf{Q})^T({}^e\mathbf{R}_h\,{}^h\mathbf{S}-{}^e\mathbf{Q}))=-2\,{}^h\mathbf{S}^T\,{}^h\mathbf{R}_e\,{}^e\mathbf{Q}^\wedge \tag{4.43}$$

As it was said before, we have to differentiate the inverse of the norm and not the norm itself. So,

$$\frac{\partial}{\partial \boldsymbol{\eta}_e}\frac{1}{\left\|{}^e\mathbf{R}_h\,{}^h\mathbf{S}-{}^e\mathbf{Q}\right\|}=-\frac{\frac{\partial}{\partial \boldsymbol{\eta}_e}\left\|{}^e\mathbf{R}_h\,{}^h\mathbf{S}-{}^e\mathbf{Q}\right\|}{\left\|{}^e\mathbf{R}_h\,{}^h\mathbf{S}-{}^e\mathbf{Q}\right\|^2}=\frac{1}{2}\frac{2\,{}^h\mathbf{S}^T\,{}^h\mathbf{R}_e\,{}^e\mathbf{Q}^\wedge}{\left\|{}^e\mathbf{R}_h\,{}^h\mathbf{S}-{}^e\mathbf{Q}\right\|^2}\frac{1}{\left\|{}^e\mathbf{R}_h\,{}^h\mathbf{S}-{}^e\mathbf{Q}\right\|} \tag{4.44}$$

Putting the whole equation together, from equations (4.33, 4.44 and 4.38), we get

$$\frac{\partial\,{}_e\vec{\boldsymbol{\phi}}}{\partial \boldsymbol{\eta}_e}=\frac{({}^e\mathbf{R}_h\,{}^h\mathbf{S})^\wedge}{\left\|{}^e\mathbf{R}_h\,{}^h\mathbf{S}-{}^e\mathbf{Q}\right\|}-\frac{1}{2}\frac{({}^e\mathbf{R}_h\,{}^h\mathbf{S}-{}^e\mathbf{Q})}{\left\|{}^e\mathbf{R}_h\,{}^h\mathbf{S}-{}^e\mathbf{Q}\right\|}\frac{2\,{}^h\mathbf{S}^T\,{}^h\mathbf{R}_e\,{}^e\mathbf{Q}^\wedge}{\left\|{}^e\mathbf{R}_h\,{}^h\mathbf{S}-{}^e\mathbf{Q}\right\|^2} \tag{4.45}$$

We can simplify to

$$\frac{\partial\,{}_e\vec{\boldsymbol{\phi}}}{\partial \boldsymbol{\eta}_e}=\frac{({}^e\mathbf{R}_h\,{}^h\mathbf{S})^\wedge}{\left\|{}^e\mathbf{R}_h\,{}^h\mathbf{S}-{}^e\mathbf{Q}\right\|}-\frac{({}^e\mathbf{R}_h\,{}^h\mathbf{S}-{}^e\mathbf{Q})}{\left\|{}^e\mathbf{R}_h\,{}^h\mathbf{S}-{}^e\mathbf{Q}\right\|^3}({}^h\mathbf{S}^T\,{}^h\mathbf{R}_e\,{}^e\mathbf{Q}^\wedge) \tag{4.46}$$

The final component to differentiate is $\frac{k}{l_{0i}}(l_i-l_{0i})\vec{\boldsymbol{\phi}}$. Since l_{0i} is a fixed value, there is no need to

expand it. Let us focus on the differentiation of l_i, since we already analysed the direction vector.

$$\frac{\partial}{\partial \boldsymbol{\eta}_e}(\frac{k}{l_{0i}}(l_i - l_{0i})\vec{\boldsymbol{\phi}}) = \frac{\partial}{\partial \boldsymbol{\eta}_e}(\frac{k}{l_{0i}} l_i \vec{\boldsymbol{\phi}} - \frac{k}{l_{0i}} l_{0i}\vec{\boldsymbol{\phi}}) \tag{4.47}$$

Expanding l_i, we get

$$\frac{\partial}{\partial \boldsymbol{\eta}_e}(\frac{k}{l_{0i}} \left\| {}^{e}\mathbf{R}_h \, {}^{h}\mathbf{S} - {}^{e}\mathbf{Q} \right\| \vec{\boldsymbol{\phi}} + \frac{k}{l_{0i}} r\theta\vec{\boldsymbol{\phi}} - \frac{k}{l_{0i}} l_{0i}\vec{\boldsymbol{\phi}}) = \frac{\partial}{\partial \boldsymbol{\eta}_e}(\frac{k}{l_{0i}}({}^{e}\mathbf{R}_h \, {}^{h}\mathbf{S} - {}^{e}\mathbf{Q}) + \frac{k}{l_{0i}}(r\theta - l_{0i})\vec{\boldsymbol{\phi}}) \tag{4.48}$$

This results in

$$\frac{\partial}{\partial \boldsymbol{\eta}_e}(\frac{k}{l_{0i}}(l_i - l_{0i})\vec{\boldsymbol{\phi}}) = ({}^{e}\mathbf{R}_h \, {}^{h}\mathbf{S}_i)^{\wedge}\frac{k}{l_{0i}} + \frac{k}{l_{0i}}(r\theta_i - l_{0i})\frac{\partial \, {}_e\vec{\boldsymbol{\phi}}_i}{\partial \boldsymbol{\eta}_e}) \tag{4.49}$$

Since the muscle torque formula is the same for the six muscles, we sum the linear result of elastic torque respective to each muscle:

$$\frac{\partial f_3}{\partial \boldsymbol{\eta}_e} = {}_e\mathbf{I}_e{}^{-1}(\sum_{i=1}^{6}({}^{e}\mathbf{Q}_i \times (({}^{e}\mathbf{R}_h \, {}^{h}\mathbf{S}_i)^{\wedge}\frac{k}{l_{0i}} + \frac{k}{l_{0i}}(r\theta_i - l_{0i})\frac{\partial \, {}_e\vec{\boldsymbol{\phi}}_i}{\partial \boldsymbol{\eta}_e})) + \mathbf{D}_{eye}({}^{e}\mathbf{R}_h \, {}^{h}\boldsymbol{\omega}_{w,h})^{\wedge}) \tag{4.50}$$

The differentiation of f_3 with respect to the the second local state variable $\boldsymbol{\eta}_h$ follows the same logic, so it is only presented in the Appendix.

Next, we will focus on the derivative of f_3 with respect to eye's velocity:

$$\begin{aligned}\frac{\partial f_3}{\partial \, {}^{e}\boldsymbol{\omega}_e} &= \frac{\partial}{\partial \, {}^{e}\boldsymbol{\omega}_{w,e}} \, {}_e\mathbf{I}_e{}^{-1}({}_e\boldsymbol{\tau}_k + {}_e\boldsymbol{\tau} d - {}^{e}\boldsymbol{\omega}_{w,e} \times {}_e\mathbf{I}_e \, {}^{e}\boldsymbol{\omega}_{w,e}) \\ &= \frac{\partial}{\partial \, {}^{e}\boldsymbol{\omega}_{w,e}} \, {}_e\mathbf{I}_e{}^{-1}(\sum_{i=1}^{6}({}^{e}\mathbf{Q}_i \times \frac{k}{l_{0i}}(l_i - l_{0i}) \, {}_e\vec{\boldsymbol{\phi}}_i) - \mathbf{D}_{eye} \, {}^{e}\boldsymbol{\omega}_{h,e} - {}^{e}\boldsymbol{\omega}_{w,e} \times {}_e\mathbf{I}_e \, {}^{e}\boldsymbol{\omega}_{w,e})\end{aligned} \tag{4.51}$$

Let us consider a function $f(\boldsymbol{\omega}) = \boldsymbol{\omega}^{\wedge}$. With this function, we are able to solve an equality that is relevant for the next few steps:

$$\frac{\partial \boldsymbol{\omega} \times {}_e\mathbf{I}_e\boldsymbol{\omega}}{\partial \boldsymbol{\omega}} = \frac{\partial}{\partial \boldsymbol{\omega}}(\boldsymbol{\omega}^{\wedge} \, {}_e\mathbf{I}_e\boldsymbol{\omega}) = \frac{\partial}{\partial \boldsymbol{\omega}}(f(\boldsymbol{\omega}) \, {}_e\mathbf{I}_e\boldsymbol{\omega}) = \frac{\partial}{\partial \boldsymbol{\omega}}(f(\boldsymbol{\omega})) \, {}_e\mathbf{I}_e\boldsymbol{\omega} + f(\boldsymbol{\omega}) \, {}_e\mathbf{I}_e \tag{4.52}$$

Doing a element-wise derivative, and knowing that the transpose of a skew symmetric matrix corresponds to its symmetric, we get

$$\frac{\partial}{\partial \boldsymbol{\omega}}(f(\boldsymbol{\omega})) \, {}_e\mathbf{I}_e\boldsymbol{\omega} = \begin{bmatrix} \frac{\partial f}{\partial \omega_x} \, {}_e\mathbf{I}_e\boldsymbol{\omega} & \frac{\partial f}{\partial \omega_y} \, {}_e\mathbf{I}_e\boldsymbol{\omega} & \frac{\partial f}{\partial \omega_z} \, {}_e\mathbf{I}_e\boldsymbol{\omega} \end{bmatrix} = -({}_e\mathbf{I}_e\boldsymbol{\omega})^{\wedge} \tag{4.53}$$

where

$$\frac{\partial f}{\partial \omega_x} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \frac{\partial f}{\partial \omega_y} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \frac{\partial f}{\partial \omega_z} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.54}$$

Therefore, we get

$$\frac{\partial \boldsymbol{\omega} \times {}_e\mathbf{I}_e\boldsymbol{\omega}}{\partial \boldsymbol{\omega}} = -({}_e\mathbf{I}_e\boldsymbol{\omega})^{\wedge} + \boldsymbol{\omega}^{\wedge} \, {}_e\mathbf{I}_e \tag{4.55}$$

So, from (3.22) and (4.55) we can write

$$\frac{\partial f_3}{\partial {}^e\boldsymbol{\omega}_{w,e}} = {}_{ee}\mathbf{I}_e^{-1}(\mathbf{D}_{eye} + {}^e\boldsymbol{\omega}_{w,e}{}^{\wedge}\,{}_{ee}\mathbf{I}_e - ({}_{ee}\mathbf{I}_e\,{}^e\boldsymbol{\omega}_{w,e})^{\wedge}) \tag{4.56}$$

The rest of the differentiations of state equations with respect to state variables follow the same logic as the ones presented above, and therefore are only presented in Chapter 8. Regarding the computation of the **B** matrix (differentiation of the state equations with respect to the inputs), the computations are quite simple and can also be seen in the Appendix.

Chapter 5

Optimal Control

Optimal control has the objective of discovering the control for a dynamical system, over a time interval, so that the system's behavior is optimal according to some constraints. These constraints are usually modeled through a cost function.

Cost functions are functions that depend on the state variables, and the input commands. Optimal control tries to minimize a cost function to get the ideal duration of the saccade and control inputs, and therefore, an optimal performance from the system. In order to attain said input for each saccade, we apply the optimal control to the linear system (local state).

Summarizing, we wish to

$$\underset{\mathbf{u},p}{\text{minimize}} \;\; J(\mathbf{u}, p, goal) = \sum^{t} \lambda_t J_t(\mathbf{u}, p, goal)$$

subject to

$$\mathbf{x}_{i+1} = \mathbf{A}\mathbf{x}_i + \mathbf{B}\mathbf{u}_i$$

$$\mathbf{y}_i = \mathbf{C}\mathbf{x}_i, \qquad i = 0, 1, ..., p$$

$$\mathbf{u} \geq 0$$

where J_t represents each cost term and λ_t is the weight associated with each individual cost term. *p* is the optimal saccade time. For each *p* we compute the costs and decide on the optimal time that has the minimum total cost for a specific saccade.

We aim to build an open-loop discrete saccadic controller for the eye, so first we need to discretize the system.

By discretizing, we are transforming our continuous model into a discrete one. Discretization is done by picking an appropriate sampling interval (Δt), in which the dynamics of the system are preserved. If the sampling time is too small or reaction time too slow, peak shapes (e.g. velocity) will not remain as narrow as they really are. We know that the sampling time has to be, at least, ten times smaller than the smallest time constant present in the continuous system [34]. A Δt of 1 ms was chosen, since it fits the conditions presented above and allows for a thorough analysis of the model. The discretization was

made resorting to Matlab's function *c2d*.

Our discrete state-space model can be represented by

$$\boldsymbol{\xi}^{(k+1)} = \mathbf{A}\boldsymbol{\xi}^{(k)} + \mathbf{B}\delta\mathbf{u}^{(k)}, \qquad k = 0, ..., p \tag{5.1}$$

$$\mathbf{y}^{(k)} = \mathbf{C}\boldsymbol{\xi}^{(k)}, \qquad k = 0, ..., p \tag{5.2}$$

In these equations **A** is the dynamic matrix which is endowed with information regarding the system's properties (Jacobian), **B** is the matrix that defines the influence of the input **u** and **C** is the output matrix, which transforms the state $\boldsymbol{\xi}$ into output **y**. In our case, this output is the orientation of the eye at a given moment.

Equation (5.1) can also be written as

$$\boldsymbol{\xi}^{k} = \mathbf{A}^{k}\boldsymbol{\xi}^{(0)} + \sum_{j=0}^{k-1} \mathbf{A}^{k-1-j}\mathbf{B}\mathbf{u}^{j} \tag{5.3}$$

If we consider the following variables

$$\mathbf{F} = \begin{bmatrix} \mathbf{A}^{p-1} & \mathbf{A}^{p-2} & \mathbf{A}^{p-3} & ... & \mathbf{I}_{3\times 3} \end{bmatrix} \tag{5.4}$$

$$\Gamma = \begin{bmatrix} \mathbf{B} & 0 & \cdots & 0 \\ 0 & \mathbf{B} & 0 & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{B} \end{bmatrix} \tag{5.5}$$

Then, we can write our state variable like this

$$\boldsymbol{\xi}^{(p)} = \mathbf{A}^{p}\boldsymbol{\xi}^{(0)} + \mathbf{F}\boldsymbol{\Gamma}\mathbf{u} \tag{5.6}$$

with

$$\mathbf{u}_h = \begin{bmatrix} \mathbf{u}^{(0)} \\ \mathbf{u}^{(1)} \\ \cdots \\ \mathbf{u}^{(p-1)} \end{bmatrix} \tag{5.7}$$

It is very important to notice that only the eye is controlled, and as such, all cost terms were thought and developed considering only its properties, since the head was kept fixed. Furthermore, knowing that we have six muscles moving the eye, this would give it six degrees of freedom. However, the eye can

only move in three degrees of orientational freedom, making this system underdetermined. This means that we would have infinite solutions, for any goal. So, to reduce the search space, some constraints were later implemented on the optimizer, along with the costs presented below.

5.1 Cost Function

The initial cost function implemented was similar to the one used in [3]. This function accounts for accuracy, duration, energy expenditure and equilibrium. Regarding accuracy and duration, these terms refer to the eye movement, whereas the energy terms refers to the total motor efforts exerted in the eye. However, since our model is now different, this was not enough: new cost terms had to be explored and implemented.

