TÉCNICO LISBOA

# Feedback control of saccades on a model of a 3D biomimetic robot eye

**Rui Eduardo Marques Cardoso**

Thesis to obtain Master of Science Degree in

## Electrical and Computer Engineering

Supervisors: Prof. Alexandre José Malheiro Bernardino
Prof. Adrianus Johannes Van Opstal

### Examination Committee

Chairperson: Prof.João Fernando Cardoso Silva Sequeira
Supervisor: Prof. Alexandre José Malheiro Bernardino
Member of the Committee: Prof. Pedro Tiago Martins Batista

**December 2019**

# Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

To my aunt Aida.

# Acknowledgements

I would like to express appreciation to my supervisor at IST, Professor Alexandre Bernardino, for all the patience and expertise he transmited to me throughout the development of this thesis. Moreover, to Professor John Van Opstal from Radboud University for his support and to my fellow colleagues at ORIENT I project, Akhil John, Carlos Aleluia and Mariana Martins.

To all my friends in different stages of life and especially the ones who shared the university times with me. I am lucky to have all of those at SIIIIIIIIIIM, you all know how important this group is to all of us.

To the best housemates I could have asked for, Ricardo, Gui and João Guilherme. You were my second family these years, and I am confident that you will always be.

To Rita, for helping me keep my head up through all the hard times. You inspire me to be a better person everyday.

Finally, I want to express the deepest gratitude to the pillars of my life, my family. Thank you for all the faith you always had on me, and for never giving up on giving me what I need to follow my path. Obrigado pai, mãe, Simão e João. Tudo o que sou devo-o a ter-vos comigo.

# Abstract

The way the human brain controls movements is a widely studied subject. Considering the specific case of saccades, the trajectories the brain tends to choose from an infinite number of possible trajectories are highly stereotypical, in which non-linear dynamic properties are observed. Also, the observation that the eyes appear to use only two Degrees-of-freedom (DOF) to direct its line of sight, instead of the possible three, provided by six extraocular muscles, was a motivation for studying these movements. The plane in which the eyes move is called Listing's plane (LP). It has also been suggested that, the brain accounts for a particular kind of disturbance which could play an important role in these saccade properties, called Signal-dependent noise (SDN). Here, a study on how saccades are controlled in the presence of SDN assuming the existence of internal feedback is presented. Simulations were carried out with a 3D biomimetic robot eye developed previously which was adapted to include a more realistic muscle model, SDN and a feedback control loop. Different methods are tested to obtain a valid linear parameterization of the eye model, which is then used to control stochastically the model using different optimization principles forwarded in the literature through optimal feedback control. The non-linear dynamic properties were observed only under SDN conditions - the existence of this kind of noise is enough to replicate saccades optimizing accuracy and duration, disregarding the effort. The trajectories are fully contained in LP only if the final position is penalized for deviations from the LP. Finding this was independent of the type of noise used.

# Keywords

Biomimetic eye, Listing's plane, Main sequence, Optimal feedback control, Signal-dependent noise

# Resumo

A forma como o cérebro controla movimentos é um tema de estudo bastante explorado. No caso específico das sacadas, as trajectórias escolhidas pelo cérebro dentro de uma infinidade de possíveis trajetórias seguem um padrão bem definido, com propriedades dinâmicas não-lineares. Além disso, o facto de que os olhos se movem apenas em dois graus de liberdade dos três possíveis que lhes são concedidos por seis musculos extraoculares, é um factor motivacional para o estudo destes movimentos. O plano onde os olhos se movem é o plano de Listing (PL). Aparentemente, o cérebro conta com um tipo de perturbação distinto que é determinante nestas propriedades das sacadas, que é dependente do sinal (SDN). Um estudo acerca do controlo de sacadas na presença de SDN assumindo a presença de realimentação é apresentado nesta dissertação. Um modelo de um robot bimimético do olho em 3D previamente desenvolvido é adaptado para incluir um modelo de musculo realista, SDN e realimentação e subsequentemente usado para simulações. Diferentes métodos são testados para obter uma linearização válida do modelo, usada para controlá-lo estocasticamente usando diferentes principios de optimização através de controlo óptimo por realimentação. As propriedades não-lineares são observadas apenas sob influência de SDN - a existência deste tipo de ruído é suficiente para replicar sacadas optimizando apenas precisão e duração. As trajectórias são inteiramente contidas no PL apenas penalizando desvios do PL na posição final, independentemente do tipo de perturbação considerado.

# Palavras Chave

Olho biomimético, Plano de Listing, *Main sequence*, Controlo óptimo por realimentação, Ruído dependente do sinal

# Contents

# List of Figures

# List of Tables

# Acronyms

| | |
|---|---|
| **SDN** | Signal-dependent noise |
| **LP** | Listing's plane |
| **KF** | Kalman Filter |
| **IDS** | Identified System |
| **LS** | Linearized System |
| **NLS** | Nonlinear System |
| **DOF** | Degrees-of-freedom |
| **LQG** | Linear-Quadratic-Gaussian |

# 1

# Introduction

## Contents

## 1.1 Motivation

The way the human brain can control movements is a complex process about which researchers have been intrigued since the early 20th century. Controlling human-like movements by such a sophisticated system where both actuators and sensors are biological mechanisms which carry such characteristic properties is a challenging problem and provides much potential for further research. Biomimetic robotics is an area of research which aims for both providing new technological capabilities and better understanding the behaviour of biological systems. Here, the saccadic system is studied and, more precisely, an attempt to understand the reproduction of 3-dimensional eye movements is made by approaching the biological structure as a robotic system using engineering tools. The oculomotor system is composed of six extraocular muscles which act in pairs, the eyeball and many circuitries throughout the brain. It is responsible for different types of eye movements, namely the vestibular, optokinetic, smooth pursuit, saccadic, vergence and short latency. In all eye-movement types, there is the clear objective of keeping (pursuit, verstibular) or bringing (saccades, vergence) the point of interest centred on the *fovea*, the highest resolution part of the retina. Eye orientation is represented by rotations about three axis, specifying horizontal, vertical and torsional components. The saccadic movements are focused here, as from the point of view of dynamics these are the most challenging. Saccades are quick and precise movements performed simultaneously by the eyes when an abrupt change in the point of fixation is required. Remarkably, these follow stereotypical trajectories which are confined to a plane where torsion is 0 [4] - the Listing's plane - and with a consistent non-linear relationship between amplitude, duration and velocity [5] - the main sequence. Several models have been developed to describe the system - starting with one-dimensional open-loop models and later, more complicated models proposed to mimick this class of eye movements. The challenge is then to create a three-dimensional model of the saccadic system, from planning to kinematics and dynamics, loyal to the main sequence and Listing's plane characteristics.

In order to perform a saccade, the brain generates a control strategy in which the goal is to achieve a more rewarding state. Different experiments have suggested that both velocity and latency of saccades are related to the value assigned by the brain to a visual stimulus [6], meaning that expected reward can influence the motor commands. Simultaneously, the change in the state produced by a motor command is influenced by disturbances which grow with its size, called Signal-dependent noise (SDN) [7–9]. Nowadays, the saccadic system is thought to rely on a combination of planning and correction of the movements. Several research works reported that, even though the brain preprograms our eyes' movement in order to reach the goal when stimulated by a visual signal, the generated trajectory might change in midflight when the stimulus moves. Since saccades are such fast movements, it has been suggested [10–13] that the brain has an internal estimate of the state of the eye, providing feedback information that relies strongly on an efferent copy of the motor commands. The way uncertainty affects

sensory inputs, as well as motor commands, is acknowledged through learning which makes it possible to tune the motor responses to different sensory information.

We begin this work with a previously built mechanical prototype of a biologically inspired eye with six muscles. However, this system has problems related to undesired effects emerging from the system mechanical implementation (e.g vibrations) [14]. So, in order to proceed with the study of saccades, a model was developed in Matlab/Simulink from the mechanical prototype. This has allowed the achievement of empirical proof of the saccadic system using optimal control in open-loop, demonstrating that this framework emulates saccades with Listing's plane and main sequence behaviour [3].

## 1.2 Problem statement and Objectives

This thesis elaborates on the control of saccades on a model of a bio-inspired 3D robot of the eye, built in the scope of ORIENT I project, a European collaboration (ERC advanced grant) between the Lisbon visual robotics group of professor Bernardino, and the Donders Center for Neuroscience of professor Van Opstal. The robot was previously projected and assembled [14], intenting to replicate the static and dynamic properties of the eye. Later, this robot was modeled and a study on open-loop optimal control of the saccades was made, expanding the existing knowledge on different cost terms and characteristics of the plant and their influence in saccade trajectories [3].

It is our primary objective to improve and extend the existing model to introduce multiplicative noise in both motor commands and sensory readings and to implement a more realistic muscle model [15]. Moreover, the model is analytically linearized, providing a reliable parametrization of the non-linear model at any working point in a time efficient way and allowing for the design of a feedback control loop.

We aim to provide evidence on the influence of signal-dependent noise and feedback in 3-dimensional saccades as well as to study the contributions of these features to the neuronal strategy for controlling saccades, having as ground truth the empirical knowledge on eye dynamics [4, 5].

## 1.3 Outline

The first chapter provides an introduction to the context of this thesis.

In chapter 2, fundamental notions about the developed work are given, together with state-of-the-art.

In chapter 3, we describe the modifications applied to the existing open-loop eye robot model and the used methods for its linearization. The carried out tests which validate these procedures are also presented.

Chapter 4 elaborates on the feedback loop design for the existing model and explains the approaches to be tested in the control of saccades, as well as the metrics used to evaluate them. Hence, the results

for the different approaches are shown.

Chapter 5 sums up the main conclusions and achievements throughout the thesis. Moreover, possible work on different aspect of this matter are suggested.

# 2

# Background and State-of-the-Art

## Contents

This chapter aims to provide the reader with basic helpful concepts to understand the developed work in the following chapters along with state-of-the-art studies on related subjects. Firstly, an explanation on the adopted notion of the oculomotor system throughout this thesis is given, followed by the behaviour of eye movements and an overview on the used 3D rigid body representations of orientation. Then multiplicative noise is explained, its bedrock and potential in this work and how it is modeled. Additionally, the concept of reward in biological systems is clarified and finally, feedback in saccades is emphasized, providing a base for the use of optimal feedback control theory in the control of saccades.

## 2.1 Oculomotor System

The oculomotor system, represented by the diagram in figure 2.1, consists of two main parts. On one hand, it has a nonlinear plant formed by the eyes and their muscles, and on the other hand a saccade controller inside the brain.

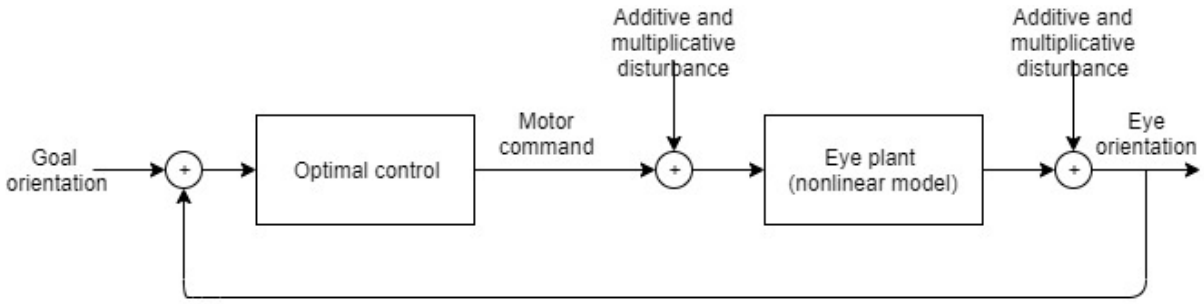



**Figure 2.1:** The oculomotor system

The extraocular muscles are responsible for eye movements and there are six of them. At first sight, this would mean that there are six Degrees-of-freedom (DOF) in the plant. However, these muscles act as agonist/antagonist pairs:

- Superior and inferior oblique - Torsional movements
- Superior and inferior rectus - Vertical movements
- Lateral and medial rectus - Horizontal movements

Therefore, because of the agonist/antagonist actions of the two muscles in each pair, only one DOF is provided to the eye per pair and so it has in total only three DOF instead of the possible six. Furthermore, these muscles are innervated by three cranial nerves, controlled by three different parts of the brain.

The point in moving the eyes is in placing the fovea, the area with the highest spatial resolution in the retina, aligned with some visual stimulus. To perform saccadic eye movements, the brain prepares a transient pulse command signal in the midbrain superior colliculus [10]. It has been suggested in several

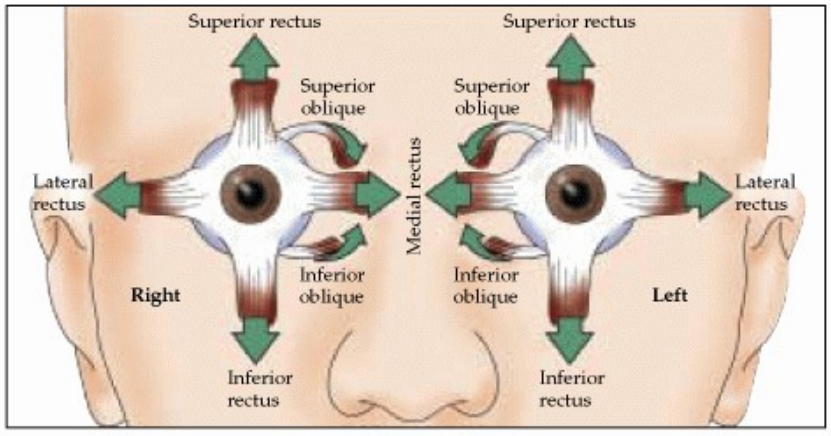



**Figure 2.2:** 6 extraocular muscles act in 3 agonist/antagonist pairs, each in one direction. [1]

models that the saccadic eye movement is controlled by an internal feedback circuit in the brainstem that drives the three muscle pairs.

Finally, as these control circuits contain neural populations that are noisy, it has been suggested that the system needs to optimize its control to deal with this noisy representation. There are two types of noise in the system: additive noise and (multiplicative) Signal-dependent noise (SDN). Multiplicative noise is a perturbation for which the standard deviation increases with the mean of the signal. In fact, this kind of disturbance is present in biological systems in general [8], and has been suggested to be of critical influence in the control of saccades [9].

## 2.2 Saccade Dynamics

Saccades have empirically observed stereotypical trajectories. Here we present the main properties of these movements, which are used to validate the research made in several studies and this one.

### 2.2.1 Main sequence

The term main sequence has been adopted in the study of saccadic movements given that these follow stereotypical relations between duration and amplitude as well as between peak velocity and amplitude for all healthy individuals. The relation between amplitude and duration of saccades has been questioned by scientists since the 19th century, but it was not until Dodge and Cline [16] that it was established that the duration of saccades (or average velocity) increases stereotypically with amplitude. This result has been confirmed later by several research works. On the other hand, it has also been shown by various researchers that the peak velocity of saccades tends to increase linearly with amplitude, reaching a saturation value for large saccades. Later, studies have shown that there is also a typical relationship

between the movement amplitude and the asymmetry of saccadic velocity profiles, which is quantified by their skewness [17]. By definition:

$$S = \frac{\int_0^\infty (t-\mu)^3 |\omega(t)| dt}{\sigma^3 \int_0^\infty |\omega(t)| dt} \tag{2.1}$$

where $\mu$ is the mean and $\sigma^2$ is the variance of the velocity profile, $\omega(t)$. The cause for the eyes to obey main sequence behaviour has been suggested to lie in a deliberate design property of the saccadic system - the kinematic characteristics can be traced back to single saccade-related cells by their burst firing patterns which have the same skewness characteristics as velocity profiles and whose shape varies noticeably with amplitude [10], suggesting that these characteristics are inherent to a strategy of higher abstraction level.

These spatial-temporal relations which characterize saccades are illustrated in figure 2.3.

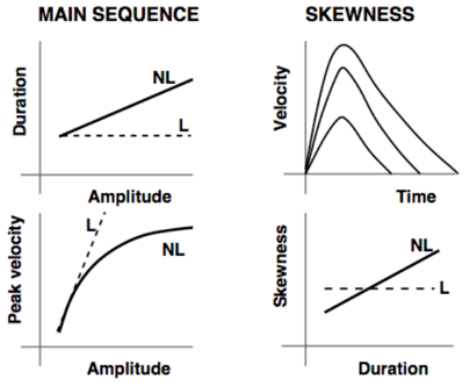



**Figure 2.3:** Non-linear dynamics of saccades - the main sequence. The dashed lines, L, represent the required behaviour for a system to be linear. NL lines represent the observed relations in saccades.

The main sequence has been crucial for the study of saccades because of its regularity, having enticed the scientific community to ask why the eyes move in such stereotypical form, and several studies were made in the framework of optimal control to understand the strategy behind saccades having it as ground truth. As a result, it is now well accepted that there is a speed-accuracy trade-off behind the neural strategy controlling saccades, and there have been several suggestions on the origin of this compromise. It is one of the main goals of this work to expand the existing knowledge on this matter, which is introduced into greater detail in 2.4.2.

### 2.2.2 Eye orientation

When the eye is fixating some point in space, the gaze direction is defined by this point's direction, according to the eye orientation relative to $y$ and $z$. Thus, the gaze direction does not specify the torsion of the eye. However, as previously stated, the six extraocular muscles are mechanically arranged in

such a way that the eye has 3 DOF rotating, as shown in figure 2.5 - torsionally when rotating about $x$, vertically for $y$ and horizontally for $z$, from a primary position where all these angles are 0.

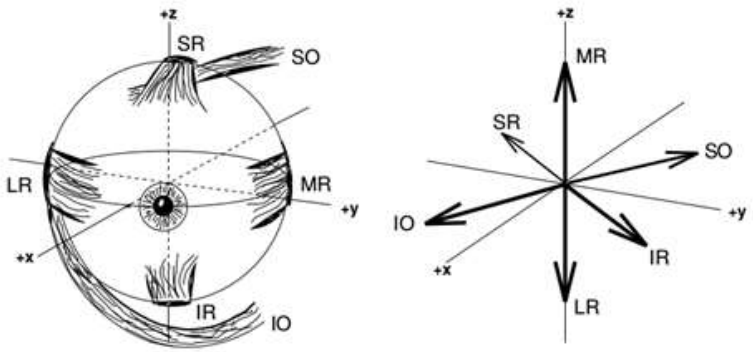



**Figure 2.4:** Eye orientation - Muscles action in a coordinate system

There has been research about eye torsion for as much as 200 years [18], when Franciscus Cornelis Donders stated that while looking at infinity with stationary, erect head, each gaze direction has a unique torsional angle regardless of the way the eye reaches that orientation. In other words, the Donders' law states that the torsional component of the eye orientation is a function of the vertical and horizontal components, meaning that the eye has not three but only two DOF

A further specification of Donders' law came with Listing's law, which states that with the head fixed and gazing at infinity in the primary position, any eye orientation can be reached by a single rotation about an axis in a plane orthogonal to the line of sight - the so-called Listing's plane [2]. Since we define the primary position as the position where the angles relative to three axes are zero, then Listing's law means that the torsion remains zero in any other eye orientation and so, the Listing's plane is the set of orientations where there is no torsion. Eye orientations are described by fictitious single-axis rotations from the primary position. However, a true eye movement is not a rotation from the primary position, but between arbitrary eye orientations. In that case, the eye orientation remains in Listing's plane (LP), but the actual axis of rotation that makes this happen is tilted out of this plane according to the so-called half-angle rule [19].

