Biologically Motivated Controllers for Robotic Eyes

by

Martin Lesmana

B.A.Sc, The University of Waterloo, 2009

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE STUDIES
(Computer Science)

The University Of British Columbia
(Vancouver)

August 2011

Abstract

We present the development of computational models of biological motor control used in two different types of eye movements — gaze shifting and gaze stabilization. They are then implemented and tested on robotic systems. The thesis also investigates the application of these biological motor control strategies in robotics applications.

We describe and test a non-linear control algorithm inspired by the behaviour of motor neurons in humans during extremely fast saccadic eye movements involved in gaze shifting. The algorithm is implemented on a robotic eye connected with a stiff camera cable, similar to the optic nerve. This adds a complicated non-linear stiffness to the plant. For high speed movement, our "pulse-step" controller operates in open-loop using an internal model of the eye plant learned from past measurements. We show that the controller approaches the performance seen in the human eye, producing fast movements with little overshoot. Interestingly, the controller reproduces the main sequence relationship observed in animal eye movements.

We also model the control of eye movements that serve to stabilize its gaze direction. To test and evaluate this stabilization algorithm, we use a camera mounted on a robotic test platform that can have unknown perturbations in the horizontal plane. We show that using models of the vestibulo-ocular and optokinetic reflexes to control the camera allows the camera to be effectively stabilized using an inertial sensor and a single additional motor, without the need for a joint position sensor. The algorithm uses an adaptive controller based on a model of the vertebrate Cerebellum for velocity stabilization, with additional drift correction. A resolution-adaptive retinal slip algorithm that is robust to motion blur was also developed. We show that the resulting system can reduce camera image motion to about one pixel per frame on average even when the platform is rotated at 200 degrees per second. As a practical robotic application, we also demonstrate how the common task of face detection benefits from active gaze stabilization.

Preface

Portions of this thesis are based on work previously published or submitted for publications by the author.

Chapter 3 on fast movements is largely based on a previously submitted work (M. Lesmana and D. K. Pai. A biologically inspired controller for fast eye movements. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, Shanghai, China, May 2011. IEEE Robotics and Automation Society, IEEE) [59]. Aside from the abstract and introduction section of the manuscript, the rest of the manuscript was written by the present author. The robotic setup was constructed by Dinesh K. Pai, Per-Erik Forssén and Bruce Dow. The rest of the work presented in the manuscript was conducted by the present author under guidance of Dinesh K. Pai.

The gaze stabilization work presented in Chapter 4 is based on collaborative work with Axel Landgren, Per-Erik Forssén and Dinesh K. Pai that has been submitted for publication (M. Lesmana, A. Landgren, P.-E. Forssén, and D. K. Pai. Active gaze stabilization for large and fast movements. Submitted (2011)) [60]. Dinesh K. Pai initiated and guided the development of the project. The robotic setup was constructed by the present author, Axel Landgren and Bruce Dow. The computer vision code was developed by Axel Landgren under the guidance of Per-Erik Forssén. The stabilization algorithm was initially developed by Axel Landgren and later extended to incorporate drift-free tracking by the present author. All the experimental results aside from that of Figure 4.6 was obtained by the present author. Figure 4.6 was created using data provided by Axel Landgren. The section on Retinal Slip Estimation was written by Per-Erik Forssén based from the thesis by Landgren (A. Landgren. A robotic camera platform for evaluation of biomimetic gaze stabilization using adaptive cerebellar feedback. Master's thesis, Linköping University, 2010) [53]. The rest of the manuscript was written largely by the present author with edits by the co-authors of the manuscript.

Table of Contents

Appendix

List of Tables

List of Figures

Glossary

DC	Direct Current
EOM	Extraocular Muscle
FIR	Finite Impulse Response
ICRLS	Inequality Constrained Recursive Least Squares
IMU	Inertial Measurement Unit
OKR	Optokinetic Response
PD	Proportional-Derivative
PID	Proportional-Integral-Derivative
QP	Quadratic Program
RBF	Radial Basis Function
RLS	Recursive Least Squares
RMS	Root Mean Square
SLAM	Simultaneous Localization and Mapping
TPS	Thin Plate Spline
VOR	Vestibulo-Ocular Reflex

Acknowledgments

I would like to thank my supervisor, Dinesh Pai, for the invaluable support and guidance he has provided during my time in the Sensorimotor Systems Lab. I also would like to acknowledge David Lowe for helping make this thesis possible.

The members of the Sensorimotor Systems Lab have been a great group of friends that helped to make research in UBC exciting and enjoyable. Many thanks to the Post-docs and Ph.D.'s, Timothy Edmunds, Atsutoshi Ikeda, Shinjiro Sueda, Sanghoon Yeo, David Levin and Bin Wang, for their wisdom and guidance all these years. I also want to thank my fellow Master's students Mahkameh Lakzadeh, Garrett Jones, Josh Litven, Ye Fan and Duo Li for their support and encouraging words.

I am also grateful to my collaborators from Linköping University, Axel Landgren and Per-Erik Forssén, for their insights and help during our collaboration.

Lastly, I am truly grateful to my parents and family for their immeasurable support and faith in me.

Chapter 1

Introduction

Human vision provides us with rich sensory information of the external world. Eye movements are inextricably linked to vision — they are a critical mechanism necessary to achieving the goal of the vision system — a clear and stable view of the world.

The repertoire of eye movements comprises some of the most interesting movements made by humans. This includes very fast, yet accurate, movements and also movements specifically produced to cancel out head movements, hence stabilizing our view of the world even as we move. We often take these movements for granted as we perform them without giving them much thought. The difficulty in generating these movements only become apparent when we try to reproduce them — hinting at the sophistication of the biological motor control system involved in producing the various types of eye movements.

The motivation for this thesis comes from two intertwined questions: (1) Can biological theories of motor control proposed in neuroscience be effective for controlling real robots? (2) Can we gain insights into biological motor control by creating computational models and building physical robots [1, 50]?

There have been many studies on the control of eye movements. However, due to the enormous difficulty in directly observing the neural mechanism involved, indirect approaches are required to understand them. By computationally modeling the proposed controller, we can test its feasibility through simulation and implementation on a robotic system. This serves as a powerful tool to better understand the controller and complements experimental studies: experimental findings can be incorporated into the computational model to improve its accuracy. Furthermore, implementing these controllers on physical robots enable us to provide the constructive evidence for its feasibility. A robotic implementation exposes the controller to the various real world constraints and factors that may have been unmodeled in a simulation setting.

Improving our understanding of how eye movements are generated helps us to comprehend the way our brain works and provides valuable clinical insights into various neurological diseases [56]. It also helps unravel some of the motor control tricks used in biology, allowing us to apply them in robotics.

1.1 Contributions

This thesis presents the computational modelling and robotic realization of two types of eye movements: the control of fast and accurate eye movements and the control of gaze stabilization movements.

With the goal of achieving fast and accurate movements in our robotic systems, we look at a class of eye movements which does exactly this — saccades. We created a controller based on the “pulse-step” paradigm proposed in neuroscience for the control of saccadic eye movements. This controller is non-linear and allows for fast movements with no overshoot as it does not follow a typical asymptotic approach to the final position. A classical linear controller such as the Proportional-Integral-Derivative (PID) controller would require very high gains to achieve these fast movements. The plant model used in designing a PID controller is also often assumed to be linear or a linear approximation of it. However, for a system such as the one used in our experiments (see section 3.5.1) a linear assumption is unsuitable due to the strong non-linearity introduced by the cable. The controller described in this paper automatically learns a general non-linear plant model by itself and skips the tuning process to create a control strategy that is best suited for the plant.

Additionally, we have developed a method to guarantee physical plausibility of the learned plant model through the use of appropriate basis functions and constraints. We have also created a recursive algorithm for solving the Quadratic Program (QP) that can accept a growing set of measurements and handle constraints (including inequality constraints), enabling the controller to learn the plant model in an online fashion. The controller implementation highlights the difficulty, and importance, of proper handling of friction in the system. This is true even on a relatively low friction setup such as ours.

We developed a computational model for the Vestibulo-Ocular Reflex (VOR) and Optokinetic Response (OKR), which are involved in gaze stabilization. The model uses a recurrent cerebellar architecture to perform feedback error learning [26]. Using this control architecture, we created a complete system for active stabilization of cameras which can reject large and fast movements and maintain drift-free object tracking. This is achieved by using both inertial measurement and vision information as inputs to the controller. The controller is adaptive and does not require a system identification step or extensive parameter tuning. It only requires a rudimentary model of the plant and we demonstrate its ability to operate even with a factor of two error in the plant model's Direct Current (DC) gain. Even under this condition, the system demonstrates rapid adaptation to a good performance. This robustness is made possible in part by the robustness of the vision algorithm used to estimate the retinal slip. We get this behaviour by applying crosschecking and resolution adaptivity to a standard feature tracking algorithm. We also describe a robust method for converting motion in the image plane into an angular velocity. The system's stabilization performance was then carefully evaluated using a high speed robotic platform that can generate general rigid motions in the horizontal plane. Finally, we quantify the effect of disturbance and the use of our gaze stabilization system on a common computer vision algorithm. We discovered that with a VOR only implementation, errors quickly accumulate and cause a drift in position. This alludes to the fact that the VOR needs to work in conjuction with other types of reflexes to achieve its functional goal.

1.2 Outline

The thesis is organized in the following manner. In **Chapter 2** we give a brief description of the anatomy and physiology of the two main biological systems involved with the two types of eye movements that is the focus of the thesis: the oculomotor system and the vestibular system. We also give an overview of the functional requirements and the different classes of eye movements.

Chapter 3 describes the computational modeling and robotic implementation of the controller used for fast and accurate eye movements. We discuss the structure used to learn a general non-linear plant model and then describe an open loop controller that uses the pulse-step paradigm to achieve very fast and accurate movements. Additionally, a method to ensure physical plausiblity in the learned model is described and an adaptive online version of the controller is also presented.

Chapter 4 describes the computational modeling and robotic implementation of the controller used for gaze stabilization. We highlight the importance of gaze stabilization in robotics and computer vision application. We then proceed to describe the controller based on a model of the cerebellum for rejecting the disturbance and the test platform used for its evaluation. The effect of disturbance and stabilization on a computer vision application is also demonstrated.

We conclude the thesis and present some avenue for future work in **Chapter 5**.

Chapter 2

Background

In this chapter, we will provide a description of the functional requirements of eye movements and discuss the two types of eye movements relevant to this thesis: saccades and VOR. In order to understand the biological motor control used for saccades and VOR, it is important to understand the oculomotor plant being controlled and the vestibular system providing the sensory signals. We have therefore included a brief description of the anatomy and physiology of the eye and the vestibular system. Please refer to [56], [99] and [62] for more detailed discussions on these topics.

2.1 Eye Anatomy and Physiology

The globe (eyeball) is supported by tissues inside the orbital cavity (eye socket) of the skull. Three agonist-antagonist pairs of Extraocular Muscles (EOMs), insert into the globe and generate the forces to rotate the globe in accordance to the desired eye movements. They consist of two oblique muscles (the superior and inferior obliques) and four rectus muscles (the lateral, medial, superior and inferior recti). Figure 2.1 illustrates the placement and insertion locations of these muscles. The lateral rectus is innervated by the abducens nerve and the superior oblique by the trochlear nerve. Meanwhile, the remaining four EOMs are connected to the oculomotor nerve. The arrangement of the EOMs gives rise to a complex coupling between the different movements directions and the different muscle pairs.

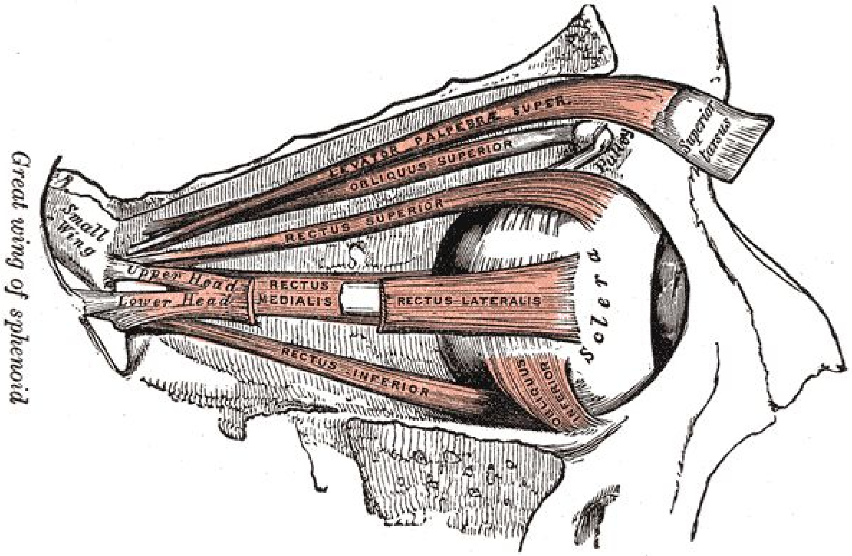


Figure 2.1: Anatomical diagram of the globe and the connected extraocular muscles. Reproduced from public domain material [37], obtained from [21].

All the tissues surrounding or connected to the globe impose a mechanical constraint on its movement. In particular, it restricts the eye movement to one that is approximately a pure rotation. Another important feature is that the globe movement is opposed by both viscous drag and elastic restoring forces [56]. There have been various studies to model and quantify the dynamics of the oculomotor plant [5, 79, 82, 86, 98]. Even though nonlinearities in the oculomotor dynamic have been experimentally observed [79, 82, 86, 89], they are often ignored in the context of motor control [16, 83, 84, 87].

The human eye's visual acuity varies across its field of vision according to the distribution of the photoreceptor on the retina. The photoreceptors density peaks at the fovea region of the retina, providing sharpest vision there. Visual acuity degrades quickly as we move away from the fovea, dropping by 50% at 2 degrees out from the center of fovea [56].

2.2 Vestibular System Anatomy and Physiology

The vestibular system lies in the inner ear and is comprised of the semicircular canals and the otolith organs. Figure 2.2 shows the inner ear structure containing the vestibular system, along with the cochlea — a part of the auditory system. The semicircular canals measure the head's rotational movement while the otolith organs measure its translational movement. Together, they provide the sensory inputs required to maintain balance and also VOR.

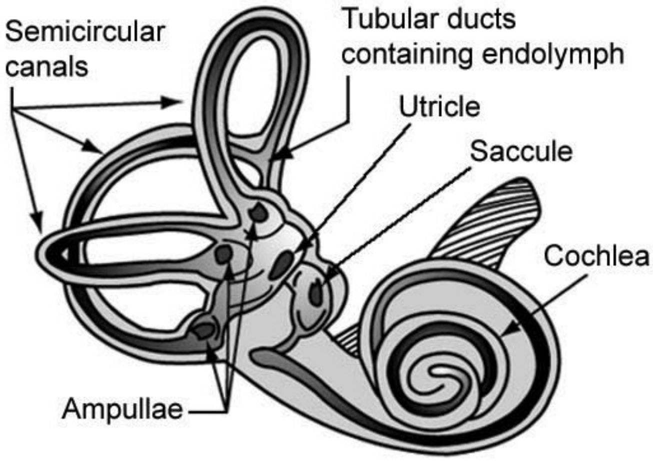


Figure 2.2: The inner ear containing the semicircular canals, otolith organs and cochlea. Image modified from public domain material obtained from [2].

The semicircular canals in the inner ear consist of three semicircular tubes oriented nearly orthogonal to one another [100]. Angular rotation causes the endolymph fluid inside the semicircular canals to flow. This in turn perturbs the sensory ephitelium that sits on the surface in the ampullae structure located at the end of the canals. The sensory ephitelium acts as a mechanical transducer, converting the movement into sensory signals [40].

The otolith organs, namely the utricular and saccular otoliths reside in the utricle and saccule of the inner ear and are sensitive to horizontal and vertical acceleration respectively. They consist of hair cells that carry otoliths, crystal structures suspended in the endolymph fluid. As the head translates, the inertia of the otoliths stimulates the hair cells that carry them, converting translational acceleration into sensory signals [8].

2.3 Classes of Eye Movements

A functional requirement of the vision system is to provide a clear and stable view of the world. This necessitates the image on the retina to be held steady and visual feature of interest to be centered at the fovea [56]. We can thus classify eye movements into the following two broad categories:

- Gaze shifting. For clear visual perception of an object of interest, its projection needs to fall on the foveal region of the retina. A saccade generates a quick shift in the eye's gaze direction, allowing us to change our visual attention from one target to another and hence scan the environment. And to keep both eyes foveated at the same visual target, the eye has to perform vergence movements as the target moves closer or further to the person.

- Gaze stabilization. To stabilize its gaze, the eye needs to maintain its gaze with respect to the world even in the presence of head movements. This enables us to perceive the environment even during locomotion. The eye has several mechanisms to address this need: the VOR, the OKR, and smooth pursuit tracking. While the VOR relies on the vestibular system, the OKR and smooth pursuit are visually-mediated. Detailed discussion of the VOR can be found below.

2.3.1 Saccades

The term saccade was first used by Javal to describe rapid eye movements that occur during reading [45]. A typical saccade duration is between 50 to 100 ms, reaching speeds of up to 500°/sec in human and up to 1000°/sec in some primates. This makes saccades one of the fastest movements in the human body. In fact, the brain cannot process the visual information during saccades due to the large amount of motion blur introduced by the high speed. Despite these rapid gaze shifts, we are able to perceive the environment in a unified and stable manner due to a combination of visual masking and visual persistence [24].

2.3.2 Vestibulo-ocular Reflex

The main goal of the VOR is to stabilize the line of sight with respect to the world in spite of head movements. It is therefore important that the VOR can respond to sudden and fast head perturbations. It achieves this through the use of the very fast vestibular system in the inner ear to measure the rotational and translational movement of the head. This signal is then used to drive the EOMs through a short pathway consisting of only three neural connections, often referred to as the three neuron arc. This results in a total delay of only 12 ms from the onset of head movement to compensating eye movement [69].

The VOR generates an eye movement in the opposite direction of the head movement. The resulting combination of both the eye and head movements should cancel each other and allow the gaze direction to remain unchanged. Although the VOR can compensate for both translational and rotational head movements, it is helpful to focus on the case of the horizontal rotational VOR. Figure 2.3 shows the main neural pathway for the horizontal rotational VOR. Note the absence of visual feedback in the pathway — the compensation relies solely on vestibular inputs.