5.1.1 Duration

This cost term aims to toll the time it takes to perform a saccade. We know that duration increases almost linearly with goal amplitude. However, our cost term does not rely on amplitude explicitly, since we want that the underlying dynamics of the eye satisfy *Main Sequence* relationships without forcing them directly. Our duration cost is given by

$$J_p = \lambda_p(1 - \frac{1}{1 + \beta \boldsymbol{p}}) \tag{5.8}$$

where λ_p is the relative weight of this cost term, β is the rate of discount parameter (temporal reward decay) and p is the saccade duration. The rate of discount can also be perceived as ”impulsiveness”, since it expresses the weight the brain links to a specific visual stimulus. So, this parameter is particular for each race and individual. Several studies have been made to investigate this weight in different fields, like psychology or economics. For instances, in [35] the authors offer a monetary reward that increases with time to several students. From the results, they were able to iterate a set of possible functions able to describe the human's time rewarding system: an hyperbolic and an exponential one. Exponential is used widely in economics since experts in that field consider there is a big risk attached to waiting on the reward. In other words if said risk is null, this type of function will underestimate reward for longer time delays. Hyperbolic functions on the other hand are used in psychology and suit human conduct better.

Although this temporal cost function fits human behaviour rather well, it is prone to causing a non-convex total cost. Therefore, it had to be iterated so that it was able to provide good results without risking convergence problems due to non-convexity.

5.1.2 Accuracy

The accuracy term seeks to penalize eye movements that deviate from the goal orientation. This makes the system strive to get as close to the objective as possible. Our cost is taken as

$$J_x = \lambda_x (y(p) - g)^2 \tag{5.9}$$

where λ_x specifies the relative cost for inaccuracies, $y(p)$ is the final position (at time p) of the eye and g is the goal position. The position of the eye is the first component of our global state, ${}^{w}\mathbf{R}_e$. However, it has to be transformed to local state(4.14), since we are working with the linearized system. Goal position is generated in rotation vector and also expressed in local state by transforming it into a rotation matrix and once again use (4.14) .

5.1.3 Energy

Here, we aim to toll the energy expenditure. Since we assume the energy to be proportional to the actuator's rotation (angular velocity), and the timesteps to be uniform, we can simplify this term as the difference between angular positions at each timestep. This can simply be written as the difference between consecutive $u^{(i)}$'s.

$$J_e(\boldsymbol{u}) = \lambda_e \|\Delta \boldsymbol{u}\|^2 \tag{5.10}$$

where J_e is the effort cost, λ_e is the weight of this cost term and $\Delta\mathbf{u}$ is the difference between the current vector of motor commands and the previous.

In matrix form, this can be written as

$$J_e(u) = \lambda_e \|\mathbf{M}\mathbf{u}\|^2 = \lambda_e \| \begin{bmatrix} I & 0 & \dots & 0 \\ -I & I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & -I & I \end{bmatrix} \mathbf{u} \|^2 = \lambda_e \| \begin{bmatrix} u_0 \\ u_1 - u_0 \\ \vdots \\ u_p - u_{p-1} \end{bmatrix} \|^2 \tag{5.11}$$

5.1.4 Equilibrium

This term is taken with the purpose of penalizing the change in state when the eye reaches the goal. With this cost it is possible to toll the change in orientation and velocity, when compared to the state reached by the optimal control. The function that translates this objective into a mathematical context is:

$$J_{eq} = \lambda_{eq} (\mathbf{x}(p) - \mathbf{x}(p+1))^2 \tag{5.12}$$

where λ_{eq} is the weight given to this cost, $x(p)$ is the final state and the other term symbolizes the subsequent states.

5.1.5 Last input accuracy

Equilibrium and accuracy cost terms are computed using the approximated linear model, so they don't stabilize at the actual required goal. In order to make sure that the system stabilizes at the goal orientation, we calculate a desired final input from equilibrium conditions and toll the deviation from that set of commands. Putting this in equation form, we get

$$J_{uacc} = \lambda_{uacc}(\mathbf{u}(p) - \mathbf{u}_{des})^2 \tag{5.13}$$

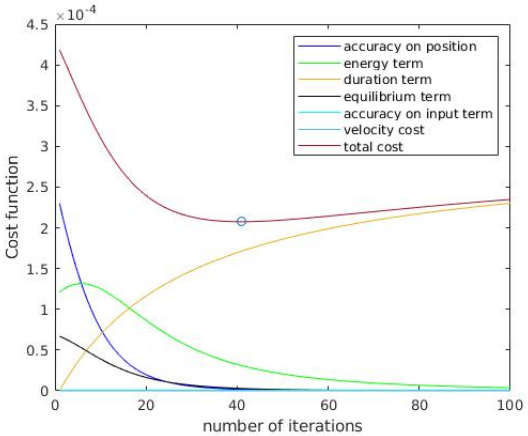
where λ_{uacc} is the relative weight of this cost, $\mathbf{u}(p)$ is the set of optimal commands at the final time step for a specific goal, given by the optimal control solver and $\mathbf{u}_{des}$ is the set of commands that ideally would lead to goal orientation given by the solution of the equilibrium equations (4.3).

5.1.6 Total cost

The complete cost function is the sum of the previous functions

$$J = J_p + J_x + J_c + J_{eq} + J_{uacc} \tag{5.14}$$

In order to have the eye behaving like a human one, the weights for each cost term had to be carefully calibrated, not giving too much importance to one term and neglecting the others. The solver used for this minimization was Matlab's *quadprog* function, since we have a linear system and quadratic costs. The balance between terms is the most important aspect of this calibration. Not only this, but we wish to minimize the whole cost function, meaning the sum of each cost term and not each one individually. An example of said minimization is shown in figure 5.1.



what is this
extra velocity cost??
It's not in Eq. 5.14

Figure 5.1: Minimization of the sum of the terms explained above, and by doing so, it is possible to find the optimal time and inputs.

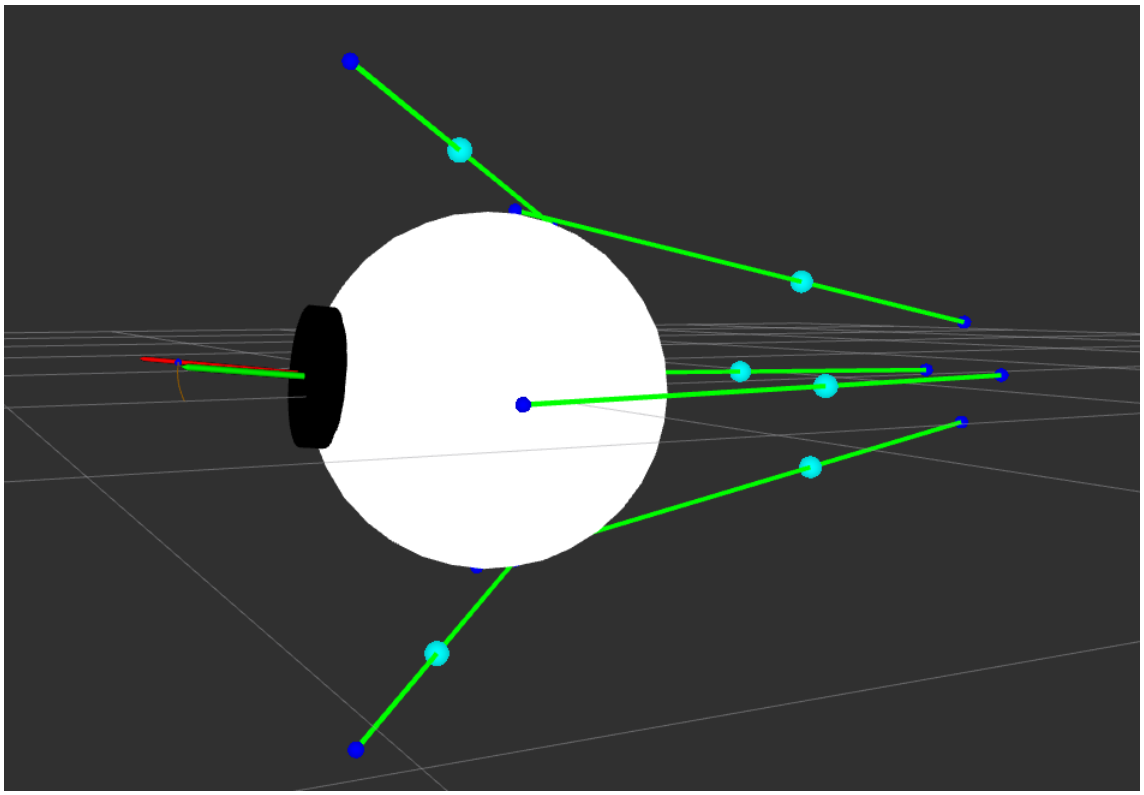


Figure 5.2: Graphical simulator developed using the visualization tool Rviz to study the slack problem.

5.2 Slack

One of the objectives of this thesis is to mitigate slack. This phenomenon can be described as the EOM having, during a saccade, smaller length than its initial length, making it unable to exert force.

Since the muscle force in this model is calculated using equation (3.5), having a negative difference in length results in an unrealistic reversal of pulling direction. It is possible to avoid this by simply implementing a constraint that makes the force zero if it would go negative, making the other muscles compensate for the specific muscle that would go slack. However, in the linear approximation of our system this is not possible to ensue since this is a non-linear constraint. The linearization not accounting for this specificity creates a necessity to control slack in a different way.

Since this proved to be a difficult problem to solve, a graphical simulator was developed resorting to ROS (Robot Operating System, [36]) graphical capabilities, namely Rviz. This proved extremely helpful since it was designed to show a difference in color when the muscles went slack, and showing their length in real time. Besides this, it was also possible to see the trajectory performed by the system in three dimensions. Figure 5.2 shows this graphical tool.

One way to deal with this is would be to put a nonlinear constraint on the optimization solver, which would check if the length of all cables are higher than their initial lengths.

$$(\frac{k}{l_{0i}}(l_i - l_{0i}) + r\mathbf{u}_i)\vec{\phi} > 1 \tag{5.15}$$

Here, the lengths and the direction vector depend on the norm between insertion points on the head

and eye and their rotation matrices. When implemented, it was quickly understood that this non-linear constraint was computationally expensive and resulted in the solver failing to find an optimal solution. Instead, it was decided to stick to strategies that used linear methods.

Since this model has six independent motors, the system has 6 degrees of freedom. However, the eye rotates in 3, which makes it an under-constrained problem. So, the inputs were divided in one fixed value (u_{common}) and one variable value for each pair of muscles ($u_{differential}$). This would allow to control the fixed value, making sure that the forces would always be positive and the variable value would make the eye perform an optimal trajectory to reach its goal. An example featuring the horizontal muscle pair is presented below.

$$u_{common} = \frac{u_{MR} + u_{LR}}{2} \tag{5.16}$$

$$u_{diff} = \frac{u_{MR} - u_{LR}}{2} \tag{5.17}$$

which leads to

$$u_{MR} = u_{common} + u_{diff} \tag{5.18}$$

$$u_{LR} = u_{common} - u_{diff} \tag{5.19}$$

Nevertheless this was not an optimal solution, because in order to mitigate slack the fixed value had to be very large, making the system incapable of stabilizing. This happened because the eye was reaching its goal with residual velocity, causing it to drift.

Another strategy tried was identifying the agonist and antagonist in each EOM pair and trying to minimize the difference between the exerted forces in each pair so that the eye wouldn't move too fast in the direction of the goal. This is justified because the initial speed in the beginning of the saccade was created due to the optimal control finding that the easiest way to reach the goal was to give increasingly negative values to the antagonist in the preponderant muscle pair for a specific saccade. For the agonist we would have

$$u_{agonist} = u_{common} + u_{diff} - u_{delta} \tag{5.20}$$

Even so, trying to minimize the difference in force between agonist and antagonist was not sufficient. To slow down the eye (in order to mitigate slack) a delay was also applied to the agonist, but this strategy also proved to be unable to avoid slack.

Two final approaches are given more relevance in this work: a solver-based approach and a memory-based one. In the first one, an external solver is used, and in the second a "lookup" table.

Solver-based approach

Since slack was persisting, even with several different approaches already tried, we resorted to an external solver to output the minimal ideal motor commands that would lead the eye to the goal, in an equilibrium state, and to a final individual muscle force bigger that a predetermined value. This can be expressed mathematically as

Minimize $u_{1-6} = ||u_{1-6}||^2$

u

subject to

$$\mathbf{f}_i > 1N$$

$$\boldsymbol{\tau}_k = 0, \qquad i = 0, 1, ..., 6$$

Then, a cost was put on the deviation from these final motor commands given by the solver, to make the system reach them (last input accuracy term). This cost substituted the accuracy term cost, since their purpose is very similar.

Memory-based approach

In this final approach a search for equilibrium points around the oculomotor range was made. For each point obtained, the associated last commands that led the eye there were stored in a ”lookup” table. A condition had to be satisfied in order for the equilibrium point to be stored: the final force for each muscle had to be higher than a given threshold in order not to have slack. So, after a random saccade goal is generated, we search the closest equilibrium point in the table and consider that the final ideal input is the one associated with that orientation in the table. More details on these two final approaches are explained in section 6.

Chapter 6

Results

This chapter aims to explain the experiments that were carried out and demonstrate their respective results. The setup of each experiment that was performed is also explained.

The main parts of this work are also validated in this chapter: the simulator, linearization and the optimal control strategies.

6.1 Simulator Validation

In order to validate the simulator, the eye-movement range (oculomotor range) and the step response were tested and compared to the empirical results that stem from the real system. We can divide this section in three parts: eye, head and both of them together. Each of these components will be analysed based on the metrics aforementioned.

Note that even if the head is not considered in the optimal control experiments, its free movement range and step response are still analysed.

6.1.1 Eye

In this subsection the the head is fixed, so that only the eye dynamics are tested.

6.1.1.1 Oculomotor Range

The full three dimensional oculomotor range is very important to analyse since it is aimed to compare the uncontrolled simulator capabilities with its behavior when optimal control is implemented. Since the eye follows Listing's law (two degrees of orientational freedom) due to neural control, one has to make sure the simulator has three degrees of freedom when moving freely. So, the torsional component in rotations when the system is moving freely can take a range of values, which doesn't happen when controlled, in order to obey Listing's law. To find the range of the oculomotor system, 1500 saccades were generated with random motor rotations between -45° and 45°, and their resulting eye orientations are presented

(including during trajectory) in figure 6.1. The range of each rotational component is described in table 6.1.

Rotational Component	Minimum (deg)	Maximum (deg)
Torsional	-45.83	51.56
Horizontal	-108.86	103.13
Vertical	-63.03	68.76

Table 6.1: Maximum and minimum values for the ocular range, presented in degrees, for each of the orientation components.