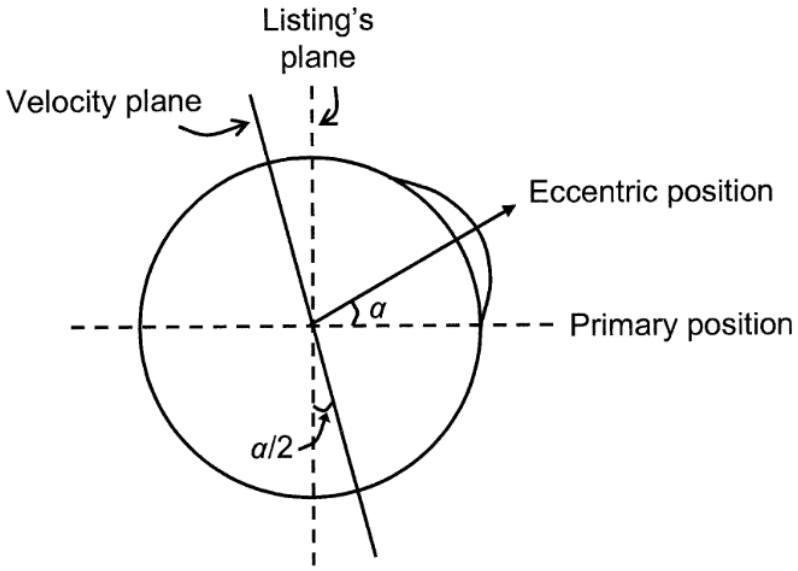



**Figure 2.5:** Listing's half-angle rule adapted from [2].The dashed horizontal line represents the line of sight when the eye is in primary position, and the dashed vertical line represents Listing's plane, which is orthogonal to the primary position. When a rotation starts from an eccentric position, the eye angular velocity vector is described by an axis tilted out of LP half as much as the line of sight.

### 2.2.3 Straightness

One-dimensional models of the saccadic system have largely been used in the study of saccades [5, 20], but these do not provide a complete framework of these movements, as oblique saccades cannot be tested and therefore do not probe how the different actuators are coupled. To account for the shortcomings in these, a common source model was formulated, explaining in the simplest possible way the straightness of olique saccades by a single vectorial velocity generator [21]. In this model, the velocity signal is decomposed trigonometrically between horizontal and vertical components, resulting in perfectly scaled velocity components and thus perfectly straight saccades. Later, other formulated alternative explanations for straight oblique saccades through cross-coupling of the horizontal and vertical velocity generators in the brainstem [22].

Human saccades are remarkably straight. However, this is not trivial given the nonlinear main sequence relations. If the control of saccades were to be done independently between horizontal and vertical components, the velocity components in the shortest dimension of the movement would last less than the opposite. However, it turns out that the duration of saccade components are nearly identical regardless their size, which can be explained by both the common source model and more complex cross-coupled models with their straight saccades.

There are several methods to measure the curvature of saccades. Here, we will compute the correlation between horizontal and vertical velocity components of oblique movements. If the correlation is unitary, then the saccade is straight since its velocity components are perfectly scaled versions of each

other. The goal is thus to have near-unitary correlation to mimic saccades correctly.

## 2.3 3D Orientation/Rotation Representations

For a complete understanding and characterization of the oculomotor system, eye orientation must be represented not only by the horizontal and vertical components but also by its torsion. Thus, different approaches have been used in the field of neuroscience - rotation matrices, quaternions and rotation vectors [23]. On the other hand, it is common in robotics to use different representations of 3D orientation and motion. In this section we present some of these representations as well as the relations between them, analysing their advantages and disadvantages given the problem at hands, considering that the previously built model already uses two kinds of representations (quaternions and rotation matrices) and that we will have to linearize it as stated in 1.2.

To linearize the saccadic system and control eye rotations we need a representation which allows us to:

1. compute derivatives of orientations with respect to the parameters
2. integrate differential equations in parameter space
3. interpolate orientations smoothly between time instants

Based on the comparison between representations made in [24] we will comment on different orientation representations - Euler angles and Rotation matrices, unit quaternions, Angle-axis and exponential map.

### 2.3.1 Euler angles and Rotation matrices

The Euler angles are three angles, $\alpha, \beta$ and $\gamma$ that describe an orientation by rotations about the three axes of a reference frame, $x$, $y$ and $z$ respectively. We define these angles according to the right-hand rule and therefore a positive angle corresponds to a clockwise rotation about the respective axis.

The most common way to represent rotations (and subsequently orientation) is with rotation matrices. A rotation matrix is, in a 3D configuration, given by a 3x3 matrix with unit length, mutually orthogonal columns and determinant 1, $det(R) = 1$. The identity element of rotation matrices is the identity matrix, $I$. The group of all such matrices is called the special orthogonal group, $SO(3)$.

A rotation matrix can be defined from the Euler angles representation as follows:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha) & -sin(\alpha) \\ 0 & sin(\alpha) & cos(\alpha) \end{bmatrix} R_y(\beta) = \begin{bmatrix} cos(\beta) & 0 & sin(\beta) \\ 0 & 1 & 0 \\ -sin(\beta) & 0 & cos(\beta) \end{bmatrix} R_z(\gamma) = \begin{bmatrix} cos(\gamma) & -sin(\gamma) & 0 \\ sin(\gamma) & cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

To execute a rotation to a vector $\mathbf{x}$ using a rotation matrix $R$ and obtain a rotated vector $\mathbf{x}'$ we simply multiply:

$$\mathbf{x}' = R\mathbf{x} \tag{2.3}$$

The multiplication of several rotation matrices leads to a composed rotation. As this is done by matrix product, the operation is noncommutative and thus, if we want to combine rotations about the three elementary axes, $R_x, R_y$ and $R_z$ in one rotation matrix $R$ then performing the operation in different orders leads to different composed rotations. Also, the use of rotation matrices is inefficient considering that nine values are used to describe a rotation instead of the necessary three.

### 2.3.2 Unit quaternions

Quaternions are a different way to describe rotations, which uses 4 numbers ($\mathbf{q} \in \mathbb{R}^4$) to give a global parametrization of SO(3). This representation generalizes complex numbers, and since these can be used to describe planar rotations in the complex plane, quaternions can be used to describe space rotations.

A quaternion can be written as:

$$\mathbf{q} = q_0 + q_x\mathbf{i} + q_y\mathbf{j} + q_z\mathbf{k} \tag{2.4}$$

Here, $q_0$, $q_x$, $q_y$ and $q_z$ are real scalars and $\mathbf{i}, \mathbf{j}$ and $\mathbf{k}$ are complex numbers related by:

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1 \tag{2.5}$$

The first element, $q_0$ is the scalar part of the quaternion and the remaining three compose its vector part:

$$\mathbf{q} = (q_0, \vec{q}) \tag{2.6}$$

The unit quaternion can encode a rotation about the unit axis of rotation $\mathbf{n}$ by an angle $\theta$ as

$$\mathbf{q} = cos(\frac{\theta}{2}) + (n_x\mathbf{i} + n_y\mathbf{j} + n_z\mathbf{k})sin(\frac{\theta}{2}) \tag{2.7}$$

The angle and axis of the rotation can be obtained from the quaternion components through:

$$\begin{aligned} \theta &= 2cos^{-1}(q_0) \\ \mathbf{n} &= \frac{(q_x, q_y, q_z)}{\sqrt{q_x^2 + q_y^2 + q_z^2}} \end{aligned} \tag{2.8}$$

In the case where the norm of the quaternion's vector part is zero, the correspondent quaternion is the identity quaternion, and an axis cannot be obtained.

This representation carries the property of being particularly straightforward to perform rotations on some ordinary 3D vector, which is done by transforming it into a quaternion by assigning to it a zero scalar part ($\mathbf{x}_0 = 0$) and then applying quaternion multiplication:

$$\mathbf{x}' = \mathbf{q} \circ \mathbf{x} \circ \mathbf{q}^{-1} \tag{2.9}$$

The resulting $\mathbf{x}'$ is also a quaternion with zero scalar part.

The multiplication between two quaternions is represented by $\circ$ and it is noncommutative. It is given by

$$p \circ q = p_0 q_0 - p \cdot q + p_0 q + q_0 p + p \times q \tag{2.10}$$

and results in a quaternion.

Unit quaternions are free from gimbal lock when used to control rotations and good for combining rotations. Furthermore, it is hard to compute derivatives of orientations with respect to the parameters, $\frac{dq}{dv}$, which is troublesome for linearizing the eye plant.

### 2.3.3 Angle-axis representation

The angle-axis representation gives a way of representing orientation described by a three-dimensional vector in which the axis, $\mathbf{n}$ is a unit vector described relative to a reference frame and the angle $\theta$ (in radians) is the amount of rotation about that axis. The two are combined as follows:

$$\mathbf{v} = \theta \mathbf{n} = \begin{bmatrix} v_x \\ v_y \\ v_x \end{bmatrix} \tag{2.11}$$

where,

$$\mathbf{n} = \begin{bmatrix} n_x \\ n_y \\ n_x \end{bmatrix} \tag{2.12}$$

Since $\mathbf{n}$ is unit-length, only two values are needed to define it the magnitude of the third value, while it is still necessary to define the sign.

Thus, the angle-axis notation has the advantage of representing a rotation with only three scalar values. While it would seem that three components would be needed for the axis and one for the angle, the fact that the axis is unit-length makes it necessary to define it only two parameters. This representation provides an easy interface to interpret rotations.

### 2.3.4 Exponential map

By using exponential mapping, a rotation matrix in $SO(3)$ can be mapped from a vector in $\mathbb{R}^3$:

$$R = e^{[v]_x} \tag{2.13}$$

where $[v]_x$ is the skew-symmetric matrix obtained from the angle-axis representation, $\mathbf{v}$:

$$[v]_x = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix} \tag{2.14}$$

The simplicity of computing derivatives of rotation with respect to the parameters, $\frac{dR}{dv}$ is obvious, which is an important feature of this representation, given the goal of linearizing the eye plant and that for this purpose it is necessary to compute Jacobians.

$$\frac{dR}{dv_x} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \qquad \frac{dR}{dv_y} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \qquad \frac{dR}{dv_z} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.15}$$

The parameter rates for interpolation (time derivative) can be obtained, for small values of $\theta$ as a function of $v$ and the angular velocity of the rigid body, $\omega$ through [24]:

$$\dot{\mathbf{v}} = \frac{1}{2}(\gamma\omega + \omega \times \mathbf{v} + \eta\mathbf{v}) \tag{2.16}$$

with

$$\begin{aligned} \gamma &= \frac{12 - \theta^2}{6} \\ \eta &= \omega \cdot \mathbf{v}\frac{60 + \theta^2}{360} \end{aligned} \tag{2.17}$$

The exponential map seems to be a valid, useful option to parameterize eye rotations, combining the perks of using angle-axis representation with the ease of computing derivatives (2.15)

### 2.3.5 Comparison between parametrizations

The three different representations present have different advantages and disadvantages between them, depending on the use we want to give them.

- **Euler angles and Rotation matrices** - Altough Rotation matrices are widely used, these are of high complexity for computation. Nevertheless, these provide an easy way of combining rotations.
- **Unit quaternions** The quaternion is a widely used representation of 3D rotations in robotics and

is especially good when dealing with problems of big rotations because of the absence of singularities. Nevertheless, to linearize a system parameterized by quaternions is quite a challenge which can easily be tackled using the exponential map.

- **Angle-axis** - It is the easiest to visualize. However, the complexity of combining rotations makes it less convenient to work with in the simulator.
- **Exponential map** - This representation is, in fact, a tool to easily go from a rotation vector ($\mathbb{R}^3$) to a rotation matrix ($SO(3)$) which makes the interpretation easy as well as combining rotations. The derivatives with respect to parameters and time are easily computed with this method [24].

The different orientation representations are used in this thesis in different contexts considering their properties and the task in which they are applied.

## 2.4 Feedback in saccades

The problem of generating motor commands to make some movement, like a saccade, can be modelled as an optimal feedback control problem:

- The movement has a cost;
- A forward model gives a prediction of the outcome generated by some action;
- Some information about the system's state is retrieved during the execution.

The influence of feedback in saccades has been proven previously in different contexts. Sparks and Mays [25], for instance, trained monkeys to look at a brief flash of light after it occurred but, in the latency period of the saccade changed the eye gaze by stimulation on the monkeys' brain in some of the trials. In this experiment, if there were no influence of feedback, the monkeys would perform a saccade with the predetermined direction and amplitude, failing the goal completely. However, the monkeys were able to perform saccades reaching the goal, proving that somehow the brain is keeping track of the orientation of the eye and taking it into account when performing movements. In various models following this experiment, the internal feedback signal has been represented as an efference copy signal of position and/or velocity coming from the brainstem [10–12]. On the other hand, studies have suggested also that saccades happen too fast for proprioceptive signals to play a significant role in feedback [26] as well as visual information [13]. To sum it up, the eyes' orientation has been shown to be monitored by the brain but with little or no help from sensory information, so this can only mean that the influence of feedback relies strongly on a precise internal forward model used to predict the consequences of

generated motor commands. On the other hand, the presence of sensory feedback has been proven in movements that combine eye and head gaze shifts [27], but that is not in the scope of this thesis. The fact that saccades rely mostly on an internal model while having such good performance makes us infer that this model must be accurate. As stated previously in 2.5, SDN is critical in motor-planning [9] and has to be considered therefore by this model - through learning the brain can understand variability and its structure and how it affects the movement.

### 2.4.1 State-space representation

The state-space representation of a system consists of a mathematical model containing its state variables and relating its inputs and outputs using first-order differential equations. The great advantage of state-space representation is that it provides a very compact way of modelling both SISO and MIMO systems. [28] The equations below can describe a discretized version of a linear dynamical system with additive noise, $\epsilon_x$ and $\epsilon_y$:

$$\begin{aligned} \mathbf{x}^{(k+1)} &= A\mathbf{x}^{(k)} + B\mathbf{u}^{(k)} + \epsilon_x \\ \mathbf{y}^{(k)} &= H\mathbf{x}^{(k)} + \epsilon_y \end{aligned} \tag{2.18}$$

Here $A$ represents the dynamic matrix, containing information about the system's dynamic properties, $B$ represents the input matrix which translates the influence of the input, $\mathbf{u}$ and $H$ is the output matrix, which transforms the state $\mathbf{x}$ into sensory readings $\mathbf{y}$.Let $m$ be the dimension of vector $\mathbf{u}$ - number of elements in the motor commands - and $n$ is the dimension of vector $\mathbf{x}$ - number of elements defining the state.

Additive noises, $\epsilon_x$ and $\epsilon_y$, are defined as Gaussian random variables with zero-mean and variance $Q_x$ and $Q_y$ respectively:

$$\begin{aligned} \epsilon_x &\sim \mathcal{N}(0, Q_x) \\ \epsilon_y &\sim \mathcal{N}(0, Q_y) \end{aligned} \tag{2.19}$$

where $Q_x$ and $Q_y$ are $n \times n$ and $m \times m$ symmetric and positive definite matrices which we assume diagonal and with values $q_x$ and $q_y$ respectively in their diagonals.

### 2.4.2 Optimal Feedback Control

To find, in a set of equations which describe a system, the values for their variables which minimize a cost, $J$, is a problem of optimal control. In mathematical terms, it deals with finding the minima for a function of $\mathbf{u}$, the action variable, and $\mathbf{x}$, the state variable:

$$J(\mathbf{u}) = \Psi(\mathbf{x}(p)) + \int_0^p l(\mathbf{x}, \mathbf{u}) dt \tag{2.20}$$

where

$$t \in [0, p]$$

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

While $J$ is the cost to be optimized, $\Psi$ represents a contribution to the cost associated with the terminal state, $\mathbf{x}(p)$. $l$ is the lagrangean function which is related to the contributions made to the cost during the optimization interval. However, the study of optimal control endowed with feedback relies more strongly on the field of stochastic optimal control for technical reasons - discrete spaces are proven to converge in a reasonable amount of time, unlike the continuous form. In this framework, it is the expected value of the cost we want to optimize, $E[J]$. Moreover, the Linear-Quadratic-Gaussian (LQG) framework [29] applies to the control of biological movements [30] and therefore an optimal control law can be obtained.

In order to understand the dynamics of these eye movements (explained in 2.2), some research has been made regarding the neuronal strategy behind them or, in other words, what the brain is trying to optimize when performing saccades. While it is well accepted that there is a trade-off between speed and accuracy originated by SDN, since it explains the main sequence [9, 10], there is still some debate regarding other cost terms such as energy consumption, which has been claimed to be critical to the way the eye behaves in [31]. This thesis tries to go over these classical saccade control approaches considering the influence of feedback, based on state-of-the-art knowledge [32] [20], a time constraint to movements and SDN not only in the motor commands but also in the sensor readings. We have as our base framework the classical approach in which the brain develops a control strategy regarding accuracy, energy consumption and speed:

$$J = J_x + J_u + J_p \tag{2.21}$$

The first term in (2.21), $J_x$, is responsible for penalizing the end-point inaccuracy, so it should increase as the eye's final state diverges from the goal set at the beginning. We define $J_x$ in the following form ($p$ represents the last time-step of the movement):

$$J_x = \mathbf{x}^{(p)T} T \mathbf{x}^{(p)} \tag{2.22}$$

Here, $T$ is a $n \times n$, positive definite, diagonal matrix, and its values define the assigned penalization to each state-variable at the end of the movement.

The second term, on the other hand, represents the cost of control - a term proportional to the sum of squared motor commands. This is how energy consumption is weighted in the movement cost:

$$J_u = \sum_{k=0}^{p} \mathbf{u}^{(k)T} L \mathbf{u}^{(k)} \tag{2.23}$$

where $L$ is a $m \times m$, positive definite, diagonal matrix. The values in this matrix weigh the penalization of each motor command, $u$ Finally, the term $J_p$ induces time-pressure in the optimization. It is thought that the brain assigns value, $\lambda$, to a stimulus which decreases hyperbolically with time, according to $\beta$ [33,34] - the cost will increase along with the delay,$p$, in reaching the goal:

$$J_p = \lambda \left(1 - \frac{1}{1 + \beta p}\right) \tag{2.24}$$

The resulting cost function is thus a function of $p$. Nevertheless, as it is our objective to study the role of feedback in controlling saccades - that is, we want to track the system's state and get the optimal motor commands as a response to it - we must consider a cost per time step, so that it is possible to decide which muscle activations should be done so as to have minimum cost given the state at time step $k$:

$$\alpha(k) = \mathbf{u}^{(k)T} L \mathbf{u}^{(k)} + \mathbf{x}^{(k)T} T^{(k)} \mathbf{x}^{(k)} + \frac{\lambda\beta}{1 + \beta p} \tag{2.25}$$

The term $T^{(k)}$ is kept null for all time-steps except for the last, since we want to penalize inaccuracy only in the endpoint and not during the whole movement.

To find the optimum policy, we consider Bellman's *principle of optimality: An optimal policy has the property that whatever the initial state and decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.* If the policy is optimal, it will minimize the sum total of costs [35]. It is thus useful to define a function, $v_{\pi*}$ which gives us the accumulated cost at each step - a function analogous to the *Bellman equation* with the difference that we want to minimize it:

$$v_{\pi*}(\mathbf{x}^{(k)}) = \min_{\mathbf{u}^{(k)}} \{\alpha^{(k)} + E[v_{\pi*}(\mathbf{x}^{(k+1)}) | \mathbf{x}^{(k)}, \mathbf{u}^{(k)}]\} \tag{2.26}$$

For any time-step, $k$, its value is given by the cost at that time-step $\alpha^{(k)}$ plus the expected value of the state resulting from applying the optimal policy since the system is endowed with random disturbances which have an influence on $E[v_{\pi*}(\mathbf{x}^{(k+1)}) | \mathbf{x}^{(k)}, \mathbf{u}^{(k)}]$. By solving this problem recursively, starting at some state $x^{(p)}$ where the optimal policy is to do nothing $u^{(p)} = 0$ then we can obtain the optimum policy for the previous time step $k - 1$ and so on.

Since we do not have access to the internal state of the system at each time step, the Kalman framework (explained in 2.4.3) is used to get an estimate of it, $\hat{\mathbf{x}}^{(k)}$. Todorov [30] proposed a form for (2.26) at each time-step:

$$v_{\pi*}(\mathbf{x}^{(k)}, \hat{\mathbf{x}}^{(k)}) = \mathbf{x}^{(k)T} W_x^{(k)} \mathbf{x}^{(k)} + (\mathbf{x}^{(k)} - \hat{\mathbf{x}}^{(k)})^T W_e^{(k)} (\mathbf{x}^{(k)} - \hat{\mathbf{x}}^{(k)}) + w^{(k)} \tag{2.27}$$

Notice that $\hat{\mathbf{x}}$ is introduced here, as it is assumed that the brain does not have access to the full state of the eye and uses an estimate of it. This is further explained in the next section.