2.3.3 Optokinetic Reflex

The OKR is a visually mediated reflex that is triggered by global visual motion [43]. It is also often referred to as the optokinetic nystagmus due to the sawtooth like pattern of eye movements generated. Although it can reach speeds up to 100°/sec, its dependence on visual processing gives it a latency of 70 ms. This means that it cannot compensate for higher frequency disturbance such as the one encountered during walking [56]. It does, however, perform well under sustained rotations, where the effect of latency is not significant.

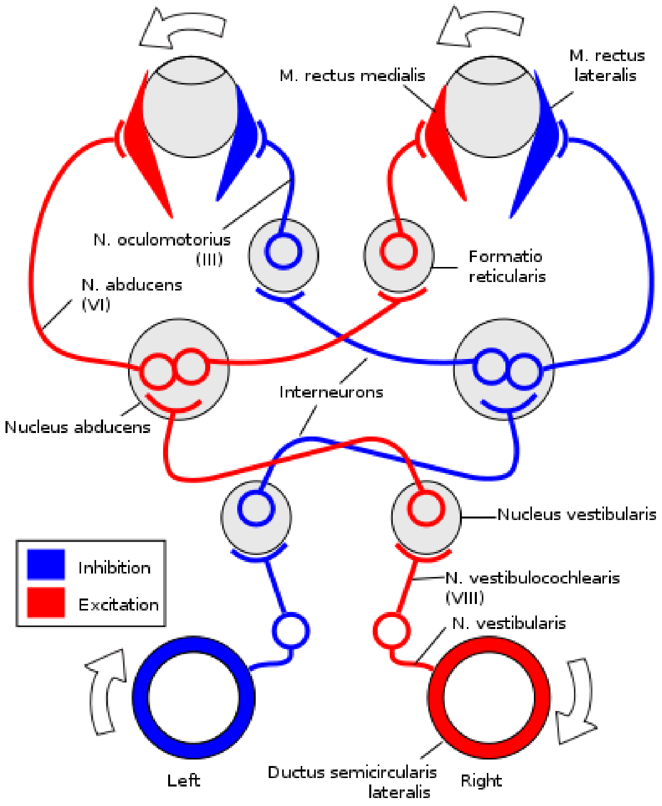


Figure 2.3: Horizontal VOR direct neuronal pathway. Reproduced from [22].

Chapter 3

A Controller for Fast and Accurate Eye Movements

3.1 Introduction

Animals do not scan their environment by continuous eye movement but by alternating fixation and saccades. Saccades are the fastest movements made by animal eyes; a saccade of less than 10 degrees can be completed in less than 50 ms [55]. One reason is that during movement, motion blur significantly degrades vision; therefore these movements must be completed as quickly as possible without losing too much positioning accuracy. There is a popular misconception that saccades are necessary only for animals with foveas and that this does not apply to robot cameras. However, even afoveate animals like rabbits need to make saccade-like movements since they have a limited field of view, and need to shift it when nearing the "joint limits" (this is called quick phase nystagmus) [19].

It is difficult to use feedback control for these high speed movements, in both animals and robots. Neural signal processing of sensory data is relatively slow in animals; there is significant evidence that saccades are performed in an open-loop fashion without visual or proprioceptive feedback [56, 105]. Signal processing in robots can be faster but traditional feedback control requires high gains and accurate estimation of derivatives to achieve the speeds seen in animal eyes, but such a controller can easily become unstable. Instead, there is strong experimental evidence that both humans [20] and animals [32] use a non-linear pulse-step controller (discussed in Section 3.3).

Outline

The remainder of the chapter is organized as follows. Section 3.3 gives the high level overview of the controller that is developed. Section 3.4 describes the technique used to estimate a non-linear model for the plant. Section 3.5 provides the details of the controller implementation and the setup used to test it. The results from the simulation and experiment are discussed in Section 3.6. We close the chapter by presenting our conclusions in Section 3.7.

3.2 Related Work

Motion control in industrial robots is often achieved through the use of a Proportional-Derivative (PD) feedback controller. This can be further augmented by an integral term to improve steady state performance and by a feedforward term to improve its dynamic performance [78]. The design of these different controller components requires some knowledge of the plant being controlled. Furthermore, an online learning system such as described in [80] can be used to improve the controller by a learning process that can capture the inaccuracy of the plant model used. Other common control techniques used for robotic manipulators include adaptive control [41, 101], sliding mode control [109], and computed torque control [4].

There has been much work on various robotic eye/gaze platforms by the active vision community. They include biologically inspired [9, 11–13, 57, 71], and also non-biologically inspired designs [36]. Specifically, the mapping of a target in the visual field to saccade target position has also been investigated [31, 106]. The actual generation of saccadic motion itself has been achieved by using *a priori* knowledge of the plant geometry and dynamic [33], tracking a spline trajectory using an inverse dynamic controller [97], or tracking a velocity profile [66]. An implementation of catch-up saccades during smooth pursuit (tracking) eye motion has been described in [29].

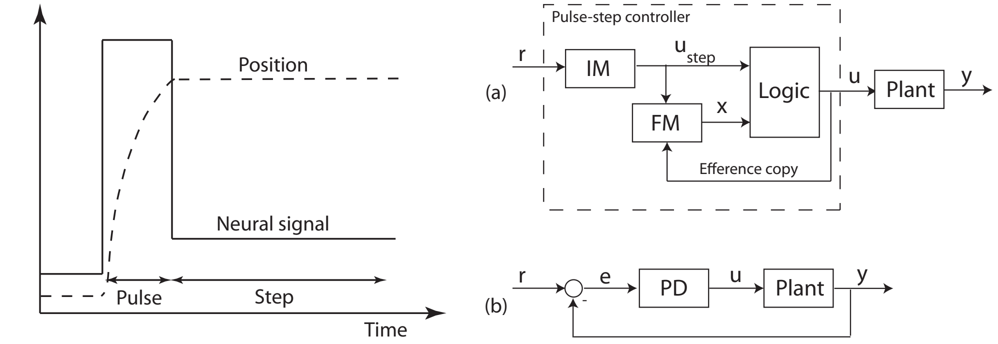
Although some of these works involve the generation of saccadic-type motion, we are not aware of any work on controlling rapid (saccadic) eye movement that exhibits the pulse-step behaviour seen in the biological counterpart for a robotic platform. It is worth noting that torque pulse, similar to the pulse-step, has been used to generate fast motion for robotic manipulator [74, 75] by learning the mapping from visual input to pulse parameters.

In the neurology of eye movements literature, there are many different motor control models for the generation of saccades. We refer the readers to an excellent review by Girard and Berthoz [34] for a thorough treatment on this topic. However, we will mention the groundbreaking model introduced by Robinson [88] introduced the use of an internal model in the generation of saccades. Owing to the duration of saccades and visual processing delays, a feedback model cannot be used to control saccades. Instead, an efference copy of the motor control signal can be used as a substitute. This idea has been further extended in a number of subsequent works [47, 94].

3.3 A Non-linear Controller For Rapid Movements

Our controller is based on the "pulse-step" controller proposed in neuroscience for control of saccadic eye movements. The control signal sent to oculomotor neurons to generate a saccade is composed of two main components as shown in Figure 3.1a. It starts with a very high constant signal called the pulse followed by a lower constant signal referred to as the step. While the step signal relates to the steady state holding force to keep the eye at the target position, the magnitude of the pulse component is roughly the same regardless of the target location or saccade amplitude [56]. Therefore, the amplitude of the saccade is modulated primarily by its duration. This model is a simplification since there is usually a "slide" region between the pulse and the step, and other variations, but it is a good first approximation.

From a control point of view, this is a switching controller which switches from a fast movement phase to a fixation phase based on the distance to target. The controller action can be physically interpreted as trying to activate the muscles to move the eye as fast as possible to the target position before applying the appropriate force level to hold it at its target. Although this controller is similar to a bang-bang controller, unlike a bang-bang controller, its output does not only switch between



(a) The neural (control) signal in the extra-ocular-muscle (EOM) during a saccade showing the characteristic pulse-step innervation. The pulse component of the signal allows for a rapid movement of the eye. Once the eye reaches the new position, the step component counteracts the forces that try to return it to its central position.

(b) A block diagram representation of the pulse-step controller that we develop (a) and a classical PD controller (b). The standard control system notation is followed for the signals: r, u, y, and e are target, control, output, and error signals respectively. Also u_{step} and x are the step level signal and state variable respectively. PD, IM and FM stands for PD controller, Inverse Model, and Forward Model.

Figure 3.1: The pulse-step signal and implemented and classical controller architecture

the boundaries of the feasible control signal $u(t) = M_+$ or $u(t) = M_-$ (where M_+ and M_- are the bounds of the feasible control signal) [51]. Instead, the pulse-step controller uses the entire range of feasible control signal.

The pulse-step controller uses an internal forward model simulation to determine the proper instance to transition from the pulse to the step. The forward model is a learned (estimated) non-linear model of the system plant. We discuss the learning process in Section 3.4. We simulate the actual plant trajectory by exciting the forward model with a copy of the input to the plant and integrate it forward in time. In neuroscience, the copy of the signal sent to the muscle is referred to as the *efference* copy.

A system level overview of the pulse-step controller as described is compared to the classical PD controller in Figure 3.1b. We can see that while the PD controller feeds the output signal back into the controller (hence the feedback), we also see a similar feedback line in the pulse-step controller. In the pulse-step controller however, the signal that is fed back is the control signal rather than the plant output

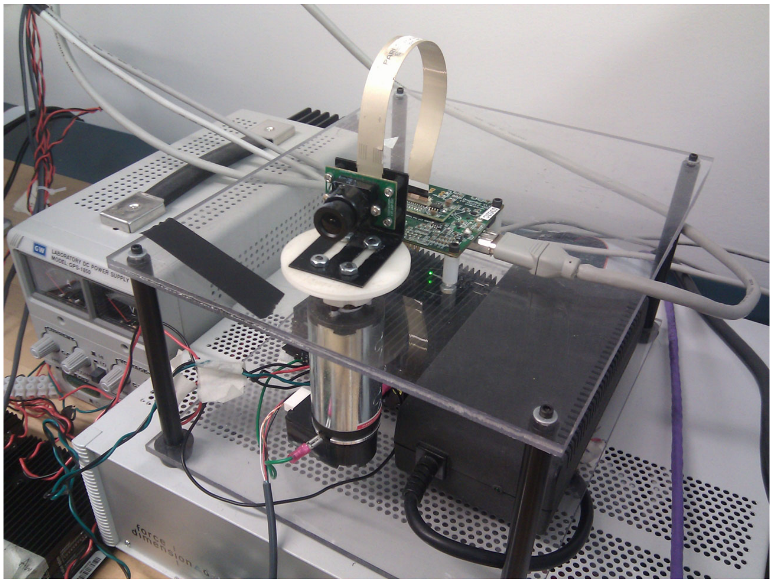


Figure 3.2: The experimental setup used to test the non-linear controller. A DC motor directly drives a camera platform about a vertical axis. The camera platform is connected by a flat-flexible cable to a processing board fixed to the ground.

(sensor measurement). The forward model then transforms the control signal into an estimate of the system state which also contains the output. This is referred to as an internal feedback [88] and it allows the controller to have knowledge of the output without having to actually measure it.

Once the internal model reaches the target position, the controller generates a step signal that is required to hold the system at that position. The appropriate magnitude of the step signal can be determined directly from the inverse model of the learned plant by evaluating the force at the target location in steady state condition.

3.4 Non-linear Plant Model Learning

In this section we describe the method used to learn a non-linear plant model.

We first assume a very general second order plant:

$$u = J\ddot{\theta} + F(\theta, \dot{\theta}). \tag{3.1}$$

This system contains a linear expression for the highest order (inertial) term J and a non-linear force term $F(\theta, \dot{\theta})$; u is the applied torque and θ, $\dot{\theta}$, and $\ddot{\theta}$ are the position, velocity, and acceleration respectively.

The last term in Equation 3.1 is a non-linear function of the state space $(\theta, \dot{\theta})$. We parameterize it by a linear combination of n Radial Basis Functions (RBFs) ($B_k(\theta, \dot{\theta})$) placed regularly in the state space. An RBF is defined as a real function that is dependent only on the distance of its input from its center [14]. Our choice of RBF was a hat function (see Appendix A) with compact support due to its low evaluation cost and sparsity of representation. These two features help to reduce the computational effort of both the learning process and open loop control.

Thus the model of the plant can be reexpressed as:

$$u \approx J\ddot{\theta} + \sum_{k=1}^{p} w_k B_k(\theta, \dot{\theta}),$$

where w_k is the weight for the k'th RBF. Hence, to learn this parameterized system model, we simply need to learn the values of J and w_k's.

Even though the system itself is non-linear, it is linear in parameters. Therefore we can perform a linear least-squares estimation on the parameters. We use a collection of recorded input signal and trajectory as samples (denoted by superscript) for the least squares regression. A typical saccade trajectory is shown in Figure 3.3. The resulting regression is as follow:

$$\underbrace{\begin{bmatrix} u^1 \\ \vdots \\ u^n \end{bmatrix}}_{\mathbf{u}} = \underbrace{\begin{bmatrix} \ddot{\theta}^1 & B_1(\theta^1, \dot{\theta}^1) & \cdots & B_p(\theta^1, \dot{\theta}^1) \\ \vdots & \vdots & \ddots & \vdots \\ \ddot{\theta}^n & B_1(\theta^n, \dot{\theta}^n) & \cdots & B_p(\theta^n, \dot{\theta}^n) \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} J \\ w_1 \\ \vdots \\ w_p \end{bmatrix}}_{\mathbf{x}}. \tag{3.2}$$

Here, A is the $m \times n$ regressor matrix, while $\mathbf{x}$ and $\mathbf{u}$ are the n-vectors of unknown

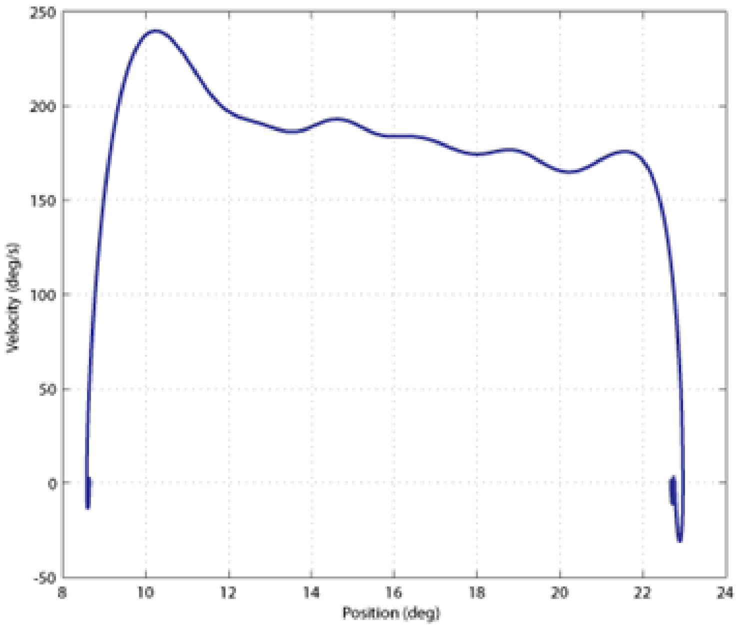


Figure 3.3: An example trajectory of a single saccade in the position-velocity state space

parameters and the measured variables respectively.

3.4.1 Physical Plausibility

The non-linear model learning method described above has the advantage of being able to represent very general systems. The structure used, however, causes it to be prone to overfitting, causing the learned model to be oscillatory. As we will discuss in section 3.5.2, this is detrimental to the controller performance and may lead to instability in the integration. An example of the learning result using compact RBFs is shown in Figure 3.4. The 'bumpiness' caused by the overfitting is clearly shown here. Another feature of the learned model seen in the plot: the magnitude of the force decreases near the edges of the plot. This is inherently caused by the shape of any choice of RBF.

Since we are dealing with non-uniformly sampled data, we can improve the interpolation and extrapolation properties of the model by using RBFs with non-

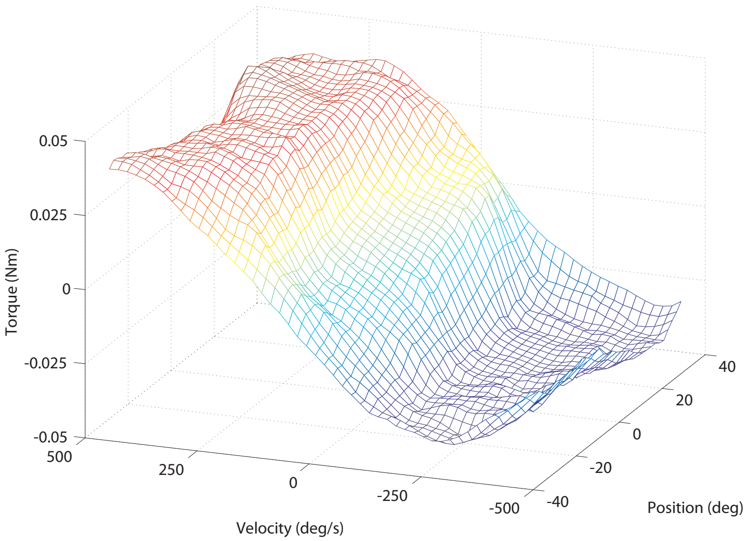


Figure 3.4: The $F(\theta, \dot{\theta})$ term learned by our algorithm for a linear mass-spring-damper system using compact RBF. The actual plant would generate a plane for this plot.

compact support [73] which improves the smoothness of the fitted function. Although there is a larger computational cost from using RBF with non-compact support, the difference is not signficant due to the relatively small size of the problem. The Thin Plate Spline (TPS) is an example of such a basis function and is the smoothest basis function, energywise [102].