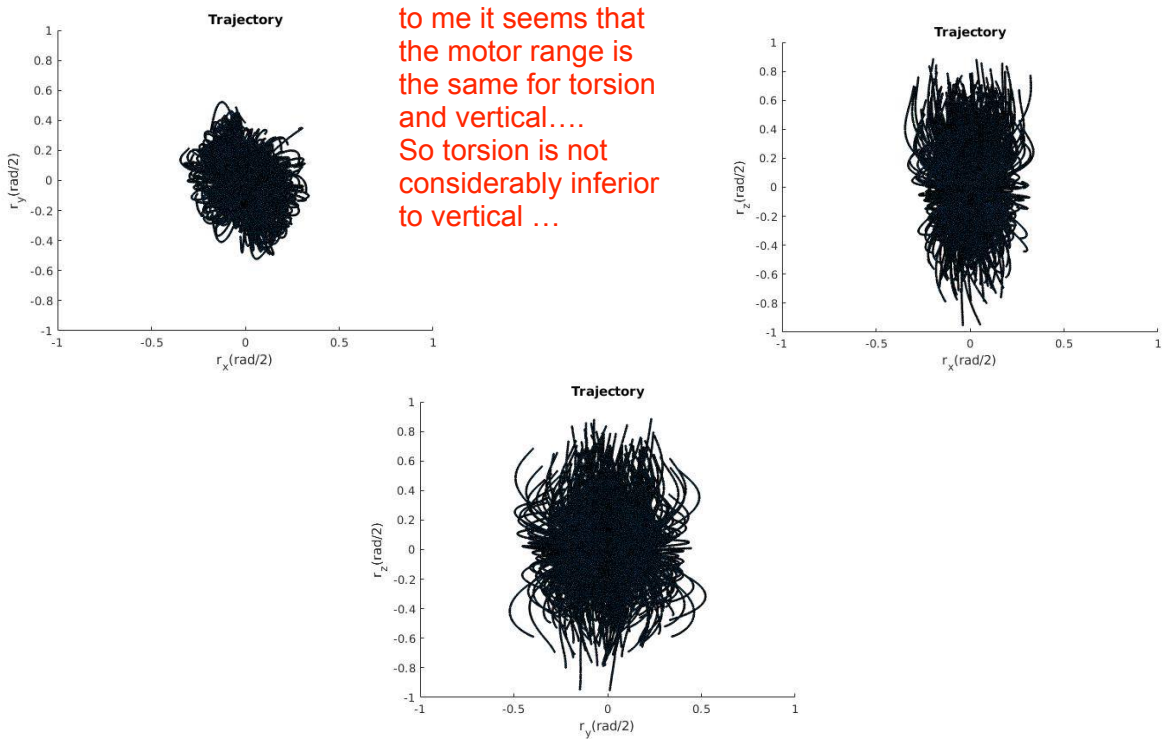


Figure 6.1: 3D ocular range of the eye model shown in the three different planes. Notice the discrepancy between the torsional component and the other two: torsional range is considerably inferior to vertical and specially horizontal. However, 45 degrees of torsional rotation (approximately 0.4 half radians) is extremely high, showing that the uncontrolled model has 3 degrees of freedom.

As can be seen in table 6.1 and figure 6.1 the range is different for all rotation components. Torsion has the smallest orientations span, while horizontal has the highest. This can be related to the norm between insertion points on the eye and head, when the eye is at the straight ahead position. Given that LR is the longest muscle, it allows the eye to rotate more to the right (negative rotation around the z axis). On the other hand, the oblique muscles are the smallest which means torsional rotations will have the smallest span of rotation. However, even though torsional limits stray around 45 to 50 degrees from the origin, this is a considerable amount of cyclotorsion. Thus, it can be concluded that the eye model implemented has three degrees of orientational freedom. It also suggests that the motor commands for horizontal components should be restrict this movement, to keep the horizontal motion of the eye within

the normal limits of about $\pm$ 45 degrees.

6.1.1.2 Step Response

The step response consists of the time evolution of the outputs of the system, when the input is a step signal. The calibration of the damping and stiffness parameters for the eye were iterated knowing that the system's slowest time constant is around 200 ms [10]. It is also known that the eye can be well represented by a second order overdamped system, which means the step response does not oscillate and has a somewhat slow stabilization process. Given these properties, the eye simulator is true to the real system, since it respects the above characteristics, as can be seen in figure 6.2.

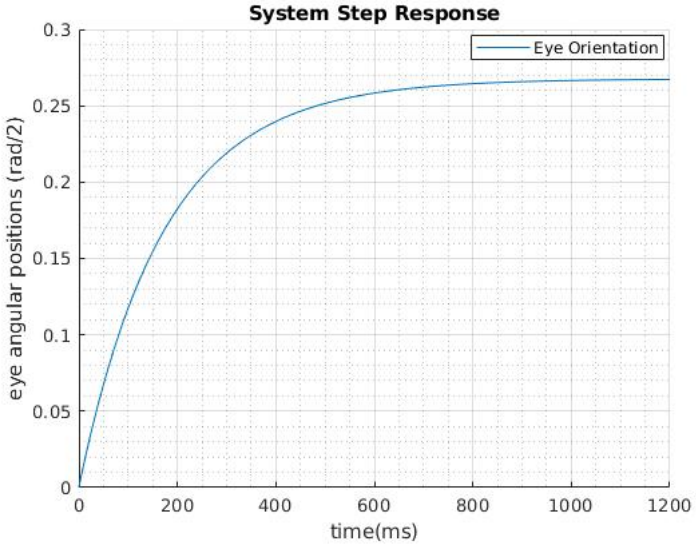


Figure 6.2: Step response for the eye, when the inputs are motor controls that lead the system to perform a 30 degree purely horizontal rotation. In order to achieve this behavior damping and elasticity matrices are isotropic and have values of 0.04 and 20, respectively. Regarding the inertia, $I_{xx} = 0.48$ $I_{yy} = 0.43$ $I_{zz} = 0.39$.

6.1.2 Head

In this subsection the the eye is fixed, so that only the head dynamics are tested. Even if this part of the system is not controlled it is interesting to verify its dynamical properties, namely, the free movement range and step response. The head has no muscle model, it was simplified as a simple sphere joint, with the same center of rotation as the eye and the inputs are direct torques in each of its three dimensions.

6.1.2.1 3D movement Range

In order to check the head's 3D movement range, the eye was fixed to head. This means the opposite reaction from eye movement that is applied on the head is neglected equation (3.11). Hence, only the head's free movement is analysed. Again, 1500 head saccades were generated by randomly applying direct torque ($\pm$ 10 N m) on each dimension of head rotation. The results can be seen below, in figure 6.3.

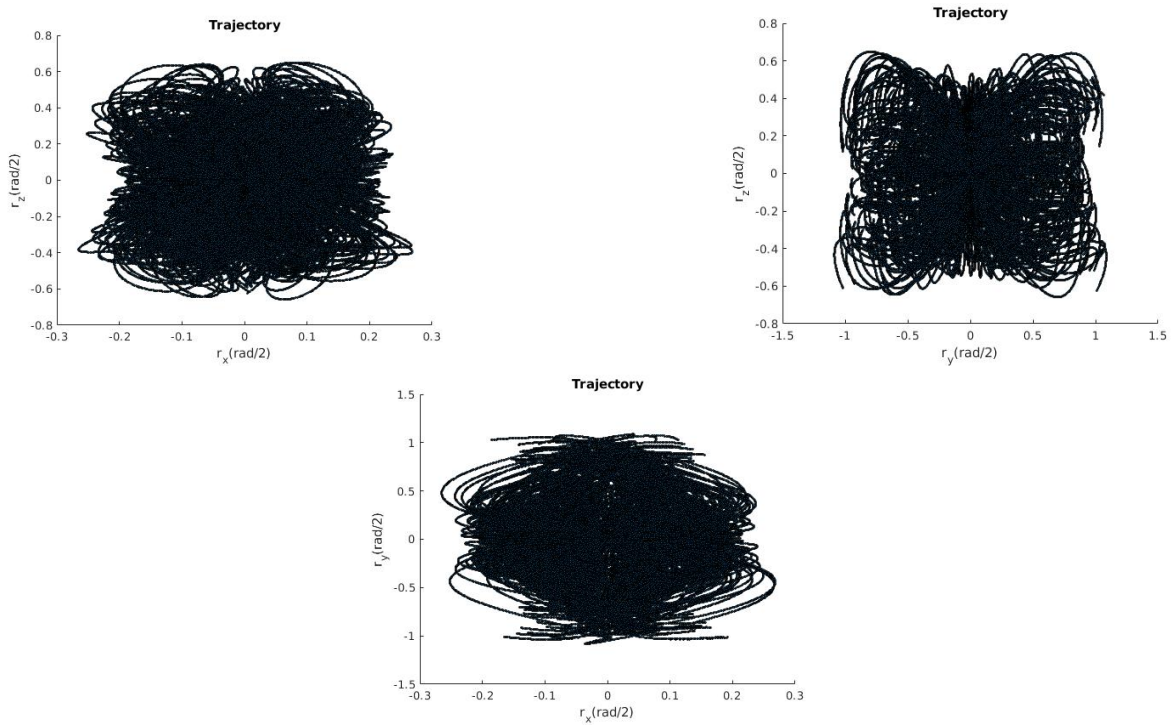


Figure 6.3: 3D movement range of the head model shown in the three different planes. Notice the difference between the horizontal component and the other two: horizontal range is slightly higher than vertical and considerably higher than torsion, as observed in humans. It can bee seen that the head has also 3 degrees of freedom.

Knowing that the head also was modeled to have elastic and damping properties, these parameters were calibrated to be closer to the human head behavior. However, this was not a trivial task since the head was assumed to be a spherical joint, having 3 degrees of freedom, and this means it would be able to move as much in one direction as the other two. Nevertheless, studies show that the head is able to perform horizontal rotations of larger magnitudes than torsional or vertical ones. Figure 6.3 depicts this fact. Horizontal rotations have a bigger range of motion than the torsional and vertical components. Vertical range is smaller than the horizontal one but considerably higher than the torsional. This comparison comes in accordance with studies made on humans [37]. However, it's important to notice that the magnitude of each rotation component is directly dependant on the magnitude of the input applied to the head (which is not controlled).

6.1.2.2 Step Response

Head inertia is modelled to be around 1000 times bigger than the eye's. The human head, without neural control is thought to be well approximated by an underdamped system [38]. This behavior can be seen in figure 6.4. The settling time is also around 10 times bigger than the eye's, even for small rotations. The head, like the eye, is modelled as an isotropic system, not privileging the dynamics in any direction. However, the neck has a higher stiffness in the torsional dimension, not allowing the head to rotate as much in this direction.

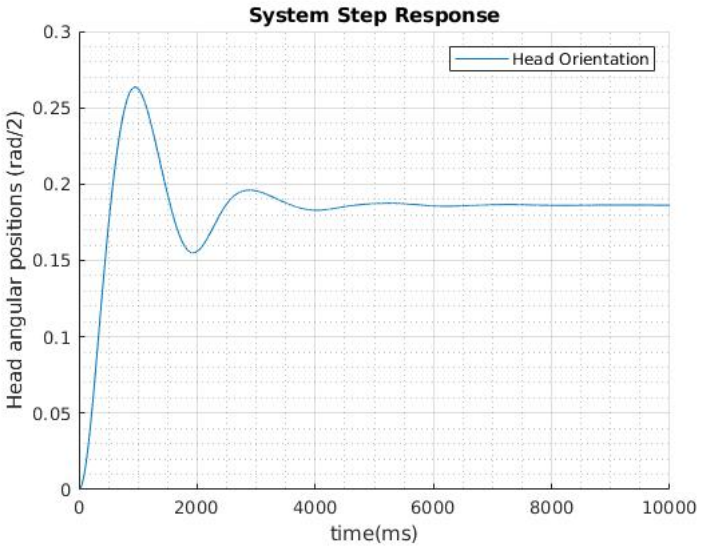


Figure 6.4: Step response for the head, when the inputs are motor controls that lead the system to perform a saccade of 20 degrees purely horizontal rotation. The values for elasticity and damping matrices are 1 and 35, respectively. The inertia was $I_{xx} = 181$ $I_{yy} = 215$ $I_{zz} = 142$.

6.1.3 Eye-Head System

The only validation left, regarding the simulator, is testing the step response for the coupled system. In order to check this, two tests were made: make the head move and check the eye's passive motion, and make the eye move and check the head's resulting motion. Since the head is a lot heavier than the eye, it is expected that the eye's passive motion is largely motion by the head. On the other hand, it is expected that the eye's motion barely influences the head, given their inertia differences. The results can be seen in figures 6.5 and 6.6.

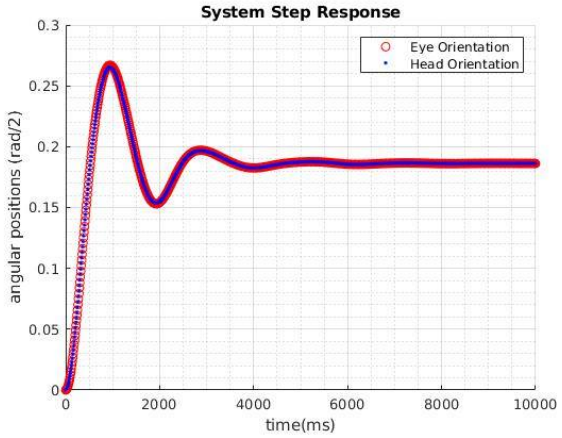


Figure 6.5: It can be seen that an input on the head to make it move, drags the eye with it, i.e., the eye passively has the exact same behavior as the head, with a few milliseconds of delay.

As expected it was observed that the eye's passive movement when the head rotates is quite accentuated, but the head's motion when the eye rotates is negligible.

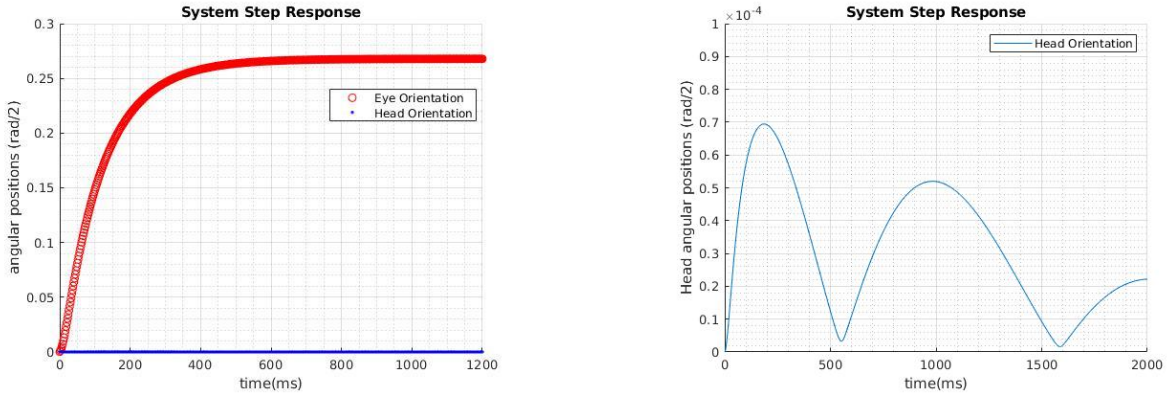


Figure 6.6: When an input is applied only to the eye in order to check the head's passive movement, it can be seen that while the eye move's with it's usual overdamped behavior, the head is almost still (it actually moves around 0.0007 degrees, which is very small compared to the eye movement).

6.2 Linearization Validation

The linearization was made to generate a linear system that approximates the functioning of the non-linear one around each equilibrium position. Here, we compare both systems. This is done by applying some perturbations on each one around the origin and in an eccentric orientation, and then comparing the results. Since it is aimed to control only the eye, a special interest is taken on how a simple perturbation affects the dynamics of the eye.

Two types of perturbations were applied to the linear system to analyse it's similarity to the non-linear one: a perturbation on orientation, represented by a magnitude-varying rotation matrix ($\delta\mathbf{R}$), and a perturbation on velocity, $\delta\boldsymbol{\omega}$. The outputs of each system are then analysed according to two metrics: comparing the direction vectors and magnitude of the acceleration ($\dot{\omega}$) of the linear and non-linear set-ups. The ideal results would be having a difference in magnitude of angular acceleration as close to zero as possible and the comparison of direction, by computing the dot product between the linear and non-linear direction vectors, as close to one as possible.

Analysing the effect of velocity perturbations on the acceleration of the eye ($\dot{\omega}$) around straight ahead orientation (figure 6.7) and around 30 degree eccentricity (figure 6.8), it can be seen that in both cases the values of the linear and non-linear systems are really close for the range of tested magnitudes. The variation in the direction and magnitude of the angular acceleration is almost identical for both cases. It can also be seen that the variation in direction is extremely small.