Setting $W_x^{(p)} = T$ and $W_e^{(p)} = 0$ and noting that under the optimal policy $\mathbf{u}^{(p)} = 0$, we get the value of cost $\alpha^{(p)}$. By recursively replacing in the *Bellman equation* (2.26) and then applying the minimization in (2.27) we can get the optimal motor commands for each time step. To simplify the notation, we set $e^{(k)} = \mathbf{x}^{(k)} - \hat{\mathbf{x}}^{(k)}$.

$$v_{\pi*}(\mathbf{x}^{(k)}, \hat{\mathbf{x}}^{(k)}) = \min_{\mathbf{u}(k)} \left\{ \begin{array}{l} \mathbf{u}^{(k)T} L \mathbf{u}^{(k)} + \mathbf{x}^{(k)T} T^{(k)} \mathbf{x}^{(k)} + \dfrac{\lambda\beta}{1+\beta p} \\ + E[\mathbf{x}^{(k+1)T} W_x^{(k+1)} \mathbf{x}^{(k+1)} | \mathbf{x}^{(k)}, \hat{\mathbf{x}}^{(k)}, \mathbf{u}^{(k)}] \\ + E[\mathbf{e}^{(k+1)T} W_e^{(k+1)} \mathbf{e}^{(k+1)} | ...] + w^{(k+1)} \end{array} \right\} \tag{2.28}$$

Here, the influence of disturbances come into the optimization since since we want to compute the expected value of the scalar quantities that depend on the quadratic form of the random variables $\mathbf{x}$ and $\mathbf{e}$:

$$E[\mathbf{x}^T A \mathbf{x}] = E[\mathbf{x}]^T A E[\mathbf{x}] + \mathrm{tr}[A \mathrm{var}[\mathbf{x}]] \tag{2.29}$$

In a system perturbed with additive noise, the variances to consider are $Q_x$ and $Q_y$. The solution comes as the commoonly seen LQG controller gains:

$$\begin{aligned} G^{(k)} &\equiv (L + B^T W_x^{(k+1)} B)^{-1} B^T W_x^{(k+1)} A \\ \mathbf{u}^{(k)} &= -G^{(k)} \hat{\mathbf{x}}^{(k)} \end{aligned} \tag{2.30}$$

where,

$$W_x^{(k)} \equiv T^{(k)} + A^T W_x^{(k+1)} A - G^{(k)T} B^T W_x^{(k+1)} A \tag{2.31}$$

So far, we have described the optimization process to obtain the optimal feedback control of saccades over a fixed movement duration, $p$. However, movements of different amplitudes have different durations as stated in section 2.2. So, it is necessary to optimize the cost function over the range of all plausible saccade durations:

$$\min_p \left\{ \sum_{k=0}^{p} v_{\pi*}(\mathbf{x}^{(k)}) \right\} \tag{2.32}$$

The duration which minimizes the expected cost is chosen, and the correspondent control law is used.

### 2.4.3 Optimal Estimation

When controlling a system, the controller often does not have access to the full state-space. For instance, if the system has in its state-space both position and velocity, it might happen that velocity cannot be measured and the controller has access only to the system's position but wants to control the two variables. State observers combine the measured output of a system with the input to provide an estimate of the system's state given the available information, with the goal of having the smallest

difference between the estimated output and the system's output. That is the case of the Kalman Filter (KF), a largely used mathematical tool with applications in many kinds of engineering devices:

$$\begin{aligned} \hat{\mathbf{x}}^{(k|k)} &= \hat{\mathbf{x}}^{(k|k-1)} + K^{(k)}(\mathbf{y}^{(k)} - H\hat{\mathbf{x}}^{(k|k-1)}) \\ \hat{\mathbf{x}}^{(k+1|k)} &= A\hat{\mathbf{x}}^{(k|k)} + B\mathbf{u}^{(\mathbf{k})} \end{aligned} \tag{2.33}$$

Here, $K^{(k)}$ represents the Kalman gain at time-step $k$, which reflects the confidence that the observer has in the measurements relative to the prediction provided by the internal model represented by $A, B$ and $H$. Hence, this tool not only provides state estimates using observations and the available information of an internal mathematical model of the device but also filters disturbances, all through the gains $K^{(k)}$- random additive noise in its most common formulation.

$$K^{(k)} = P^{(k|k-1)}H^T(HP^{(k|k-1)}H^T + Q_y)^{-1} \tag{2.34}$$

$P^{k|k-1}$ is the prior state uncertainty. The posterior state uncertainty, $P^{k|k}$ is calculated in the following form, according to the model's dynamics:

$$\begin{aligned} P^{(k|k)} &= P^{(k|k-1)}(I - H^T K^{(k)T}) \\ P^{(k+1|k)} &= AP^{(k|k)}A^T + Q_x \end{aligned} \tag{2.35}$$

As can be seen in equations (2.34) and (2.35) the sources of variability, $Q_x$ and $Q_y$, are included and influence the values of gain which translate the confidence on the sensor readings (matrix $H$), given an internal model of the system dynamics (matrix $A$) - an increase in variability results in a decrease in the Kalman gain.

## 2.5 Signal-dependent Noise

Systems are usually perturbed by additive gaussian noise, in different contexts like control, communication or data acquisition systems. However, some systems are endowed not only with additive but also with SDN as is the case of biological systems, in which the variance of the noise depends on the size of the signal. This effect has been studied before in different research projects from diverse scientific contexts such as psychology and neurophysics and was firstly hypothesized as being originated by muscle contractions [7] but later proven to be a result of neuronal commands unrelated to how muscles produce force [8]. These experiments showed most importantly that the standard deviation of the force produced by humans grows linearly as a function of the mean force.

Regarding specifically the control of saccades, some research has also been made on the effect of SDN. Saccades have stereotypically asymmetric velocity profiles, producing larger commands earlier in

the movement rather than its opposite. The reason for this peculiar behaviour lies on the neural strategy of control, which tries to minimize variability [9] by producing large commands early in the movement in order to let variability dissipate naturally throughout the movement by the viscous dynamics of the eye and reaching, therefore, the end of the movement with maximum accuracy. In other words, the variability produced by SDN early in the movement has a smaller influence on the end-point accuracy of the eye. Indeed, it has been shown in [20] that a noise-free saccade has a fairly symmetrical profile, which does not correspond to the biological eye movement.

However, it is not only in biological systems that multiplicative noise is present. It can be seen also in fields of image processing, in which there is the notorious case of speckle-noise [36], caused by the roughness of the object being captured. In digital photography, there is also SDN when capturing surfaces with shadows or caused by dust on the lens.

### 2.5.1 Control of saccades with SDN

With SDN, the system has the influence of a kind of noise which is amplified by the size of both the motor commands and the sensory readings. By adding these terms to the normal system state-space representation (2.18), the result is the following system equations:

$$\begin{aligned} \mathbf{x}^{(k+1)} &= A\mathbf{x}^{(k)} + B(\mathbf{u}^{(k)} + \epsilon_u^{(k)}) + \epsilon_x \\ \mathbf{y}^{(k)} &= H(\mathbf{x}^{(k)} + \epsilon_s^{(k)}) + \epsilon_y \end{aligned} \tag{2.36}$$

One can notice the introduction of terms $\epsilon_u$ and $\epsilon_s$. These noise terms are characterized with zero-mean magnitude and variance that depends on the motor commands $\mathbf{u}$ and state $\mathbf{x}$ in the following form:

$$\begin{aligned} \epsilon_u^{(k)} &\equiv \begin{bmatrix} c_1 u_1^{(k)} \phi_1^{(k)} \\ c_2 u_2^{(k)} \phi_2^{(k)} \\ \vdots \\ c_m u_m^{(k)} \phi_m^{(k)} \end{bmatrix} & \epsilon_s^{(k)} &\equiv \begin{bmatrix} d_1 x_1^{(k)} \mu_1^{(k)} \\ d_2 x_2^{(k)} \mu_2^{(k)} \\ \vdots \\ d_n x_n^{(k)} \mu_n^{(k)} \end{bmatrix} \\ \phi &\sim \mathcal{N}(0,1) & \mu &\sim \mathcal{N}(0,1) \\ c_i &\geq 0 & d_i &\geq 0 \end{aligned} \tag{2.37}$$

A more straightforward way of defining the SDN terms comes if we make

$$C_1 \equiv \begin{bmatrix} c_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \ddots \end{bmatrix} \qquad C_2 \equiv \begin{bmatrix} 0 & 0 & 0 \\ 0 & c_2 & 0 \\ 0 & 0 & \ddots \end{bmatrix} \tag{2.38}$$

$$D_1 \equiv \begin{bmatrix} d_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \ddots \end{bmatrix} \qquad D_2 \equiv \begin{bmatrix} 0 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & \ddots \end{bmatrix} \tag{2.39}$$

and so we have

$$\begin{aligned}\epsilon_u^{(k)} &= \sum_{i=1}^{m} C_i \mathbf{u}^{(k)} \phi_i^{(k)} \\ \epsilon_s^{(k)} &= \sum_{i=1}^{n} D_i \mathbf{x}^{(k)} \mu_i^{(k)}\end{aligned} \tag{2.40}$$

and since $\phi$ and $\mu$ are Gaussian random variables, then also $\epsilon_u$ and $\epsilon_s$ are gaussian random variables as follows:

$$\begin{aligned}\epsilon_u^{(k)} &\sim \mathcal{N}(0, \sum_{i=1}^{m} C_i \mathbf{u}^{(k)} \mathbf{u}^{(k)T} C_i) \\ \epsilon_x^{(k)} &\sim \mathcal{N}(0, \sum_{i=1}^{n} D_i \mathbf{x}^{(k)} \mathbf{x}^{(k)T} D_i)\end{aligned} \tag{2.41}$$

Finally, replacing the noise terms in (2.36) by (2.40) we obtain a final form for the system equations endowed with both additive noise and SDN, firstly approached by Emo Todorov [32] and later adapted by Reza Shadmehr et Al. as one can find in [20]

$$\begin{aligned}\mathbf{x}^{(k+1)} &= A\mathbf{x}^{(k)} + B(\mathbf{u}^{(k)} + \sum_{i=1}^{m} C_i \mathbf{u}^{(k)} \phi_i^{(k)}) + \epsilon_x \\ \mathbf{y}^{(k)} &= H(\mathbf{x}^{(k)} + \sum_{i=1}^{n} D_i \mathbf{x}^{(k)} \mu_i^{(k)}) + \epsilon_y\end{aligned} \tag{2.42}$$

SDN affects the optimization of $v_{\pi *}$ since it alters the variance in the model: we now have both the additive noise and SDN variances, $\sum_{i=1}^{m} C_i \mathbf{u}^{(k)} \mathbf{u}^{(k)T} C_i$ and $\sum_{i=1}^{n} D_i \mathbf{x}^{(k)} \mathbf{x}^{(k)T} D_i$ (2.41). The solution of (2.27) with the presence of SDN is given by:

$$\begin{aligned}G^{(k)} &\equiv (L + C_x^{(k+1)} + C_e^{(k+1)} + B^T W_x^{(k+1)} B)^{-1} B^T W_x^{(k+1)} A \\ \mathbf{u}^{(k)} &= -G^{(k)} \hat{\mathbf{x}}^{(k)}\end{aligned} \tag{2.43}$$

where,

$$\begin{aligned}C_x^{(k+1)} &\equiv \sum_i C_i^T B^T W_x^{(k+1)} B C_i \\ C_e^{(k+1)} &\equiv \sum_i C_i^T B^T W_e^{(k+1)} B C_i\end{aligned} \tag{2.44}$$

and

$$\begin{aligned}W_e^{(k)} &\equiv (A - AK^{(k)}H)^T W_e^{k+1} (A - AK^{(k)}H) + G^{(k)T} B^T W_x^{(k+1)} A \\ W_x^{(k)} &\equiv T^{(k)} + A^T W_x^{(k+1)} A + D_e^{(k+1)} - G^{(k)T} B^T W_x^{(k+1)} A\end{aligned} \tag{2.45}$$

We can see that control gains are inversely proportional to the motor noise and the energy penalization term. Intuitively, this makes sense - if the motor noise term is higher it is advantageous to produce smaller motor commands, and the same applies to the energy term. Notice that the control gains obtained from the optimization of $v_{\pi *}(\mathbf{x}^{(k)}, \hat{\mathbf{x}}^{(k)})$ must depend on the observer gains, $K$ since we are also

optimizing the estimation error.

SDN influences not only the control but also the state estimation. The equations for computing the Kalman gains are thus different, as we have to include in them the sources of variability which depend on the motor commands and the state:

$$K^{(k)} = P^{(k|k-1)}H^T(HP^{(k|k-1)}H^T + Q_y + \sum_i HD_i\hat{\mathbf{x}}^{(k)}\hat{\mathbf{x}}^{(k)T}D_i^TH^T)^{-1} \tag{2.46}$$

$$\begin{aligned} P^{(k|k)} &= P^{(k|k-1)}(I - H^TK^{(k)T}) \\ P^{(k+1|k)} &= AP^{(k|k)}A^T + Q_x + \sum_i BC_i\mathbf{u}^{(k)}\mathbf{u}^{(k)T}C_i^TB^T \end{aligned} \tag{2.47}$$

The introduction of SDN in the system results that the Kalman gains at each time step are dependent on the state and the motor command at the same time step. To solve this problem, Todorov [32] suggested an iterative method for computing the Kalman gains, which as been adapted later to the notation used here [20][1]. It consists of firstly, using equations (2.34) and (2.35) ignoring the presence of multiplicative noise and then using these Kalman gains to compute the control gains from equation (2.43). After that, a new sequence of Kalman gains is calculated from the following equation, with the unconditional covariances $S_e^{(k)} \triangleq E[\mathbf{e}^{(k)}\mathbf{e}^{(k)T}]$ and $S_x^{(k)} \triangleq E[\hat{\mathbf{x}}^{(k)}\hat{\mathbf{x}}^{(k)T}]$:

$$\begin{aligned} K^{(k)} &= S_e^{(k)}H^T(HS_e^{(k)}H^T + Q_y + \sum_i HD_iS_x^{(k)}D_i^TH^T)^{-1} \\ S_e^{(k+1)} &= Q_x + (A - AK^{(k)}H)S_e^{(k)} + \sum_i B_iC_iG^{(k)}S_x(k)G^{(k)T}C_i^T \\ S_x^{(k+1)} &= AK^{(k)}HS_e^{(k)}A^T + (A + BG^{(k)})S_x^{(k)}(A + BG^{(k)})^T \end{aligned} \tag{2.48}$$

Todorov showed that, by repeating this method, the Kalman and control gains are guaranteed to converge in few iterations.

[1] Although expressions (2.48) were originally from [20], the equations for $K^{(k)}$ and $S_e^{(k)}$ had mistakes in this reference which were corrected to the present expressions.

# 3

# Eye model

## Contents

In this chapter, we present the work developed regarding the eye model. Firstly, a muscle model proposed in [15] is introduced to the existing eye plant model, followed by a description of the physical nonlinear plant. Subsequently, two methods are used to identify the system - through Matlab/System Identification Toolbox and analytical linearization of the state equations. Finally, the behaviour of both the muscle model alone and the 3D biomimetic robot model is presented, along with a comparison between the two identification methods.

## 3.1 Muscle model

Extraocular muscles behave according to the law of reciprocal innervation [37]. This law states that an increase in the innervation of an agonist muscle is accompanied by a simultaneous decrease in the innervation to the antagonist muscle. Thus, the six extraocular muscles can be seen as three pairs of agonist/antagonist muscles acting in three distinct directions, $x$, $y$ and $z$. When the agonist muscle is innervated, it contracts, applying a pulling force in the eye. Simultaneously, the antagonist muscle relaxes, but still produces a force in the opposite direction thanks to its elasticity.

In this work, we assume that there are 3 identical systems (one per pair of muscles) which receive a neuronal activation signal and transform it into force - this is the force production part or the agonist component of our muscle model. The model used in this work was extracted from [15] as follows

$$\alpha_1 \dot{f} + \alpha_2 f = u \tag{3.1}$$

where $u$ is an activation signal and $f$ is the force applied by the agonist muscle. The sign of the activation defines which of the two muscles in the pair will act as agonist. For $u_y$, for example, a positive value represents an activation of the superior rectus and therefore an upward movement occurs.

The equation denotes a $1^{st}$ order system in which the time constant is defined by the values of $\alpha_1$ and $\alpha_2$, meaning that the time our muscles take to reach the desired value is defined by these constants. Figure 3.1 presents the implementation of the force production part of the muscle in our simulator.

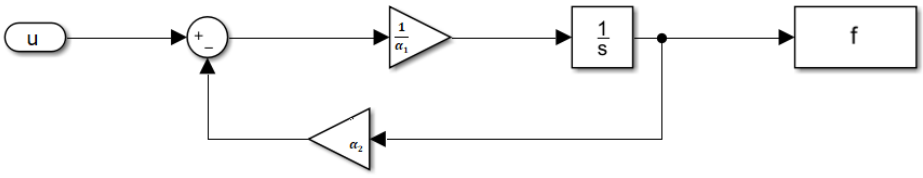



**Figure 3.1:** Muscle model

However, this muscle model would be incomplete without the antagonist component of the pair. In other words, the antagonist muscle's elasticity and the influence it has on the system's dynamics. In

fact, it is this part of the model which causes the coupling between the three dimensions in hand: eye orientation displacement causes the antagonist muscles to produce a force contrary to the sense of the displacement, which is modelled as an elastic for each muscle. Further explanation on the antagonist component of our muscle model is presented in the following section.

## 3.2 3D Biomimetic robot eye model

In a previous work of this project, the model illustrated by the diagram in figure 3.2 was implemented [3]. The model represents a biomimetic robot of the eye where three motors are responsible for the movement of six points, $P_i$ where springs are attached to replicate the six extraocular muscles. By moving these insertion points, tension is produced in these springs and therefore eye movement is generated.

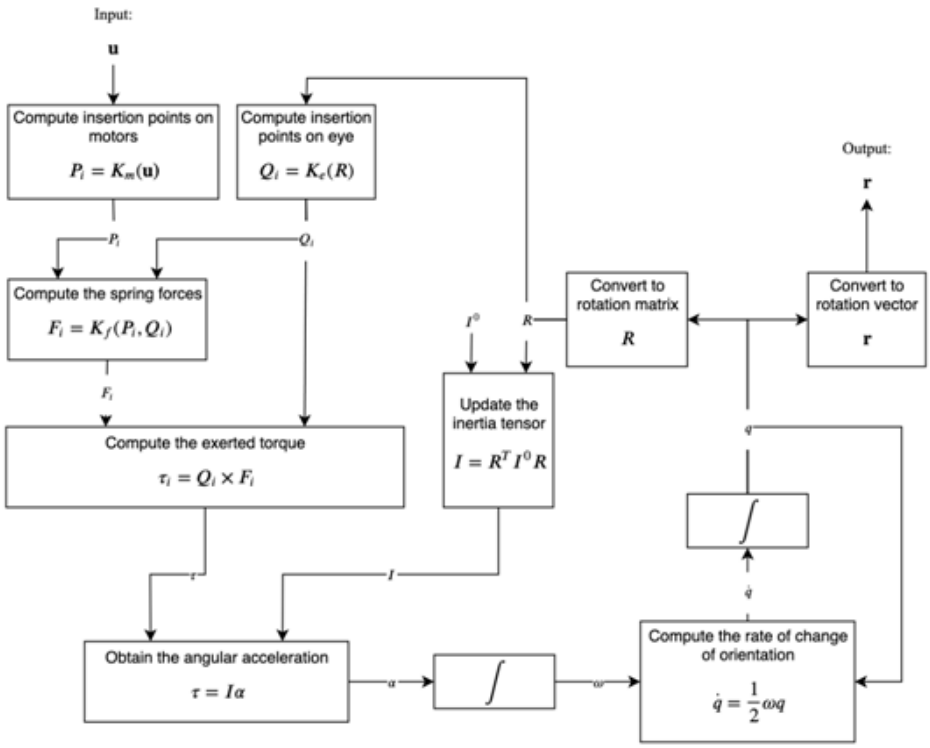



**Figure 3.2:** Diagram of the initial nonlinear 3D model of the human eye, with position inputs. Adapted from [3].

Here, the input generates a change in the motor insertion points, $P_i$ (see figure 3.3). To compute the torque created by the elastics, one must first compute the position of the insertion points in the eye, $Q_i$, which vary with the eye orientation, $\mathbf{v}$. This is done by rotating the initial position of these points to the current position, using a rotation matrix:

$$Q_i = RQ_{i0} \tag{3.2}$$

The force exerted by each elastic on each insertion point can be computed as follows:

$$F_i = \frac{k_i}{l_{0i}}\big(\,\|P_i - X_i\| + \|X_i - Q_i\|\,\big)\frac{X_i - Q_i}{\|X_i - Q_i\|} \tag{3.3}$$

Where, as seen in figure 3.3, $X_i$ represents the intermediate points through which the elastics pass. Finally, we can calculate the torque applied in the eyeball by the elastics

$$\tau_{ela=} \sum_i Q_i \times F_i \tag{3.4}$$

The total torque applied to the eyeball consists not only of the one created by the elastics, $\tau_{ela}$, but also to the friction and gravity torques

$$\tau = \tau_{ela} + \tau_f + \tau_g \tag{3.5}$$

where,

$$\begin{aligned} \tau_f &= -\beta\omega \\ \tau_g &= C_m \times F_g \end{aligned} \tag{3.6}$$

$C_m$ is the center of mass, given by $C_m = R(\mathbf{v})C_{m0}$, and $F_g$ is the gravity force.