Even though we can improve the smoothness and extrapolation behaviour of the learned model by using non-compact RBF, we can further imbue the learned model with additional desirable properties. It is useful to constrain the system stiffness and damping such that they remain positive in the entire state space. This follows the typical stiffness and damping behaviour of most mechanical systems including, in particular, the orbit of the eye [86] [1]. We can do this by constraining the system forces ($F(\theta, \dot{\theta})$) to be monotonically increasing in each coordinate direction.

[1] It is important to note that systems do exist which exhibit negative stiffness and damping behaviour [30]. But this occurs very rarely.

To obtain monotonicity (increasing), we place a constraint on the gradient of the force to be always non-negative:

$$\nabla F \geq 0\,. \tag{3.3}$$

We decompose F into the spring and damping forces such that the monotonicity constraint can be applied separately:

$$F(\theta, \dot{\theta}) = F_{\text{spring}}(\theta) + F_{\text{damping}}(\dot{\theta})\,. \tag{3.4}$$

Since the derivation and result for the spring and damping forces are similar, the subsequent description will only mention the spring force.

We can approximate the gradient of the spring force using a set of non-negative RBFs ($\tilde{B}_k$) and apply non-negativity constraint on the weights:

$$\nabla F_{\text{spring}} = \sum_{k}^{p} \tilde{w}_k \tilde{B}_k, \quad \text{where } \tilde{w}_k \geq 0 \ \forall k\,. \tag{3.5}$$

We then integrate them to obtain a set of monotonically increasing (non-radial) basis functions (B_k) to approximate the force. These basis functions (with non-negative weights) can be used to describe a monotonically increasing force over the entire state space:

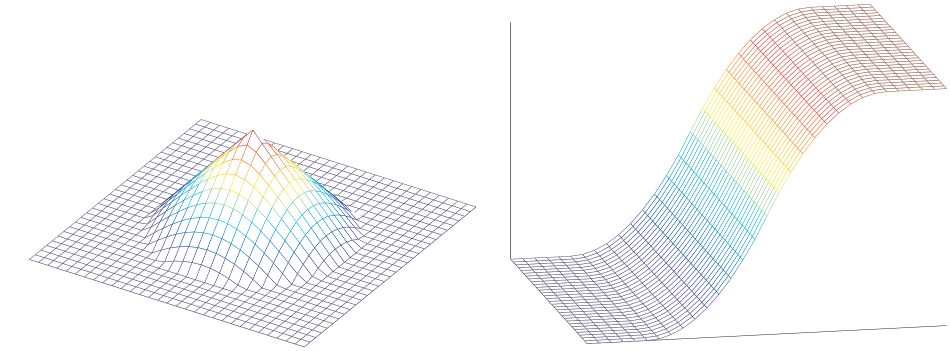
$$F_{\text{spring}} = \sum_{k}^{p} w_k B_k, \quad \text{where } \tilde{w}_k \geq 0 \ \forall k\,. \tag{3.6}$$

The shape of the original RBF and the resulting basis functions is shown in Figure 3.5b. The full derivation of the basis functions used in our implementation can be found in Appendix A.

The resulting form used for approximating the non-linear force is as follow:

$$u \approx J\ddot{\theta} + \sum_{k=1}^{p} w_k B_k(\theta) + \sum_{j=1}^{q} v_j D_j(\dot{\theta}) - c\,,$$

where B_k and D_j are the monotonically increasing basis functions for the stiffness and damping terms respectively. w_k and v_j are the non-negative basis functions



(a) Compactly supported radial basis function

(b) Basis function guaranteeing physically plausibility

Figure 3.5: Basis functions used in model learning. (a) An RBF with compact support (hat function), used in 3.4. (b) A basis function derived (from the hat function) which enforces physical plausibility behaviour in the learned plant model.

weights, and c is a constant offset term.

To enforce non-negativity on the weights, a non-negative least squares solver [54] is used. An example of the model learning result is shown in Figure 3.6. Comparing this result with the one obtained using compact RBFs (Figure 3.4), we see a vast improvement in the smoothness of the fit, the similarity to actual plant and the good extrapolation behaviour.

An alternative approach to obtain positive stiffness and damping characteristic is to constraint the weights of the RBFs themselves to be monotonically increasing. However this approach does not extend the desired characteristic beyond the space covered by the set of RBFs since the RBF function rolls off to zero or negative infinity away from the center. Hence, the extrapolation behaviour might become problematic for the controller described in Section 3.5.2. An example where this behaviour negatively affects the controller is shown in Figure 3.7.

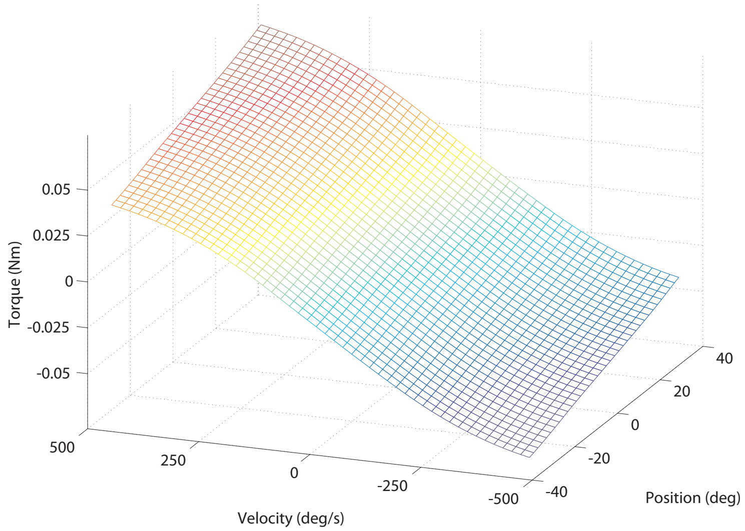


Figure 3.6: The $F(\theta, \dot{\theta})$ term learned by our algorithm for a linear mass-spring-damper system using physically plausible basis function. The actual plant would generate a plane for this plot.

3.4.2 Online Learning

Although the model learning process described above is capable of capturing very general plant dynamics, the learning happens only in an offline (batch) manner. For the control system to be truly adaptable, it needs to be capable of adapting to a continuously changing plant. This is especially critical for the pure open loop strategy we employ — differences between the learned and actual plant dynamics will adversely affect the tracking performance. It is therefore important that our learning algorithm can learn in an online or continuous manner.

To perform online learning, the learning algorithm needs to belong in the class of recursive algorithms. A recursive algorithm accepts a growing set of measurements while still occupying only a small and constant memory footprint. Moreover, we also require that the algorithm updates its result quickly and places more importance on the newer set of data so that it can track a changing system.

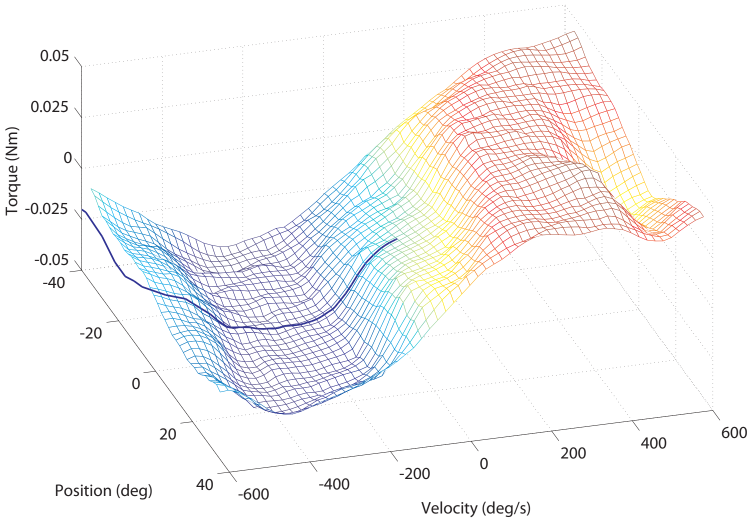


Figure 3.7: The effect of poor extrapolation behaviour of the learned plant model on the forward integration used by the controller. The surface represents the $F(\theta, \dot{\theta})$ and the line shows the resulting integration path through the state space. As the state of the integration steps past the "hill" of the RBFs, it quickly falls into the steep valley behind it.

Unconstrained Online Learning

An obvious choice of technique for doing online learning is to use the recursive version of the least squares technique previously used in the offline learning process. The original Recursive Least Squares (RLS) technique [63] can be reduced in complexity from $O(n^3)$ to $O(n^2)$, where n is the number of parameters in the model, by applying the Sherman-Morrisson-Woodbury formula for the rank-1 update of the matrix inverse. More recent development has led to fast RLS techniques which reduces the complexity further to $O(n)$ [17, 28, 64].

Constrained Online Learning

The above techniques however, are unable to solve constrained problems encountered in our formulation for physical plausibility. Unfortunately they are not easily extended to incorporate even the simple non-negative constraints that we need.

There have been several extensions proposed to the RLS technique to solve the constrained least squares problems. A modification to the RLS was presented by Resende et al. [85] to solve least squares problem with equality constraints. Zhu and Li [111] described an algorithm which solves the RLS problem with linear constraints, including inequality constraints. However, the algorithm described for solving Inequality Constrained Recursive Least Squares (ICRLS) requires running 2^d (where d is the number of constraints) separate RLS solvers in parallel and selecting the solution with the smallest error while still satisfying the constraints. This method thus becomes computationally expensive very quickly. In our case, d reflects the number of parameters to solve and this is typically between 10 to 20, rendering this particular method prohibitively expensive to use.

Due to the lack of suitable existing methods to solve the ICRLS problem, we reformulated it under the more general QP framework since constrained QP is a well studied problem. Our constrained least squares problem is as follow:

$$\underset{\mathbf{x}}{\text{minimize}} \quad ||A\mathbf{x} - \mathbf{b}|| \tag{3.7a}$$

$$\text{subject to} \quad \mathbf{x} \geq 0\,, \tag{3.7b}$$

where A is the $m \times n$ the regressor matrix, $\mathbf{b}$ the n elements force vector, and $\mathbf{x}$ the unknown parameters vector from Equation 3.2. Rewriting it as a QP, we get:

$$\underset{\mathbf{x}}{\text{minimize}} \quad \frac{1}{2}\mathbf{x}^T H\mathbf{x} + \mathbf{c}^T\mathbf{x} \tag{3.8a}$$

$$\text{subject to} \quad \mathbf{x} \geq 0\,, \tag{3.8b}$$

where

$$H = 2A^T A \tag{3.9a}$$

$$\mathbf{c}^T = -2\mathbf{b}^T A\,. \tag{3.9b}$$

Here, H is the $m \times m$ Hessian matrix and $\mathbf{c}$ is the m-vector linear term. This constrained QP problem can be solved using a number of different methods [77]. In our implementation we use the standard primal active set method to solve the individual QP problem.

We extend this standard QP formulation to handle a growing set of measurements by creating an update rule for the Hessian matrix and the linear term for every new measurement. Augmenting the regressor matrix and measurement vector with the new data, we get:

$$A_{t+1} = \left[\begin{array}{c} A_t \\ \hline \mathbf{a}_{t+1}^T \end{array} \right] \tag{3.10a}$$

$$\mathbf{b}_{t+1}^T = \left[\begin{array}{c|c} \mathbf{b}_t^T & b_{t+1} \end{array} \right] , \tag{3.10b}$$

where the subscript indicates the measurement index. Putting this into the QP formulation, we can then get the update rule for the QP terms:

$$\begin{aligned} H_{t+1} &= 2(A_{t+1}^T A_{t+1}) && (3.11a) \\ &= 2A_t^T A_t + 2\mathbf{a}_{t+1}\mathbf{a}_{t+1}^T \\ &= H_t + 2\mathbf{a}_{t+1}\mathbf{a}_{t+1}^T && (3.11b) \\ \mathbf{c}_{t+1}^T &= -2\mathbf{b}_{t+1}^T A_{t+1} && (3.11c) \\ &= -2(\mathbf{b}_t^T A_t + b_{t+1}\mathbf{a}_{t+1}^T) \\ &= \mathbf{c}_t^T - 2b_{t+1}\mathbf{a}_{t+1}^T \, . && (3.11d) \end{aligned}$$

Even though the initial solution search to the constrained QP problem would require several iterations to converge to the correct working set, we can minimize the number of iterations in the subsequent solves by leveraging similarity between one QP problem to the next. It is possible to warm start the primal active set solver at each new measurement by passing both the working set and solution obtained in the previous QP solve. This greatly speeds up the algorithm — as long as the working set remains unchanged from one solve to the next, the algorithm only need to perform a single linear system solve. It turns out that for the ICRLS problem this is almost always the case.

Additionally, we can apply an exponential forgetting factor, similar to the one used in RLS, to our ICRLS formulation. An exponential forgetting factor will assign less weight to older measurements. This enables the learning algorithm to better track the system changes over time. Appendix B describes the derivation of the exponential forgetting factor in the above algorithm.

3.4.3 Friction Learning

As described above, the model should be able to capture friction (Coulomb and viscous). Due to the discontinuity at zero velocity introduced by Coulomb friction, however, separating the model learning for positive and negative velocity improves its identification by making the discontinuity explicit in the structure.

3.5 Implementation

3.5.1 Experimental Setup

In this section we describe the test bed used to test the non-linear pulse-step controller as shown in Figure 3.2. The setup consists of a single-degree-of-freedom actuated camera platform, directly driven by a brushed DC motor (A-max 236668, Maxon Motor AG). The axis of the motor is set to be perpendicular to the ground and the attached encoder provides a resolution of 0.18°. A linear motor amplifier (LSC 30/2, Maxon Motor AG) provides the motor with a current corresponding to the command signal from the controller. The motor drives a Dragonfly 2 remote head camera (Point Grey) that is connected to a ground-fixed processing board by a flat-flexible-cable of substantial stiffness. This cable introduces a parasitic non-linear torque to the setup.

A networked computer running xPC target (MathWorks Inc.) is connected to the motor amplifier and encoder. It acts as a hard real-time controller with a sample time of 0.5 msec. It implements a virtual spring-damper system for the camera platform by means of a PD controller. The spring and damper coefficients are chosen so as to match the dominant time constant of the human eyeball (0.299 seconds) as obtained experimentally by Robinson [86]. Consequently, the overall plant that we are controlling is overdamped, similar to the behaviour exhibited by the mechanics of the human eye.

3.5.2 Pulse-Step Controller

The real-time controller also executes the pulse-step controller algorithm, independent of the virtual spring-damper. The target locations are specified to the pulse-step controller through an external command sent over the network. The overdamped characteristic of the plant allows the transition from pulse to step to be decided in a simple fashion: we can rely on the damping to bring the system to a stop almost instantaneously when the controller transitions from pulse to step (provided the step level chosen corresponds to the correct amount of holding torque at that position). For a more general case, a look-ahead integration can be performed in addition to the current state estimation at each time step. This simulates the system trajectory up to a short interval in the future for the scenario whereby the control signal is switched to the step level. The integration of the forward model is performed numerically using a Störmer-Verlet integrator which provides us with a better accuracy than an Euler integrator. Due to the very fast motion (about 0.2 degrees per millisecond), the size of the timestep for the forward model integration needs to be sufficiently small to minimize error in the steady state estimate. We use a timestep size of 2 milliseconds in our implementation.

An issue encountered with the controller was the larger than expected steady state error caused by the motor friction. For a given step input sent to the plant, the steady state position reached would have an error of approximately 2 degrees due to friction. Therefore the controller cannot rely on the step level to correct any error in the pulse component of the trajectory. Furthermore, as a result of the rapid trajectory motion, the fast higher order dynamics affects the accuracy of the estimated trajectory of the pulse, causing an RMS error of 1.5 degrees. However, a correlation between the saccade amplitude and the steady state error was observed and nearly all of the steady state errors are caused by undershoots as shown in Figure 3.8. This correlation is likely to be caused by the higher order dynamics having a damping effect. This unmodelled damping effect causes the system to lose more energy than the forward model simulation and this error accumulates with increasing pulse duration (saccade amplitude). We compensate for this extra damping by using the above correlation in the controller to modify the saccade targets by amounts corresponding to the amplitude.

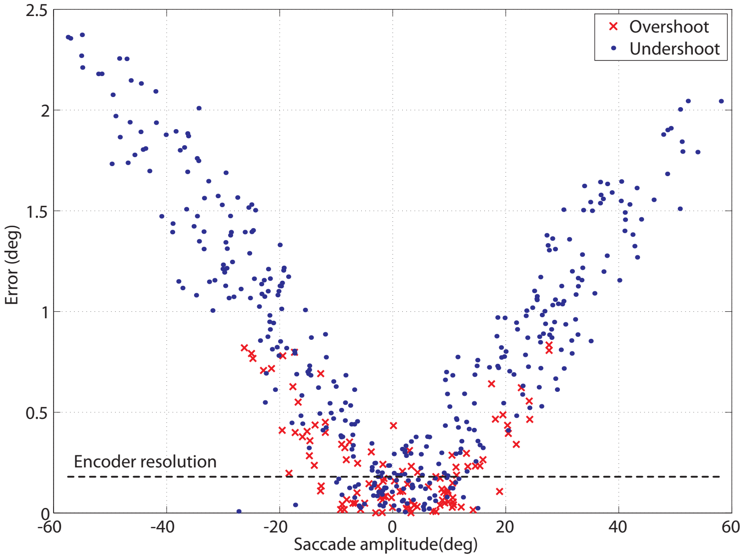


Figure 3.8: Correlation between the steady state error and the amplitude of the saccade obtained from our experimental result.

3.6 Results

3.6.1 Simulation Results

We first tested our plant model learning system in simulation. A synthetic pulse-step data set is generated with a system composed of a linear mass-spring-damper. Following the dominant time constant in Robinson's human eye mechanic model, we implement a linear spring and damper system on the experimental setup with a spring constant of 0.5 mNm/deg and a damping constant of 0.15 mNm/(deg/s). The inertia is matched to the experimental setup motor load inertia ($2.08 \times 10^{-7}\text{Nm} \cdot \text{s}^2/\text{deg}$). Figure 3.9 shows a series of pulse-step trajectories and force generated on this system. Applying the non-linear model learning gives us a very good estimate of the system parameters with a Root Mean Square (RMS) residual of 6.88×10^{-4} Nm. The estimated mass has a relative error of 10% and the learned $F(\theta, \dot{\theta})$ is approximately linear, as expected from the linear spring-damper.