As for the effects of orientation perturbations, they can be seen in figure 6.9 for straight ahead orientation and in figure 6.10 for the eccentric orientation. It is noticeable that the linear system is much more sensitive to this type of perturbation. The variation in the direction has a higher error in case the system is linearized at 30 degrees eccentricity, even for smaller perturbations. Regarding the difference in magnitude of the angular acceleration, it can be seen that when the system is linearized around the origin, it tends to underestimate the magnitude, whereas when linearized at an eccentric orientation it could either under or overestimate the value of $\dot{\omega}$.

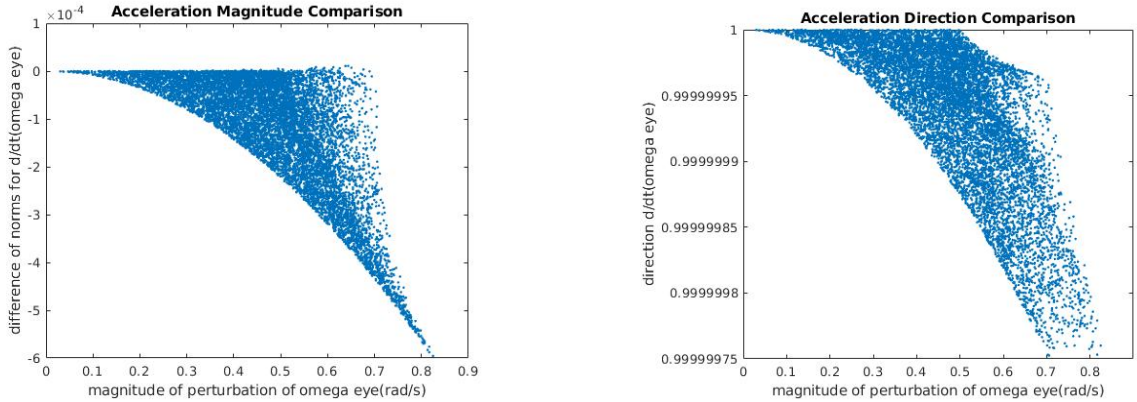


Figure 6.7: Comparison between the linear and non-linear systems, regarding acceleration direction and magnitude when a perturbation on the velocity is applied around the origin.

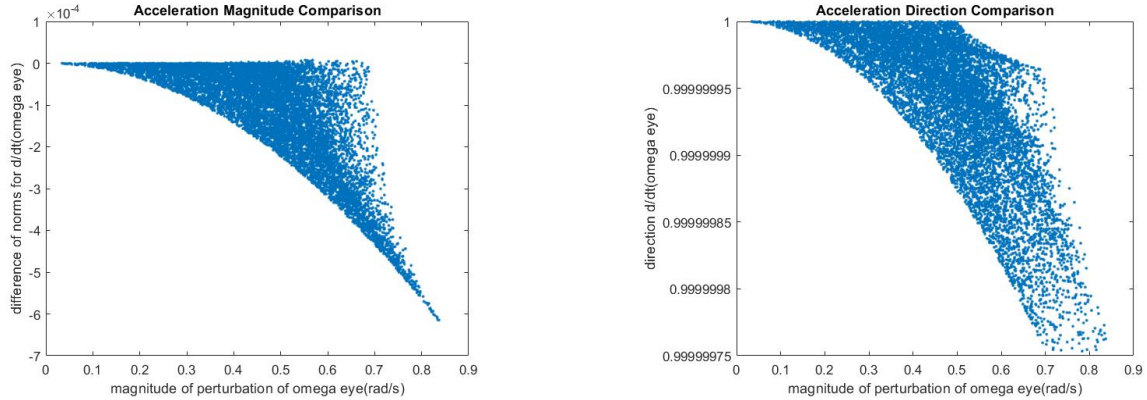


Figure 6.8: Comparison between the linear and non-linear systems, regarding acceleration direction and magnitude when a perturbation on the velocity is applied at an eccentric position (horizontal 30 deg).

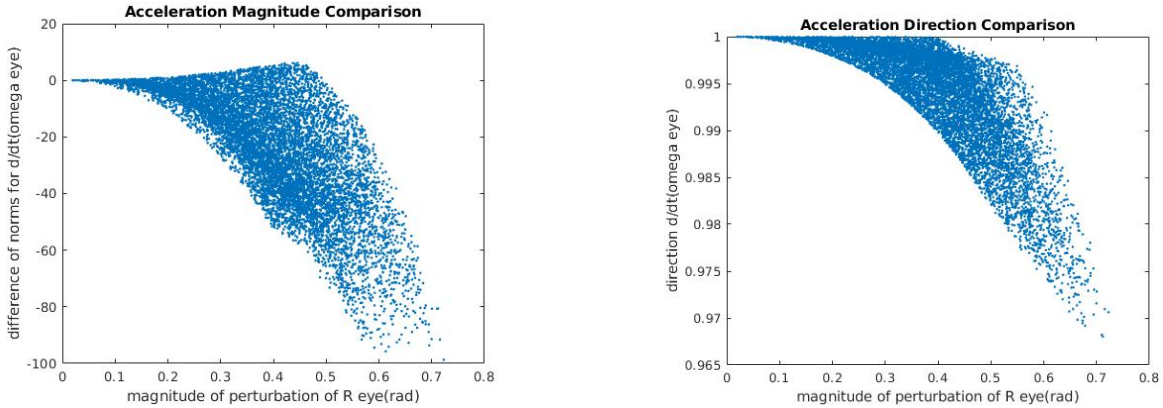


Figure 6.9: Comparison between the linear and non-linear systems, regarding acceleration direction and magnitude when a perturbation on the orientation is applied at straight ahead orientation.

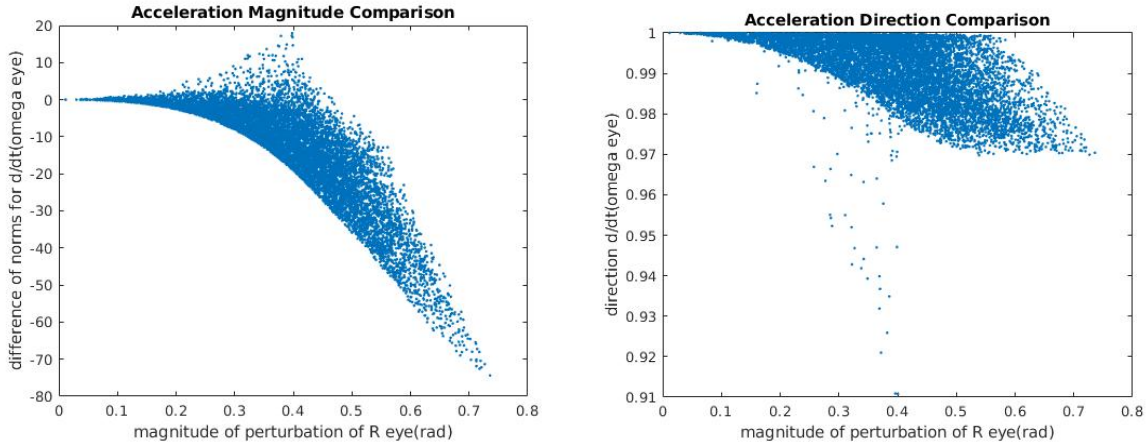


Figure 6.10: Comparison between the linear and non-linear systems, regarding acceleration direction and magnitude when a perturbation on the orientation is applied at an eccentric position (horizontal 30 deg).

Given the comparisons between the linear and the non-linear systems (from ten thousand random perturbations), it can be noticed that this type of analytical linearization is very reliable for small perturbations around the origin, but loses some of its benefits when one strays from the equilibrium point. For better understanding of the difference in reliability of the linearization in different orientations, the minimum perturbation for which the comparison between the linear and non-linear systems has an error greater that five per cent is quantified in table 6.2.

Perturbation	Max Magnitude at Origin	Max Magnitude at 30 deg
δR_{eye}	0.137 rad (7.85°)	0.103 rad (5.89°)
$\delta \omega_{eye}$	No max	No max

Table 6.2: Maximum magnitude of each perturbation that makes the linearization give an error bigger than 5%.

6.3 Optimal Control Results

In this section the two main approaches (solver-based and memory-based) tested are explained and their respective results presented.

It has been observed that the linearization implemented in this work is efficient for a small interval around the linearization point. So, for larger saccades, it is natural that the results have a bigger error. The strategy exercised is to identify the motor commands required to fixate the eye at the goal in the non-linear model and use this in a feedforward method. This means that the optimal inputs for the trajectory are computed based on the final motor rotations in addition to the classical costs (energy, duration, accuracy and equilibrium).

6.3.1 Solver-based Approach

This approach computes the target steady state motor commands by solving the equilibrium equations (3.24) at the desired eye orientation. Becuse the solution of the motor commands is under-constrained (there are many possible actuations resulting in the same eye orientation), we need to include additional constraints in the solution.

By putting some constraints on the output of the solver (based on Matlab's fmincon function), it was desired that the these end commands would make the cables tense (to help mitigate slack) and that the eye would reach its goal orientation and fixate there. With this purpose, the outputs of the external solver had to make the state functions f_1 and f_3 (equations 3.20 and 3.22) equal to zero. Assuming that the velocity would be null at goal orientation, all that was needed was that the elastic torque would be zero. So, the two constraints applied were that the sum of the six muscle torques had to be zero, but at the same time each muscle had to exert a minimum force at the end orientation. Since the optimal controller gets the final motor commands from the solver to reach its target, it is possible to penalize the deviation from those desired commands. This was done by computing the difference between the final optimal inputs generated by an external solver, and the commands the optimal controller generates by minimizing the cost terms (5.13).

6.3.1.1 Obedience to Listing's law

200 random saccades were performed (with amplitudes from 5 to 40 degrees), using the optimal controller and the final eye orientations were displayed in three different planes (xy, xz and yz). Each saccade started from the orientation reached by the eye in the previous saccade. It is possible to see from figure 6.11 that the red dots (saccades with slack) make the Listing's plane thicker. These results show a total standard deviation (σ_x) of 1.36 degrees. This value accounts for both slack-free saccades and ones with slack.

6.3.1.2 Obedience to *Main Sequence*

Regarding the *Main Sequence*, increasingly higher magnitude (horizontal) saccades were tested to see if the system's behavior resembled the human eye's dynamic properties. The results can be seen in figure 6.12. Duration increases almost linearly with amplitude and peak velocity increases until it saturates. These are behaviors that go in accordance to what has been studied in humans. Skewness was also tested, but didn't present results as good as the *Main Sequence* relations. It is noticeable that peak velocity is not reached in the same time frame for any two or more saccades, even the ones with a higher amplitude. However, in humans, for higher amplitude saccades, peak velocity is reached at around the same time frame but larger ones still take more time to decelerate, causing them to take more time.

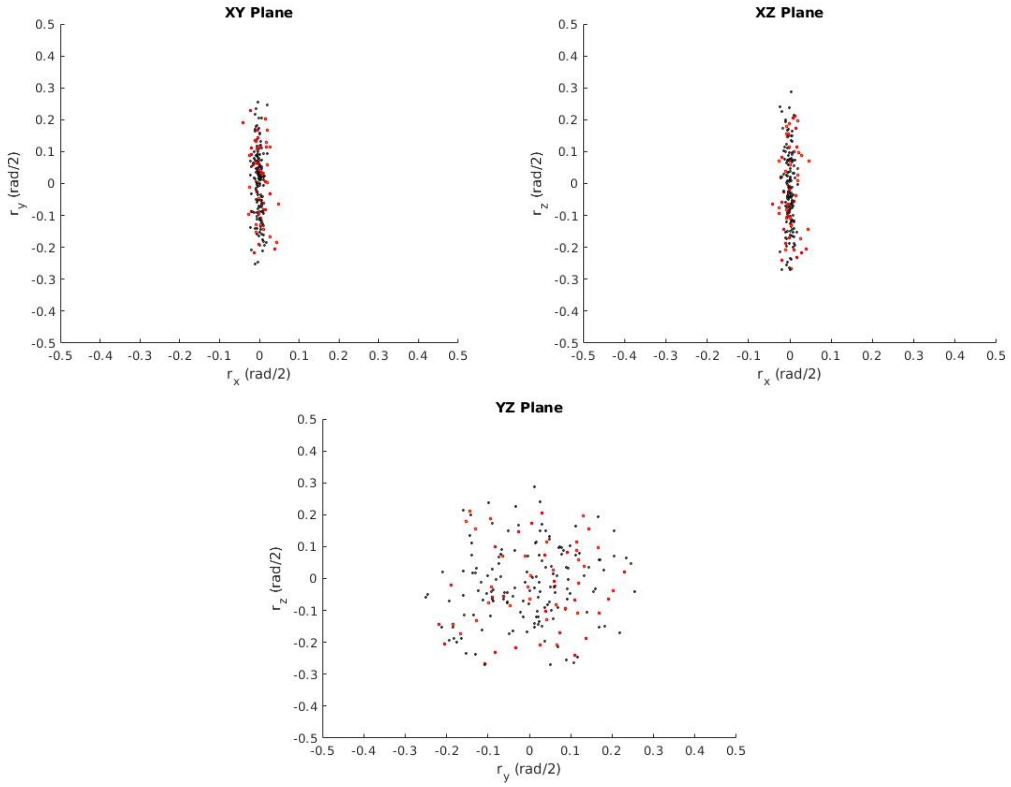


Figure 6.11: Final orientation of the eye for 200 saccades. The red dots represent eye movements in which slack occurred.

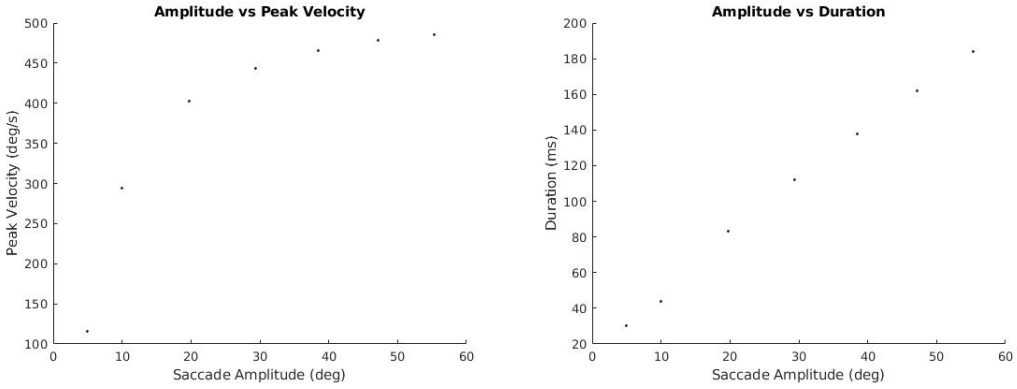


Figure 6.12: Main Sequence relations for solver-based approach.