We know from Newton's second law that torque is related to angular acceleration according to

$$\tau = J\alpha \tag{3.7}$$

finally, integrating $\alpha$, we get the angular velocity, which can be combined to the orientation inside the model, expressed in quaternions, $\mathbf{q}$, to obtain the rate of change in quaternions. This is achieved by representing the angular velocity $\boldsymbol{\omega}$ by a quaternion with null scalar part, $\omega$. [1]

$$\dot{q} = \frac{1}{2}\omega \circ q \tag{3.8}$$

This value is further integrated and combined with the initial orientation in quaternions, rotating it. With this, we arrive at a new orientation of the eye in the simulator.

However, it is our goal to adapt this model, changing the muscle model so that the system input becomes a neuronal signal given to muscles which produce force, without disregarding their elastic characteristics. To do so, we use the assumption that the six extraocular muscles act in three pairs - the oblique pair (superior and inferior) cause torsional movements, the superior and inferior rectus pair is responsible for vertical movements while the lateral and medial rectus pair account for the horizontal component of saccades so respectively, we assume that these three muscle pairs produce separately

[1] Starting from the initial time-step, $k = 0$, the orientation of the subsequent time-step is given by $q(1) = \omega \circ q(0)$. Thus, to get the orientation at any time-step $k$ we do $q(k) = \omega^k \circ q(0)$ and, representing the angular velocity quaternion as $\omega = e^{\frac{\theta}{2}\mathbf{n}}$ it is simple to arrive at $\frac{d}{dt}q(k) = \frac{\theta}{2}\mathbf{n} \circ q(k)$ and finally, noting that $\frac{\theta}{2}\mathbf{n} = \omega$ then we get (3.8)

the torsional, vertical and horizontal components of the torque to be applied in the eyeball, but have coupled elastic components, dependent on the eye orientation

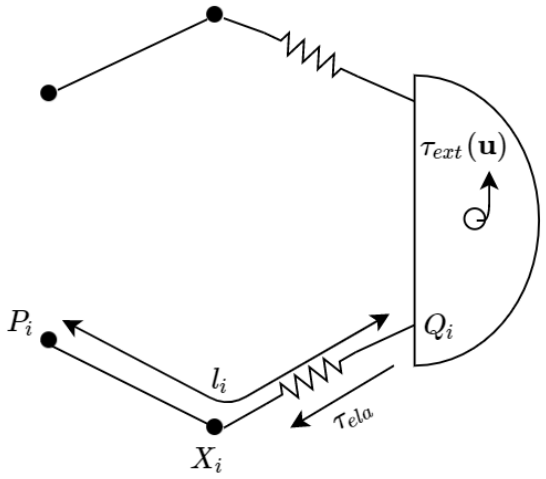



**Figure 3.3:** Biomimetic robot eye one-dimensional model. $P_i$ represents the points where elastics are fixed, $X_i$ are points through which the elastics pass and $Q_i$ are the insertion points of the elastics in the eyeball. These elastics represent the elastic part of muscles which act in pairs. The force these muscles produce is applied directly in the eyeball as an external force, $\tau_{ext}$

In this model, when one of the muscles of each pair receives an activation signal for generating force, it contracts, applying a force in the eyeball(agonist).The other muscle receives no input and is simply extended (antagonist) creating a force in the direction contrary to the movement, as represented in figure 3.3. In the three-dimensional case, the forces created by the three muscle pairs form a torque which we called $\tau_{ext}$. The action of the muscles translates into the sum of two torques applied to the eyeball - the external torque (agonist) and the elastic torque (antagonist):

$$\tau_{musc} = \tau_{ext} + \tau_{ela} \tag{3.9}$$

The total torque is now different from the one in the initial model (3.5):

$$\tau = \tau_{musc} + \tau_f + \tau_g \tag{3.10}$$

Furthermore, as stated in 2.3, it is desired to obtain the output in the angle-axis representation. This is done by converting the quaternion to angle-axis using equation (2.8). We finally arrive at the new form of the model with the requires changes, seen in figure 3.4.

Using the exponential map [24] to parameterize the coordinates of the eye according to (2.16) and (2.17) and Newton's $2^{nd}$ law to model rotations (3.7), we arrive at the following system of differential

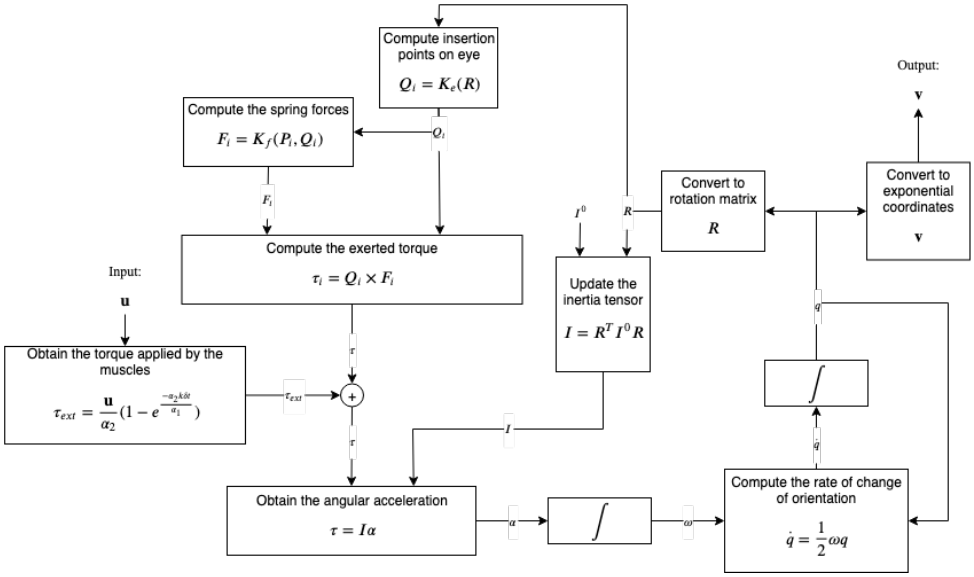



**Figure 3.4:** Diagram of the nonlinear 3D model of the human eye

equations which rules the system.

$$\begin{cases} \dot{\mathbf{v}} = \frac{1}{2}(\gamma\boldsymbol{\omega} + \boldsymbol{\omega} \times \mathbf{v} + \eta\mathbf{v}) \\ \dot{\boldsymbol{\omega}} = J^{-1}(\tau_{ext}(\mathbf{u}) + \tau_{ela}(\mathbf{v}) + \tau_g(\mathbf{v}) - \tau_f(\boldsymbol{\omega})) \end{cases} \tag{3.11}$$

We have presented the analytical formulae to compute all the quantities present in (3.11) except for the input defined $\tau_{ext}$. By solving equation (3.1) in the 3D case where $f = \tau_{ext}(\mathbf{u})$ we have

$$\tau_{ext}(\mathbf{u}) = \frac{\mathbf{u}}{\alpha_2}(1 - e^{\frac{-\alpha_2 t}{\alpha_1}}) \tag{3.12}$$

considering equation 3.12, it becomes clear that when $\alpha_1$ tends to zero, the torque is simply

$$\tau_{ext}(\mathbf{u}) = \frac{\mathbf{u}}{\alpha_2} \tag{3.13}$$

For simplicity of modeling the system in the state-space, $\alpha_1$ was assumed to be close to zero since the

time constant in the muscle actuators is much smaller than that of the remaining system and this removes the burden of having a time dependency in the input matrix of the system. Also as time increases in (3.12), the torque value also tends to (3.13).

Thus, defining the state, $\mathbf{x}$ as the orientation and angular velocity of each dimension, and the input as the neuronal commands, $u$ we have the following system

$$\begin{aligned} \dot{\mathbf{v}} &= f_1(\mathbf{v}, \boldsymbol{\omega}) \\ \dot{\boldsymbol{\omega}} &= f_2(\mathbf{v}, \boldsymbol{\omega}, \mathbf{u}) \end{aligned} \tag{3.14}$$

Here, $\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$ and $\boldsymbol{\omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}$ thus this leads us to the $6^{th}$-order state-space:

$$\mathbf{x} = \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{3.15}$$

with inputs:

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \tag{3.16}$$

Moreover, we consider the brain to have access to a position signal in its feedback loop (see section 2.4) and thus we have

$$\mathbf{y} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \tag{3.17}$$

To design a feedback loop for our nonlinear model of the eye it is necessary to get a linear estimate of the system's dynamics, $A$, and input influence, $B$. For this, two main approaches were considered: (i) using MATLAB/System Identification Toolbox and (ii) the analytical linearization of the described system. These are explained in detail below.

## 3.3 System Identification

Having the model built with the necessary specifications as previously stated, it is time to estimate a representation of the system's dynamic properties to be used when building the system's controller and also a state estimator - in this case, its state-space matrices, $A, B$ and $H$. The identification of linear

models is done here around the equilibrium point $(\mathbf{x}_0, \mathbf{u}_0)$ and are thus valid near this point:

$$\mathbf{x}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \mathbf{u}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.18}$$

The estimation of these parameters is called system identification and is performed through mathematical methods using observed input-output pairs of signals. Using Matlab's System Identification toolbox, the model of the robot was identified with the subspace state-space estimation approach (n4sid) by using, as an input signal in $\mathbf{u}$, a pseudo-random binary signal (PRBS) and observing its output's characteristics, $\mathbf{y}$. The PRBS is a binary sequence ranging from two values whose periods are pseudo-random. Thus, there are some restrictions which have to be obeyed to get a reasonable identification of the system. Here, the main parameters to take into account were both the input's clock period, i.e. the time period for a change in the input value to be allowed, which should not be too low to give the system time to stabilize before changing the input value again, and the input range in the 3 different channels.

Furthermore, we imposed the identification to be made in a $6^{th}$-order state-space since each of the 3 dimensions of the movement can be described by a second-order differential equation.

In order to keep the relation between the hidden state, $\mathbf{x}$ and the output, $\mathbf{y}$, a a restriction on the values in the output matrix had to be imposed:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

This way, we can force the values of $x_1$, $x_3$ and $x_5$ to represent the eye's angular position components in the eye reference frame axes $x$, $y$ and $z$.

However, this way of parameterizing the system cannot force the other state variables to be the system velocities in the different directions. Therefore, we compute a linear approximation to the system analytically, where the state variables are completely known.

## 3.4 System Linearization

The state equations (3.11) which rule the motion of the system are nonlinear. Through Jacobian linearization of the system, we can attain a linear approximation of the system for perturbations, $\delta$,

around some operating point $(\mathbf{x}_0, \mathbf{u}_0)$ using the following method:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ \mathbf{f}(\mathbf{x}, \mathbf{u}) &= \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) + \left.\frac{df}{d\mathbf{x}}\right|_{\mathbf{x}_0, \mathbf{u}_0} \delta\mathbf{x} + \left.\frac{df}{d\mathbf{u}}\right|_{\mathbf{x}_0, \mathbf{u}_0} \delta\mathbf{u} \\ \delta\mathbf{x} &= \mathbf{x} - \mathbf{x_0} \\ \delta\mathbf{u} &= \mathbf{u} - \mathbf{u_0} \end{aligned} \tag{3.19}$$

In the case of the operating point, $(\mathbf{x}_0, \mathbf{u}_0)$, being an equilibrium point, we know that the system is not moving and therefore we have

$$\mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) = 0 \tag{3.20}$$

which leaves us with

$$\delta\dot{\mathbf{x}} = \frac{d(\mathbf{x} - \mathbf{x_0})}{dt} = \dot{\mathbf{x}} \tag{3.21}$$

Finally, substituting in (3.19) we arrive at a way of computing the linearized state equation of the system around the equilibrium point $(\mathbf{x}_0, \mathbf{u}_0)$

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = \left.\frac{df}{d\mathbf{x}}\right|_{\mathbf{x}_0, \mathbf{u}_0} \delta\mathbf{x} + \left.\frac{df}{d\mathbf{u}}\right|_{\mathbf{x}_0, \mathbf{u}_0} \delta\mathbf{u} \tag{3.22}$$

Analogously to (2.18), where the system is formulated in the state space, and (3.14), the state equation, we have then:

$$\begin{aligned} A &= \begin{bmatrix} \frac{df_1}{d\mathbf{v}} & \frac{df_1}{d\boldsymbol{\omega}} \\ \frac{df_2}{d\mathbf{v}} & \frac{df_2}{d\boldsymbol{\omega}} \end{bmatrix} \\ B &= \begin{bmatrix} \frac{df_1}{d\mathbf{u}} \\ \frac{df_2}{d\mathbf{u}} \end{bmatrix} \end{aligned} \tag{3.23}$$

The dimensions to each of the derivatives in these matrices are $(3 \times 3)$ since both $f_1$ and $f_2$ are defined in $\mathbb{R}^3$ as well as the states $\mathbf{v}$ and $\boldsymbol{\omega}$, leading to the dimensions of $(6 \times 6)$ and $(6 \times 3)$ for matrices $A$ and $B$, respectively. To compute these derivatives, some support from the MATLAB/Symbolic Math Toolbox was used given the extensiveness of these calculations.

Regarding (3.11) and (3.14), for $f_1$ the jacobian was calculated exclusively using this tool. On other hand, for $f_2$ we considered the inertia moment to be constant around the equilibrium point (in reality it changes slightly with the eye orientation as it moves the centre of mass), and so there are two parts of it which depend on $\mathbf{v}$ - the torque produced by the elastics, $\tau_{ela}$ and the gravity torque, $\tau_g$. In fact, as the system is making use of rotation matrices for computing both the torques, the chain rule applies - the symbolic tool computes the derivatives with respect to a rotation matrix variable of the muscle insertion points, $Q$, and then making use of the exponential map the derivative of $Q$ with respect to the exponential

variables, $v$, is analytically computed and multiplies the first:

$$\frac{df_2(Q(\mathbf{v}))}{d\mathbf{v}} = \frac{df_2(Q(\mathbf{v}))}{d\mathbf{Q}(\mathbf{v})}\frac{dQ(\mathbf{v})}{d\mathbf{v}} \tag{3.24}$$

The derivative with respect to $\omega$ is very straightforward to compute since the only part of $f_2$ depending on it is the friction torque, $\tau_f = -\beta\omega$.

In the end, we get a linearization of the continuous system which we want to control stochastically, so it has to be discretized. For this purpose, we use MATLAB/Control System Toolbox function *c2d()* with zero-order hold method and a sampling time of 0.001s, which was considered adequate for the purpose of this work. However, this value can easily be changed if needed depending on the particular control architecture to be used in the real system. The linearized system matrices can be found in appendix A.

## 3.5 Results

### 3.5.1 Muscle responses

The $1^{st}$ order agonist muscle system response is presented in figure 3.5, for different values of $\alpha_1$, with fixed $\alpha_2 = 1$. We know from control systems theory that a first-order system has settling time of $3\tau$, and $\tau$ is the time constant, given in this case by $\tau = \frac{\alpha_1}{\alpha_2}$. So, the with fixed $\alpha_2$, the settling time is dependent on the value of $\alpha_1$ alone - as its value increases, the system becomes slower.

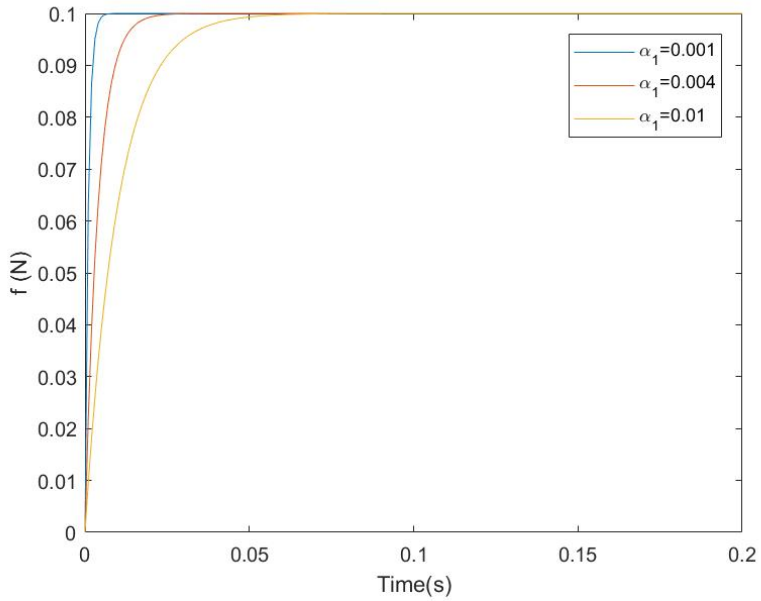



**Figure 3.5:** Agonist muscle response to a neuronal signal of amplitude 0.1, with $\alpha_2 = 1$ and different values of $\alpha_1$

The responses in 3.5 confirm what was expected: with a low value of $\alpha_1$ the muscles respond almost instantaneously, while with higher values the settling time increases. In section 3 we considered $\alpha_1 = 0$ to simplify the state-space model, since the time-constant of the remaining eye system is high enough to discart the delay in the muscle response.

Let us now analyse the combined response of the agonist and antagonist pair of muscles. For this, we will simulate a neuronal step signal activating the agonist muscle: lateral rectus - horizontal component in the positive sense of the force. This results in a positive torque in the eye ball that moves horizontally and pulls the antagonist muscle that counteracts this force passively via its elastic characteristic. We expect that this torque will have the opposite sign in the horizontal direction, leading to the stabilization of the muscle torque at zero $\tau_{z_{musc}} = 0$ and thus demonstrating the agonist/antagonist action of the muscles.