The online learning algorithm is also tested in simulation and compared to the batch learning algorithm. While the Hessian and linear term is updated at every

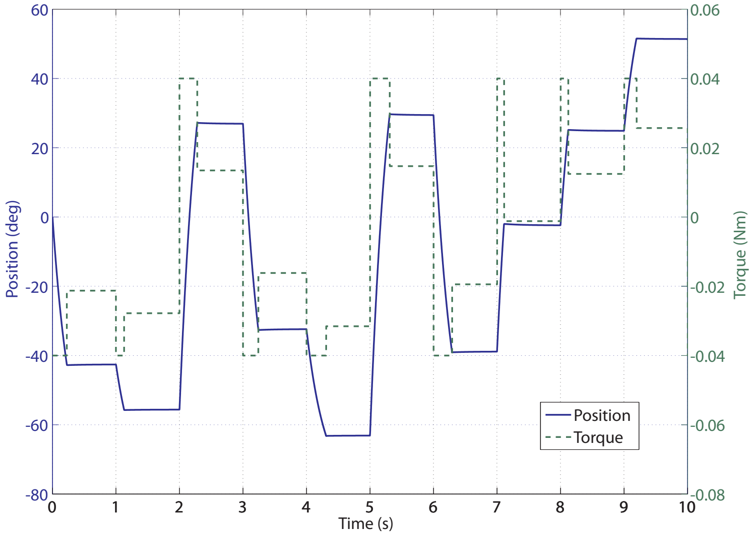


Figure 3.9: Simulated pulse-step trajectory and input signal used as a learning dataset for testing our learning system.

new measurement, the constrained QP solve is performed only once after every saccade. The online learning algorithm is typically first initialized with measurements from approximately 10 saccades so as to avoid poorly conditioned KKT matrix in the QP solve. Applying the two algorithms to the above plant, the resulting RMS residuals are plotted in Figure 3.10. As expected, the online learning residual decreases as more measurements are incorporated. Although the performance of the online learning algorithm approaches the batch version as more measurements are used, it remains slightly worse. This slightly lower accuracy can be attributed to the drop in numerical accuracy as we use the Hessian matrix instead of using a more numerically stable least squares method such as the QR decomposition. This is the trade-off that was made between compressing the data (forming the fixed-size Hessian from a growing regressor matrix) and the numerical conditioning. We also look at the online algorithm performance in handling a sudden change in plant dynamics. We first start by simulating a system as described before, we then suddenly perturb the system by permanently reducing the stiffness in the plant to 20% of the original value. Figure 3.11 shows the learning performance of the online algorithm. Pre-perturbation, the algorithm reaches some steady state performance

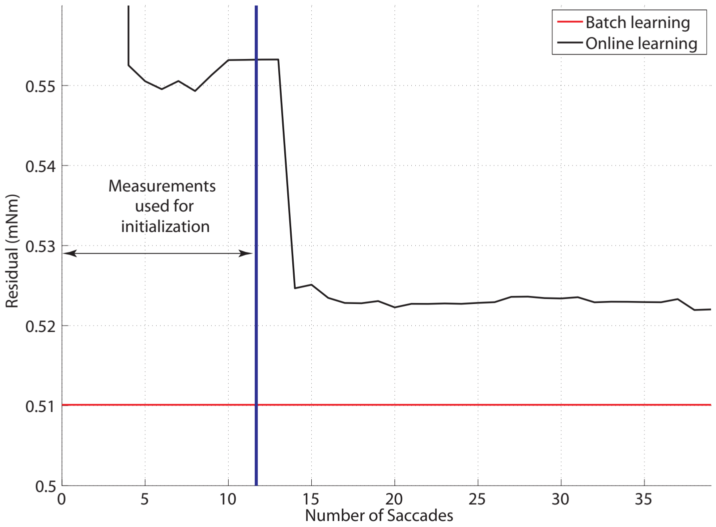


Figure 3.10: The RMS residuals of the models as learned by the batch and online learning algorithm. The residuals are calculated from all the measurements in the dataset. Also shown is the residuals obtained from the model learned online as it is still being initialized by the first 12 saccades measurements.

level early on. Immediately after perturbation, the learned plant model is incorrect and this is reflected by the large residual value. However, the learning algorithm incorporates the newer data into the model, putting more importance on them than the older data. This continuously improve the accuracy of the learned model as reflected by the decreasing residuals. We also compare the results with the batch learning performed on the entire dataset and part of the dataset before and after the plant perturbation. As expected, the performance of batch learning on the entire data set is significantly worse since it averages out the behaviour of two different plants. On the other hand, performing the batch learning only on data for the plant before or after perturbation yields much improved results.

We then cross validate the learned plant model with a new set of simulated pulse-steps trajectory. This gives us a good result, as shown in Figure 3.12, with a

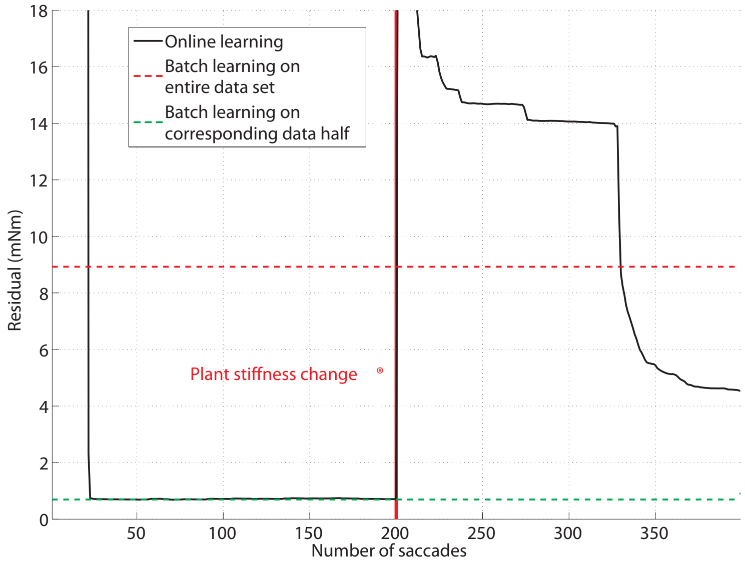


Figure 3.11: The RMS residuals of the models learned by the batch and online algorithms. The model residuals during the initialization are not shown. The residuals for the first 100 saccades are computed from the measurements before perturbation while the rest are computed from post-perturbation measurements.

fit of 99.8%. Fit is defined as:

$$\text{fit} = 100 \times \frac{1 - ||y - \hat{y}||}{||y - \bar{y}||},$$

where y and $\bar{y}$ are the cross-validation trajectory and its mean, while $\hat{y}$ is the trajectory generated from the learned model.

3.6.2 Experimental Results

The parameters of the virtual spring-damper in our experiment are matched to those used in the simulation. However, the cable introduces a significant amount of parasitic torque as discussed below.

We first learn a non-linear plant model from a set of pulse-steps trajectory. Even though the implemented virtual spring-damper system is linear, we are affected by

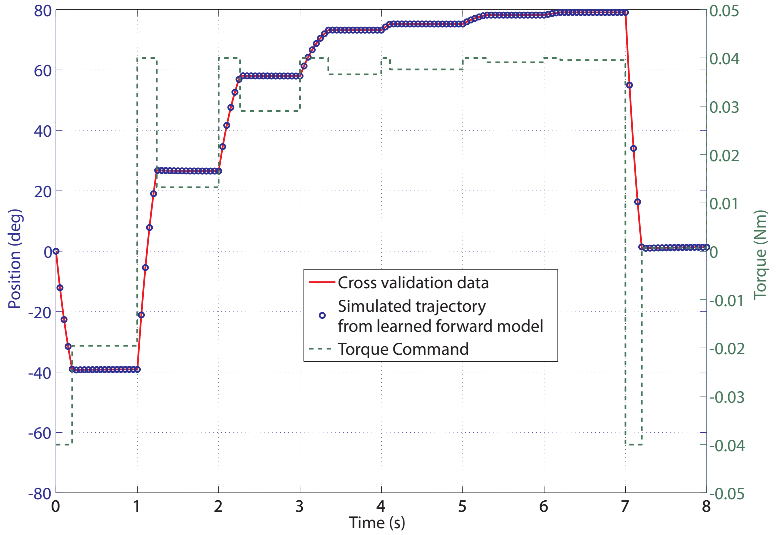


Figure 3.12: Result for cross-validation of the learned system model in our simulation. The cross validation dataset is generated from a system with the same parameters as the learning dataset. For visualization purposes, the plot only shows a downsampled version of the trajectory data generated from the learned forward model.

unmodelled system dynamics that can come from the amplifier, friction and the camera cable. These dynamics cause the overall system to behave in a non-linear fashion even though a DC motor with load may appear to be linear at first. The non-linearity of such a system is usually hidden away by the robustness of the feedback controller for slow dynamics. However our setup requires us to deal with fast system dynamics, nullifying the above feedback controller advantage.

With the implemented pulse-step controller from Section 3.5.2, we command the system to move to a series of targets. Figure 3.13 shows an example of the controller performance. The steady state RMS error (measured prior to the start of each new commanded target) is 0.99°. However the accuracy of the controller improves with the cable removed, giving us a steady state RMS error of 0.24°. We attribute this improvement to the uncertainty involved with the parasitic torque introduced by the cable.

Our experimental data exhibits a relationship between saccade amplitude and duration, that is similarly observed in saccadic eye motion of animals. This rela-

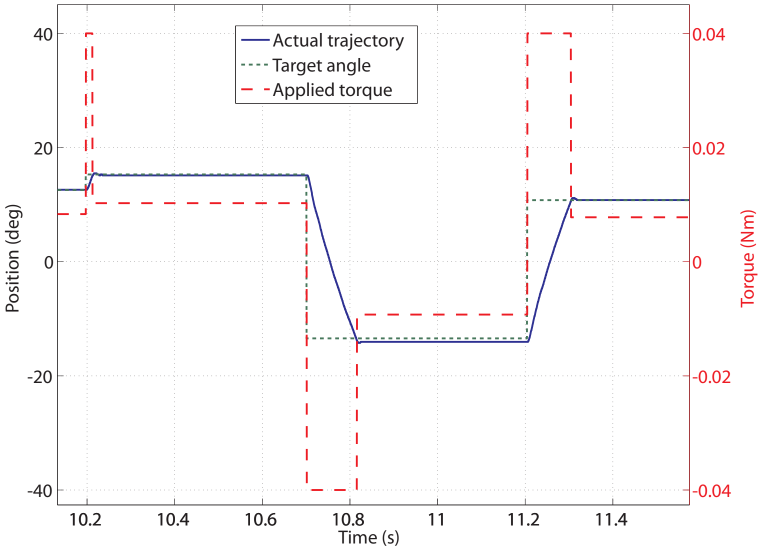


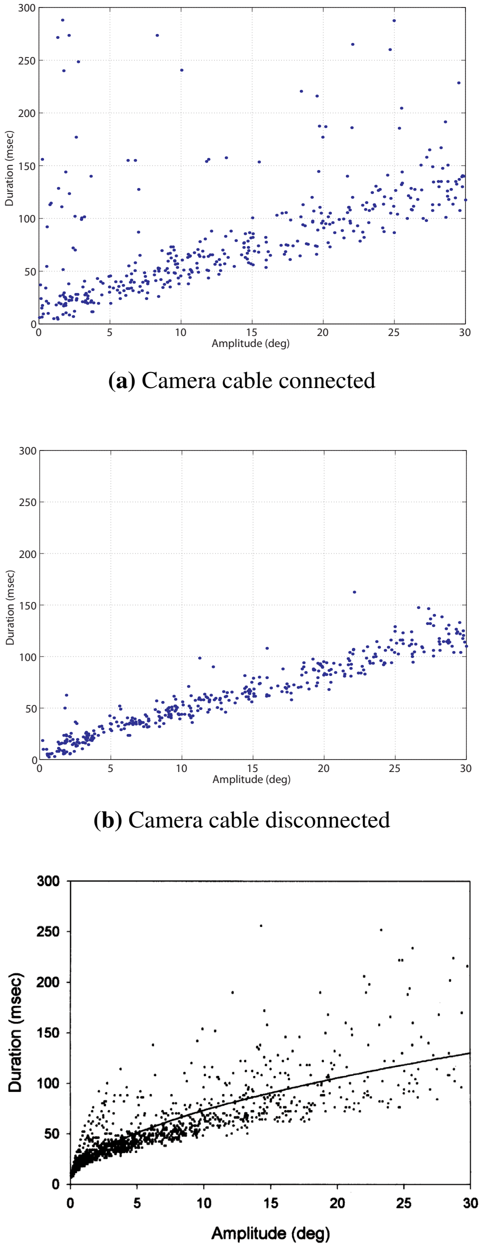
Figure 3.13: The experimental system output with the implemented nonlinear pulse-step controller. The input is a series of commanded target positions.

tionship, often referred to as the main sequence [6], is shown in Figure 3.15.

To demonstrate friction learning, we counducted an experiment with a geared (262:1) DC motor (1624012S, MICROMO, Clearwater, FL). At an efficiency rating of 59%, the gearhead introduces a significant amount of friction into the plant. We excite the plant with a band-limited white noise source. The F term at 0 velocity is shown in 3.16. The plot shows the two sides of the discontinuity at zero velocity. The observed Coulomb friction remains approximately constant across the motor range at 10 mNm.

3.7 Conclusions

There are many control strategies commonly used for the control of motion. However most of them only work well with a linear system and stability issues tends to arise as the system performance requirement and the associated controller gains increase. They often require knowledge of the plant which can be difficult to obtain or hobbled by parameter inaccuracies and unmodelled dynamics. We have devel-



(a) Camera cable connected

(b) Camera cable disconnected

(c) Human saccades data. Reproduced from [55], by permission of Oxford University Press.

Figure 3.15: The main sequence relationship between saccade amplitude and duration. (a) The amplitude vs. duration plot of our experimental setup with the camera cable attached shows a trend similar to the main sequence in the eye. (b) With the camera cable disconnected, the resulting main sequence relationship has a smaller variance. (c) Measured human saccades data fitted with a power equation.

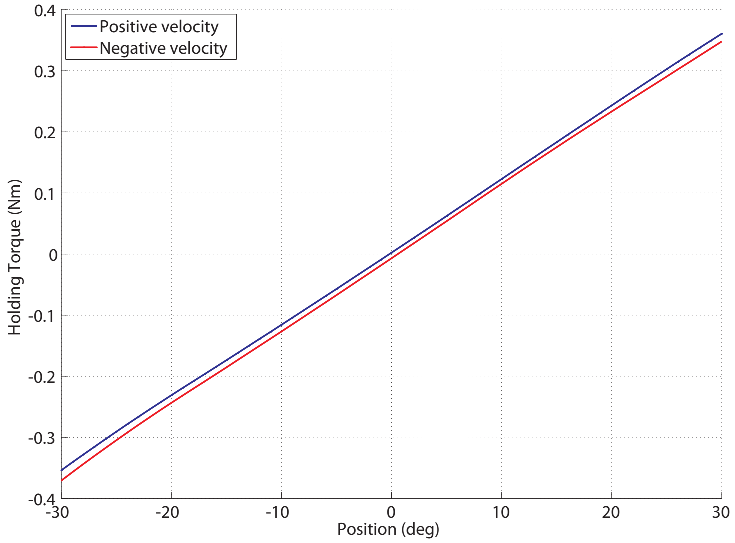


Figure 3.16: The learned holding torque at 0 velocity for a gearhead motor.

oped a control system strategy learned from observations of fast eye movements in animals. This controller is unhindered by many of the issues and limitations associated with a typical controller such as the one described above. In fact, it is able to apply sharp changes to the control signal which is not possible with a PD controller for example, without setting its feedback gains to be very high. By implementing this controller on a robotic platform, we have also at the same time shown the plausibility of the use of such a controller for fast eye movements in animals.

A challenge that we encountered in our implementation is with the dynamics introduced by the camera cable. The parasitic torque that it introduces has been at least partially modeled by our learning algorithm. Even so, the performance of the system is still affected by the cable. This indicates that the dynamics of the cable has not been captured fully by our model, possibly due to its high order dynamics or dependence on initial conditions.

Chapter 4

A Controller for Gaze Stabilization Movements

4.1 Introduction

The human vision system has to function in a dynamic environment. Motion of either the imaged objects or the eye itself causes two significant problems: (1) motion blur, which degrades vision, and (2) disappearance of objects from the field of view, which makes vision impossible.

Similar challenges are encountered in robotic vision system, and a common practice to deal with these is by using quick fixes. To avoid camera motion blur, the camera motion is restricted. To avoid object motion blur, strong illumination and short shutter speeds are used (alternatively the resolution is reduced). To prevent objects from moving out of view, wide field of view optics are used (with an associated loss of spatial resolution).

In contrast, biological systems deal with fast motions by active movements of the eye to counteract the unwanted motions and keep the image stable on the retina. This *active gaze stabilization* relies on measuring both the acceleration of the head using the vestibular system and image motion in the form of a *retinal slip* signal. Applying this trick from biology to robotic vision systems could, for example, allow a humanoid to recognize objects while walking, or allow a visual Simultaneous Localization and Mapping (SLAM) system in a car to work while driving in rough terrain (by suppressing motion blur from bumps and vibrations).