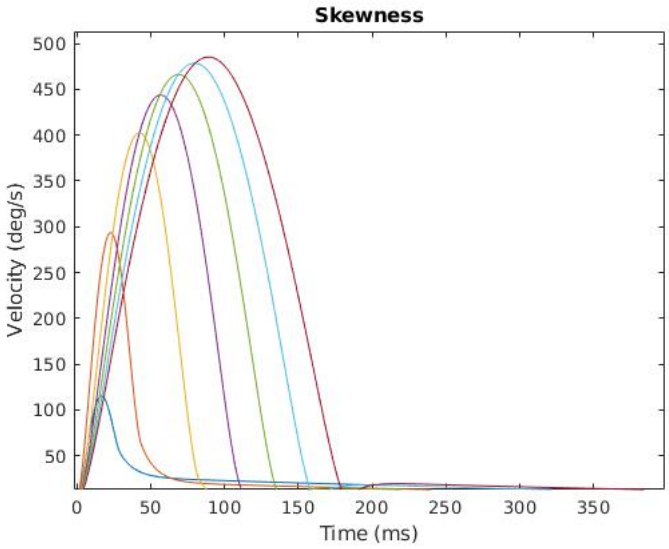


Figure 6.13: Skewness analysis for the solver-based approach.

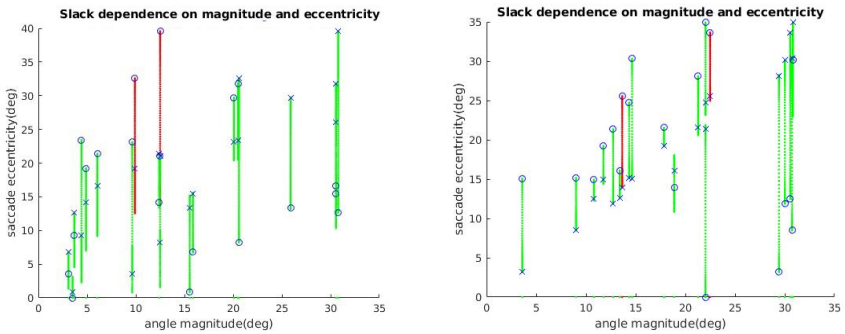


Figure 6.14: Relation between angle magnitude and saccade eccentricity. The circles denote the starting position of the eye for that saccade, and the crosses are the final position. The lines in between show the total rotation made. Red lines are trajectories where slack happened and green lines are normal trajectories.

6.3.1.3 Causes of Slack

A trial with only a few saccades was attempted, to study how slack depended on the eccentricity and amplitude of the saccade. This result can be seen in figure 6.14. It is possible to see from the same figure that although slack happens more frequently at larger eccentricity (40 degrees away from the origin), it is not necessary for the eye to be so far from the straight ahead orientation for this phenomenon to occur: there is also a saccade starting at around 30 degrees (one of several) that went slack, while other saccades starting from the same degree of eccentricity appear to have a normal behavior.

Resorting to the graphical simulator, it was possible to see that the trajectory of the eye doesn't move only between the initial and final points. At times, the eye's motion goes beyond the expected, by reaching the goal and then drifting to another direction (mainly horizontal). This fact lead to understand that the solver gives final commands that do not solve exactly the equilibrium conditions, meaning that when the eye reaches the supposed equilibrium point with non-zero velocity, it diverges as seen in figure 6.15. This caused some accuracy error and slack, so there was a need for a different approach.

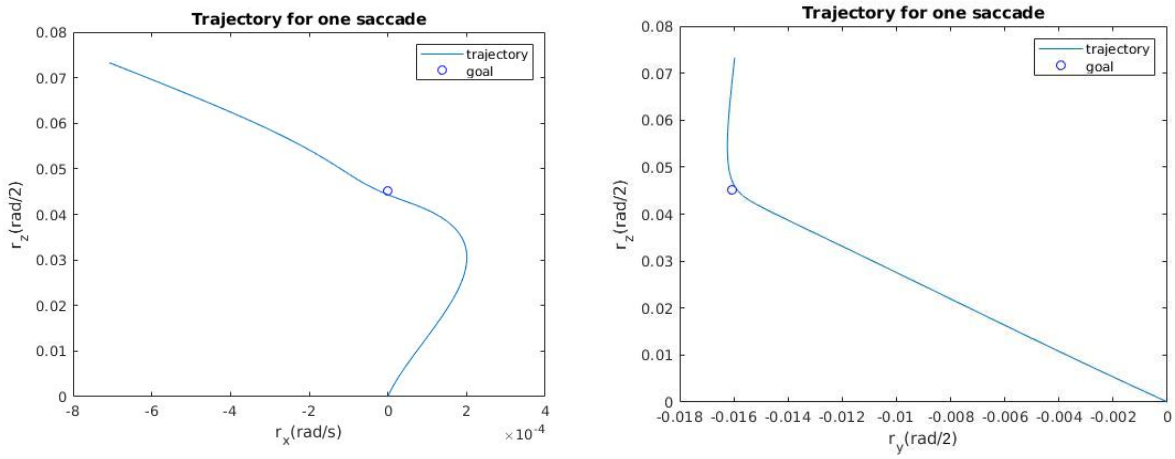


Figure 6.15: Depiction of horizontal drift caused by the system reaching the goal with residual velocity. The saccade starts from the origin and the blue circle represents the goal orientation. The drift was caused by the high values provided by the solver to the horizontal muscle pair. Note differences in scale.

6.3.2 Memory-based Approach

Since the solver based approach had problems due to slack and unstable equilibrium points, it was decided to change approaches. A more biologically plausible strategy might be to learn combinations of muscle extensions associated with specific eye orientations.

In order to gather equilibrium points, random step input combinations were given to the all motors in a range of $\pm$ 85 degrees, with the aim to find equilibrium orientations all around the eye's oculomotor range. If the combination of commands provided resulted in a stable equilibrium orientation, that set of inputs and resulting rotation vector was saved.

Figure 6.16 shows the distribution of the points stored in three dimensions. A skewness in equilibrium points can be observed, having less points on the right side of the plot. This is due to the asymmetry of the extraocular muscles, mainly between the horizontal pair. Furthermore, these equilibrium points were plotted with respect to the forces that each muscle exerts in that specific equilibrium rotation, in order to validate the functioning of the system. In a specific saccade, when a muscle applies more force on the eye than its counterpart, it's automatically labelled as the agonist. Hence, for instance, in a horizontal saccade to the left, LR would be the agonist, meaning that it exerts more force for that specific movement and in the equilibrium state than the others EOM. In figure 6.17 this can be seen for IR, MR and all muscles together.

However, since it is aimed to mitigate slack, the combinations of inputs that would lead the muscles being tense below a certain threshold at the final orientation were discarded. So, when a random saccade goal is generated, a search for the closest equilibrium rotation was made and the commands associated to that target were considered the ideal final motor input. These command inputs were then compared to the ones provided by the previous solver for the same goal and it was noticed that there were significant differences in the values for the horizontal muscle pair: the solver output much bigger values for this pair. So, the value from the memory approach was used instead of the commands provided by the solver in the previous approach.

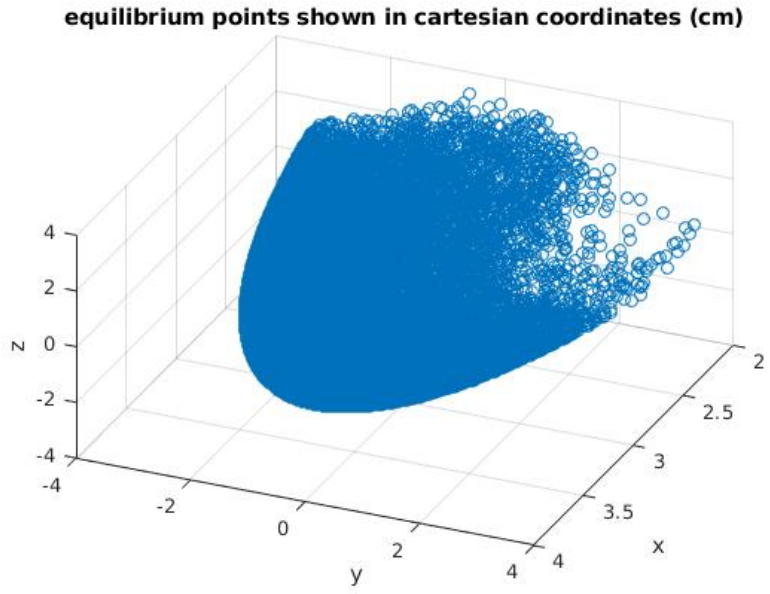


Figure 6.16: Equilibrium points in the eye's visual range. There is a skewness in their distribution due to the asymmetry of the horizontal muscles. Around 20000 points were obtained and are shown here.

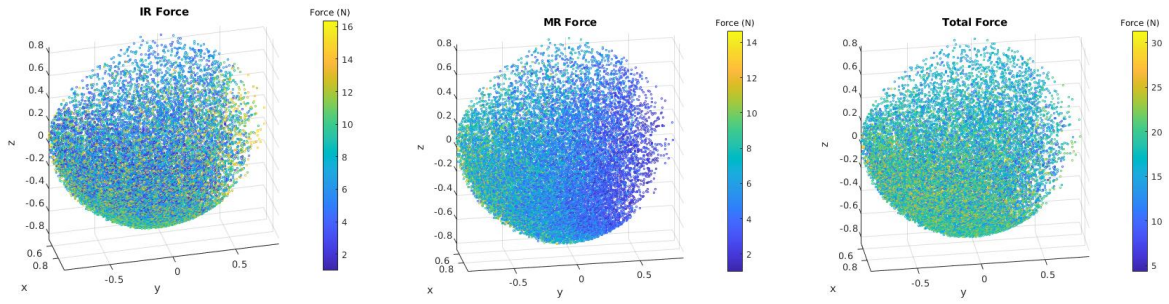


Figure 6.17: To the utmost left one can see the forces applied by the Inferior Rectus in equilibrium positions all around the eye's visual range. Since the IR rotates the eye downward (positive y), for equilibrium points lower in the vertical axis, it exerts more force. In the center the same logic may be applied, since the Medial Rectus rotates the eye leftward (positive z), so more force is applied by this muscle for equilibrium orientations to the left. To the right the full force of the muscles is shown, having a well-distributed force among equilibrium orientations.

The metrics to test this approach were the same as before.

6.3.2.1 Obedience to Listing's law

250 random saccades were generated (with amplitudes from 5 to 40 degrees), using the optimal controller and their obedience to Listing's plane was evaluated, along with the verification of slack. The results from this experiment can be seen in figure 6.18. As it is noticeable, the biggest difference between the results from the previous approach is the mitigation of slack. Pre-tension had an important role in this task, by not letting the negative commands make the cables relaxed. Additionally, by identifying equilibrium orientations and storing them it was possible for the system to reach the goal orientation with almost zero velocity. The major limitation that arises from this approach is the possibility of having a relevant accuracy error, since the goals are generated randomly and that specific orientation might not be an equilibrium position. However, from the tests performed it was observed that the error was almost always lower than 5 degrees. However, the maximum error was 7 degrees because the saccade goals implied a rotation to the right side, where we have fewer equilibrium points (figure 6.16). This is not optimal, but it is a trade-off so that it would be possible to eliminate slack. The standard deviation from Listing's plane was 1.17 degrees. For primates, studies have shown that this values is around 0.6 to 1.2 degrees. Even though our result is closer to the maximum value, proves the model and this approach are valid, and quite accurate.

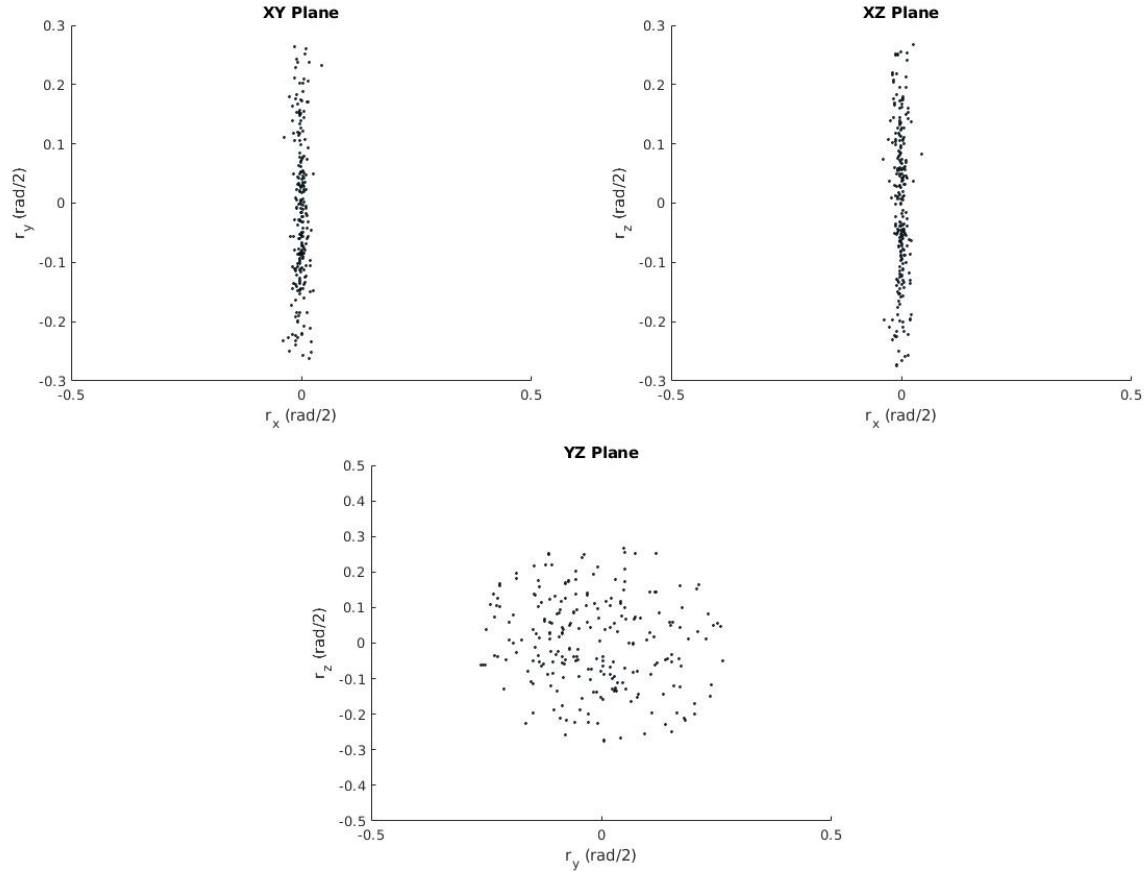


Figure 6.18: 250 saccades' final eye orientations shown in 3 different planes. The standard deviation was of 1.17 degrees.

The trajectories for these saccades can also be observed, to analyse if they also obey Listing's law. They can be seen in figure 6.19. During the trajectories there was no slack, and during the eye's movement until the goal, all orientations were in Listing's plane, and all trajectories were mainly straight, having low to null curvature. This shows that the controller makes the eye move in the optimal way, i.e., through the shortest path to the goal.