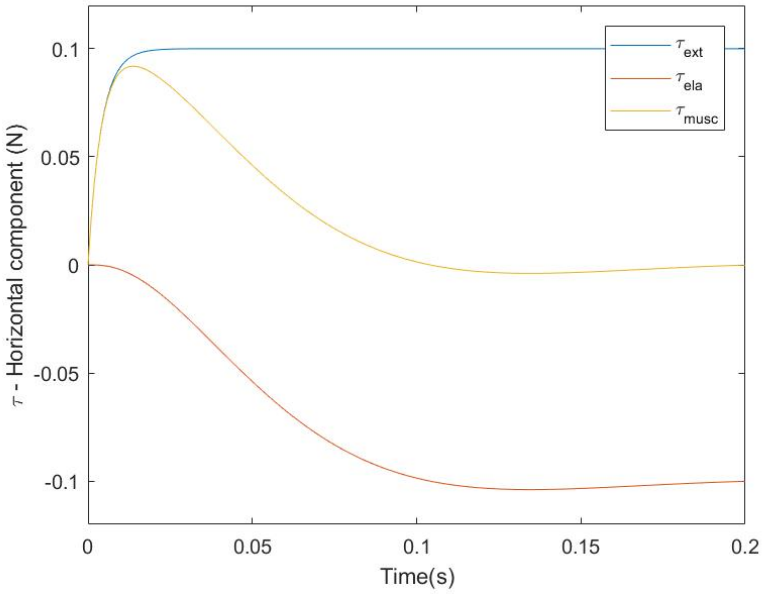



**Figure 3.6:** Agonist/antagonist action of the muscle model. Here, $\tau_{ext}$ represents the lateral rectus as it simulates a contraction (agonist) in the positive sense of the horizontal component of the torque, while $\tau_{ela}$ simulates the action of the medial rectus (antagonist). The action of both results in a pulse, $\tau_{musc}$

As seen in figure 3.6, the simulated response is as expected. The step neuronal signal originates a step response of the agonist muscle, simulating its contraction, while the elastics produce a contrary torque stabilizing the total torque produced by the pair of muscles over a time period of roughly $130ms$.

### 3.5.2 Identification and Linearization

To validate if the 3D Nonlinear System (NLS) is correctly modelled by the state equations (3.11) a step input was put into the NLS in MATLAB/Simulink and its output was observed. From the output, the rate

of change of the exponential coordinates and the angular acceleration were both calculated through the presented formula. Simultaneously, the values of $\omega$ were obtained from the simulation of the system as well as the orientation output and its numerical differentiations were made in MATLAB.

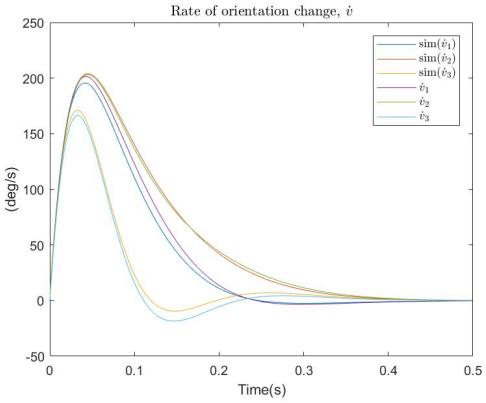



**Figure 3.7:** Comparison between the numerically and analytically calculated values of the rate of change of the exponential coordinates

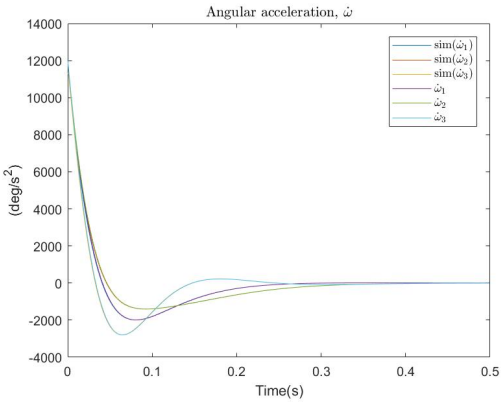



**Figure 3.8:** Comparison between the numerically and analytically calculated values of the angular acceleration

| | $\dot{v}_1$ | $\dot{v}_2$ | $\dot{v}_3$ | $\dot{\omega}_1$ | $\dot{\omega}_2$ | $\dot{\omega}_3$ |
|---|---|---|---|---|---|---|
| Relative error(%) | 1.23 | 0.66 | 1.10 | 2.19 | 3.41 | 8.23 |

**Table 3.1:** Relative error between analytical computations and experimental values of $\dot{\mathbf{v}}$ and $\dot{\omega}$ in the 3 different dimensions

The results for these calculations are shown in figures 3.7 and 3.8 for $\dot{v}$ and $\dot{\omega}$ respectivelly. The relative error between the numerical and the analytical results was calculated using the following method and the resultant values are presented in table 3.1.

$$\begin{aligned} RE\dot{v}(\%) &= \frac{\sum_{k=1}^{p} \frac{\dot{\mathbf{v}}_{LS}(k) - \dot{\mathbf{v}}_{NLS}(k)}{\dot{\mathbf{v}}_{NLS}(k)}}{p} \times 100\% \\ RE\dot{\omega}(\%) &= \frac{\sum_{k=1}^{p} \frac{\dot{\boldsymbol{\omega}}_{LS}(k) - \dot{\boldsymbol{\omega}}_{NLS}(k)}{\dot{\boldsymbol{\omega}}_{NLS}(k)}}{p} \times 100\% \end{aligned} \tag{3.25}$$

As one can see, the error values are quite low, confirming what can be seen in the plots - the system is well described by the state equations and, therefore, we can proceed for the linearization of the system.

The analytical linearization of the system was done as described in 3.4. A Matlab function was created, which allows for the linearization of the system around any given operating point, $(\mathbf{v}, \mathbf{u})$, in a time-efficient manner - the whole linearization is done with symbolic variables and these can then be replaced by the operating point very rapidly, providing a valid linearization of the eye model around the whole ocular range. The reason why it is important to have a fast linearization is discussed further in the

document.

The linearization around the equilibrium point

$$\mathbf{x} = \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.26}$$

was tested together with the model obtained from system identification in two ways - firstly, using the *compare()* function from Matlab and then by performing saccades controlled in open-loop (without any disturbance), using the Linearized System (LS) parameters for the optimization. The commands obtained from the optimal control approach were then used to generate the saccades in the three different systems: the LS, the Identified System (IDS) and the NLS, which is our ground truth for comparing the identification and linearization.

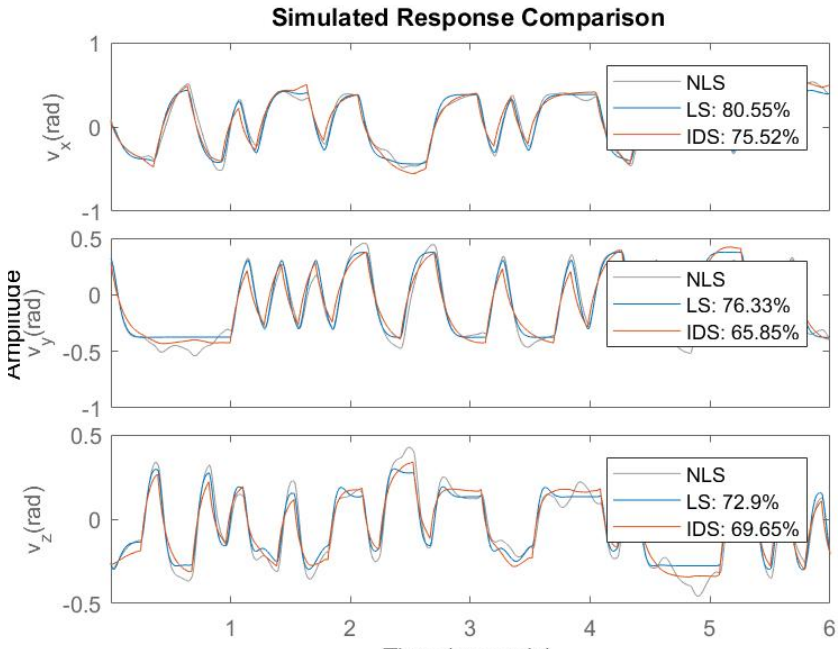



**Figure 3.9:** Simulated response of IDS and LS superimposed in their 3 dimensions - the percentages represent the normalized mean squared goodness measure of each model in each dimension, the values of the output, $v$, are expressed in radians

Figure 3.9 shows the plot obtained from Matlab/System Identification Toolbox's function *compare()*. The fitness value is calculated by

$$fit(i) = 1 - \frac{\|\mathbf{x}_{zval}(i) - \mathbf{x}(i)\|}{\|\mathbf{x}_{zval}(i) - mean(\mathbf{x}_{zval}(i))\|} \tag{3.27}$$

and it indicates how close are the systems' outputs relative to the measured output in the NLS, contained in $zval$. This result suggests that the NLS is better identified by the linearization than by Matlab's System Identification Toolbox. To get a better grasp on how this results while performing saccades, we followed to the second experiment.

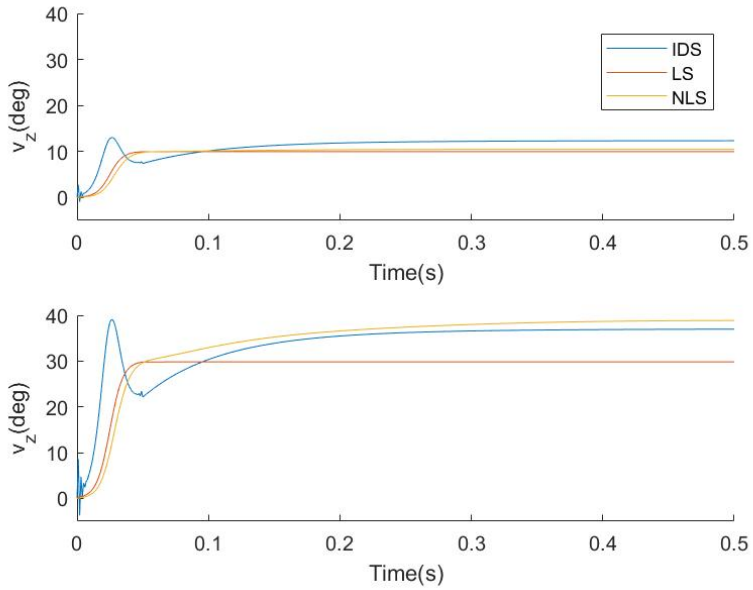



**Figure 3.10:** Response of the three different systems to the same saccade inputs. (Up) Horizontal saccade with goal set to $10^{\circ}$, in which it can be seen that the NLS behaves similarly to the LS. (Down) Horizontal saccade of large amplitude $30^{\circ}$, where although the IDS has a bad transient phase, its steady state value is closer to the NLS the one displayed by the LS.

In figure 3.10 the responses from the three systems to the saccade input is shown for a small amplitude and a bigger amplitude saccade. It becomes clear when analysing these plots that when the LS is working further away from the linearization operating point, it performs worse than for smaller magnitudes when compared to the NLS. However in bigger amplitude saccades, in fact, the IDS seems to stabilize in a similar value to the NLS, unlike the linearized version. Also, it is straightforward to notice that the LS behaves better than the IDS for small amplitude saccades both in the rise and in the static phases - while the IDS overshoots and oscillates, the LS is seemingly critically damped. To quantify how much worse the LS gets for bigger amplitudes, the mean relative error was calculated for a range of different amplitudes in horizontal and vertical saccades, and plotted as seen in figures 3.11 and 3.12.

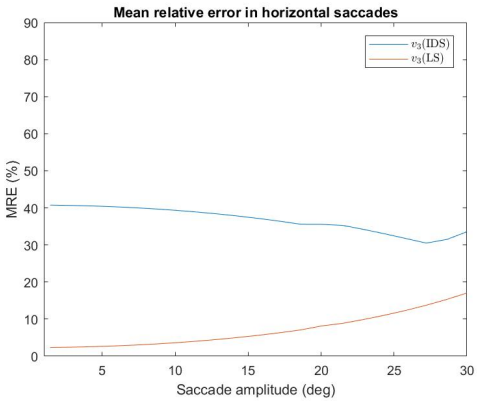



**Figure 3.11:** The relative error in a horizontal saccade is below 20% in the LS but always greater than 30% in the IDS in a ocular range of 30º

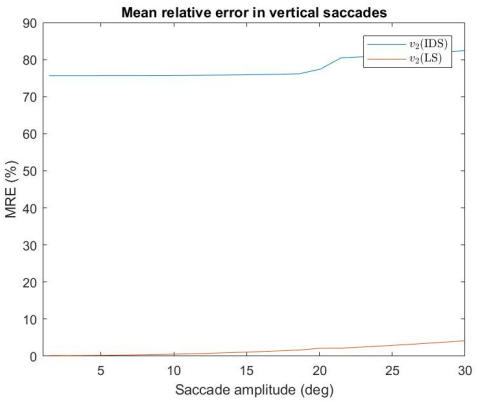



**Figure 3.12:** The relative error in a horizontal saccade is very low (<5%) in the LS but always greater than 70% in the IDS in a ocular range of 30º

# 4

# Optimal Feedback Control of Saccades

## Contents

In 2.4, we presented state-of-the-art work on optimal feedback control of biogological movements. Here, we elaborate on the various approaches considered in this thesis and analyze each different approach relative to the real human eye behaviour described in 2.2 by testing them in our 3D eye simulator.

## 4.1 Control design

The notion of the oculomotor system adopted in this thesis is presented in section 2.1, and an overview of its components is given through the rest of chapter 2. Here, we elaborate more on the design of the feedback loop implemented to the eye plant.

To perform a saccade to a certain goal orientation $\mathbf{y_{ss}}$, we first optimize the expected value of the cost function, as in (2.29), for a fixed movement duration, $p$. The expected cost $E[J(p)]$, in the framework of stochastic feedback control, is given by (2.27) for all time steps in the saccade movement $k \in [0; p]$

This process is repeated by searching in the whole plausible space of $p$ - we assume that a saccade cannot be shorter than a minimum duration, $min_p$, nor longer than a maximum duration, $max_p$. The movement duration which minimizes the expected cost is then selected, and the corresponding gains are used in the control. Detailed information on the optimization process is given in section 2.4.2 and the considered forms of the cost function are described later in section 4.2.

The state-space equations of the closed-loop system are

$$\begin{aligned}
\mathbf{x}^{(k+1)} &= A\mathbf{x}^{(k)} + B(\mathbf{u}^{(k)} + \epsilon_u^{(k)}) + \epsilon_x \\
\mathbf{y}^{(k)} &= H(\mathbf{x}^{(k)} + \epsilon_s) + \epsilon_y \\
\hat{\mathbf{x}}^{(k+1)} &= A\hat{\mathbf{x}}^{(k)} + AK^{(k)}(\mathbf{y}^{(k)} - H\hat{\mathbf{x}}^{(k)}) + B\mathbf{u}^{(\mathbf{k})} \\
\mathbf{u}^{(k+1)} &= -G^{(k+1)}\hat{\mathbf{x}}^{(k+1)}
\end{aligned} \tag{4.1}$$

The optimal control gains, $G^{(k)}$, are computed only for the time-steps $k = 1 \dots p$. To keep the system in steady state after instant $p$ we have to continue giving actuation to the system, otherwise it will return to the rest position. We do this by switching to a control scheme where the control gains are zero and the reference input forces a new equilibrium point at the target position.

The feedback of the system is in form of a position signal and the actuation is made through the neuronal activation signal which generates force. Given that the control architecture of the system is based on proportional actuation facing the state error (fig.4.1), when this error is null the controller does not generate any actuation and hence the system does not stabilize in the goal orientation. In other words, the system is type 0, which means that zero steady-state error is not assured without integral action and there is, then, the need to introduce a gain in the reference value and a feedforward term $\mathbf{u}_{ff}$

Introducing a reference input in a feedback control loop is a well-studied problem in control theory. To

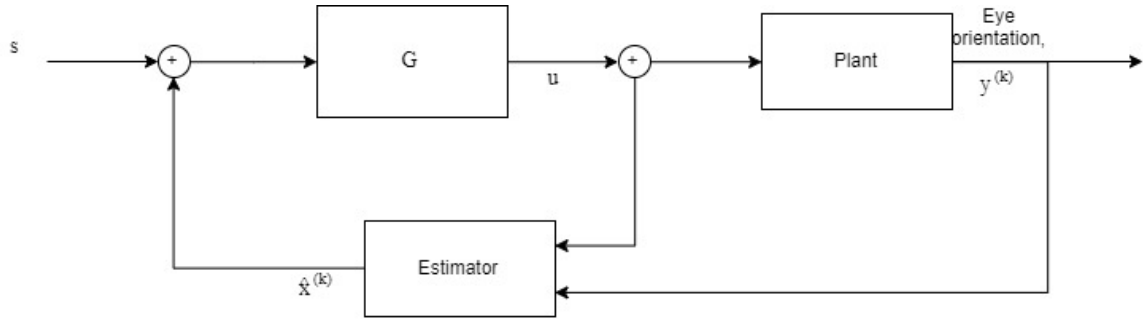



**Figure 4.1:** Type 0 system with proportional controller - this system cannot stabilize with zero steady-state error.

implement this, we based ourselves on the explanation given in [28] on Reference Inputs with Estimators (Chapter 8). These contributions are added to the system employing two gains, $N_x$ and $N_u$, which are obtained based on the system's dynamics

$$\begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} A - I & B \\ H & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{4.2}$$

The gains $N_x$ and $N_u$ are implemented in the system as seen in figure 4.2. By setting as reference the desired goal orientation, it generates a reference state, $\mathrm{s}$ and a reference motor command $\mathbf{u}_{ff}$, as seen in the block diagram in figure 4.2.

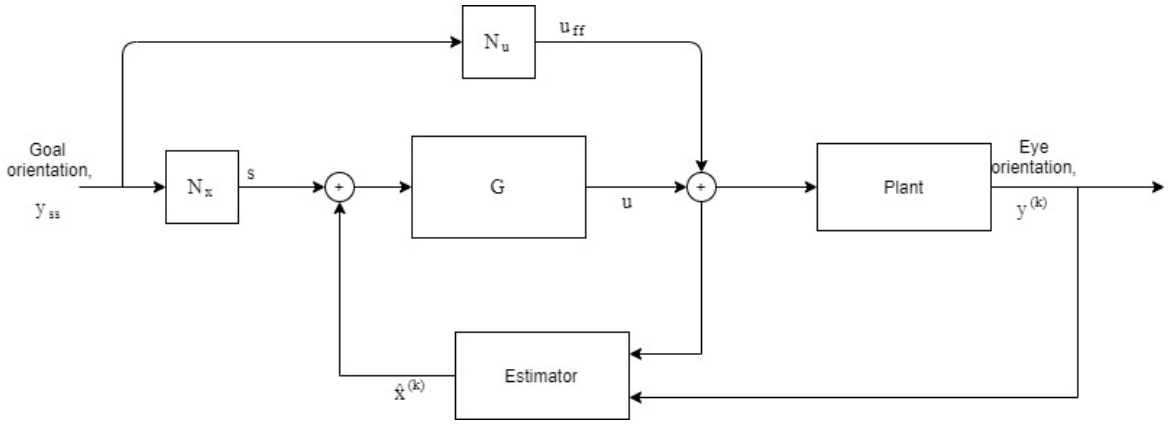



**Figure 4.2:** Diagram of system with reference tracking gains $N_x$ and $N_u$.

Hence, this implies that at the end of the movement, as the saccade optimal control gains are no longer defined, the system is controlled only by the reference input. Since we are studying fundamentally the saccadic movement and not the subsequent period of fixation, the noise is removed at the end of

the movement, $p$, for simplification purposes.

$$
\begin{aligned}
& k \leq p \\
\mathbf{x}^{(k+1)} &= A\mathbf{x}^{(k)} + B(\mathbf{u}^{(k)} + \mathbf{u_{ff}} + \epsilon_u^{(k)}) + \epsilon_x \\
\hat{\mathbf{x}}^{(k+1)} &= A\hat{\mathbf{x}}^{(k)} + AK^{(k)}(\mathbf{y}^{(k)} - H\hat{\mathbf{x}}^{(k)}) + B(\mathbf{u^{(k)}} + \mathbf{u_{ff}}) \\
& k > p \\
\mathbf{x}^{(k+1)} &= A\mathbf{x}^{(k)} + B\mathbf{u_{ff}} + \epsilon_x \\
\hat{\mathbf{x}}^{(k+1)} &= A\hat{\mathbf{x}}^{(k)} + B\mathbf{u_{ff}}
\end{aligned}
\tag{4.3}
$$

The final system has the following equations:

$$
\begin{aligned}
& k \leq p \\
\mathbf{x}^{(k+1)} &= A\mathbf{x}^{(k)} + B(\mathbf{u}^{(k)} + \mathbf{u_{ff}} + \epsilon_u^{(k)}) + \epsilon_x \\
\mathbf{y}^{(k)} &= H(\mathbf{x}^{(k)} + \epsilon_s) + \epsilon_y \\
\hat{\mathbf{x}}^{(k+1)} &= A\hat{\mathbf{x}}^{(k)} + AK^{(k)}(\mathbf{y}^{(k)} - H\hat{\mathbf{x}}^{(k)}) + B(\mathbf{u^{(k)}} + \mathbf{u_{ff}}) \\
\mathbf{u}^{(k+1)} &= G^{(k+1)}(\mathbf{s} - \hat{\mathbf{x}}^{(k+1)}) \\
& k > p \\
\mathbf{x}^{(k+1)} &= A\mathbf{x}^{(k)} + B\mathbf{u_{ff}} + \epsilon_x \\
\mathbf{y}^{(k)} &= H\mathbf{x}^{(k)} + \epsilon_y
\end{aligned}
\tag{4.4}
$$

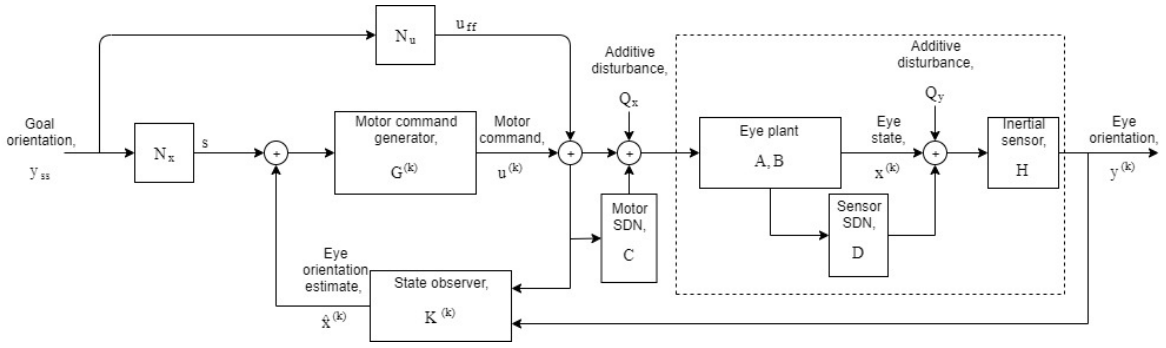



**Figure 4.3:** Block diagram of the oculomotor system. The optimal control splits between an optimal estimator and an optimal command generator. The generated commands are disturbed with additive and multiplicative noise. The resulting command is applied in the eye plant and its state, which is hidden, is also disturbed with additive noise and SDN. The available information to the observer is the eye orientation.