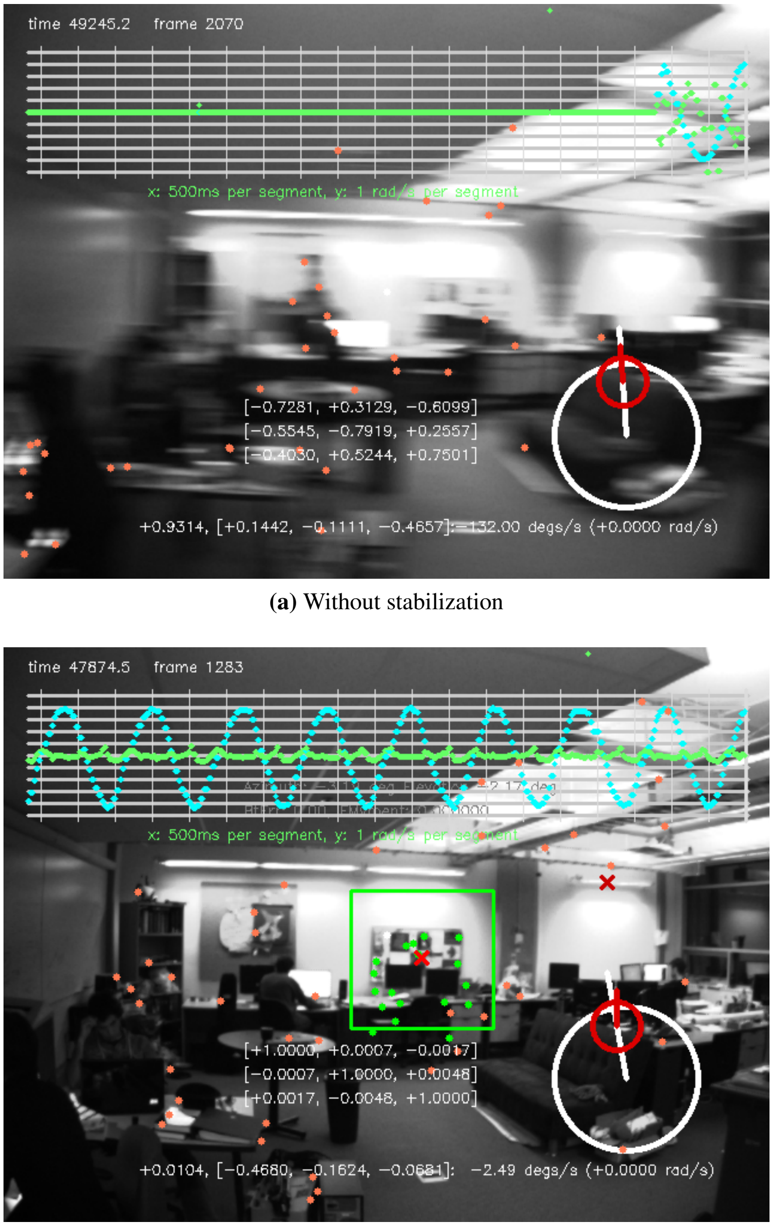
Following the terminology in biology, we will use the word “head” to refer to the platform on which both an actuated camera and an Inertial Measurement Unit (IMU) are mounted. Figure 4.1a shows the results when the head is subjected to a high speed rotation; Figure 4.1b shows the same motion of the head, but using our system to stabilize gaze. We use a simple setup to test and analyse the performance of our stabilization system. Although the degree of freedom is limited (1), this enables us to perform rigorous and reproducible performance evaluation and comparison.

Outline

The rest of the chapter is organized as follows. We briefly discuss the related work in Section 4.2. In Section 4.3 we provide an overview of the hardware used in the head and the test platform. Section 4.4 describes our adaptive controller and Section 4.5 presents the vision algorithm used to estimate the retinal slip. In Section 4.6 we discuss several stabilization experiments and their results. Section 4.7 looks at the effect of fast movements on computer vision algorithms. We conclude the chapter in Section 4.8.

4.2 Related Work

We have previously (Chapter 2) mentioned the three neuron arc that provides the direct pathway for the VOR. This short latency pathway is well understood, and for horizontal rotational VOR, connects the semicircular canal to the vestibular neuron which then projects to the oculomotor neuron [18]. However, there is a lot of interest in trying to understand the mechanism and pathway for learning in the VOR. This is a part of the bigger question of where and how motor learning occurs in the brain. One hypothesis suggests that the cerebellar flocculus is the main site for learning [44]. This aligns well with the models by Marr and Albus on the role of climbing fibres in sending the learning signals to affect the learned motor model in the cerebellum [3, 67]. More recent works [25, 26, 48, 108] further expand the learning process model in the brain, solving some of the shortcomings of earlier works. In particular, a model of the cerebellum uses a recurrent architecture to


time 49245.2 frame 2070
x: 500ms per segment, y: 1 rad/s per segment
[−0.7281, +0.3129, −0.6099]
[−0.5545, −0.7919, +0.2557]
[−0.4030, +0.5244, +0.7501]
+0.9314, [+0.1442, −0.1111, −0.4657]: 132.00 degs/s (+0.0000 rad/s)
time 47874.5 frame 1283
x: 500ms per segment, y: 1 rad/s per segment
[+1.0000, +0.0007, −0.0017]
[−0.0007, +1.0000, +0.0048]
[+0.0017, −0.0048, +1.0000]
+0.0104, [−0.4680, −0.1624, −0.0681]: −2.49 degs/s (+0.0000 rad/s)

(a) Without stabilization

(b) With our stabilization method

Figure 4.1: Image frames captured by the camera at 60 fps. (a) shows a frame captured near the peak velocity (230°/sec) of a sinusoidal disturbance without stabilization. (b) shows a corresponding frame captured with gaze-stabilization turned on.

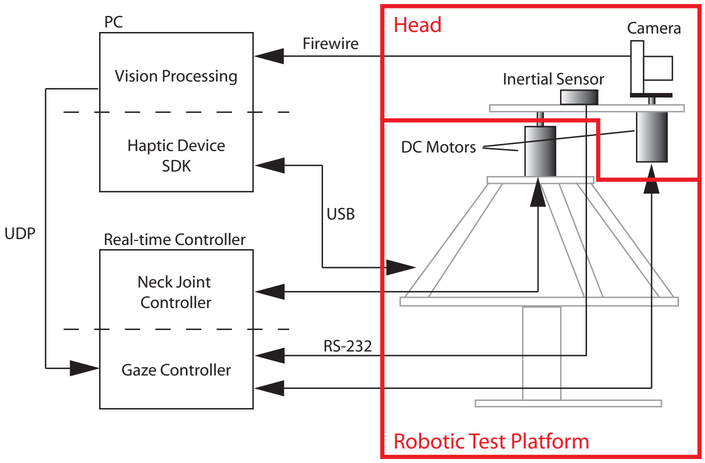


Figure 4.2: Overview of the hardware setup used to test the gaze control algorithm. The gaze controller uses only the visual, inertial, and camera position information to compensate the camera motion generated by the test platform.

solve the motor-error problem associated with feedback error learning [25, 26].

On the non-biological side of things, most of the techniques used for active gaze stabilization, also referred to as image or video stabilization, are targeted at small motions. A common application is suppression of vibrations of hand-held cameras, e.g. by moving the internal camera optics in response to the sensed inertial motion [15, 76]. In [38], an onboard camera is stabilized against vehicle vibration by panning and tilting either a set of external mirrors or the camera itself. Alternatively, one can capture the image as is (without stabilization) and use digital post-processing to correct the motion blurred image [46, 107]. This has been demonstrated to be effective in stabilizing video sequences captured from moving vehicles [61, 70].

With the exception of [38], the above methods are not suited to compensating for large motions encountered, for example, in walking. Consequently, humanoid robots often rely on the larger range of whole camera motions to perform gaze stabilization [13, 27, 58, 81, 96]. The Cog robotic platform uses a learned mapping between retinal displacement to motor command to compensate for motion sensed from optical flow and vestibular signals [13]. The Babyrobot platform [68] uses a neural-network stabilization algorithm incorporating both visual and inertial input [81]. The iCub robot [10] uses a chaotic controller that relies on visual feedback to stabilize the camera motion [27]. The camera stabilization system in [96] incorporates a controller based on the feedback error learning model of the cerebellum as proposed by Kawato [49]. A stabilization system based on the recurrent cerebellar model was implemented on a pneumatic-air-muscle driven eye in [58].

Purely image-based stabilization and tracking has been studied for a long time under the name *visual servoing* [23, 42, 52]. Compared to systems using inertial sensors, visual servoing is better suited to tracking of slow phenomena, due to the limited sampling rate of most cameras. A notable exception is the 1000 frames per second camera system described in [72]. The system relies on a custom hardware setup consisting of 128 Field Programmable Gate Array chips to process an image with a resolution of 128x128 pixels. It would also require strong scene illumination or wide lens aperture to compensate for the high frame rate. Therefore this approach is not suitable for many applications. In [91], a head mounted camera system is stabilized using the eye's own VOR and OKR by continuously tracking and reproducing the eye orientation. Although this system has useful clinical [93] and interactive [90] applications, it does not model the motor control system involved in eye movements and is also not suitable as a method to stabilize the vision system in most robotic applications.

Even though there has been a considerable amount of related work, to our knowledge the system described in this chapter is the first (non-biological) stabilization system capable of handling large and fast movements. Furthermore, its performance has been carefully evaluated using reproducible disturbances, something that is often absent in the existing literature. This provides a means for a fair comparison with other stabilization algorithms.

4.3 Hardware Overview

The system hardware consists of the following major components (see Figure 4.2).

- Vision system.

- Head, consisting of a camera, motor, and IMU.
- Test platform, a high speed robot (3 DOF translational + 1 DOF rotational).
- Control computers.

4.3.1 Vision System

The vision system uses a Firewire camera running at 60 frames per second with a resolution of 648×488 (Dragonfly 2, Point Grey, Richmond, BC). The lens used provides a 74° horizontal viewing angle. The load inertia of the camera is minimized by only mounting the remote head on the motor while the processing board remains fixed to the head platform. The vision software described in Section 4.5 was developed using the OpenCV library.

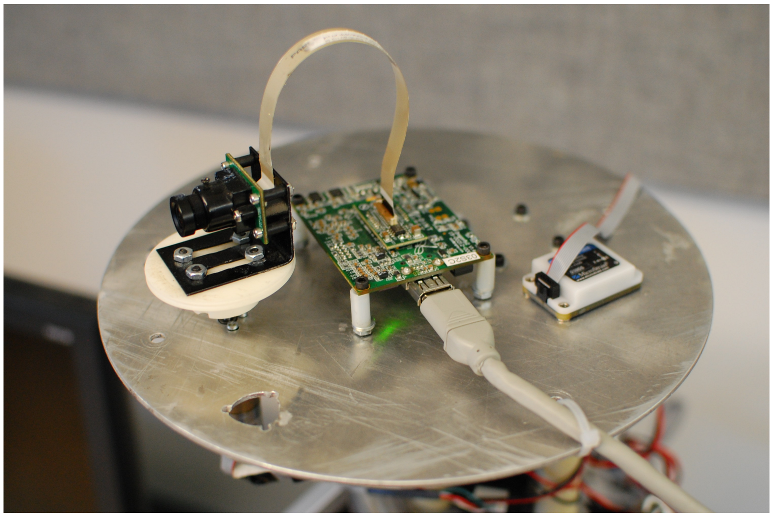


Figure 4.3: The head platform carrying the actuated camera system (left), and the IMU (right).

4.3.2 Head

The camera is actuated using a geared (262:1) DC motor (1624012S, MICROMO, Clearwater, FL) approximately (< 1cm away) about its optical center. The gearing gives the attached 16 lines per revolution magnetic encoder a resolution of 0.02°. This is much higher than the backlash due to the gearbox itself (up to 3°). The IMU (3DM-GX3-25, MicroStrain, Williston, VT) provides both translational acceleration and angular rate information. It has a full-scale range of ±5g and ±300°/sec with a 17-bit resolution. The inertial measurements are sent to the real-time controller through a serial (RS-232) connection at 1000 Hz.

4.3.3 Test Platform

The head can be used while mounted on any mobile platform. However, to carefully test system performance we built a platform consisting of a high-speed translation stage which can move the head with 3 DOF, on which is mounted a rotation stage (called the "neck") that can rotate the head in the horizontal plane.

The translation stage consists of a commercial haptic device based on the delta parallel mechanism (delta.3, Force Dimension, Nyon, Switzerland). Its parallel kinematic structure enables a high end-effector force (20 N continuous) and a large workspace (40 cm diameter). These properties are useful in moving the rest of the platform (including the head) quickly. The base of the delta mechanism is oriented such that it is parallel to the ground as shown in Figure 4.4 and controlled through USB by a PC. The neck is driven by a geared (14:1) DC Motor (2342012CR, MICROMO) with a 1024 lines per revolution optical encoder.

4.3.4 Control Computers

The vision software and haptic device control execute on a generic PC. The rest of the setup is controlled by a hard real-time controller equipped with various I/O modules running at a sample rate of 0.5 msec (xPC Target, MathWorks Inc., Natick, MA). A 16-bit digital-to-analog board drives the linear motor amplifiers (LSC 30/2, Maxon motor AG, Sachseln, Switzerland) connected to the camera and neck joint DC motors.

4.4 Adaptive Controller for Gaze Stabilization

Gaze stabilization requires the camera orientation to be maintained with respect to the world or object of interest. While the camera orientation can be measured

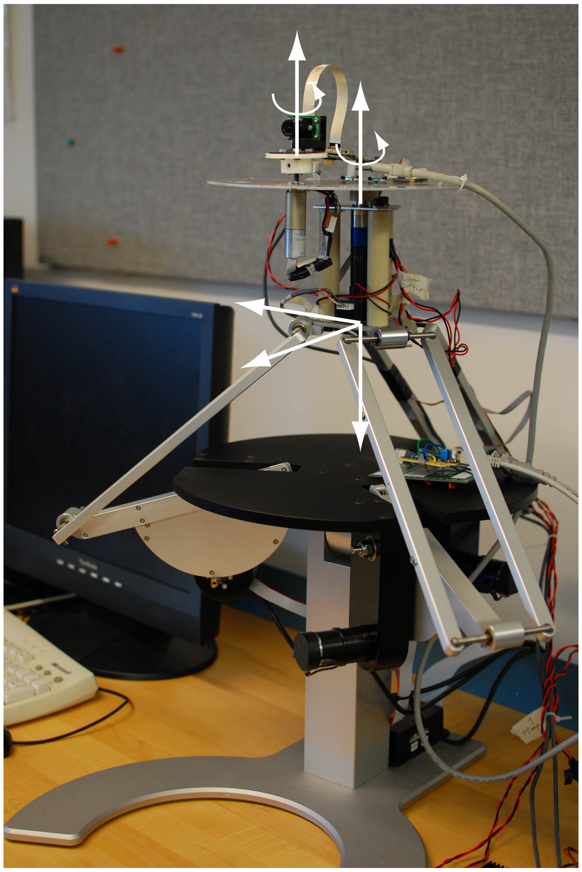


Figure 4.4: The head platform mounted on the robotic test platform with the axes of motion indicated. The camera head is actuated by a DC motor allowing a rotation about an axis perpendicular to the ground. The test platform rotational axis (at the neck joint) is parallel to the camera's and is colinear to one of its translational axis.

using visual information, this is not always feasible when fast motion is involved due to its limited sampling rate. This is an important and subtle point: the sampling rate has to be limited due to the optical requirements of the camera, available illumination, and delays in visual processing. However, this results in motion blur in the images, making visual processing even more difficult. Instead, our gaze stabilization algorithm relies on the much faster IMU velocity data to drive a velocity compensation loop. We then add an outer position tracking loop, driven by vision information, to avoid drift.

To perform velocity compensation, we implemented a model of the rotational VOR similar to the one described in [26] (see box in Figure 4.5a). The model's recurrent architecture allows for an adaptive controller which employs gradient descent learning without requiring a complicated inverse plant model. However, unlike [26], our brainstem model is a simple gain, representing the simplest model of the plant inverse since this is easier to obtain from the DC motor's specification (as the inverse of the speed constant). The brainstem performs basic compensation to the velocity sensed by the vestibular system (inertial sensor). This simple plant inverse model is improved through the modulation of the *inputs* to the brainstem controller by the cerebellum. The cerebellum is modeled as an adaptive filter trained using a *distal* (sensory) signal (here the retinal slip).

Retinal slip. This term is somewhat ambiguously used in the literature, and needs some clarification before we start. We define the term to mean the residual angular velocity of the eye relative to the visual world. This is a vector quantity in general, though in the planar case it is a scalar. It has units of radians (or degrees) per second. In a discrete setting, it is more useful to interpret this angular velocity in terms of the image motion between consecutive video frames. Therefore we will, on occasion, change the unit of angle to be the angle subtended by a standard pixel (typically at the center of the image), and unit of time to be the inter-frame interval, and express retinal slip as pixels/frame. We discuss the method used for estimating retinal slip in Section 4.5.

Notation. We use leading superscripts and subscripts to denote coordinate frames: ${}^{w}_{h}\boldsymbol{\omega}$ refers to the angular velocity $\boldsymbol{\omega}$ of frame h with respect to frame w. We use w, h, l to denote world (inertial), head, and camera frame respectively.

The plant inverse modeled by the brainstem and cerebellum forms a feedforward controller to track an input angular velocity signal, ${}^{h}_{l}\omega_{\text{des}}$. To achieve disturbance cancellation in VOR, this signal should correspond to the negative of the head velocity measured by the vestibular system

$$ {}^{h}_{l}\omega_{\text{des}} = -{}^{w}_{h}\hat{\omega}. \tag{4.1}$$

The distal error signal is defined as

$$e_c = {}^{h}_{l}\omega - {}^{h}_{l}\omega_{\text{des}}. \tag{4.2}$$

One of the advantages of VOR stabilization is that this error in the motion of the camera in the head can now be estimated from the retinal slip ${}^{w}_{l}\hat{\omega}$. This makes sense since, ignoring the non-colinearity of the rotational axes for the moment, the kinematic relationship between the measured variables for rotational VOR is given by

$${}^{w}_{l}\omega = {}^{w}_{h}\omega + {}^{h}_{l}\omega\,. \tag{4.3}$$

By substituting (4.1) and (4.3) in (4.2) we get

$$e_c = {}^{w}_{h}\hat{\omega} + {}^{h}_{l}\omega = {}^{w}_{l}\hat{\omega}. \tag{4.4}$$

This analysis could be extended to include camera translations (or equivalently, by Chasles' theorem, rotation about a different point). However, it is well known that errors due to camera rotation are much more significant than those due to translation (e.g., [107]), so we can safely ignore translations and small offsets of the rotation axis unless the visual targets are very close to the camera.

As mentioned above, we implement the brainstem filter as the inverse of the motor speed constant of the DC motor driving the camera; this is a parameter that is easily obtained from the motor specifications. The cerebellum is implemented as a Finite Impulse Response (FIR) filter with the following transfer function

$$C(q) = \sum_{k=1}^{K} w_k q^{-kT_t}\,, \tag{4.5}$$

where q is the forward shift operator, T_t is the tap delay, and w_k are the adaptive weights that define the filter. We use $K = 160$ taps with a tap delay of $T_t = 10$ ms, resulting in a filter of length 1.6 sec.