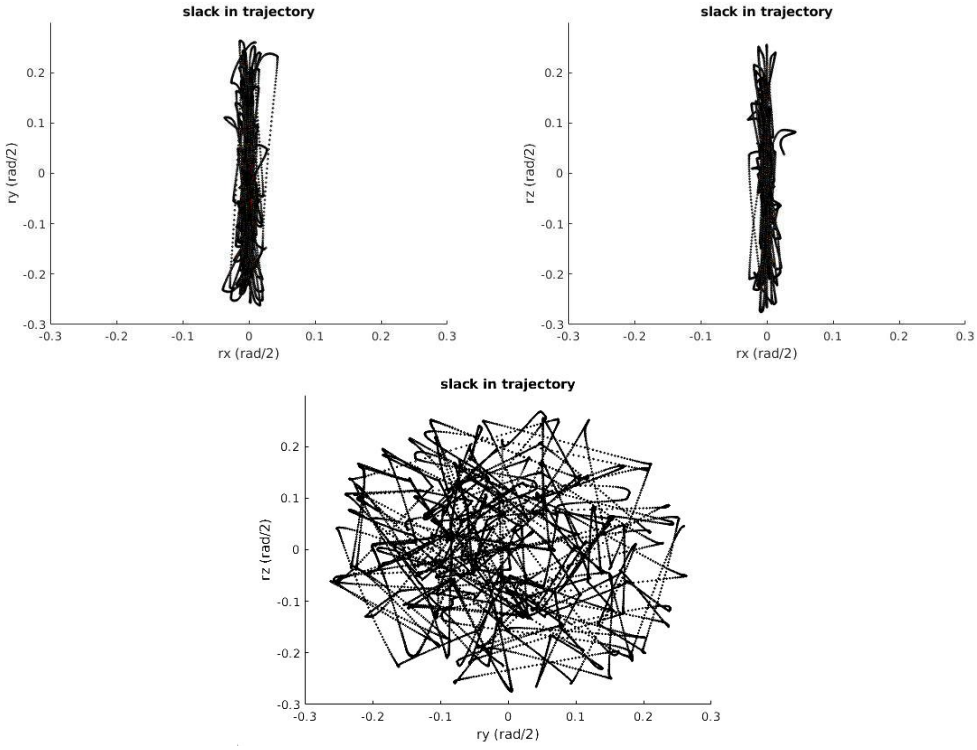


Figure 6.19: Trajectories of the eye to reach the final orientations seen in figure 6.18. Maximum deviation from Listing's plane is 4.9 degrees.

Since there were no saccades with slack, and the results showed good accuracy, this approach was validated, and a deeper analysis followed, to compare the properties of the controlled simulation with the properties of the human eye. This comparison was based on the *Main Sequence* and the skewness of velocity profiles.

6.3.2.2 Obedience to *Main Sequence*

As it can be observed in figure 6.20, the expected *Main Sequence* relations apply. Duration increases almost linearly with saccade amplitude due to the trade-off between duration and energy. The values of duration are also in agreement with neurological studies, having a value of around 200 ms for large saccades. Regarding the relation between peak velocity and amplitude, if one inspects the right plot from this figure, it is possible to notice that there is a linear increase on maximum angular velocity until around 400 deg/s. However, after this value, peak velocity saturates since the optimal controller does not allow the eye to reach greater speeds.

Another metric used to evaluate the functioning of this system is the skewness of saccades. In humans, peak velocity saturates at some point, but bigger saccades take longer to be executed. Further-

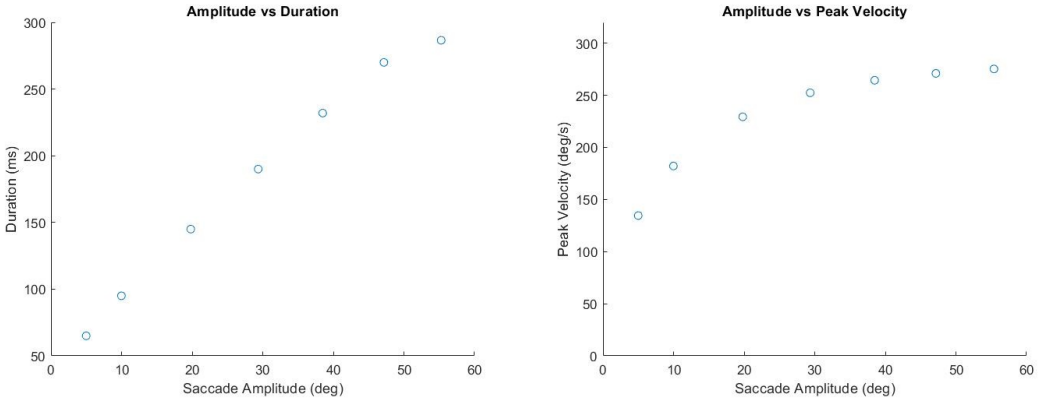


Figure 6.20: Main Sequence behavior tested on the controlled simulator.

more, the acceleration velocity is similar for all amplitudes, which means that the difference of duration is due to the deceleration phase. This is the skewness of saccades. So, higher amplitude saccades have the same peak velocity, but take longer to reach equilibrium. The controlled simulator respects this behavior, as seen in figure 6.21. Leftward horizontal saccades with increasing amplitude were performed in order to test the skewness of saccades.

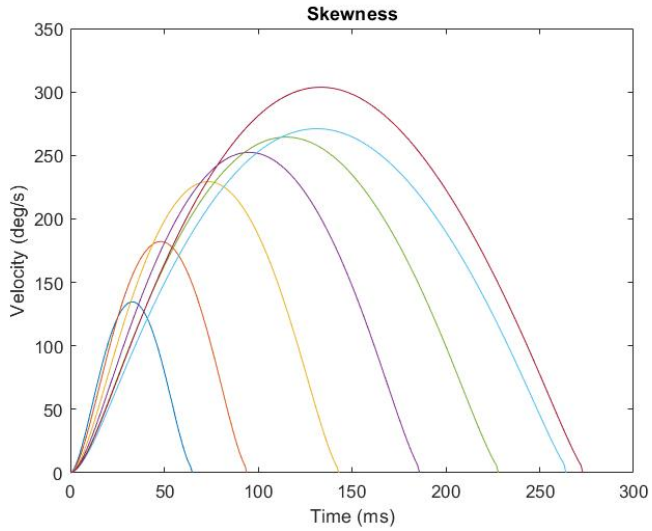


Figure 6.21: Skewness of velocity profiles of saccades with increasing amplitudes. The rotations were all horizontal, starting from straight ahead orientation, and their magnitudes were, respectively 5, 10, 20, 30, 40, 50 and 55 degrees.

Since it was aimed to build a biomimetic eye model, the simulator had to follow the properties of oculomotor system, both mechanic and dynamical ones. This approach allows us to obtain results that are in accordance to actual eye behavior, while having a low accuracy error.

6.3.3 Muscle Responses

The objective of this work is to study neural control, resorting to robotic models. Hence, a relevant test is to check how the inputs make the muscle force change. In the actual system, it has been shown that the neural signals that make the eye perform saccades are a combination between a pulse and a step

signal. This means the muscle force should have a similar behavior. The results of this test can be observed in figure 6.22.

these are absolute values; should be signed…
and better look at the torques

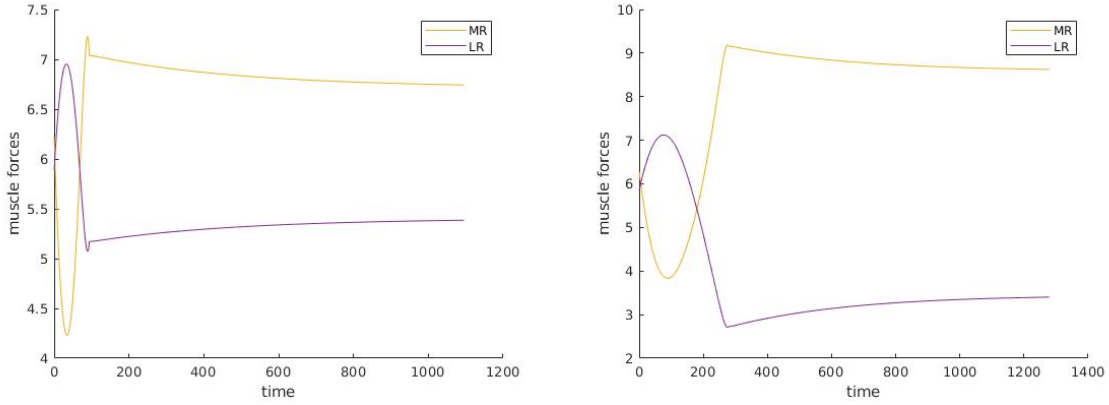


Figure 6.22: Medial and Lateral rectus response in N to the commands that lead to rightwards horizontal saccades of 10 degres (left) and 55 degrees (right). Time axis is in ms.

The initial forces are due to the pre-tension applied to the system so that the saccades would not have slack. The saccades performed for this test were negative horizontal rotations, which make the Medial Rectus act as the antagonist and the Lateral Rectus act as the agonist. It is obvious that since the right graphic is for a bigger saccade, it takes more time to stabilize, and its force has a bigger magnitude. Regarding the shape of the muscle response, it can be seen that there is a pulse form, followed by a step shape. This is in accordance to what is thought to be the shape of the neural input signals.

It's interesting to see that during the saccade, the antagonist muscle (MR) relaxes, exerting less force, but during the fixation time, it actually has to apply more force on the eye so that it fixates on the target (rightwards orientation).

I would like to understand this…; How would the plot be if you would show the net difference between the forces? LR - MR

Chapter 7

Conclusions

In this work, it was proposed to implement an open-loop saccadic control system applied to a three-dimensional model of the human eye. This control gave inputs to the system to make the eye reach a goal orientation, by minimizing a cost function. This comes as a way to study the terms optimized by the brain to perform saccadic movements. The terms present in this function directly influence the mechanic and dynamic properties of the eye, so it was important to choose carefully the terms to be minimized. The human eye's attributes aimed to mimic were the obedience of Listing's law and the non-linear dynamic properties shown to be followed by the human and primate oculomotor system (main sequence and skewness of the velocity profiles). Results were compared with empirical data on the human's eye behavior.

The model was developed completely in Matlab. Extraocular muscles (EOM) geometry was inspired in the human eye, resulting in an asymmetrical model. These six muscles were modelled as six independent elastic strings whose length changes based on individual motor rotations. The dynamics of the model were derived from rigid body's rotation equations, along with simple physics modelling of the springs. Even though a mechanical prototype with similar EOM was built recently in the ORIENT project, it was chosen to use a numerical simulator, since this way it was possible to iterate parameters like damping coefficients or inertial properties.

The approximation of the non-linear simulator by a linear state-space model was done using a variation based analytical linearization. For a small perturbation around the linearization point, this technique provided great results. However, by straying away from the initial point, its results were increasingly less satisfactory. Even so, this type of approximation allows to linearize the system at any point, even during the eye's movement, which creates the possibility of having linear approximations closer to the non-linear system.

One of the objectives of this work was to study the conditions that led the system to have slack. In order to do this several approaches were analysed. The one that mitigated this phenomenon was a memory-based approach, which might be a biologically valid approach, since it is reasonable to assume that it is easier for the brain to learn from experience an inverse kinematic map of the eye, than solving a non-linear system of equations at every single eye movement. The storage of optimal commands that

lead to a specific eye orientation is a more realistic assumption.

The approaches tested resulted in a thin Listing's plane. However, slack tended to thicken said plane. Therefore, by erasing slack, we were able to reduce Listing's plane thickness, making the model more biomimetic. The non-linear properties that the human eye follows were also obtained using this approach, mainly due to the minimization of the energy and duration for each saccade.

Several cost terms were also studied, in order to control the eye. The best results were obtained by minimizing a cost function with terms on energy, duration, equilibrium and command accuracy.

7.1 Contributions

The main contributions of this work are:

1. Analytical variation-based linearization, making possible a linear approximation in any orientation, during trajectories.
2. Geometrically realistic eye saccadic simulator, with simplified head modelling.
3. Empirical proof of 3D asymmetric saccadic system control using optimal open-loop control considering slack.
4. Developing of a graphical simulator.

7.2 Future Work

After this work, some paths are opened to explore. A similar model could be used to study the influence of noise in the control signal, in order to get closer to understanding the neural control of the oculomotor system. Also, a feedback controller would allow to use the linearization made here to get a linearized system at each orientation of each trajectory performed, and it would also help reducing accuracy error. Here, a head was modelled but not controlled. This is an important next step, in order to get a more complete auditory system by controlling the eye and head coupled system.

Bibliography

[1] C. M. Harris and D. M. Wolpert. The main sequence of saccades optimizes speed-accuracy trade-off. *Biological cybernetics*, 95(1):21–29, 2006.

[2] R. Cardoso. Feedback control of saccades on a model of a 3d biomimetic robot eye. Master's thesis, Instituto Superior Técnico, 2019.

[3] C. Aleluia. Control of saccades with model of artificial 3d biomimetic eye. Master's thesis, Instituto Superior Técnico, 2019.

[4] M. Lucas. Construction and characterization of a biomimetic robotic eye model with three degrees of rotational freedom. Master's thesis, Instituto Superior Técnico, 2017.

[5] J. M. Miller and D. A. Robinson. A model of the mechanics of binocular alignment. *Computers and Biomedical Research*, 17(5):436–470, 1984.

[6] G. Westheimer and S. P. McKee. Visual acuity in the presence of retinal-image motion. *JOSA*, 65 (7):847–850, 1975.

[7] E. Chekaluk and K. R. Llewellyn. Masking effects in saccadic eye movements. In *Studies in Visual Information Processing*, volume 5, pages 45–54. Elsevier, 1994.

[8] J. A. Cromer and D. M. Waitzman. Neurones associated with saccade metrics in the monkey central mesencephalic reticular formation. *The Journal of physiology*, 570(3):507–523, 2006.

[9] R. W. Baloh, A. W. Sills, W. E. Kumley, and V. Honrubia. Quantitative measurement of saccade amplitude, duration, and velocity. *Neurology*, 25(11):1065–1065, 1975.

[10] J. Van Opstal. *The auditory system and human sound-localization behavior*. Academic Press, 2016.

[11] F. C. Donders and W. D. Moore. *On the anomalies of accommodation and refraction of the eye: With a preliminary essay on physiological dioptrics*, volume 22. New Sydenham Society, 1864.

[12] D. Tweed and T. Vilis. Implications of rotational kinematics for the oculomotor system in three dimensions. *Journal of Neurophysiology*, 58(4):832–849, 1987.

[13] J. van Opstal. *The Gaze Control System*, pages 47–95. Springer New York, New York, NY, 2002. ISBN 978-0-387-21703-1. doi: 10.1007/978-0-387-21703-1_2. URL https://doi.org/10.1007/978-0-387-21703-1_2.

[14] M. R. Harwood, L. E. Mezey, and C. M. Harris. The spectral main sequence of human saccades. *Journal of Neuroscience*, 19(20):9098–9106, 1999.

[15] H. Collewijn, C. J. Erkelens, and R. M. Steinman. Binocular co-ordination of human horizontal saccadic eye movements. *The Journal of physiology*, 404(1):157–182, 1988.

[16] S.-N. Yang and G. W. McConkie. Eye movements during reading: A theory of saccade initiation times. *Vision research*, 41(25-26):3567–3585, 2001.

[17] S. Sarabandi and F. Thomas. A survey on the computation of quaternions from rotation matrices. *Journal of Mechanisms and Robotics*, 11(2), 2019.

[18] S. L. Altmann. *Rotations, quaternions, and double groups*. Courier Corporation, 2005.

[19] S. Traversaro and A. Saccon. Multibody dynamics notation (version 2). 2019.

[20] H. Goossens and A. J. van Opstal. Optimal control of saccades by spatial-temporal activity patterns in the monkey superior colliculus. *PLoS Comput Biol*, 8(5):e1002508, 2012.

[21] H. Goossens and A. Van Opstal. Dynamic ensemble coding of saccades in the monkey superior colliculus. *Journal of neurophysiology*, 95(4):2326–2341, 2006.

[22] R. Shadmehr and S. Mussa-Ivaldi. *Biological learning and control: how the brain builds representations, predicts events, and makes decisions*. Mit Press, 2012.

[23] R. Shadmehr and S. Mussa-Ivaldi. *Biological learning and control: how the brain builds representations, predicts events, and makes decisions*, chapter 11. Mit Press, 2012.