Figure 4.3 summarizes the implementation of the feedback loop on the existing eye model, which was done using Matlab/Simulink. The derivation of the control and estimation gains, $G$ and $K$ can be found in sections 2.4.2 and 2.4.3 respectively, in which we use the stochastic optimal control Approach

developed in [32]. The description for matrices $A, B, H, C, D, Q_x$ and $Q_y$ is presented in section 2.4.1, where we give the background on the generic state-space representation of systems as well as on the specific case of systems with multiplicative terms of noise.

## 4.2 Optimal control Approaches

While the theoretical background for feedback optimal control was given in section 2.4.2, here we elaborate on the principles of the brain when performing saccades - the cost function of our optimal control problem. As stated previously, the parameters for which the brain tries to optimize when performing movements is a widely studied and debatable subject. Many hypotheses have been analyzed through optimal control, by matching different optimality principles with the stereotypical eye dynamics, but very few of these studies considered the specific case of optimal feedback control of saccades with SDN. However, there have been studies on this subject [13, 20] which are used here as the main background. In both these works, the cost assigned to saccades was considered to depend on three different aspects - Accuracy, Energy and Duration (**AED**).

$$J = J_x + J_u + J_p \tag{4.5}$$

In the framework of stochastic feedback control, this translates into the cost per step described by (2.25). This formulation of the optimal control problem has allowed researchers to obtain main sequence behaviour of their models while respecting Listing's law both in the mentioned works and in open-loop optimal control formulations.

Recall that the expected value of the cost in any step $k$ is given by the cost per step, $\alpha^{(k)}$, plus the expected value of the state resulting from applying the policy at that step, $\mathbf{u}^{(k)}$:

$$v_{\pi*}(\mathbf{x}^{(k)}, \hat{\mathbf{x}}^{(k)}) = \min_{\mathbf{u}^{(k)}} \left\{ \begin{array}{l} \mathbf{u}^{(k)T} L \mathbf{u}^{(k)} + \mathbf{x}^{(k)T} T^{(k)} \mathbf{x}^{(k)} + \dfrac{\lambda\beta}{1+\beta p} \\ + E[\mathbf{x}^{(k+1)T} W_x^{(k+1)} \mathbf{x}^{(k+1)} | \mathbf{x}^{(k)}, \hat{\mathbf{x}}^{(k)}, \mathbf{u}^{(k)}] \\ + E[\mathbf{e}^{(k+1)T} W_e^{(k+1)} \mathbf{e}^{(k+1)} | ...] + w^{(k+1)} \end{array} \right\} \tag{4.6}$$

Also, the expected values in the above equation have the form (2.29) which depends on the variability of the state and the estimation error and therefore, in the presence of SDN depend on the quadratic form of the motor commands:

$$\begin{aligned} E[\mathbf{x}^{(k+1)} | \mathbf{x}^{(k)}, \hat{\mathbf{x}^{(k)}}, \mathbf{u}^{(k)}] &= A\mathbf{x}^{(k+1)} + B\mathbf{u}^{(k)} \\ \mathrm{var}[\mathbf{x}^{(k+1)} | ...] &= Q_x + \sum_i BC_i \mathbf{u}^{(k)} \mathbf{u}^{(k)T} C_i^T B^T \end{aligned} \tag{4.7}$$

| Approach | Cost terms | Signal-dependent noise |
|---|---|---|
| 1 | AED | - |
| 2 | AED | ✓ |
| 3 | AD | - |
| 4 | AD | ✓ |

**Table 4.1:** Different Approaches considered in the formulation of the optimal feedback control problem.

$$\begin{aligned} E[\mathbf{e}^{(k+1)}|...] &= (A - AK^{(k)}H)e^{(k)} \\ \text{var}[\mathbf{e}^{(k+1)}|...] &= Q_x + \sum_i BC_i\mathbf{u}^{(k)}\mathbf{u}^{(k)T}C_i^T B^T + AK^{(k)}Q_y K^{(k)T}A^T \\ &\quad + \sum_i AK^{(k)}HD_i\mathbf{x}^{(k)}\mathbf{x}^{(k)T}D_i^T H^T K^{(k)T}A^T \end{aligned} \tag{4.8}$$

This gives us the hint that the brain might not be minimizing effort but only endpoint accuracy and duration while considering the influence of SDN, as has been proposed previously [9, 38]:

$$J = J_x + J_p \tag{4.9}$$

which, in the stochastic framework leads us to the cost per step

$$\alpha(k) = \mathbf{x}^{(k)T}T^{(k)}\mathbf{x}^{(k)} + \frac{\lambda\beta}{1+\beta p} \tag{4.10}$$

We call this Approach **AD** as it optimizes accuracy and duration, although with the presence of SDN we are implicitly optimizing effort as well.

In sum, we expect that a system without SDN which minimizes effort will behave similarly to one with SDN but disregarding the effort in the cost function. In both cases the system is penalizing the sum of squared motor commands, $\mathbf{u}^2$, either through minimization of effort or variability, respectively. Although some results have been obtained suggesting this analogy [9, 31, 39], these works have not considered a three-dimensional model of the eye as ours does.

Table 4.1 summarizes the Approaches considered in experiments. In all these Approaches it is necessary to tune variables $T$, $\lambda$ and $\beta$ which play a role in the cost functions to minimize as explained in section 2.4.2. In the AED optimization Approaches it is necessary to consider another variable correspondent to the effort penalization cost, $L$.

Moreover, to validate the correct behaviour of saccades in each Approach, it is necessary to define a set of metrics with the human physiological ground which allow us to match the obtained results with the normal human saccade behaviour.

## 4.3 Saccade metrics

Here, we present the factors to consider when evaluating the performance of our system in emulating the behaviour of the human eye. In section 2.2 we presented the most commonly used characteristics of this group of eye movements on research works with the same goal as ours: (i) the main sequence, (ii) velocity profile asymmetry and (iii) the Listing's plane. However, as we are implementing optimal feedback control for the first time to a three-dimensional model of the saccadic system, there is another item which we have to consider in the validation of the performed simulations - the straightness of saccades. To sum up, there are four elements to consider in this work:

- **Main sequence** - the stereotypical relation between amplitude and duration and between amplitude and peak velocity of saccades must be observed.
- **Asymmetry** - as the duration of saccades gets longer, their temporal velocity profiles tend to become more asymmetrical, with the peak velocity being attained early in the movement.
- **Listing's plane** - the eye torsion is zero in head-fixed saccades - $v_x = 0$.
- **Straightness** - Oblique saccades exhibit a slight curvature in their trajectories.

To analyse if saccades performed by the simulator obey the main sequence, we perform multiple simulations from random initial orientations to random goal orientation within a range of $[-15°; 15°]$. This is a range where our linearization is valid, with low error between the linearized and nonlinear system as seen in figures 3.11 and 3.12. The amplitude, duration and peak velocity of each trial is stored and then plotted. Then we compare with the results from human trials: duration increases linearly with amplitude and that peak velocity increases with amplitude.

The asymmetry of the velocity profiles is measured through a method suggested in [17] which consists on obtaining a relation between the acceleration period, i.e. the time until the peak velocity is attained, and the total saccade duration,

$$S_{coef} = \frac{k_{peak}}{p} \tag{4.11}$$

The resulting value should be, for saccades, at most $S_{coef} = 0.5$, and should decrease towards $S_{coef} = 0$ as saccade duration increases, meaning that the peak is attained early in the movement and never after half of the duration.

To check for Listing's plane constraint, the whole trajectory is recorded in each saccade. Then, the horizontal components, $v_z$ are plotted against the torsion values, $v_x$ and the same is done between vertical, $v_y$ and torsional components. It is expected that these plots form a cluster around the zero

torsion line in both the graphs - the Listing's plane. A good measurement for the thickness of Listing's plane is its standard deviation, $\sigma_x$, which is around $0.6^\circ$ for humans.

The straightness of a saccade can be measured quantitatively by computing the correlation between its vertical and horizontal velocity components. Our system is considerably coupled, so only with coordinated control of each muscle is it possible to attain straight trajectories. In human saccades, only a slight curvature is observed in oblique saccades - with a near unitary correlation between vertical and horizontal velocities in oblique saccades. To analyse the curvature of saccades we perform oblique saccades in which the angle between horizontal and vertical components is at least $25^\circ$, and the movement amplitude is within the acceptable range of our approximation, shown in section 3.5.2 (we use the value of $20^\circ$). The correlations between horizontal and vertical velocity components are computed and the minimum correlation trial is plotted so that it is possible to visualize the most curved saccade.

## 4.4 Results

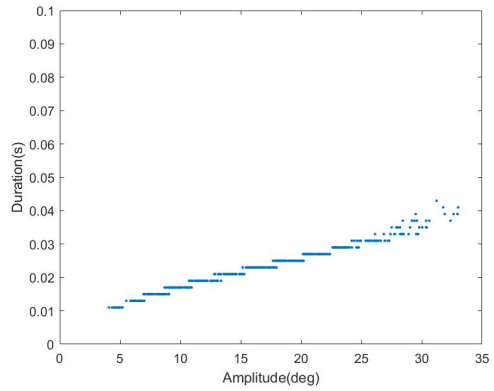
In this section, we present the results on the four different criterions presented in 4.3 obtained using the different Approaches taken and specified in table 4.1. The values of variables $T$, $\lambda$ and $\beta$ were kept constant through the different experiments, varying only the energy penalty term, $L$ and the motor noise matrix, $C$. A further analysis on LP is given for a case in which torsion is unconstrained. It is important to note that the search for the optimal movement duration was made in multiples of two sampling periods, and the sampling period is $\Delta t = 0.001s$

### 4.4.1 Approach 1 - AED optimization without SDN

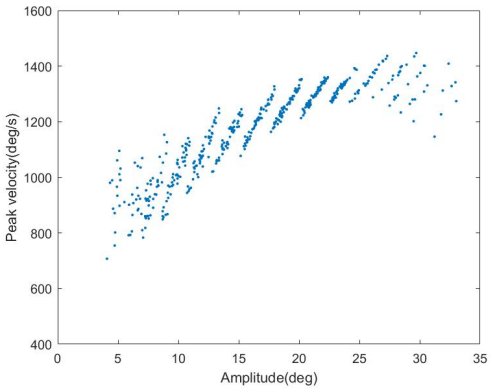
Multiple saccades were simulated setting the multiplicative noise matrices $C$ and $D$ to zero, and keeping the additive noise variances with the values $q_x = 0.001$ and $q_y = 0.001$ The data recorded from each trial - saccade duration, amplitude, peak velocity and final position - were used to plot the results regarding the specified saccade metrics.

• **Main sequence** In figures 4.4, the relations between amplitude, duration and peak velocity are displayed, in order to analyse if the non-linear kinematic properties are respected in this Approach.

As expected, the compromise between accuracy, duration and effort originates movements with main sequence properties - saccade duration grows linearly with amplitude as seen in 4.4a while the maximum attained velocity during the movement grows nonlinearly (figure 4.4b), reaching a saturation point around 1300 (deg/s). Because the saccades have randomly generated initial and goal orientations, there is some variability in the velocity for movements with the same amplitude since the different directions lead to slightly different costs.

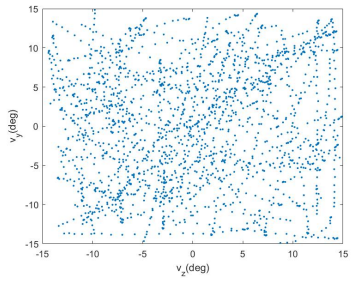



**(a)** Amplitude vs. duration

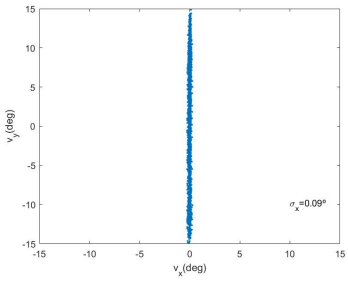



**(b)** Amplitude vs. peak velocity

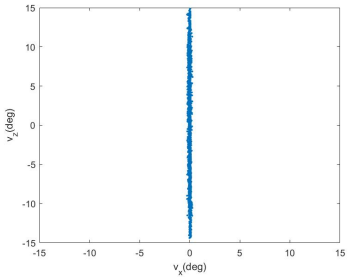
**Figure 4.4:** Approach 1 - Main sequence



**(a)** Horizontal component vs. Vertical component




**(b)** Torsion vs. Horizontal component



**(c)** Torsion vs. Vertical component

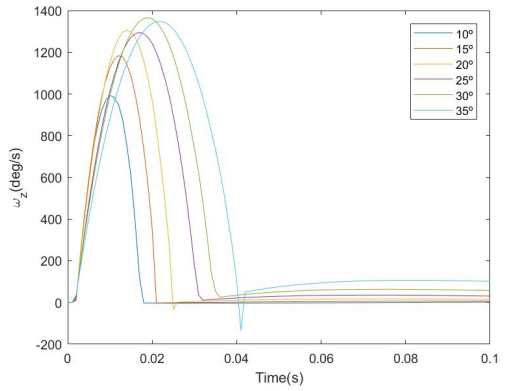
**Figure 4.5:** Approach 1 - Listing's plane

• **Listing's plane** The trajectories of saccades were recorded and their torsion values through each trial were plotted against the vertical and horizontal components, to analyse if the torsion lies in a plane and if it corresponds to the Listing's plane (figure 4.5).

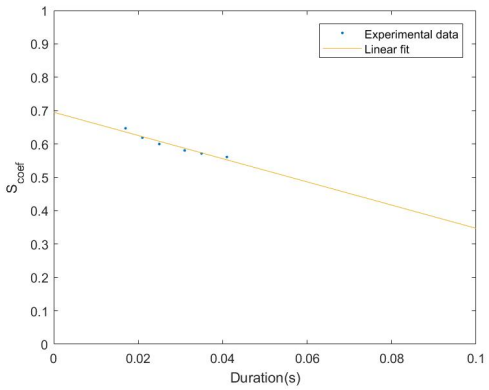
The obtained plots show roughly a line in both the XY and XZ planes. The obtained value for the standard deviation of the torsion is $\sigma_x = 0.09°$, which is a very low value. The reason for the low thickness of the plane is that we are constraining the endpoint torsion as much as the vertical and horizontal components, in matrix $T$.

• **Velocity profiles** Horizontal saccades of different amplitudes were performed to check how their velocity profiles differ with amplitude. $S_{coef}$ was calculated with equation 4.11 and a trend line was fitted to facilitate the visualization of its evolution.

Figure 4.6a shows the effect of the effort penalization in this Approach with the saturation on peak velocity, and the linear increase in duration with saccade amplitude. Notice that for small amplitude saccades, when velocity decreases towards zero it remains in zero (e.g. $10°$ line), in contrast to what happens with bigger amplitudes (e.g. $35°$ line). The reason why this happens is that our controller is




**(a)** Temporal velocity profiles of horizontal saccades with different amplitudes




**(b)** $S_{coef}$ relation with movement duration

**Figure 4.6:** Approach 1 - Velocity profiles and their skewness

based on the linearization made and, as stated in 3.5.2, the error between the linearization and the nonlinear model increases when the saccade amplitude moves away from the equilibrium point in which the model was linearized. This, together with the fact that after the saccade as reached its duration, $p$, the controller has no effect and the model is thus controlled by the feedforward command alone, yields the sudden change in velocity observed in larger saccades.

Another feature to be analysed from the plots in figure 4.6 is the asymmetry of velocity profiles. Empirical observations on humans have shown that the velocity peak is attained early in the movement, resulting in asymmetrical curves for larger movement amplitudes. This was not observed using Approach 1 - although the value of $S_{coef}$ decreases with amplitude, it is always greater than 0.5, meaning that the peak velocity is reached after half of the movement duration. We assume this to happen only in the presence of multiplicative noise, so this result is as expected.

• **Curvature** The saccade with the most curved trajectory from the set of oblique movements simulated is shown in figure 4.7. The value of the correlation between horizontal and vertical velocity components is very high and it can be seen in the figure that the movement is almost perfectly straight as expected.

### 4.4.2 Approach 2 - AED optimization with SDN

After adding signal-dependent noise to the simulator, with $c_i = 0.01$ for $i = 1, 2, 3$, i.e with equal noise magnitudes in the 3 input dimensions, again multiple saccades were simulated optimizing accuracy, effort and duration, using the same values in the accuracy penalty matrix T, effort matrix L and temporal reward discounting $\alpha$ and $\beta$.