Each time a retinal slip signal arrives, the weights are modified with the update rule

$$\Delta w_k = -\gamma u(t - kT_t) e_c(t)\,, \tag{4.6}$$

where u is the control signal sent to the motor and γ is the learning rate. Appropriate values of γ were found to be in the range $\gamma \in [10^{-5}, 3 \times 10^{-5}]$. Lower values give very slow learning, and higher values result in an unstable filter due to the noise present in the retinal slip signal.

As the controller runs faster than the tap delay, learning is handled asynchronously. The weights are updated as soon as more than one tap delay has passed and a new retinal slip value is available.

Position tracking is performed similar to Shibata et al. [2001], who add a proportional, or *P-controller* to convert the position error into a velocity error signal (e_p). We calculate the position error as the difference between a reference position ${}^{w}_{l}\theta_{\text{ref}}$ and the camera position ${}^{w}_{l}\hat{\theta}$ as determined by a multi-resolution KLT tracker (see Section 4.5 for more details). This is added with the negative of the head velocity estimate to form the new signal tracked by the inner velocity loop. The distal error signal is also modified to reflect the change in desired velocity. The modified system architecture is illustrated in Figure 4.5a. We have in effect implemented a model of the OKR: the camera tracks the motion of the visual field even in the absence of head motion.

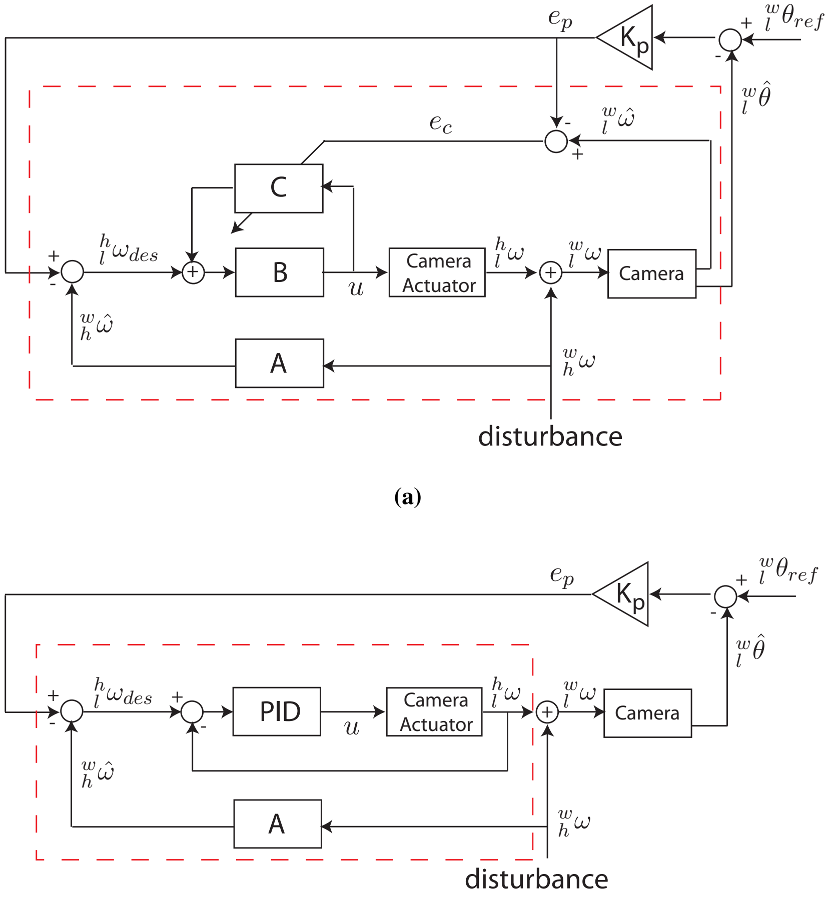
4.5 Retinal Slip Estimation

Retinal slip is the main signal that facilitates learning of all aspects of gaze stabilization, including VOR, fixation, and OKR. Here we describe how it is estimated robustly.

4.5.1 Multi-resolution KLT

We make use of the multi-resolution KLT tracker [65, 95] as implemented in OpenCV. The KLT algorithm tracks small rectangular patches by finding the subpixel shift that aligns the Taylor expansion of each patch at two different time instances. The original KLT tracker is accurate, but can only handle small displacements, as it requires a good initial guess in order to work.

The OpenCV implementation makes use of coarse to fine search to also handle large displacements. Here alignment happens first at a coarse resolution. This first alignment is then used as an initial guess at the next finer resolution. This process



(a)

(b)

Figure 4.5: Alternative gaze stabilization architectures consisting of the velocity tracking inner loop (inside the dashed box) and a position tracking outer loop. The two architectures differ in the design of the inner loop. The angular rate inertial sensor (A) provides an estimate of the rotational velocity of the head platform. (a) Cerebellar-like recurrent controller. In the recurrent design, the desired camera velocity is augmented with the output of the cerebellum (C) and used by the brainstem (B) to generate the plant control signal (*u*). (b) Feedback controller with PID. This controller uses the velocity derived from the encoder data as a fast feedback signal.

is repeated until the finest resolution is reached.

4.5.2 Crosschecking and Resolution Adaptivity

As the KLT algorithm tracks small patches, it has problems near depth-discontinuities (where half the patch may look different in one of the frames) it also frequently matches the wrong regions if a repetitive texture is viewed. Most of these erroneous matches can be detected, and subsequently eliminated by adding a crosschecking step [7]. This step tracks each patch, first forwards and then backwards in time. Points that do not end up where they started (within a small tolerance) are rejected.

During fast rotational motions the KLT algorithm often fails to find correspondences due to motion blur. For this reason we have added a layer on top of OpenCVs KLT. This layer detects frames where many regions fail to be tracked, and subsequently reduces the resolution of the images fed into KLT one octave at a time.

4.5.3 Estimation of 3D Camera Rotation

The coordinates of points tracked by KLT are expressed in the image grid, and thus depend on both the focal length and the optics of the camera. We convert the coordinates to normalised image coordinates using calibration parameters found from calibration using a planar target [110]. The projection model assumes that image coordinates $\mathbf{x}$ are generated from 3D points $\mathbf{X}$ as:

$$\mathbf{x} = f(\tilde{\mathbf{x}}, k_1, k_2) \quad \text{where} \quad \tilde{\mathbf{x}} = \mathbf{K}\mathbf{X}\,. \tag{4.7}$$

Here k_1 and k_2 are the lens distortion parameters, and $\mathbf{K}$ is the intrinsic camera matrix, all provided by the calibration.

Using the camera calibration, we find the normalised image coordinates $\mathbf{u}$, as:

$$\mathbf{u} = \mathbf{K}^{-1} f^{-1}(\mathbf{x}, k_1, k_2)\,. \tag{4.8}$$

The normalised coordinates are homogeneous 3-element vectors, and they have the desirable property that they are proportional to the 3D coordinates $\mathbf{X}$. By normalising them to unit length, we obtain the projection of the 3D points onto the unit sphere.

$$\hat{\mathbf{u}} = \mathbf{u}/\sqrt{u_1^2+u_2^2+u_3^2}\,. \tag{4.9}$$

Using projections of a set of points onto the unit sphere at two different time instances, we can compute the relative 3D rotation of the camera using the Orthogonal Procrustes algorithm [92]. Consider a set of normalised points $\mathbf{U} = [\hat{\mathbf{u}}_1 \ \ldots \ \hat{\mathbf{u}}_N]$ and the corresponding points at a different time instant $\mathbf{V} = [\hat{\mathbf{v}}_1 \ \ldots \ \hat{\mathbf{v}}_N]$. Assuming that the camera has undergone a pure rotation, they should be related as $\mathbf{U} = \mathbf{RV}$. The Orthogonal Procrustes algorithm finds the least-squares approximation of the unknown rotation $\mathbf{R}$ by solving the following problem:

$$\arg\min_{\mathbf{R}} ||\mathbf{U}-\mathbf{RV}||^2 \text{ , subject to } \mathbf{R}^T\mathbf{R} = \mathbf{I}\,. \tag{4.10}$$

Using the singular value decomposition (SVD) of the matrix $\mathbf{UV}^T$ the solution becomes [35]:

$$\mathbf{R} = \mathbf{AB}^T \text{ where } \mathbf{ADB}^T = \text{svd}(\mathbf{UV}^T)\,. \tag{4.11}$$

This is how we find the 3D camera rotation $\mathbf{R}$ between two consecutive frames.

The rotation matrix $\mathbf{R}$ is related to the angular velocity vector $\omega = [\omega_1\ \omega_2\ \omega_3]^T$ through the matrix exponent

$$\mathbf{R} = \exp\left(\begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} t\right), \tag{4.12}$$

where t is the inter-frame time interval. For rotation matrices the logarithm has a closed form expression, see e.g. [39], result A4.6. As our camera currently only has one axis of rotation, we compute the angular velocity about that axis by projection of the angular velocity vector onto the camera rotation axis, $\hat{\mathbf{n}}$,

$$\omega = \hat{\mathbf{n}}^T \omega\,. \tag{4.13}$$

Note that this gives us the angular velocity in rad/frame.

4.5.4 Retinal Slip Estimation Performance

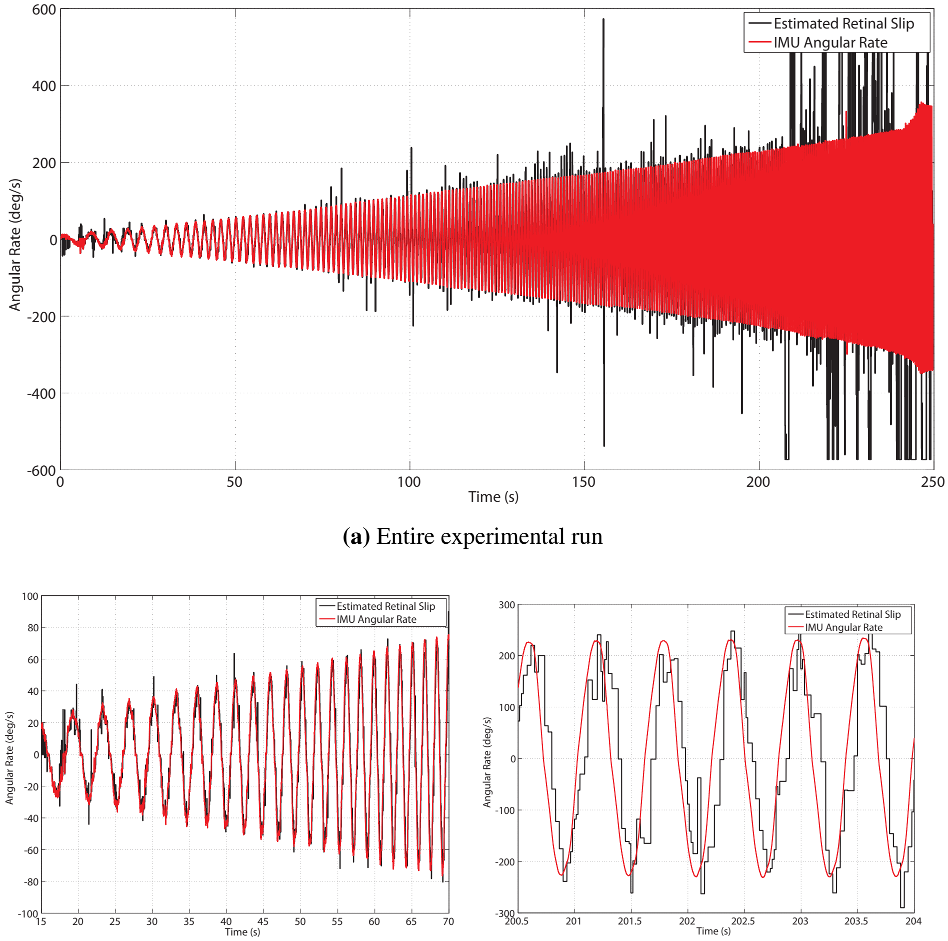
To evaluate the performance of our retinal slip estimation algorithm, we carried out an experiment comparing the retinal slip estimation results with the true angular velocity. We apply sinusoidal signals to the neck motor with increasing peak velocity starting from 10°/sec all the way to 300°/sec. During this experiment, the eye motor is disabled and we use the IMU reading as ground truth for the camera angular velocity. This experiment was conducted inside an office environment and the result is experiment is shown in Figure 4.6. We can see that the algorithm is able to estimate the retinal slip robustly up to 200°/sec and the estimate lags behind by the amount of time it takes to process a frame. This result demonstrates the algorithm robustness in the presence of motion blur even at high speed.

4.6 Experiments and Discussion

4.6.1 Stabilization Performance

We checked the controller stabilization performance by first disturbing the system with a sinusoidal position signal of varying frequency. The frequency ranges from 0.16 to 0.64 Hz with a corresponding maximum speed from 57°/sec to 230°/sec respectively. The maximum velocity encountered in this test is close to the maximal speed of the camera motor. Figure 4.7a shows a snapshot of the disturbance velocity profile and the corresponding stabilization response from the camera. The retinal slip trace in the Figure corresponds to the velocity of the features as seen by the camera. We believe that the retinal slip is the most important metric for evaluating the stabilization performance since the goal of a camera stabilization system is to minimize image movement. Note that the results are obtained after the cerebellar weights have been learned. We observe that the difference between disturbance velocity and camera velocity corresponds to the amount of retinal slip as expected. Most of the error occurs as the camera velocity lags behind during a change in orientation.

Although velocity tracking such as shown in Figure 4.7a is commonly used to evaluate VOR performance, an often overlooked aspect of VOR is position tracking. The position tracking performance of the system, corresponding to the same time



(a) Entire experimental run

(b) Initial segment with lower velocity

(c) Final segment with higher velocity

Figure 4.6: Retinal slip estimation performance under sinsuoidal motion with increasing velocity. The estimation result is compared against the angular rate measured by the IMU. Measurement data are obtained from [53].

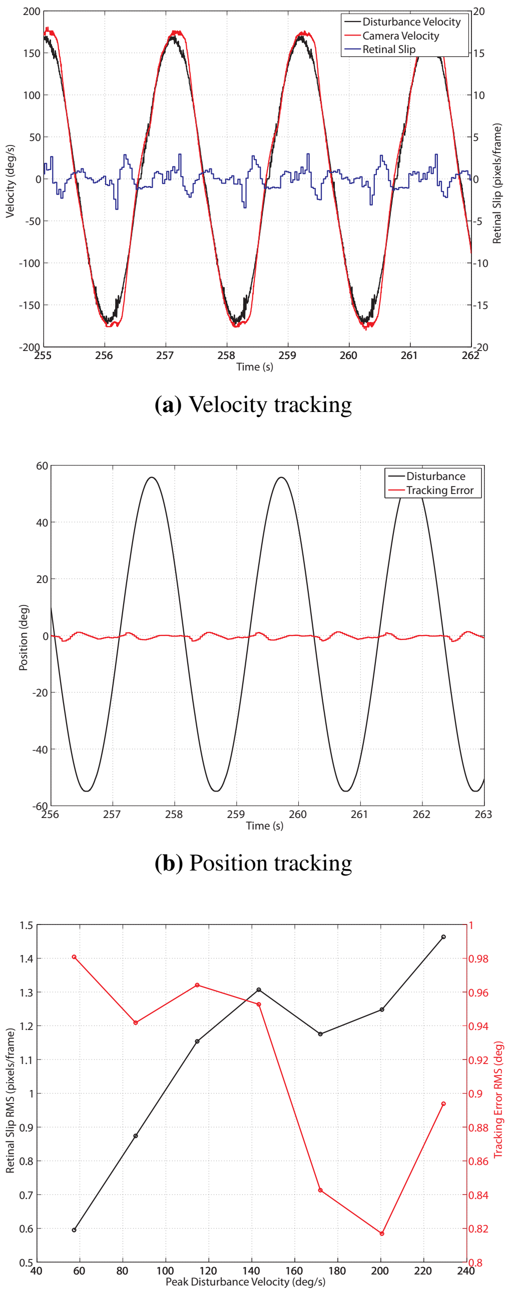
period as Figure 4.7a is shown in Figure 4.7b. Here, the disturbance causes a maximum rotational displacement of ±56° while the stabilization system tracking error is bounded to 2°.

The system performance for different disturbance speeds is shown in Figure 4.7c. As expected, the retinal slip tends to increase as the disturbance velocity increases. On the other hand the tracking error appears to be more robust to changes in the disturbance speed. We see that the stabilization system gives, for a disturbance up to 230°/sec, a RMS tracking error of less than 1° and a frame-to-frame image motion of less than 1.5 pixels. Figure 4.1 illustrates the motion blur effect caused by a disturbance of such speed and the large improvement obtained from using the gaze-stabilization system.

We also tested the stabilization algorithm with a coloured noise disturbance. This is representative to the disturbance experienced by, for example, a vehicle travelling over a rough terrain. In this test we use white noise bandpass filtered from 0.5 to 1.5 Hz as a disturbance, similar to [58]. Figure 4.8 shows the response of the setup to the noise disturbance. After the cerebellar weights have been learned, the system has a retinal slip RMS of 2.1 pixels/frame.

We can clearly observe the effect of learning in the cerebellum by setting the gain in the brainstem path to be 50% of the correct gain. This causes the initial stabilization performance, before the cerebellum weights have been trained, to be very undergained. As the cerebellum becomes trained, the retinal slip quickly decreases. Figure 4.9a shows the speed at which the cerebellum learns and compensates for the deficient inverse plant model assumed by the brainstem. It is interesting to observe the evolution of the cerebellar filter weights during the course of adaptation. Initially the cerebellar filter has no influence on the output of the stabilization system as the filter weights start off as zero. With the arrival of the retinal slip signal, the update rule begins modifying the weights to minimize retinal slip. Figure 4.9b shows the progression of the filter weights during the initial course of this experiment. We can see that the system converges quickly — the weights reach a steady state distribution approximately 5 seconds into the experiment. This time period corresponds to the result in Figure 4.9a — steady state retinal slip performance is reached in approximately 7.5 seconds[1]. Note that, as expected, the filter weights converge to have the standard form of a low pass filter. In this case it has a ringing behaviour due to the negative lobe captured within the tap length.