[24] A. Karami, M. Eghtesad, and S. A. Haghpanah. Prediction of muscle activation for an eye movement with finite element modeling. *Computers in Biology and Medicine*, 89:368–378, 2017.

[25] G. Cannata, M. D'Andrea, and M. Maggiali. Design of a humanoid robot eye: models and experiments. In *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pages 151–156. IEEE, 2006.

[26] D. Tweed. Three-dimensional model of the human eye-head saccadic system. *Journal of Neurophysiology*, 77(2):654–666, 1997.

[27] E. S. Maini, L. Manfredi, C. Laschi, and P. Dario. Bioinspired velocity control of fast gaze shifts on a robotic anthropomorphic head. *Autonomous Robots*, 25(1-2):37–58, 2008.

[28] Q. Wei. *Biomechanical modeling and simulation of human eye movement*. Citeseer, 2010.

[29] D. Biamino, G. Cannata, M. Maggiali, and A. Piazza. Mac-eye: A tendon driven fully embedded robot eye. In *5th IEEE-RAS International Conference on Humanoid Robots, 2005.*, pages 62–67. IEEE, 2005.

[30] D. A. Robinson. A quantitative analysis of extraocular muscle cooperation and squint. *Investigative Ophthalmology & Visual Science*, 14(11):801–825, 1975.

[31] H. Goldstein, C. Poole, and J. Safko. *Classical mechanics*. American Association of Physics Teachers, 2002.

[32] G. Wu and K. Sreenath. Variation-based linearization of nonlinear systems evolving on $so(3)$ and s^2, 2015. *IEEE Access*, 3:1592–1604, 2015.

[33] Y. Yu and X. Ding. Trajectory linearization control on so(3) with application to aerial manipulation. *Journal of the Franklin Institute*, 355(15):7072 – 7097, 2018. ISSN 0016-0032. doi: https://doi.org/10.1016/j.jfranklin.2018.06.043. URL `http://www.sciencedirect.com/science/article/pii/S0016003218305222`.

[34] A. V. Oppenheim and D. H. Johnson. Discrete representation of signals. *Proceedings of the IEEE*, 60(6):681–691, 1972.

[35] J. Myerson and L. Green. Discounting of delayed rewards: Models of individual choice. *Journal of the experimental analysis of behavior*, 64(3):263–276, 1995.

[36] M. Quigley, J. Faust, T. Foote, and J. Leibs. Ros: an open-source robot operating system.

[37] C. Sforza, G. Grassi, N. Fragnito, M. Turci, and V. F. Ferrario. Three-dimensional analysis of active head and cervical spine range of motion: effect of age in healthy male subjects. *Clinical Biomechanics*, 17(8):611–614, 2002.

[38] G. Peng, T. Hain, and B. Peterson. A dynamical model for reflex activated head movements in the horizontal plane. *Biological cybernetics*, 75(4):309–319, 1996.

Chapter 8

Appendix

Constant values used in this work are shown in the table below.

Physical properties		
	Eye	Neck
Stiffness (N)		
K_{xx}	20	60
K_{yy}	20	30
K_{zz}	20	50
Inertia (kg/m^2)		
I_{xx}	0.00048	0.181
I_{yy}	0.00043	0.215
I_{zz}	0.00039	0.142
Damping (N m s)		
D_{xx}	0.04	1.1
D_{yy}	0.04	1.1
D_{zz}	0.04	1.1

Table 8.1: Constants for eye and head used in this work.

8.1 Differentiation

From 4.18 we get

$$\frac{\partial f_1}{\partial \boldsymbol{\eta}_e} = -\,{}^{e}\boldsymbol{\omega}^{\wedge}_{w,e} \tag{8.1}$$

$$\frac{\partial f_1}{\partial \boldsymbol{\eta}_h} = 0 \tag{8.2}$$

$$\frac{\partial f_1}{\partial\, {}^{e}\boldsymbol{\omega}_{w,e}} = \mathbf{I}_{3\times 3} \tag{8.3}$$

$$\frac{\partial f_1}{\partial\, {}^{h}\boldsymbol{\omega}_{w,h}} = 0 \tag{8.4}$$

and from 4.19 we get

$$\frac{\partial f_2}{\partial \boldsymbol{\eta}_e} = 0 \tag{8.5}$$

$$\frac{\partial f_2}{\partial \boldsymbol{\eta}_h} = -\,{}^{h}\boldsymbol{\omega}^{\wedge}_{w,h} \tag{8.6}$$

$$\frac{\partial f_2}{\partial\, {}^{e}\boldsymbol{\omega}_{w,e}} = 0 \tag{8.7}$$

$$\frac{\partial f_2}{\partial\, {}^{h}\boldsymbol{\omega}_{w,h}} = \mathbf{I}_{3\times 3} \tag{8.8}$$

Regarding the next state equation, if we expand 3.22 we have

$${}^{e}\dot{\omega}_{w,e} = {}_{e}I_{e}{}^{-1}({}_{e}\tau_k + {}_{e}\tau_d - {}^{e}\omega_{w,e} \times {}_{e}I_{e}\,{}^{e}\omega_{w,e}) \tag{8.9}$$

$$= {}_{e}I_{e}{}^{-1}(\sum_{i=1}^{6}({}^{e}Q_i \times \frac{k}{l_{0i}}(l_i - l_{0i})\,{}_{e}\vec{v_i}) - D_{eye}\,{}^{e}\omega_{h,e} - {}^{e}\omega_{w,e} \times {}_{e}I_{e}\,{}^{e}\omega_{w,e}) \tag{8.10}$$

The most complex derivative is the one of the direction vector $\vec{\phi_i}$ w.r.t our first two state variables $\boldsymbol{\eta}_e$ and $\boldsymbol{\eta}_h$. Thus, we will explore this derivative a bit deeper.

$${}_{e}\vec{\boldsymbol{\phi}}_i = \frac{({}^{e}R_h\,{}^{h}S - {}^{e}Q)}{\left\|{}^{e}R_h\,{}^{h}S - {}^{e}Q\right\|} = ({}^{e}R_h\,{}^{h}S - {}^{e}Q)\frac{1}{\left\|{}^{e}R_h\,{}^{h}S - {}^{e}Q\right\|} \tag{8.11}$$

$$\frac{\partial\, {}_{e}\vec{\phi}}{\partial \eta_e} = \frac{\partial}{\partial \eta_e}({}^{e}R_h\,{}^{h}S - {}^{e}Q)\frac{1}{\left\|{}^{e}R_h\,{}^{h}S - {}^{e}Q\right\|} + ({}^{e}R_h\,{}^{h}S - {}^{e}Q)\frac{\partial}{\partial \eta_e}\frac{1}{\left\|{}^{e}R_h\,{}^{h}S - {}^{e}Q\right\|} \tag{8.12}$$

Let us focus on the first term of the previous equation, $\frac{\partial}{\partial \boldsymbol{\eta}_e}({}^{e}\mathbf{R}_h\,{}^{h}\mathbf{S} - {}^{e}\mathbf{Q})$, and remember Eq.(3.29). This way, we can write that first term as:

$$\frac{\partial}{\partial \eta_e}({}^{e}R_h\,{}^{h}S - {}^{e}Q) = \frac{\partial}{\partial \eta_e}({}^{e}R_w\,{}^{w}R_h\,{}^{h}S - {}^{e}Q) \tag{8.13}$$

We wish to apply a perturbation on the eye axis, and as such, we need to apply it in the rotation matrix of the eye in the world frame.

$$\frac{\partial}{\partial \eta_e}(({}^{h}R_w\,{}^{w}R_e\tilde{R}_e)^{T}\,{}^{h}S - {}^{e}Q) = \frac{\partial}{\partial \eta_e}(\tilde{R}_e^{T}\,{}^{e}R_w\,{}^{w}R_h\,{}^{h}S - {}^{e}Q) \tag{8.14}$$

Our perturbation, according to a Taylor Series expansion can be seen as

$$\tilde{\mathbf{R}}(t) = \exp(\boldsymbol{\eta}^{\wedge}) = \mathbf{I} + \boldsymbol{\eta}^{\wedge} + \mathcal{O}(\left\|\tilde{\mathbf{x}} - \bar{\tilde{\mathbf{x}}}\right\|^2) \tag{8.15}$$

Remember that the transpose of a skew-symmetric matrix is it's symmetric. Thus, from (8.15), we can replace $\tilde{\mathbf{R}}_e$ with

$$\frac{\partial}{\partial \eta_e}({}^{e}R_h\,{}^{h}S - {}^{e}Q) = -(I_{3\times 3} + \eta_e^{\wedge} + \mathcal{O}(\tilde{\eta_e}^2))\,{}^{e}R_h\,{}^{h}S - {}^{e}Q \tag{8.16}$$

We can dispose of the terms that don't depend on $\boldsymbol{\eta}_e$ (since their derivatives would be 0), simplify the ones that do depend on this variable and neglect the higher order terms since we aim to get linear expressions, leaving us with:

$$\frac{\partial}{\partial \eta_e}({}^{e}R_h\,{}^{h}S - {}^{e}Q) = -\eta_e^{\wedge}\,{}^{e}R_h\,{}^{h}S = ({}^{e}R_h\,{}^{h}S)^{\wedge} \tag{8.17}$$

Let us now focus on the second term from Eq.(8.12). However, we are making this term simpler. So, let us find the derivative of the norm and, in the end, invert it.

$$\begin{aligned}\frac{\partial}{\partial \eta_e}\left\|{}^{e}R_h\,{}^{h}S - {}^{e}Q\right\| = \frac{\partial}{\partial \eta_e}(({}^{e}R_h\,{}^{h}S - {}^{e}Q)^T({}^{e}R_h\,{}^{h}S - {}^{e}Q))^{\frac{1}{2}} = \\ \frac{1}{2}\frac{1}{\left\|{}^{e}R_h\,{}^{h}S - {}^{e}Q\right\|}\frac{\partial}{\partial \eta_e}(({}^{e}R_h\,{}^{h}S - {}^{e}Q)^T({}^{e}R_h\,{}^{h}S - {}^{e}Q))\end{aligned} \tag{8.18}$$

Let's take the last term of the previous equation.

$$\begin{aligned}&\frac{\partial}{\partial \eta_e}(({}^{e}R_h\,{}^{h}S - {}^{e}Q)^T({}^{e}R_h\,{}^{h}S - {}^{e}Q)) = \\ &\frac{\partial}{\partial \eta_e}(-{}^{e}Q^T\,{}^{e}R_h\,{}^{h}S + {}^{e}Q^T\,{}^{e}Q + {}^{h}S^T\,{}^{h}R_e\,{}^{e}R_h\,{}^{h}S - {}^{h}S^T\,{}^{h}R_e\,{}^{e}Q)\end{aligned} \tag{8.19}$$

Again, let's consider only the terms that depend on ${}^{w}\mathbf{R}_e$, because it's where our perturbation is applied. As such, the other terms' derivative will be zero, including the higher order terms, since we wish to have, exclusively, linear terms. Furthermore, the square of the perturbation results in a minimal value. Given these reasons we can consider these terms negligible.

Let us also expand our perturbation variable.

$$\frac{\partial}{\partial \eta_e}(-{}^{e}Q^T\,{}^{e}R_h\,{}^{h}S - {}^{h}S^T\,{}^{h}R_e\,{}^{e}Q) = \tag{8.20}$$

$$= \frac{\partial}{\partial \eta_e}(-{}^{e}Q^T({}^{h}R_w\,{}^{w}R_e\tilde{R}_e)^T\,{}^{h}S - {}^{h}S^T\,{}^{h}R_w\,{}^{w}R_e\tilde{R}_e\,{}^{e}Q) = \tag{8.21}$$

$$= \frac{\partial}{\partial \eta_e}\,{}^{e}Q^T(I_{3\times 3} + \eta_e^{\wedge} + \mathcal{O}(\tilde{\eta_e}^2))\,{}^{e}R_h\,{}^{h}S - {}^{h}S^T\,{}^{h}R_e(I_{3\times 3} + \eta_e^{\wedge} + \mathcal{O}(\tilde{\eta_e}^2))\,{}^{e}Q = \tag{8.22}$$

$$= \frac{\partial}{\partial \eta_e}\,{}^{e}Q^T\eta_e^{\wedge}\,{}^{e}R_h\,{}^{h}S - {}^{h}S^T\,{}^{h}R_e\eta_e^{\wedge}\,{}^{e}Q = \tag{8.23}$$

$$= -\frac{\partial}{\partial \eta_e}\,{}^{e}Q^T({}^{e}R_h\,{}^{h}S)^{\wedge}\eta_e + {}^{h}S^T\,{}^{h}R_e\,{}^{e}Q^{\wedge}\eta_e \tag{8.24}$$

According to (**??**), we get

$$\frac{\partial}{\partial \eta_e}((^eR_h\,{}^hS - {}^eQ)^T(^eR_h\,{}^hS - {}^eQ)) = -\,{}^eQ^T(^eR_h\,{}^hS)^\wedge + {}^hS^T\,{}^hR_e\,{}^eQ^\wedge \tag{8.25}$$

The terms from Eq.(8.25) are actually the same term written in a different form. Therefore, we have

$$\frac{\partial}{\partial \eta_e}((^eR_h\,{}^hS - {}^eQ)^T(^eR_h\,{}^hS - {}^eQ)) = 2\,{}^hS^T\,{}^hR_e\,{}^eQ^\wedge \tag{8.26}$$

As it was said before, we have to differentiate the inverse of the norm and not the norm itself. So,

$$\frac{\partial}{\partial \eta_e}\frac{1}{\left\|{}^eR_h\,{}^hS - {}^eQ\right\|} = -\frac{\frac{\partial}{\partial \eta_e}\left\|{}^eR_h\,{}^hS - {}^eQ\right\|}{\left\|{}^eR_h\,{}^hS - {}^eQ\right\|^2} = -\frac{2\,{}^hS^T\,{}^hR_e\,{}^eQ^\wedge}{\left\|{}^eR_h\,{}^hS - {}^eQ\right\|^2} \tag{8.27}$$

Putting the whole equation together, from (8.12), we get

$$\frac{\partial\,{}_e\vec{\phi}}{\partial \eta_e} = \frac{(^eR_h\,{}^hS)^\wedge}{\left\|{}^eR_h\,{}^hS - {}^eQ\right\|} - \frac{1}{2}\frac{(^eR_h\,{}^hS - {}^eQ)}{\left\|{}^eR_h\,{}^hS - {}^eQ\right\|}\frac{2\,{}^hS^T\,{}^hR_e\,{}^eQ^\wedge}{\left\|{}^eR_h\,{}^hS - {}^eQ\right\|^2} \tag{8.28}$$

We can simplify to

$$\frac{\partial\,{}_e\vec{\phi}}{\partial \eta_e} = \frac{(^eR_h\,{}^hS)^\wedge}{\left\|{}^eR_h\,{}^hS - {}^eQ\right\|} - \frac{(^eR_h\,{}^hS - {}^eQ)}{\left\|{}^eR_h\,{}^hS - {}^eQ\right\|^3}(^hS^T\,{}^hR_e\,{}^eQ^\wedge) \tag{8.29}$$

Our full derivative is:

$$\frac{\partial f_3}{\partial \boldsymbol{\eta}_e} = {}_e\mathbf{I}_e{}^{-1}(\sum_{i=1}^{6}(^e\mathbf{Q}_i \times ((^e\mathbf{R}_h\,{}^h\mathbf{S}_i)^\wedge\frac{k}{l0_i} + \frac{k}{l0_i}(r\theta_i - l0_i)\frac{\partial\,{}_e\vec{\boldsymbol{\phi}}}{\partial \boldsymbol{\eta}_e})) + \mathbf{D}_{eye}(^e\mathbf{R}_h\,{}^h\boldsymbol{\omega}_{w,h})^\wedge) \tag{8.30}$$