• **Main sequence** The main sequence plots, seen in figure 4.8, have the desired shapes, demon-

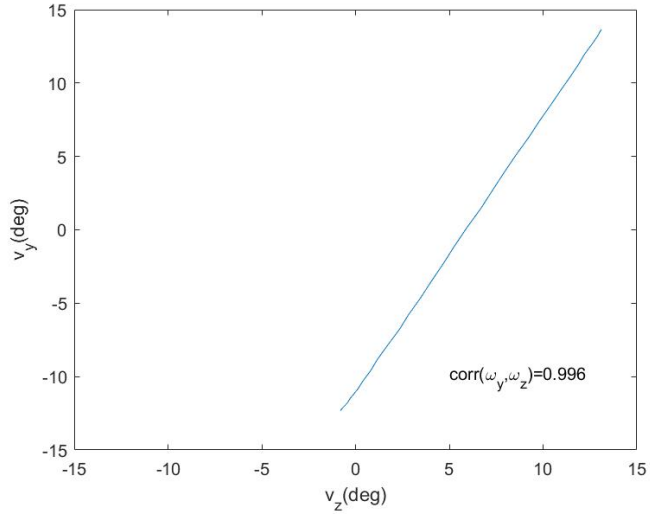
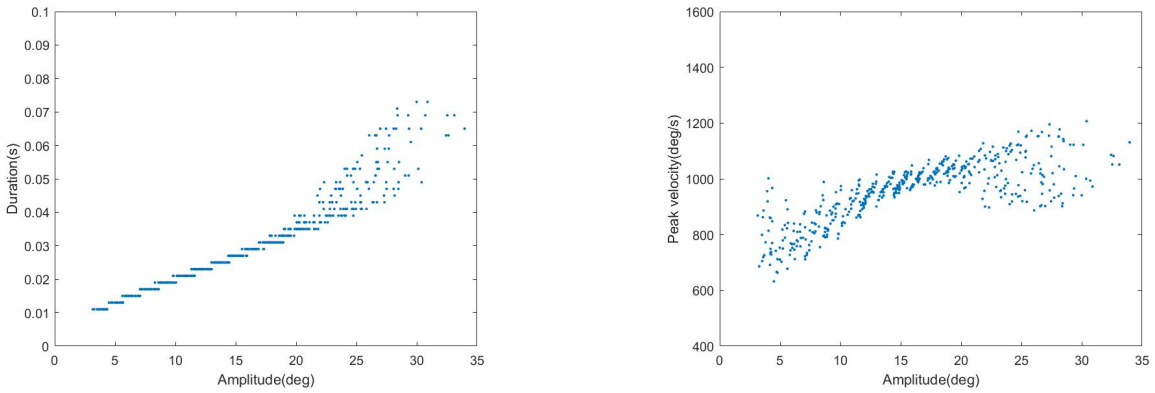



**Figure 4.7:** Saccade with lowest correlation between horizontal and vertical velocity components from a set with mean correlation of 0.998



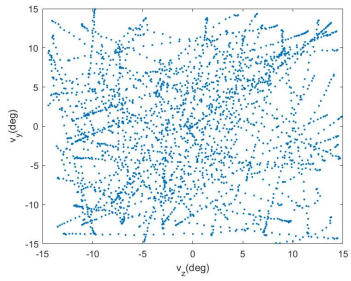

**(a)** Amplitude vs. duration

**(b)** Amplitude vs. peak velocity

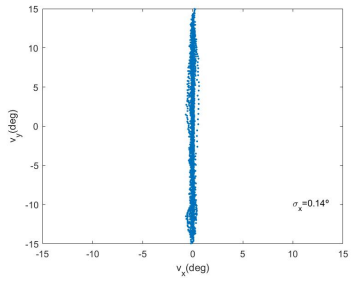
**Figure 4.8:** Approach 2 - Main sequence

strating the linear relation between amplitude and duration (4.8a) and the saturating relation between amplitude and peak velocity (4.8b). However, these have differences when compared to the main sequence relations obtained in Approach 1, where multiplicative noise was inexistent – the slope of the linear relation between amplitude and duration is bigger and the peak velocity attains lower values for the same amplitudes, saturating at around 1100(deg/s). These differences reflect the influence of multiplicative noise in the system. By introducing this kind of disturbance which is as stated proportional to the squared motor commands, the system finds it more advantageous to produce lower motor commands, reducing the effort. The result is a bigger movement duration and analogously lower maximum velocity for a saccade of the same amplitude, hence explaining the bigger increase of duration with amplitude and the lower value of saturation in the peak velocity.

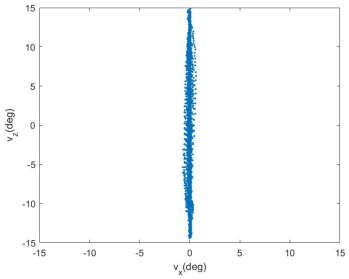
- **Listing's plane** By plotting the orientations in the XY and XZ plane, the listing's plane is shown in




**(a)** Horizontal component vs. Vertical component

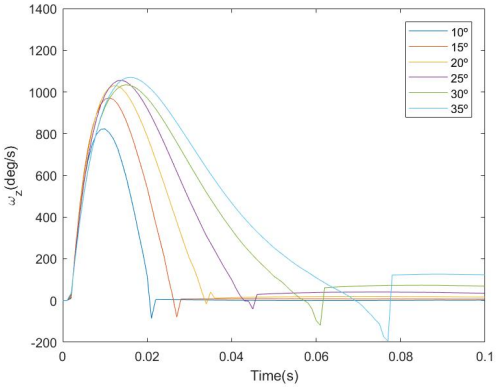



**(b)** Torsion vs. Horizontal component

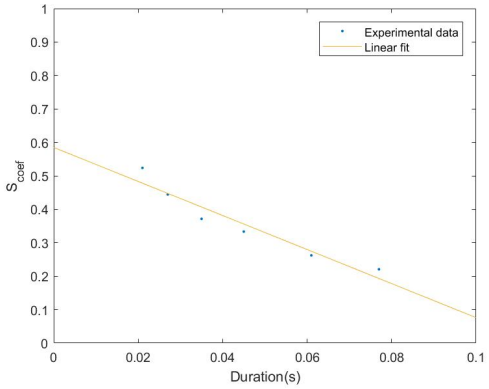



**(c)** Torsion vs. Vertical component

**Figure 4.9:** Approach 2 - Listing's plane




**(a)** Temporal velocity profiles of horizontal saccades with different amplitudes




**(b)** $S_{coef}$ relation with movement duration

**Figure 4.10:** Approach 2 - Velocity profiles and their skewness

figure 4.9. Again, the Listing's plane displays a very low thickness, with an insignificant increase in the standard deviation. ($\sigma_x = 0.14°$)

• **Velocity profiles** The temporal velocity profiles here show the lower saturation values on the peak velocity as well as the increased difference between the durations of different amplitude saccades, confirming the conclusions attained by analyzing the main sequence plots. Furthermore, it is noticeable that in high amplitude movements the velocity profiles show asymmetry, reaching its maximum value early in the saccade. This is further confirmed by the $S_{coef}$ plotted against movement duration in figure 4.10b, with values under 0.5 which decrease with movement duration. This result confirms the expectation that the asymmetry in temporal velocity profiles is originated by SDN.

• **Curvature** The saccade obtained in the lowest correlation trial, seen in figure 4.11, is slightly curved. The mean value of correlation is roughly the same as in Approach 1, but the trials in Approach 2 have greater variability also in the correlation between velocity components given the introduction of SDN. In fact, the saccade presented here is more realistic relatively to the ones performed by humans.

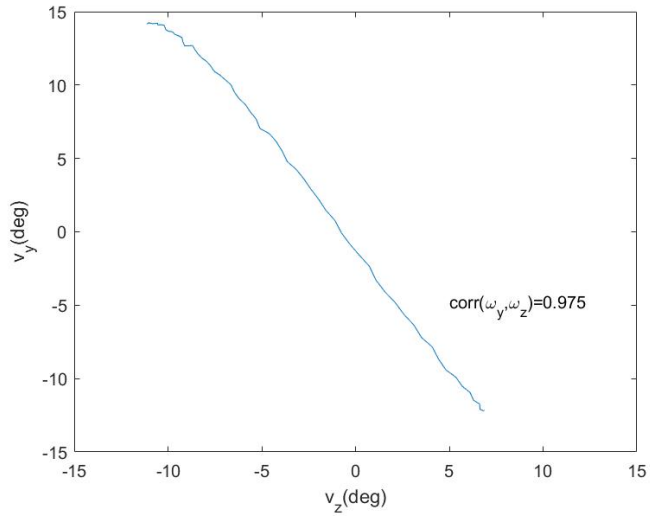



**Figure 4.11:** Saccade with lowest correlation between horizontal and vertical velocity components from a set with mean correlation of 0.990

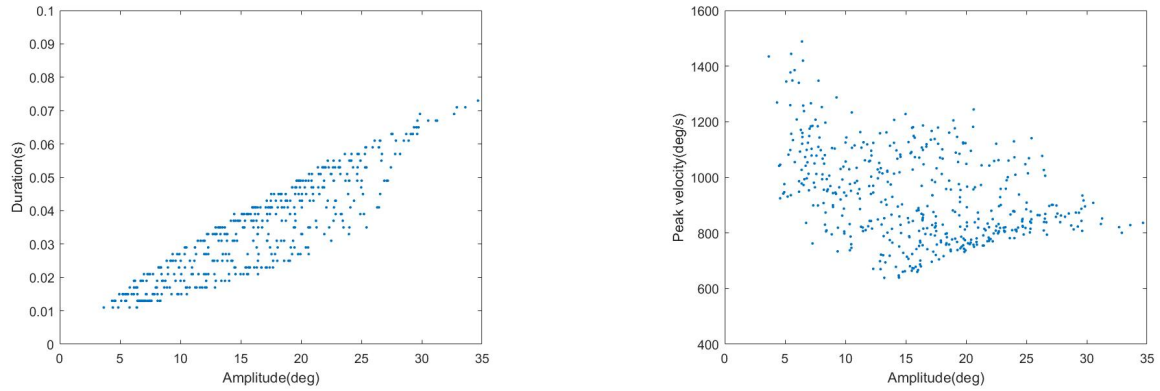
### 4.4.3 Approach 3 - AD optimization without SDN

Approach 3 was experimented by removing the cost term on effort ($L = 0$), thus optimizing accuracy and duration. As in Approach 1, SDN was set to zero and saccades with random initial and goal orientations were simulated.

• **Main sequence** The main sequence dependencies are in this Approach, as expected, not followed. Analysing figure 4.12, although the duration seems to increase linearly with amplitude, the relation between amplitude and peak velocity is strange - bigger velocities are attained given the complete disregard on the values of the motor commands in this Approach. The absence of effort penalization and signal-dependent noise implicitly in the accuracy term causes the system to see no advantage in producing low motor commands, resulting in big peak velocity values.

• **Listing's plane** The trajectories obtained in this Approach also obey the Listing's plane restriction, once more with very low thickness as seen in figure 4.13

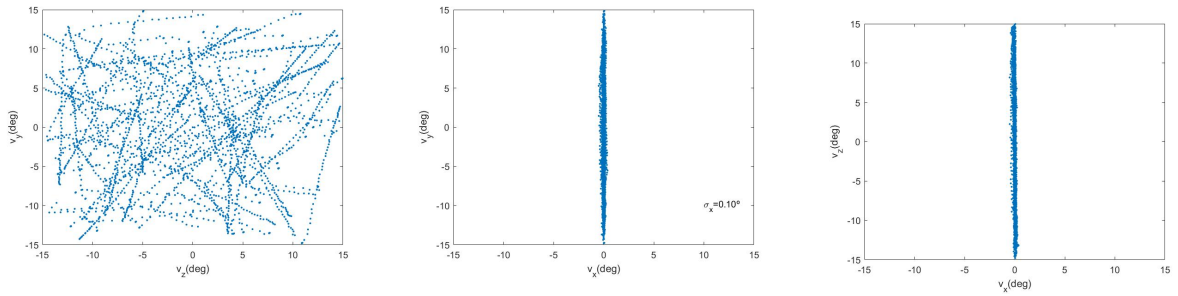
• **Velocity profiles** In simulations with no SDN and not optimizing effort, there is no reason not to produce big motor commands since we are in no way interested in minimizing $\mathbf{u}$. The temporal velocity profiles in figure 4.14a reflect this effect, with the maximum velocity of the saccade being reached as soon as the movement begins as a result of the system finding itself in the state with the biggest value of error, to which the motor commands are proportional (see equation 4.4). This fast reaction of the controller produces velocity profiles in the system which do not correspond to the shape of the ones observed in human saccades, although their asymmetry increases with movement duration (figure 4.14b. On the other hand, we can see that for bigger amplitudes the peak velocities decrease, as a result of the trade-off between endpoint accuracy and duration - the system finds it more convenient to




**(a)** Amplitude vs. duration

**(b)** Amplitude vs. peak velocity

**Figure 4.12:** Approach 3 - Main sequence

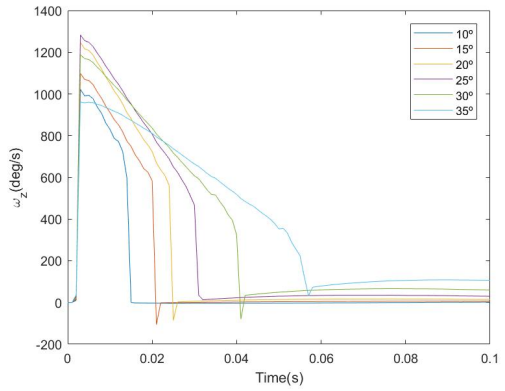



**(a)** Horizontal component vs. Vertical component

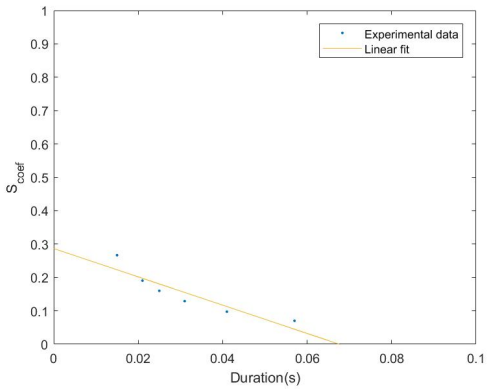
**(b)** Torsion vs. Horizontal component

**(c)** Torsion vs. Vertical component

**Figure 4.13:** Approach 3 - Listing's plane




**(a)** Temporal velocity profiles of horizontal saccades with different amplitudes




**(b)** $S_{coef}$ relation with movement duration

**Figure 4.14:** Approach 3 - Velocity profiles and their skewness

achieve the goal orientation accurately and hence takes more time to reach it, which results in lower gain values given that we use $T^{(k)} = 0$ for all time-steps except the last, $k = p$. To get a better grasp on the influences these parameters have on the controller gains, see 2.4.2. Approach 3 fails to replicate velocity profiles of human saccades.

• **Curvature** The most curved saccade simulated in Approach 3 has the lowest correlation between velocity components in all the Approaches, with the reasonably lower value of $0.8744$. This is most likely caused by the fact that motor commands are not being minimized and the produced velocities are thus more irregular as seen in figure 4.14a. Nevertheless, the trajectory described by the eye, displayed in figure 4.15 is still reasonably straight.

### 4.4.4 Approach 4 - AD optimization with SDN

Finally, we added SDN to the system and experimented it optimizing accuracy and duration in several random saccades. The results are presented below.

• **Main sequence** As in Approaches 1 and 2, the nonlinear dynamic properties of saccades are respected in this case as shown in figure 4.16. Although the effort is not constrained, the addition of motor signal-dependent noise is enough to make the main sequence relations similar to Approach 2, where effort was minimized. Notice that these plots are much the same as the ones obtained in Approach 2 - meaning that the values of noise used make the introduction of this disturbance play a more significant role in the optimization than the effort term.

• **Listing's plane** In figure 4.17 on can see that the torsion in experimented saccades stays in Listing's plane throughout the movements, once again resulting in a very low thickness of the plane.

• **Velocity profiles** The obtained plots of velocity profiles for different amplitude horizontal saccades

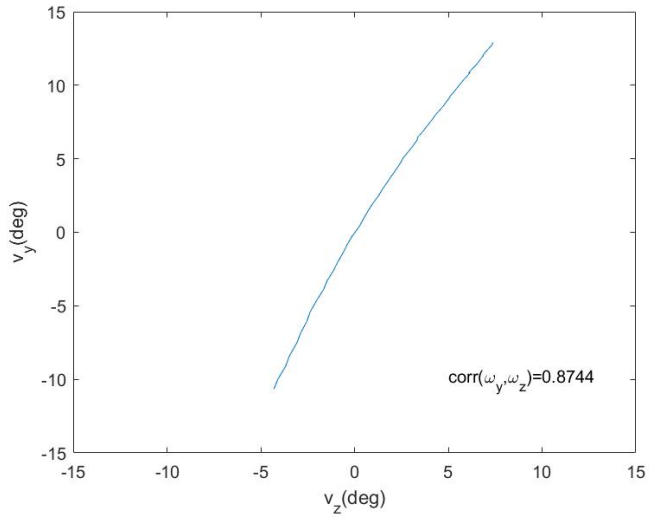
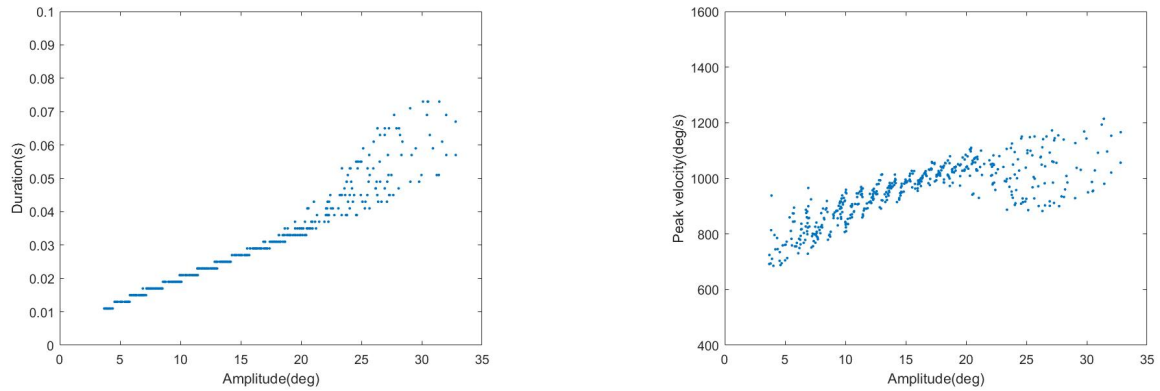


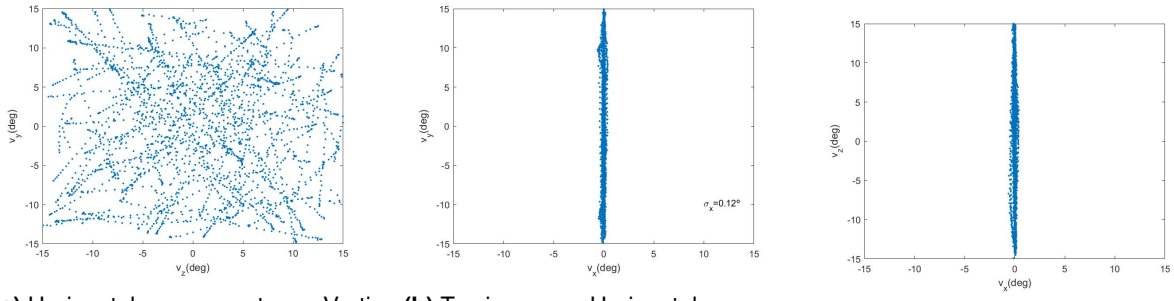

**Figure 4.15:** Saccade with lowest correlation between horizontal and vertical velocity components from a set with mean correlation of 0.839




**(a)** Amplitude vs. duration

**(b)** Amplitude vs. peak velocity

**Figure 4.16:** Approach 4 - Main sequence

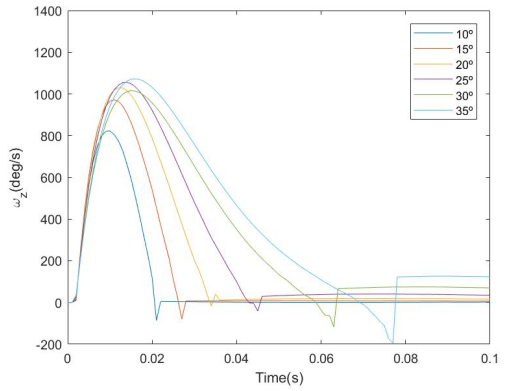



**(a)** Horizontal component vs. Vertical component

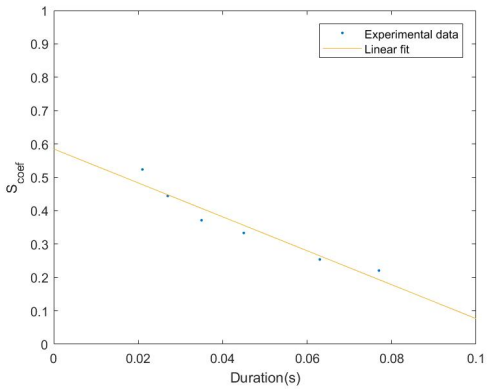
**(b)** Torsion vs. Horizontal component

**(c)** Torsion vs. Vertical component

**Figure 4.17:** Approach 4 - Listing's plane




**(a)** Temporal velocity profiles of horizontal saccades with different amplitudes




**(b)** $S_{coef}$ relation with movement duration

**Figure 4.18:** Approach 4 - Velocity profiles and their skewness

are shown in figure 4.18a. These are very similar to the ones obtained using Approach 2, presenting an increased asymmetry in bigger amplitude movements as well as peak velocity saturation, as expected since this Approach also considers the influence of SDN.