[1] 5 seconds filter convergence time plus half of the window's length



(a) Velocity tracking

(b) Position tracking

(c) RMS error of velocity and position tracking for different velocities

Figure 4.7: Tracking performance to a sinusoidal disturbance.

(a) and (b) show tracking for a disturbance with a maximum velocity of 172°/sec.

(c) shows the stabilization performance to different peak velocities caused by sinusoidal disturbance of various frequencies. The sign of the camera velocity has been changed for easier visualization.

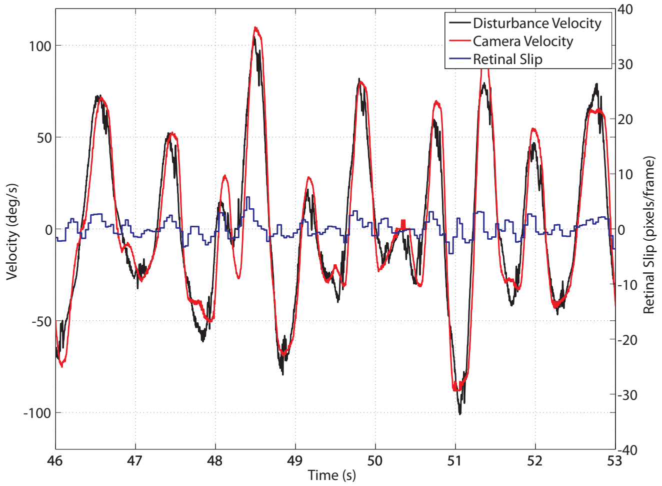
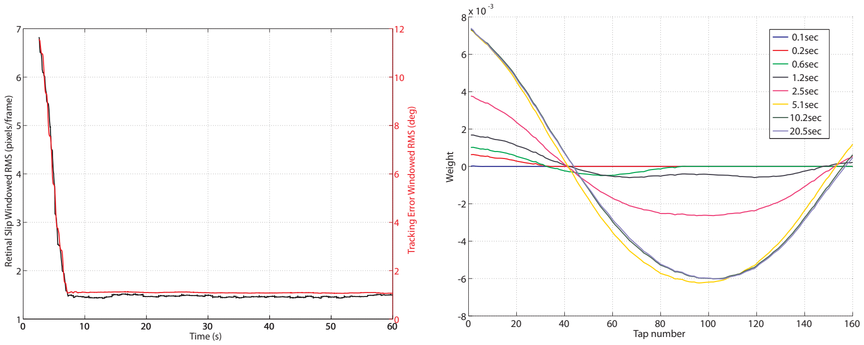


Figure 4.8: System performance to a bandpass noise disturbance. The sign of the camera velocity has been changed for easier visualization.



(a) Adaptation performance

(b) Filter weights evolution

Figure 4.9: Stabilization system performance under sinusoidal disturbance (maximum velocity of 230°/sec) with the brainstem gain set to 50% of the correct value. (a): Position and velocity tracking RMS error (with 5 seconds window). (b): The evolution of filter weights as the stabilization system adapt from an initial brainstem gain set at 50% of the correct value. Note that the point in times shown are logarithmic.

	Our adaptive controller	PID controller
Retinal Slip (pixels/frame)	1.65	1.69
Tracking Error (°)	0.91	2.03

Table 4.1: Comparison of RMS tracking performance of a PID controller and our adaptive controller. The disturbance signal is a 0.8 Hz sinusoid with a maximum velocity of 230°/sec.

4.6.2 Comparison with a Feedback Controller

A standard robotics approach to the stabilization problem posed is to use a feedback mechanism such as a PID controller. We implemented such a controller as illustrated in Figure 4.5b and compared its performance to our adaptive controller. A system identification was first performed to obtain the transfer function of the plant. We use the Matlab System Identification toolbox (MathWorks Inc.) to estimate a linear process model of the plant based on its response to a noise signal. Using the identified plant model, the PID gains were then tuned using the robust response time automated PID tuning tool in Matlab with a specified bandwidth of 72 rad/s and phase margin of 60 degrees. The result shows that the performance of our adaptive controller is comparable to that of the PID controller (see Table 4.1).

Although we have found the two controllers to have similar performance, there are several advantages with using our adaptive control structure over the more traditional feedback loop. One advantage is that it does not need an encoder in the camera actuator. This reduces cost, weight, and space requirements and improves system robustness — factors which are critical in some applications. Note that the inertial measurement system is needed in both controllers to measure the movement of the head. Furthermore, the self-learning nature of the controller enables good performance without controller gains tuning, a process which often requires system identification to get an accurate model of the plant. Instead, it is sufficient to use a coarse, low order plant model estimate as the brainstem controller. We have demonstrated the system's ability to accomodate a very simplistic (a zero order model for a geared DC motor) and inaccurate (50% of actual plant DC gain) plant model prior.

4.6.3 User-driven Disturbance

Along with the dynamic platform, we have also tested our gaze-stabilization system by mounting it on a wheeled-platform and disturbing it by hand as shown in Figure 4.10. Even though this does not have the repeatability offered by the dynamic platform, it provides a clear delineation between the disturbance source and the stabilization algorithm. Figure 4.11 shows a sample user-driven disturbance (as measured by the IMU) and the corresponding stabilization result. The results are also clearly seen in the video (Extension 1).

4.7 Effect of Fast Movements on Computer Vision Algorithms

Common computer vision tasks, such as detection and recognition, work better with sharp images, and their use on mobile platforms should thus benefit from active gaze stabilization. In this Section we demonstrate this with the very common task of face detection. The OpenCV implementation of a Haar Feature-based cascade classifier, first proposed by Viola and Jones [103, 104], is used as the face detector.

We test for the effect of fast movements on the face detection algorithm by placing a set of printed faces within the camera field of view at a fixed distance from the camera. The printed faces were used as substitute for human subjects as they can remain fixed in the scene across the different experiments for fair comparison. We then apply sinusoidal disturbances with increasing velocity and capture the images recorded by the camera. This is then repeated with the faces placed at a different distance. The captured frames in Figure 4.12 show that aside from the targets disappearing from the camera field of view, significant motion blurring also occurs at higher speeds. Both of these effects negatively affect the face detection algorithm. Even at $46^\circ/\text{sec}$, the face detection algorithm already misses some of the faces, at $230^\circ/\text{sec}$ none of the faces are detected anymore. Figure 4.14 shows the results for the frames which have the same field of view as the undisturbed case. We see that the face detection performance deteriorates as the speed of the disturbance, and consequently motion blur, increases.

We also repeated the experiment with the stabilization system turned on. The

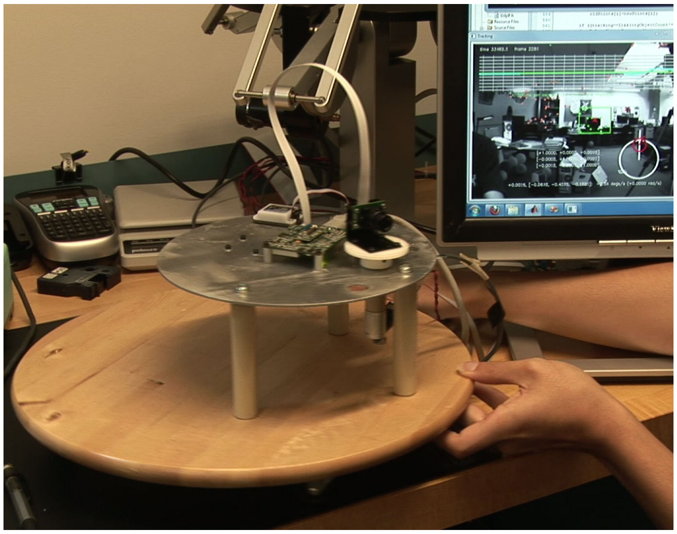


Figure 4.10: The wheeled-platform used to subject our gaze-stabilization system with user-driven disturbance.

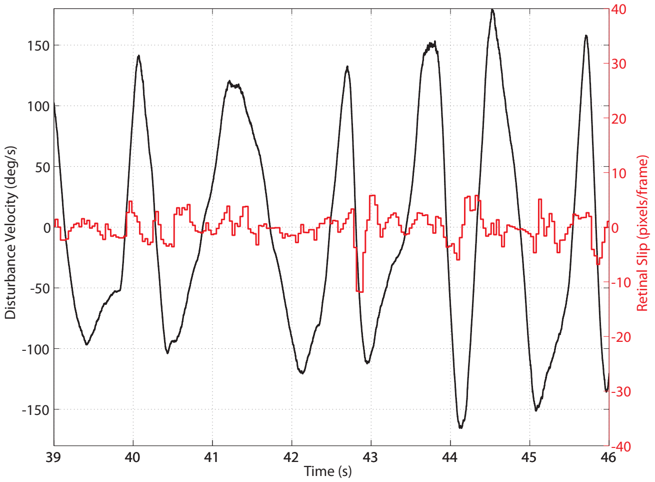


Figure 4.11: System performance to a user-driven disturbance. The disturbance velocity is obtained directly from the angular rate measurement of the IMU.

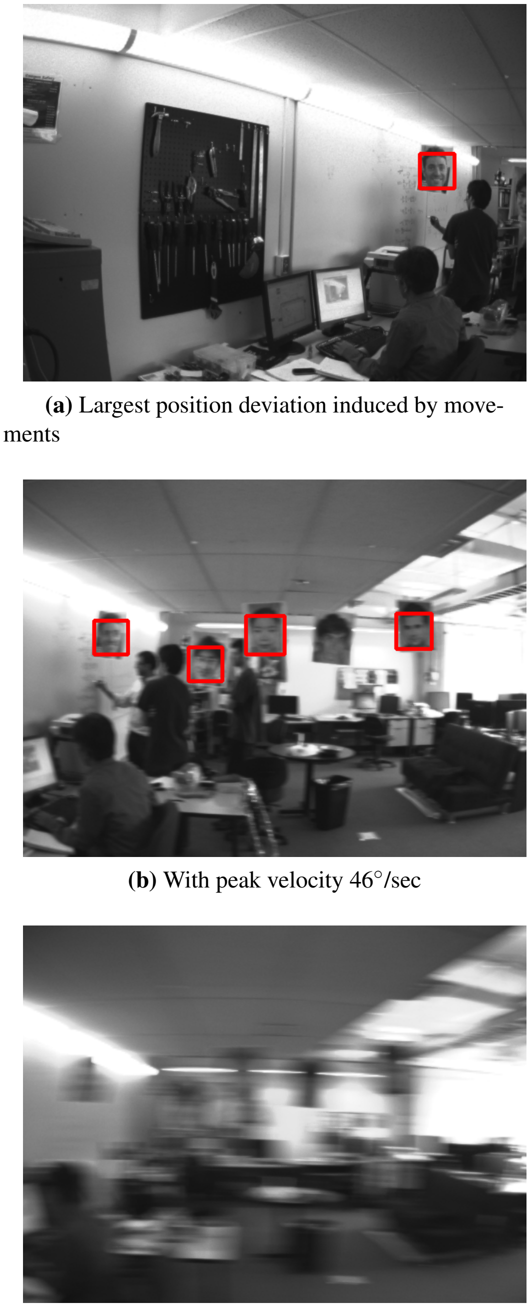
stabilization causes the field of view to be maintained under fast movements and also minimizes the blurring as shown in Figure 4.13. The face detection algorithm performs consistently well even under increasing disturbance speed (see Figure 4.14).

4.8 Conclusions

We described a computational model for the motor control of the Vestibulo-Ocular Reflex in humans and implemented it on a highly dynamic robotic system. The VOR model consists of both the "direct" pathway from the vestibular organ to the EOM and the site for motor learning. It employs a recurrent architecture model of the cerebellum which solves the distal error learning problem typically associated with feedback error learning.

Based on the VOR model above, we implemented a stabilization system for cameras mounted on highly dynamic robots. To focus on the stabilization algorithm and careful performance evaluation, we used a camera mounted on a robotic test platform that can have unknown perturbations in the horizontal plane. We showed that the camera can be effectively stabilized using an inertial sensor and a single additional motor, without a joint position sensor. Our stabilization algorithm is composed of an inner velocity loop driven by the fast IMU velocity data and an outer position tracking loop which provides drift correction. The velocity loop mimics the architecture of the biological VOR system whereby inertial information provides velocity compensation and the system is continuously tuned using the retinal slip error signal. The outer position loop uses position information from the camera to track an object, giving a behaviour similar to the OKR. By adding a fast velocity loop instead of just using the position loop, we can incorporate inertial information to compensate for fast platform motion. This keeps our setup simple and affordable as inertial measurements can be obtained at high sample rate much more easily than vision data at the same rate. We also developed a resolution-adaptive retinal slip algorithm that is robust to motion blur.

The controller design offers several advantages over traditional feedback control without sacrificing performance. These include reduced hardware requirements and a simpler design process. Although the stabilization system relies on

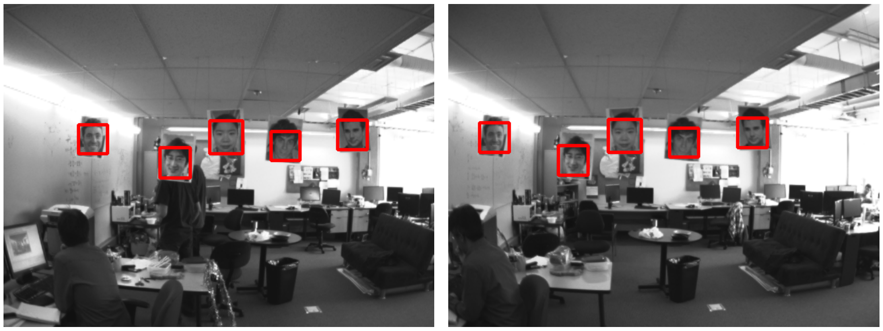


(a) Largest position deviation induced by movements

(b) With peak velocity 46°/sec

(c) With peak velocity 230°/sec

Figure 4.12: Image frames captured by the camera (60 fps) with sinusoidal disturbance without stabilization.
(a) shows a frame captured with the camera at its most extreme orientation (nearly 0 velocity).
(b) and (c) show the images captured near the peak disturbance velocity



(a) With peak velocity 46°/sec **(b)** With peak velocity 230°/sec

Figure 4.13: Image frames captured by the camera (60 fps) with sinusoidal disturbance with the stabilization system activated. (a) and (b) show the images captured near the peak disturbance velocity, corresponding to the camera pointing in the undisturbed orientation.

the result of visual processing, many of the applications which can benefit from it already perform this processing. Thus, the visual processing needed is likely to be available at no extra cost.

We evaluated the performance quantitatively using another high speed robot to generate repeatable sequences of large and fast movements that a gaze stabilization system can attempt to counteract. Thanks to the high-accuracy repeatability, we can make a fair comparison of algorithms for gaze stabilization. We show that the resulting system can reduce camera image motion to about one pixel per frame on average even when the platform is rotated at 200 degrees per second.

We also demonstrated how active gaze stabilization allows face detection to work in circumstances where it would otherwise fail completely. The reason for this is that visual acuity can be sustained. Humanoids and other interactive robots should directly benefit from this, as they make heavy use of face detection. In general, loss of visual acuity is a real and important issue encountered in all robotic systems with moving cameras. Similar performance improvements are to be expected for any vision algorithm that requires good visual acuity.

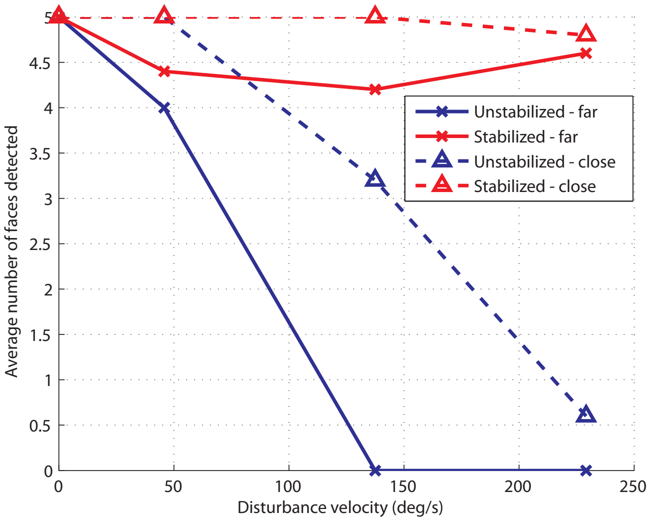


Figure 4.14: Relationship between disturbance velocity and face detection performance when all 5 printed faces are within field of view. Each data point corresponds to an average of over 5 samples, ignoring false positives.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this thesis we developed and described computational models of biological motor controllers used in eye movements. These controllers are then implemented on robotic systems, exposing them to various real world conditions and limitations, providing the acid test for the models.

In Chapter 3 we looked at saccades — a type of eye movements used in gaze shifting involving very fast yet accurate movements. A critical characteristic of these types of eye movements is that they are too fast for the use of a feedback controller in the presence of neural delays. This necessitates an open loop controller and consequently a model learning algorithm that can determine the unknown plant dynamics. We developed a learning algorithm that can identify a general second order non-linear plant and incorporate constraints to ensure that the learned system is physically plausible. The controller itself employs the pulse-step model observed in neurological recordings to enable fast movements even though the system is highly overdamped. The correct transition from pulse to step is made possible by a forward simulation of the learned plant dynamics, generating fast and accurate movements. Interestingly, our results reproduce the main sequence relation found in human eye movements.

The focus of Chapter 4 is on eye movements for gaze stabilization. This class of eye movements is particularly useful for both humans and mobile robots as it can improve the quality of the perceived image by reducing motion blur. We modeled the cerebellum adaptation mechanism for the fast vestibulo-ocular reflex and coupled that with the slower optokinetic response to achieve tracking. We demonstrated the robustness of the vision algorithm and stabilization controller to very fast disturbances on a high-speed robotic system. The benefit of such a system in robotics and computer vision is clear in the face-detection experiment we carried out.