We now wish to differentiate the same state equation, with respect to the second local state variable $\boldsymbol{\eta}_h$. Once again, the most complex step is to differentiate the elastic torque's direction vector. The logic is the same as before. However, the variable to which we apply the perturbation changes to ${}^w\mathbf{R}_h$, since we are aiming to linearize the error dynamics for the eye with respect to the head orientation, at this point. The initial expansion is exactly the same as before, the first difference appears when we apply the perturbation. Therefore, let us start from equation (8.12).

$$\frac{\partial\,{}_e\vec{\phi}}{\partial \eta_h} = \frac{\partial}{\partial \eta_h}(^eR_h\,{}^hS - {}^eQ)\frac{1}{\left\|{}^eR_h\,{}^hS - {}^eQ\right\|} + (^eR_h\,{}^hS - {}^eQ)\frac{\partial}{\partial \eta_h}\frac{1}{\left\|{}^eR_h\,{}^hS - {}^eQ\right\|} \tag{8.31}$$

Again, let's use the divide and conquer technique. First, let's focus on the first term's differentiation

$$\frac{\partial}{\partial \eta_h}(^eR_h\,{}^hS - {}^eQ) = \frac{\partial}{\partial \eta_h}(^eR_w\,{}^wR_h\,{}^hS - {}^eQ) \tag{8.32}$$

Now, let us apply and expand the perturbation term

$$\frac{\partial}{\partial \eta_h}({}^eR_w\,{}^wR_h\tilde{R}_h\,{}^hS - {}^eQ) = \frac{\partial}{\partial \eta_h}\,{}^eR_h(I_{3\times 3} + \eta_h^\wedge + \mathcal{O}(\tilde{\eta_h}^2))\,{}^hS - {}^eQ \tag{8.33}$$

Once again, we shall neglect the terms that don't depend on our local state variable, as well as higher order terms. This way, we end up with

$$\frac{\partial}{\partial \eta_h}({}^eR_h\,{}^hS - {}^eQ) = \frac{\partial}{\partial \eta_h}({}^eR_h\eta_h^\wedge\,{}^hS) = -\frac{\partial}{\partial \eta_h}({}^eR_h\,{}^hS^\wedge\eta_h) = -\,{}^eR_h\,{}^hS^\wedge \tag{8.34}$$

Like before, let's differentiate the norm and then invert it.

$$\begin{aligned}\frac{\partial}{\partial \eta_h}\left\|{}^eR_h\,{}^hS - {}^eQ\right\| = \frac{\partial}{\partial \eta_h}(({}^eR_h\,{}^hS - {}^eQ)^T({}^eR_h\,{}^hS - {}^eQ))^{\frac{1}{2}} = \\ \frac{1}{2}\frac{1}{\left\|{}^eR_h\,{}^hS - {}^eQ\right\|}\frac{\partial}{\partial \eta_h}(({}^eR_h\,{}^hS - {}^eQ)^T({}^eR_h\,{}^hS - {}^eQ))\end{aligned} \tag{8.35}$$

The expansion of the above equation is the same as before (Eq.(8.18)). Therefore let's start from Eq.(8.19) and apply the perturbation to it

$$\frac{\partial}{\partial \eta_h}\left\|{}^eR_h\,{}^hS - {}^eQ\right\| = \frac{\partial}{\partial \eta_h}(-\,{}^eQ^T\,{}^eR_h\,{}^hS - {}^hS^T\,{}^hR_e\,{}^eQ) = \tag{8.36}$$

$$=\frac{\partial}{\partial \eta_h}(-\,{}^eQ^T\,{}^eR_h\tilde{R}_h\,{}^hS - {}^hS^T({}^wR_h\tilde{R}_h)^T\,{}^wR_e\,{}^eQ) = \tag{8.37}$$

$$=\frac{\partial}{\partial \eta_h}(-\,{}^eQ^T\,{}^eR_h\tilde{R}_h\,{}^hS - {}^hS^T\tilde{R}_h^T\,{}^hR_e\,{}^eQ) = \tag{8.38}$$

$$=\frac{\partial}{\partial \eta_h}(-\,{}^eQ^T\,{}^eR_h(I_{3\times 3} + \eta_h^\wedge + \mathcal{O}(\tilde{\eta_h}^2))\,{}^hS + {}^hS^T(I_{3\times 3} + \eta_h^\wedge + +\mathcal{O}(\tilde{\eta_h}^2))\,{}^hR_e\,{}^eQ) \tag{8.39}$$

Let us disregard the terms that do not depend on our state variable. Hence, we get

$$\frac{\partial}{\partial \eta_h}(-\,{}^eQ^T\,{}^eR_h\eta_h^\wedge\,{}^hS + {}^hS^T\eta_h^\wedge\,{}^hR_e\,{}^eQ) = \tag{8.40}$$

$$= \frac{\partial}{\partial \eta_h}({}^eQ^T\,{}^eR_h\,{}^hS^\wedge\eta_h - {}^hS^T({}^hR_e\,{}^eQ)^\wedge\eta_h) = \tag{8.41}$$

$$= \frac{\partial}{\partial \eta_h}({}^eQ^T\,{}^eR_h\,{}^hS^\wedge - {}^hS^T({}^hR_e\,{}^eQ)^\wedge) = \tag{8.42}$$

$$= 2\,{}^eQ^T\,{}^eR_h\,{}^hS^\wedge \tag{8.43}$$

As for the inverse of the norm, we get

$$\frac{\partial}{\partial \eta_h}\frac{1}{\left\|{}^eR_h\,{}^hS - {}^eQ\right\|} = -\frac{\frac{\partial}{\partial \eta_h}\left\|{}^eR_h\,{}^hS - {}^eQ\right\|}{\left\|{}^eR_h\,{}^hS - {}^eQ\right\|^2} = -\frac{2\,{}^eQ^T\,{}^eR_h\,{}^hS^\wedge}{\left\|{}^eR_h\,{}^hS - {}^eQ\right\|^2} \tag{8.44}$$

Putting the whole equation together, from (8.12), we get

$$\frac{\partial\,{}_e\vec{\phi}}{\partial \eta_h} = \frac{-\,{}^eR_h\,{}^hS^\wedge}{\left\|{}^eR_h\,{}^hS - {}^eQ\right\|} - \frac{1}{2}\frac{({}^eR_h\,{}^hS - {}^eQ)}{\left\|{}^eR_h\,{}^hS - {}^eQ\right\|}\frac{2\,{}^eQ^T\,{}^eR_h\,{}^hS^\wedge}{\left\|{}^eR_h\,{}^hS - {}^eQ\right\|^2} \tag{8.45}$$

We can simplify to

$$\frac{\partial\, {}_e\vec{\phi}}{\partial \eta_h} = \frac{-\,{}^eR_h\,{}^hS^\wedge}{\left\|{}^eR_h\,{}^hS - {}^eQ\right\|} - \frac{({}^eR_h\,{}^hS - {}^eQ)}{\left\|{}^eR_h\,{}^hS - {}^eQ\right\|^3}({}^eQ^T\,{}^eR_h\,{}^hS^\wedge) \tag{8.46}$$

Our full derivative is

$$\frac{\partial f_3}{\partial \boldsymbol{\eta}_h} = -\,{}_e\mathbf{I}_e{}^{-1}(\sum_{i=1}^{6}({}^e\mathbf{Q}_i \times ({}^e\mathbf{R}_h\,{}^h\mathbf{S}_i{}^\wedge \frac{k}{l0_i} + \frac{k}{l0_i}(r\theta_i - l0_i)\frac{\partial\, {}_e\vec{\boldsymbol{\phi}}}{\partial \boldsymbol{\eta}_h})) + \mathbf{D}_{eye}\,{}^e\mathbf{R}_h\,{}^h\boldsymbol{\omega}_{w,h}{}^\wedge) \tag{8.47}$$

An equality that is relevant for the next few steps is the following

$$\frac{\partial \omega \times I_e\omega}{\partial \omega} = \frac{\partial}{\partial \omega}(\omega^\wedge I_e\omega) = \frac{\partial}{\partial \omega}(f(\omega)I_e\omega) = \frac{\partial}{\partial \omega}(f(\omega))I_e\omega + f(\omega)I_e \tag{8.48}$$

Doing a element-wise derivative, we get

$$\frac{\partial}{\partial \omega}(f(\omega))I_e\omega = \begin{bmatrix} \frac{\partial f}{\partial \omega_x}I_e\omega & \frac{\partial f}{\partial \omega_y}I_e\omega & \frac{\partial f}{\partial \omega_z}I_e\omega \end{bmatrix} = ((I_e\omega)^\wedge)^T \tag{8.49}$$

Knowing that the transpose of a skew symmetric matrix corresponds to its symmetric, it results in

$$\frac{\partial \omega \times I_e\omega}{\partial \omega} = \omega^\wedge I_e - (I_e\omega)^\wedge \tag{8.50}$$

So, from (8.10) and (8.50) we can write

$$\frac{\partial f_3}{\partial\, {}^e\omega_{w,e}} = {}_eI_e{}^{-1}(D_{eye} + {}^e\omega_{w,e}{}^\wedge\, {}_eI_e - ({}_eI_e\,{}^e\omega_{w,e})^\wedge) \tag{8.51}$$

Also, knowing that

$${}^e\omega_{h,e} = {}^e\omega_{w,e} - {}^eR_h\,{}^h\omega_{w,h} = -\,{}^e\omega_{e,h} \tag{8.52}$$

we derive

$$\frac{\partial f_3}{\partial\, {}^w\omega_{w,h}} = {}_eI_e{}^{-1}(D_{eye}\,{}^eR_h) \tag{8.53}$$

All that's left are the partial derivatives of the last state function. Following the same logic as before, if we expand 3.23, we get

$${}^h\dot{\omega}_{w,h} = {}_hI_h{}^{-1}({}_h\tau_{eye-head} + {}_h\tau_{dneck} + {}_h\tau_{kneck} + {}_h\tau_{ext} - {}^h\omega_{w,h} \times {}_hI_h\,{}^h\omega_{w,h}) \tag{8.54}$$

$$= {}_hI_h{}^{-1}(\sum_{i=1}^{6}({}^hS_i \times \frac{k}{l_{0i}}(l_i - l_{0i})\,{}_h\vec{v_i}) - D_{neck}({}^h\omega_{w,h}) \tag{8.55}$$

$$- K_{neck}\log({}^wR_h) + {}_h\tau_{ext} - {}^h\omega_{w,h} \times {}_hI_h\,{}^h\omega_{w,h}) \tag{8.56}$$

The differentiation of the equation above with respect to the first local state variable is

$$\frac{\partial f_4}{\partial \boldsymbol{\eta}_e} = -\,{}_h\mathbf{I}_h{}^{-1}(\sum_{i=1}^{6}({}^h\mathbf{S}_i \times ({}^h\mathbf{R}_e\,{}^e\mathbf{Q}_i{}^\wedge \frac{k}{l0_i} + \frac{k}{l0_i}(r\theta_i - l0_i)\frac{\partial\,{}_h\vec{\phi}}{\partial \boldsymbol{\eta}_e})) + \mathbf{D}_{eye}\,{}^h\mathbf{R}_e\,{}^e\boldsymbol{\omega}_{h,e}{}^\wedge) \tag{8.57}$$

where

$$\frac{\partial\,{}_h\vec{\phi}}{\partial \eta_e} = \frac{\partial}{\partial \eta_e}\frac{({}^hS - {}^hR_e\,{}^eQ)}{\left\|{}^hS - {}^hR_e\,{}^eQ\right\|} \qquad = \frac{{}^hR_e\,{}^eQ^\wedge}{\left\|{}^hS - {}^hR_e\,{}^eQ\right\|} - \frac{({}^hS - {}^hR_e\,{}^eQ)}{\left\|{}^hS - {}^hR_e\,{}^eQ\right\|^3}({}^hS^T\,{}^hR_e\,{}^eQ^\wedge) \tag{8.58}$$

The differentiation of the fourth state function with respect to the second state variable is

$$\frac{\partial f_4}{\partial \boldsymbol{\eta}_h} = {}_h\mathbf{I}_h{}^{-1}(\sum_{i=1}^{6}({}^h\mathbf{S}_i \times ({}^h\mathbf{R}_e\,{}^e\mathbf{Q}_i{}^\wedge \frac{k}{l0_i} + \frac{k}{l0_i}(r\theta_i - l0_i)\frac{\partial\,{}_h\vec{\phi}}{\partial \boldsymbol{\eta}_h})) + \mathbf{D}_{eye}({}^h\mathbf{R}_e\,{}^e\boldsymbol{\omega}_{h,e})^\wedge + \mathbf{K}_{neck}) \tag{8.59}$$

where

$$\frac{\partial\,{}_h\vec{\phi}}{\partial \eta_h} = \frac{\partial}{\partial \eta_h}\frac{({}^hS - {}^hR_e\,{}^eQ)}{\left\|{}^hS - {}^hR_e\,{}^eQ\right\|} = \frac{({}^hR_e\,{}^eQ)^\wedge}{\left\|{}^hS - {}^hR_e\,{}^eQ\right\|} + \frac{({}^hS - {}^hR_e\,{}^eQ)}{\left\|{}^hS - {}^hR_e\,{}^eQ\right\|^3}({}^hQ^T\,{}^eR_h\,{}^hS^\wedge) \tag{8.60}$$

From 8.56 and 8.50 (applied to the head) we can write the next entry of our Jacobian matrix as

$$\frac{\partial f_4}{\partial\,{}^e\omega_{w,e}} = {}_hI_h{}^{-1}(D_{eye}\,{}^hR_e) \tag{8.61}$$

and as for the final entry, we have

$$\frac{\partial f_4}{\partial\,{}^h\omega_{w,h}} = {}_hI_h{}^{-1}(D_{eye} + D_{neck} + {}^h\omega_{w,h}{}^\wedge\,{}_hI_h - ({}_hI_h\,{}^h\omega_{w,h})^\wedge)) \tag{8.62}$$

And for matrix **B**:

$$\frac{\partial f_1}{\partial \mathbf{u}} = 0 \tag{8.63}$$

$$\frac{\partial f_2}{\partial \mathbf{u}} = 0 \tag{8.64}$$

$$\frac{\partial f_3}{\partial \mathbf{u}} = {}_e\mathbf{I}_e{}^{-1}({}^e\mathbf{Q}_i{}^\wedge \frac{k}{l0_i} r \frac{{}^e\mathbf{R}_h\,{}^h\mathbf{S}_i - {}^e\mathbf{Q}_i}{\left\|{}^e\mathbf{R}_h\,{}^h\mathbf{S} - {}^e\mathbf{Q}\right\|}) \tag{8.65}$$

$$\frac{\partial f_4}{\partial \mathbf{u}} = {}_e\mathbf{I}_h{}^{-1}(-\,{}^h\mathbf{S}_i{}^\wedge \frac{k}{l0_i} r \frac{{}^h\mathbf{S} - {}^h\mathbf{R}_e\,{}^e\mathbf{Q}}{\left\|{}^h\mathbf{S} - {}^h\mathbf{R}_e\,{}^e\mathbf{Q}\right\|}) \tag{8.66}$$

The last three columns of matrix **B** will be zeros, except for

$$\mathbf{B}(10:12, 7:9) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

This is because of the external torque term for the head. Since this torque is applied directly on the motors, it is considered to be an input. So, the derivative of said torque is basically the identity matrix.