• **Curvature** The curvature of the performed saccades is here very similar to what has been observed in Approach 2, confirming once more that these two Approaches are highly connected. Both the minimum correlation trial and the mean of correlations have almost the same values. The most curved saccade, seen in 4.19, presents a reasonably straight trajectory as expected.

By matching the results obtained in this Approach to the ones presented for Approach 1 and 2, we can take the conclusion that, elegantly, the minimization of endpoint accuracy in a system endowed with noise that depends on the square of the motor commands, $\mathbf{u}^2$, produces the same realistic saccadic movements as optimizing effort, confirming the hypothesis suggested in [9, 38] which motivates this thesis. Moreover, all the Approaches considered optimal feedback control and positive results were obtained in this framework, which is consistent with our assumption that the oculomotor system relies on a feedback loop.

Moreover, the trajectories described by saccades in all of the tested Approaches are slightly curved and present near-unitary correlations between horizontal a vertical velocity components. In a system where no restrictions were imposed to assure straightness of trajectories, the optimality principles were sufficient to generate realistic saccades with scaled components of velocity. We conclude from these facts that at the origin of generating saccades is a vectorial source which is decomposed, generating nearly straight movements.

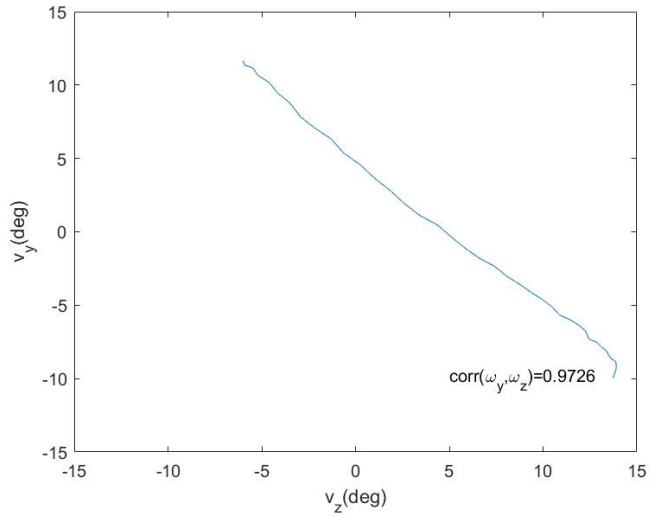
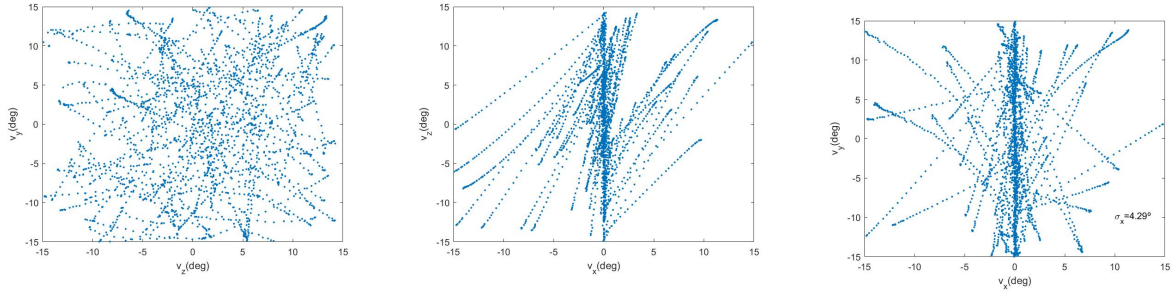



**Figure 4.19:** Saccade with lowest correlation between horizontal and vertical velocity components from a set with mean correlation of 0.986



**(a)** Horizontal component vs. Vertical component

**(b)** Torsion vs. Horizontal component

**(c)** Torsion vs. Vertical component

**Figure 4.20:** Eye trajectories with unconstrained torsional component.

### 4.4.5 Unconstrained torsion

After obtaining the Listing's plane (LP) results shown in the previous sections for Approaches 1-4, we tested the system behaviour upon removal of the weight related to the torsion in the endpoint accuracy term of the cost function, given that the obtained LPs in the different Approaches present much the same thicknesses. Furthermore, the obtained standard deviation values for the torsion are too low considering the measurements made in humans in other research works.

With the purpose of testing the effect of this modification in our Approach, again multiple saccade trajectories with random goal directions were recorded. The performance of the system was, in this case, bad considering the previous Approaches - the torsion values are much higher and thus the eye does not move in a plane.

The reason for this might be related to the coupling between the 3 dimensions in the model, especially

the torsional and horizontal components, which results in highly coupled gains for $u_x$ and $u_z$ and thus causing the torsion values to vary much in the favour of horizontal gains when it is not constrained. This aspect is illustrated by the values of the dynamic matrix obtained in the linearization, in appendix A.1. The results in figure 4.20 suggest that high torsion values correspond to high horizontal amplitude saccades. Another hypothesis is that since no importance is given to the torsional component, the variability is accumulated in this irrelevant dimension.

Nonetheless, further research on this topic would be useful to confirm or contradict these hypotheses.

# 5

# Conclusion

## Contents

This work proposed to implement an optimal feedback controller of saccades in a three-dimensional model of a mimetic robot of the eye plant, resembling the human oculomotor system. Through optimal feedback control, a controller and an estimator were designed to perform saccades optimally according to a cost function. To study the principles optimized by the brain to perform these movements, different cost functions were tested, with different noise conditions, includind Signal-dependent noise (SDN). The results obtained were put against empirical data on the behaviour of human eyes with repect to kinematics properties (Listing's plane constraint, trajectory curvature) and dynamical properties (the main sequence and asymmetric temporal velocity profiles).

Using the model previously built in the scope of ORIENT I project, we adapted the model to a more realistic input of force (muscle activations), instead of position. Having the goal of controlling it using linear control techniques, we had to perform its linearization. Firstly, we used Matlab's System Identification Toolbox to obtain a linear parametrization of the model. However, this method was far from perfect, and another approach was taken by analytical linearization of the model, which proved to fit the needs of this work better.

Having a reliable parameterization of the system, we proceeded to the design of the control loop of the saccadic movements. This was made through optimal feedback control for the duration of the saccade. The problem of fixating the eye in the goal orientation after the saccade was not the main focus of the thesis so, to tackle it, no biological considerations were made and a common control technique based on feedforward was used.

To understand the neural principles behind saccadic movements, a study was made on the cost terms to optimize. In parallel, a study on the contribution of SDN was made for the considered cost functions. The combinations of these two topics resulted in four different approaches to study - accuracy, effort and duration optimization considering either the classical additive noise or the more biologically plausible SDN (Approaches 1 and 2) and accuracy and effort penalization, again with additive noise or SDN (Approaches 3 and 4). To validate each of these approaches we evaluated the obtained results according to a set of criteria defined by empirical observations on humans.

## 5.1 Discussion

After scrutinizing the performance of each approach in the different criteria, we concluded that the minimization of effort is redundant in the presence of SDN, as expected from an analytical perspective - both effort minimization and SDN presence with accuracy minimization imply that the motor commands are penalized. While in the absence of SDN it is necessary to minimize effort to obtain main sequence properties, in its presence, minimizing accuracy and duration is sufficient. Furthermore, we conclude that the asymmetry of velocity profiles arises from the influence of SDN alone - in both AED and AD op-

timizations, an increase of skewness with movement duration was only observed when the multiplicative term of disturbance was present.

The curvatures of the trajectories in the performed experiments by different approaches suggest a confirmation that 3D saccades are generated by a common vectorial generator [21], as the velocity profiles obtained for oblique saccades are scaled versions between the horizontal and vertical components. The most realistic trajectories are observed in approaches with SDN, once more supporting the hypothesis that this disturbance is determinant in saccade dynamics and kinematics.

The results obtained on the Listing's plane analysis were inconclusive. In the performed trials, a weight on the torsional values was assigned, resulting in almost perfect Listing's planes for every approach. By removing the weight on this component, however, the system stopped using exclusively 2 Degrees-of-freedom (DOF), moving on the torsional dimension as well. This seems to be a consequence of using feedback controlling saccades in this system, with coupling between the 3 pairs of muscles but more research on this topic has to be made to confirm this hypothesis.

## 5.2 Contribution

The constributions made with this thesis are the following

- **First empirical proof of 3D saccadic system control using optimal feedback control and considering signal-dependent noise;**
- **Demonstration of the influence of signal-dependent noise on saccade behaviour, together with a study on the influence of different cost terms.**
- Adaptation of mimetic eye model to more realistic input and its analytical linearization;
- Optimal feedback control scripts.

In bold text, the scientific contributions and in normal text, the more technical contribtuions for further work in the project.

## 5.3 Future Work

Although this thesis accomplishes its main goals, some questions still remain.

Firstly, the linearization of the 3D biomimetic robot model which is made in this thesis, although theoretically allowing for linearization in different operating points, it was only done around the equilibrium point $(\mathbf{x}_0, \mathbf{u}_0)$. The scripts developed in this thesis provide a useful base to compute fast linearizations of the system at arbitrary operating points.

Second, we did not address thoroughly the post-saccadic period of the simulation. One option is to include the fixation period in the optimization, by dividing the cost function into two parts, one for the saccade cost and the other for the fixation cost.

Third, it would be interesting to obtain a more detailed figure on how the extraocular muscles are coupled to understand how much effort the brain has to make to prevent eye torsion.

Finally, after concluding the eye system, further work is planned on the ORIENT project to develop an auditory system and the integration of these two systems with a head-neck joint.

# Bibliography


[1] D. Purves, G. Augustine, D. Fitzpatrick, W. Hall, A. LaMantia, and L. White, "The actions and innervation of extraocular muscles," *Neuroscience. 5th ed: Sinauer Associates, Inc*, pp. 436–440, 2012.

[2] A. M. Wong, "Listing's law: Clinical significance and implications for neural control," *Survey of Ophthalmology*, vol. 49, no. 6, pp. 563–575, 2004.

[3] C. A. Tavares, "Control of Saccades with Model of Artificial 3D Biomimetic Eye," no. October, 2019.

[4] T. Haslwanter, D. Straumann, B. J. M. Hess, and V. Henn, "Static roll and pitch in the monkey: Shift and rotation of listing's plane," *Vision Research*, vol. 32, no. 7, pp. 1341–1348, 1992.

[5] A. T. Bahill, M. R. Clark, and L. Stark, "The main sequence, a tool for studying human eye movements," *Mathematical Biosciences*, vol. 24, no. 3-4, pp. 191–204, 1975.

[6] K. Hikosaka, "Delay Activity of Orbital and Lateral Prefrontal Neurons of the Monkey Varying with Different Rewards," *Cerebral Cortex*, vol. 10, no. 3, pp. 263–271, 2000.

[7] R. A. Schmidt and et al, "Motor-output variability: A theory for the accuracy of rapid motor acts," *Psychological Review*, vol. 86, no. 5, pp. 415–451, 1979.

[8] K. E. Jones, A. F. d. C. Hamilton, and D. M. Wolpert, "Sources of signal-dependent noise during isometric force production," *Journal of Neurophysiology*, vol. 88, no. 3, pp. 1533–1544, 2002.

[9] C. M. Harris and D. M. Wolpert, "Signal-dependent noise determines motor planning," *Nature*, vol. 394, no. 6695, pp. 780–784, 8 1998. [Online]. Available: http://www.nature.com/articles/29528

[10] H. H. L. M. Goossens and A. J. Van Opstal, "Optimal Control of Saccades by Spatial-Temporal Activity Patterns in the Monkey Superior Colliculus," *PLoS Comp Biology*, vol. 8, no. 5, p. 1002508, 2012. [Online]. Available: www.ploscompbiol.org

[11] R. Jürgens, W. Becker, and H. H. Kornhuber, "Natural and drug-induced variations of velocity and duration of human saccadic eye movements: Evidence for a control of the neural pulse generator by local feedback," *Biological Cybernetics*, vol. 39, no. 2, pp. 87–96, 1981.

[12] C. A. Scudder, “A new local feedback model of the saccadic burst generator,” *Journal of Neurophysiology*, vol. 59, no. 5, pp. 1455–1475, 5 1988. [Online]. Available: https://www.physiology.org/doi/10.1152/jn.1988.59.5.1455

[13] H. Chen-Harris, W. M. Joiner, V. Ethier, D. S. Zee, and R. Shadmehr, “Adaptive control of saccades via internal feedback.” *The Journal of neuroscience*, vol. 28, no. 11, pp. 2804–13, 3 2008. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/18337410http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2733833

[14] M. Ruiz, “Construction and Characterization of a Biomimetic Robotic Eye Model with Three Degrees of Rotational Freedom : A Testbed for Neural Control of Eye Movements,” no. October, 2017.

[15] R. Shadmehr and S. Mussa-Ivaldi, “Cost of Time in Motor Control,” in *Biological Learning and Control*. The MIT Press, 1 2012, pp. 307–334. [Online]. Available: http://mitpress.universitypressscholarship.com/view/10.7551/mitpress/9780262016964.001.0001/upso-9780262016964-chapter-12

[16] R. Dodge and T. S. Cline, “The angle velocity of eye movements.” *Psychological Review*, vol. 8, no. 2, pp. 145–157, 1901. [Online]. Available: http://content.apa.org/journals/rev/8/2/145

[17] A. J. Van Opstal and J. A. Van Gisbergen, “Skewness of saccadic velocity profiles: A unifying parameter for normal and slow saccades,” *Vision Research*, vol. 27, no. 5, pp. 731–745, 1987.

[18] A. J. Van Opstal, “200 years Franciscus Cornelis Donders,” *Strabismus*, vol. 26, no. 4, pp. 159–162, 2018. [Online]. Available: https://doi.org/10.1080/09273972.2018.1551770

[19] D. Tweed and T. Vilis, “Geometric relations of eye position and velocity vectors during saccades,” *Vision Research*, vol. 30, no. 1, pp. 111–127, 1 1990. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/0042698990901314

[20] R. Shadmehr and S. Mussa-Ivaldi, “Optimal Feedback Control,” in *Biological Learning and Control*. The MIT Press, 1 2012, ch. Optimal Fe, pp. 335–366.

[21] J. A. van Gisbergen, A. J. van Opstal, and J. J. Schoenmakers, “Experimental test of two models for the generation of oblique saccades,” *Experimental Brain Research*, vol. 57, no. 2, pp. 321–336, 1985.

[22] W. Becker and R. Jürgens, “Human oblique saccades: Quantitative analysis of the relation between horizontal and vertical components,” *Vision Research*, vol. 30, no. 6, pp. 893–920, 1990.

[23] T. Haslwanter, “Mathematics of three-dimensional eye rotations,” *Vision Research*, 1995.

[24] F. S. Grassia, “Practical Parameterization of Rotations Using the Exponential Map 1 Introduction 2 Evaluation of Common Parameterizations,” *The Journal of Graphics Tools*, vol. 3, pp. 1–13, 1998.

[25] D. L. Sparks and L. E. Mays, “Spatial localization of saccade targets. I. Compensation for stimulation-induced perturbations in eye position,” *Journal of Neurophysiology*, vol. 49, no. 1, pp. 45–63, 1983.

[26] E. L. Keller and D. A. Robinson, “Absence of a Stretch Reflex in Extraocular Muscles of the Monkey,” *Journal of neurophysiology*, vol. 34(5), pp. 908–919, 1971.

[27] D. Guitton and M. Volle, “Gaze control in humans: eye-head coordination during orienting movements to targets within and beyond the oculomotor range,” *Journal of Neurophysiology*, vol. 58, no. 3, pp. 427–459, 9 1987. [Online]. Available: https://www.physiology.org/doi/10.1152/jn.1987.58.3.427

[28] G. F. Franklin, J. D. Powell, and M. Workman, *Digital Control of Dynamic Systems*, 1 1985, vol. 6, no. 1. [Online]. Available: http://doi.wiley.com/10.1002/oca.4660060111

[29] M. Athans, “The Role and Use of the Stochastic Linear—Quadratic—Gaussian Problem in Control System Design,” *IEEE Transactions on Automatic Control*, vol. 16, no. 6, pp. 529–552, 1971.

[30] E. Todorov, “Stochastic optimal control and estimation methods adapted to the noise characteristics of the sensorimotor system,” *Neural Computation*, vol. 17, no. 5, pp. 1084–1108, 2005. [Online]. Available: www.cogsci.ucsd.edu/

[31] A. A. Kardamakis and A. K. Moschovakis, “Optimal control of gaze shifts,” *Journal of Neuroscience*, vol. 29, no. 24, pp. 7723–7730, 2009.

[32] E. Todorov and M. I. Jordan, “Optimal feedback control as a theory of motor coordination.” *Nature neuroscience*, vol. 5, no. 11, pp. 1226–35, 2002. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/12404008

[33] J. Myerson and L. Green, “Discounting of delayed rewards: Models of individual choice,” *Journal of the Experimental Analysis of Behavior*, vol. 64, no. 3, p. 263, 1995. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1350137/

[34] S. Kobayashi and W. Schultz, “Influence of reward delays on responses of dopamine neurons,” *Journal of Neuroscience*, vol. 28, no. 31, pp. 7837–7846, 7 2008.

[35] R. Bellman, “Dynamic programming and stochastic control processes,” *Information and Control*, vol. 1, no. 3, pp. 228–239, 9 1958. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0019995858800030

[36] M. Tur and J. Goodman, “When is speckle noise multiplicative? Speckle Phenomena in Optics-Second Edition View project Structural Health Monitoring of Aircraft / Civil Structures View project,” 1982. [Online]. Available: https://www.researchgate.net/publication/43134848

[37] K. J. Ciuffreda and L. Stark, “Descartes’ law of reciprocal innervation,” *Optometry and Vision Science*, vol. 52, no. 10, pp. 663–673, 10 1975. [Online]. Available: https://insights.ovid.com/crossref?an=00006324-197510000-00003

[38] J. Diedrichsen, R. Shadmehr, and R. B. Ivry, “The coordination of movement: optimal feedback control and beyond,” *Trends in Cognitive Sciences*, vol. 14, no. 1, pp. 31–39, 1 2010. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1364661309002587

[39] C. M. Harris and D. M. Wolpert, “The main sequence of saccades optimizes speed-accuracy trade-off,” *Biological Cybernetics*, vol. 95, no. 1, pp. 21–29, 7 2006.

# Appendix A

## A.1 Linearized system

### A.1.1 Continuous

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -521.1 & 0 & -68.2 & -42.0 & 0 & 0 \\ 0 & -560.7 & 0 & 0 & -42.0 & 0 \\ -184.6 & 0 & -1049.5 & 0 & 0 & -42.0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 2101.3 & 0 & 0 \\ 0 & 2101.3 & 0 \\ 0 & 0 & 2101.3 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

### A.1.2 Discrete

$$A = \begin{bmatrix} .9997 & 0 & -3.3611e-5 & 9.7919e-4 & 0 & -1.1126e-8 \\ 0 & 0.9997 & 0 & 0 & 9.7919e-4 & 0 \\ -9.1032e-5 & 0 & 0.9995 & -3.0134e-8 & 0 & 9.7911e-4 \\ -0.5102 & 0 & -0.0667 & 0.9586 & 0 & -3.3143e-5 \\ 0 & -0.5490 & 0 & 0 & 0.9586 & 0 \\ -0.1808 & 0 & -1.0276 & -8.9766e-5 & 0 & 0.9583 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.001 & 0 & -5.8694e-9 \\ 0 & 0.001 & 0 \\ -1.5897e-8 & 0 & 0.001 \\ 2.0576 & 0 & -2.3379e-5 \\ 0 & 2.0575 & 0 \\ -6.3319e-5 & 0 & 2.0574 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$