A theme of this thesis is to apply ideas and techniques from biological motor control to solve several robotics tasks and challenges. Concurrently, we have also used robotics to explore and gain insight into different neuroscience ideas.

5.2 Future Work

There are several future work venues for both controllers presented in this thesis. We discuss them in the following sections.

5.2.1 Controller for Fast Movements

Feedback. Even with a good system model, steady state error will still persist due to noise. The classical approach to deal with steady state error is by adding an integral action to the feedback controller. Without a feedback system, it is impossible to eliminate steady state error completely. Therefore a feedback mechanism can be combined with the developed pulse-step controller to improve the steady state tracking performance. This can be in the form of a corrective step taken after the main motion (akin to a corrective saccade in human), or with the feedback mechanism taking into effect during the saccadic motion.

Pulse-step transition. A current limitation of the pulse-step control signal is the sharp transition between the pulse and the step. A slide region can be introduced between the two to give a smooth transition. This parallels the observed biological neural pattern.

Using vision information. An alternative to obtaining the trajectory data from instrumenting the actuator (an encoder in our setup) is to instead use the existing camera information. Steady state (targeting) error can be obtained by using a feature detector and determining the distance between the centre of the image to the position of the target of interest on the camera. Furthermore, optical flow information during camera motion can possibly be used to provide velocity (and consequently trajectory) information.

5.2.2 Controller for Gaze Stabilization

In the future, we want to explore the use of a second camera to provide binocular vision and hence depth perception. This will enable the system to perform translational VOR on top of the current rotational VOR. In addition, increasing the degrees of freedom of the camera will extend the different types of movements that the system can reject. Another avenue for future work is to improve the camera actuation which currently suffers from a large (3°) backlash due to the DC motor gearbox. Alternatives to the geared DC motor include cable driven systems, piezo actuators, and harmonic drives.

Bibliography

[1] B. Adams, C. Breazeal, R. Brooks, and B. Scassellati. Humanoid robots: A new kind of tool. *IEEE Intelligent Systems and Their Applications*, 15(4): 25–31, 2000. → pages 1

[2] N. Aeronautics and S. Administration. The effects of space flight on the human vestibular system. http://weboflife.nasa.gov/learningResources/vestibularbrief.htm, July 2011. → pages 7

[3] J. Albus. A theory of cerebellar function. *Mathematical Biosciences*, 10 (1-2):25–61, 1971. → pages 36

[4] C. An, C. Atkeson, J. Griffiths, and J. Hollerbach. Experimental evaluation of feedforward and computed torque control. *IEEE TRANS. ROBOT. AUTOM.*, 5(3):368–373, 1989. → pages 12

[5] S. Anderson, J. Porrill, S. Sklavos, N. Gandhi, D. Sparks, and P. Dean. Dynamics of primate oculomotor plant revealed by effects of abducens microstimulation. *Journal of neurophysiology*, 101(6):2907, 2009. → pages 6

[6] A. Bahill, M. Clark, and L. Stark. The main sequence, a tool for studying human eye movements. *Mathematical Biosciences*, 24(3-4):191–204, 1975. → pages 32

[7] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *Proceedings 2007 IEEE International Conference on Computer Vision (ICCV)*, 2007. → pages 47

[8] R. Baloh and V. Honrubia. *Clinical neurophysiology of the vestibular system*. Oxford University Press, USA, 2001. → pages 8

[9] K. Bassett, M. Hammond, and L. Smoot. A fluid-suspension, electromagnetically driven eye with video capability for animatronic applications. In *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, pages 40 –46, 2009. doi:10.1109/ICHR.2009.5379606. → pages 12

[10] R. Beira, M. Lopes, M. Praça, J. Santos-Victor, A. Bernardino, G. Metta, F. Becchi, and R. Saltarén. Design of the robot-cub (icub) head. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 94 –100, Orlando, Florida, May 2006. IEEE Robotics and Automation Society, IEEE. → pages 39

[11] L. Berthouze, P. Bakker, and Y. Kuniyoshi. Learning of oculo-motor control: a prelude to robotic imitation. In *IEEE/RSJ International Conference on Robotics and Intelligent Systems (IROS97), Osaka, Japan*, pages 376–381. Citeseer, 1997. → pages 12

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

[13] R. Brooks, C. Breazeal, M. Marjanović, B. Scassellati, and M. Williamson. The Cog project: Building a humanoid robot. *Computation for metaphors, analogy, and agents*, pages 52–87, 1999. → pages 12, 38, 39

[14] M. Buhmann. Radial basis functions. *Acta numerica*, 9:1–38, 2001. → pages 16

[15] Canon. What is optical image stabilizer? http://www.canon.com/bctv/faq/optis.html, December 2010. → pages 38

[16] H. Chen-Harris, W. Joiner, V. Ethier, D. Zee, and R. Shadmehr. Adaptive control of saccades via internal feedback. *The journal of neuroscience*, 28 (11):2804, 2008. → pages 6

[17] J. Cioffi and T. Kailath. Fast, recursive-least-squares transversal filters for adaptive filtering. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 32(2):304–337, 1984. → pages 22

[18] B. Cohen. The vestibulo-ocular reflex arc(induced eye movements). *Vestibular system. Part 1.(A 75-23302 09-51) Berlin, Springer-Verlag, 1974,*, pages 477–540, 1974. → pages 36

[19] H. Collewijn. Eye-and head movements in freely moving rabbits. *The Journal of Physiology*, 266(2):471, 1977. ISSN 0022-3751. → pages 11

[20] C. Collins. The human oculomotor control system. *Basic mechanisms of ocular motility and their clinical implications*, pages 145–180, 1975. → pages 12

[21] W. Commons. Public domain obtained from http://commons.wikimedia.org/wiki/File:Gray889.png, July 2011. → pages 6

[22] W. Commons. Public domain obtained from http://commons.wikimedia.org/wiki/File:Vestibulo-ocular_reflex_EN.svg, July 2011. → pages 10

[23] P. I. Corke. *High-performance visual closed-loop robot control*. PhD thesis, Mechanical and Manufacturing Engineering, University of Melbourne, 1994. → pages 39

[24] M. Davidson, M. Fox, and A. Dick. Effect of eye movements on backward masking and perceived location. *Attention, Perception, & Psychophysics*, 14(1):110–116, 1973. → pages 9

[25] P. Dean and J. Porrill. Oculomotor anatomy and the motor-error problem: the role of the paramedian tract nuclei. *Progress in Brain Research*, 171: 177–186, 2008. ISSN 0079-6123. → pages 36, 38

[26] P. Dean, J. Porrill, and J. Stone. Decorrelation control by the cerebellum achieves oculomotor plant compensation in simulated vestibulo-ocular reflex. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 269(1503):1895–1904, 2002. ISSN 0962-8452. → pages 3, 36, 38, 43

[27] B. Durán, Y. Kuniyoshi, and G. Sandini. Eyes-neck coordination using chaos. In H. Bruyninckx, L. Preucil, and M. Kulich, editors, *European Robotics Symposium 2008*, volume 44 of *Springer Tracts in Advanced Robotics*, pages 83–92. Springer Berlin / Heidelberg, 2008. URL http://dx.doi.org/10.1007/978-3-540-78317-6_9. → pages 38, 39

[28] P. Fabre and C. Gueguen. Improvement of the fast recursive least-squares algorithms via normalization: A comparative study. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 34(2):296–308, 1986. → pages 22

[29] E. Falotico, D. Zambrano, G. Muscolo, L. Marazzato, P. Dario, and C. Laschi. Implementation of a bio-inspired visual tracking model on the iCub robot. In *RO-MAN, 2010 IEEE*, pages 564–569. IEEE. → pages 12

[30] E. Fasse, N. Hogan, S. Gomez, and N. Mehta. A novel variable mechanical impedance electromechanical actuator. *Dynamic Systems and Control*, 55 (1):311–318, 1994. → pages 18

[31] P.-E. Forssén. Learning Saccadic Gaze Control via Motion Prediciton. In *Computer and Robot Vision, 2007. CRV'07. Fourth Canadian Conference on Computer and Robot Vision*, pages 44–54. IEEE, 2007. ISBN 0769527868. → pages 12

[32] A. Fuchs and E. Luschei. Firing patterns of abducens neurons of alert monkeys in relationship to horizontal eye movement. *Journal of neurophysiology*, 33(3):382–392, 1970. ISSN 0022-3077. → pages 12

[33] C. Fung and B. Shi. A Biomimetic Active Stereo System with Torsional Control and Saccade-like Speed. In *Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. The First IEEE/RAS-EMBS International Conference on*, pages 389–392. IEEE, 2006. ISBN 1424400406. → pages 12

[34] B. Girard and A. Berthoz. From brainstem to cortex: Computational models of saccade generation circuitry. *Progress in Neurobiology*, 77(4): 215–251, 2005. → pages 13

[35] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, 1983. → pages 48

[36] C. Gosselin and J. Hamel. The agile eye: a high-performance three-degree-of-freedomcamera-orienting device. In *1994 IEEE International Conference on Robotics and Automation, 1994. Proceedings.*, pages 781–786, 1994. → pages 12

[37] H. Gray. Anatomy of the human body, by henry gray. 20th ed., thoroughly rev. and re-edited by warren h. lewis., 1918. → pages 6

[38] W. Gunthner, P. Wagner, and H. Ulbrich. An Inertially Stabilised Vehicle Camera System-Hardware, Algorithms, Test Drives. In *IEEE Industrial Electronics, IECON 2006-32nd Annual Conference on*, pages 3815–3820. IEEE, 2007. ISBN 1424403901. → pages 38

[39] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision, 2nd ed.* Cambridge University Press, New York, NY, USA, 2003. → pages 48

[40] S. Highstein, R. Fay, and A. Popper. *The vestibular system*, volume 19. Springer Verlag, 2004. → pages 7

[41] N. Hung, H. Tuan, T. Narikiyo, and P. Apkarian. Adaptive control for nonlinearly parameterized uncertainties in robot manipulators. *Control Systems Technology, IEEE Transactions on*, 16(3):458–468, 2008. ISSN 1063-6536. → pages 12

[42] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *Robotics and Automation, IEEE Transactions on*, 12(5):651–670, 1996. ISSN 1042-296X. → pages 39

[43] U. Ilg. Slow eye movements. *Progress in neurobiology*, 53(3):293–329, 1997. → pages 9

[44] M. Ito. Neurophysiological aspects of the cerebellar motor control system. *International Journal of Neurology*, 7(2):162, 1970. → pages 36

[45] E. Javal. Essai sur la physiologie de la lecture. In *Annales D'Oculistique*, 1879. → pages 8

[46] N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski. Image deblurring using inertial measurement sensors. *ACM Trans. Graph.*, 29:30:1–30:9, July 2010. ISSN 0730-0301. → pages 38

[47] R. Jürgens, W. Becker, and 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*, 39(2):87–96, 1981. → pages 13

[48] M. Kawato. Feedback-error-learning neural network for supervised motor learning. *Advanced neural computers*, 6(3):365–372, 1990. → pages 36

[49] M. Kawato. Feedback-error-learning neural network for supervised motor learning. *Advanced neural computers*, 6(3):365–372, 1990. → pages 39

[50] M. Kawato. From Understanding the Brain by Creating the Brain towards manipulative neuroscience. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 363(1500):2201, 2008. → pages 1

[51] D. Kirk. *Optimal control theory: an introduction*. Dover Pubns, 2004. → pages 14

[52] D. Kragic and H. Christensen. Survey on visual servoing for manipulation. Technical Report ISRN KTH/NA/P-02/01-SE, Computational Vision and Active Perception Laboratory, Royal Institute of Technology, Stockholm, Sweden, 2002. → pages 39

[53] A. Landgren. A robotic camera platform for evaluation of biomimetic gaze stabilization using adaptive cerebellar feedback. Master's thesis, Linköping University, 2010. → pages v, 50

[54] C. Lawson and R. Hanson. *Solving least squares problems*, volume 15. Society for Industrial Mathematics, 1995. → pages 20

[55] R. Leigh and C. Kennard. Using saccades as a research tool in the clinical neurosciences. *Brain*, 127(3):460, 2004. → pages 11, 33

[56] R. Leigh and D. Zee. *The neurology of eye movements*. Oxford Univ Pr, 1999. → pages 2, 5, 6, 8, 9, 11, 13

[57] A. Lenz, S. Anderson, A. Pipe, C. Melhuish, P. Dean, and J. Porrill. Cerebellar-inspired adaptive control of a robot eye actuated by pneumatic artificial muscles. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(6):1420–1433, 2009. → pages 12

[58] A. Lenz, S. R. Anderson, A. Pipe, C. Melhuish, P. Dean, and J. Porrill. Cerebellar-inspired adaptive control of a robot eye actuated by pneumatic arificial muscles. *IEEE Transactions on Systems, man and cybernetics*, 39 (6):1420–1433, December 2009. → pages 38, 39, 51

[59] M. Lesmana and D. K. Pai. A biologically inspired controller for fast eye movements. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, Shanghai, China, May 2011. IEEE Robotics and Automation Society, IEEE. → pages iv

[60] M. Lesmana, A. Landgren, P.-E. Forssén, and D. K. Pai. Active gaze stabilization for large and fast movements. Submitted (2011). → pages iv

[61] Y. Liang, H. Tyan, S. Chang, H. Liao, and S. Chen. Video stabilization for a camcorder mounted on a moving vehicle. *Vehicular Technology, IEEE Transactions on*, 53(6):1636–1648, 2004. ISSN 0018-9545. → pages 38

[62] H. Lindeman. Anatomy of the otolith organs. *Advances in oto-rhino-laryngology*, 20:405, 1973. → pages 5

[63] L. Ljung and E. Ljung. *System identification: theory for the user*, volume 11. Prentice-Hall Englewood Cliffs, NJ, 1987. → pages 22

[64] L. Ljung, M. Morf, and D. Falconer. Fast calculation of gain matrices for recursive estimation schemes. *International Journal of Control*, 27(1): 1–19, 1978. → pages 22

[65] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2*, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc. → pages 45

[66] L. Manfredi, E. Maini, P. Dario, C. Laschi, B. Girard, N. Tabareau, and A. Berthoz. Implementation of a neurophysiological model of saccadic eye movements on an anthropomorphic robotic head. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 438–443. IEEE, 2007. ISBN 142440200X. → pages 12

[67] D. Marr. A theory of cerebellar cortex. *The journal of physiology*, 202(2): 437, 1969. → pages 36

[68] G. Metta. *Babyrobot: A study into sensori-motor development*. PhD thesis, University of Genoa, LIRA-Lab, DIST, Genoa, Italy, 2000. → pages 39

[69] F. Miles and S. Lisberger. Plasticity in the vestibulo-ocular reflex: a new hypothesis. *Annual Review of Neuroscience*, 4(1):273–299, 1981. → pages 9

[70] C. Morimoto and R. Chellappa. Fast electronic digital image stabilization for off-road navigation. *Real-Time Imaging*, 2(5):285–296, 1996. ISSN 1077-2014. → pages 38

[71] D. Murray, P. McLauchlan, I. Reid, and P. Sharkey. Reactions to peripheral image motion using a head/eye platform. In *Proc. 4th Intl Conf. on Computer Vision, Berlin*, pages 403–411. → pages 12

[72] Y. Nakabo, M. Ishikawa, H. Toyoda, and S. Mizuno. 1ms column parallel vision system and it's application of high speed target tracking. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*. → pages 39

[73] O. Nelles. *Nonlinear system identification: from classical approaches to neural networks and fuzzy models*. Springer Verlag, 2001. → pages 18

[74] J. Neubert and N. Ferrier. Direct mapping of visual input to motor torques. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 4, pages 634–638. IEEE, 2006. ISBN 0769525210. → pages 13

[75] J. Neubert and N. Ferrier. Motion generation through biologically-inspired torque pulses. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4157–4162. IEEE, 2010. → pages 13

[76] Nikon. Nikon vr. http://imaging.nikon.com/products/imaging/technology/vr/, December 2010. → pages 38

[77] J. Nocedal and S. Wright. *Numerical optimization*. Springer verlag, 1999. → pages 23

[78] K. Ohnishi, M. Shibata, and T. Murakami. Motion control for advanced mechatronics. *IEEE/ASME Transactions on mechatronics*, 1(1):56–67, 1996. → pages 12

[79] L. Optican. Dynamic eye plant models and the control of eye movements Christian Quaia. *Strabismus*, 927(3972/03):16–00, 2003. → pages 6

[80] G. Otten, T. De Vries, J. Van Amerongen, A. Rankers, and E. Gaal. Linear motor motion control using a learning feedforward controller. *IEEE/ASME transactions on mechatronics*, 2(3):179–187, 1997. → pages 12

[81] F. Panerai, G. Metta, and G. Sandini. Learning visual stabilization reflexes in robots with moving eyes. *Neurocomputing*, 48:323–337, 2002. → pages 38, 39

[82] K. D. Pfann, E. L. Keller, and J. M. Miller. New models of the oculomotor mechanics based on data obtained with chronic muscle force transducers. *Annals of biomedical engineering*, 23(4):346–58, 1995. ISSN 0090-6964. → pages 6

[83] C. Quaia and L. Optican. Dynamic eye plant models and the control of eye movements. *Strabismus*, 11(1):17–31, 2003. → pages 6

[84] T. Raphan, V. Matsuo, and B. Cohen. Velocity storage in the vestibulo-ocular reflex arc (vor). *Experimental Brain Research*, 35(2): 229–248, 1979. → pages 6

[85] L. Resende, J. Romano, and M. Bellanger. A fast least-squares algorithm for linearly constrained adaptive filtering. *Signal Processing, IEEE Transactions on*, 44(5):1168–1174, 1996. → pages 23

[86] D. Robinson. The mechanics of human saccadic eye movement. *The Journal of Physiology*, 174(2):245, 1964. → pages 6, 18, 25

[87] D. Robinson. Models of the saccadic eye movement control system. *Biological Cybernetics*, 14(2):71–83, 1973. → pages 6

[88] D. Robinson. Oculomotor control signals. *Basic mechanisms of ocular motility and their clinical implications*, 24:337–374, 1975. → pages 13, 15

[89] D. Robinson, D. O'meara, A. Scott, and C. Collins. Mechanical components of human eye movements. *Journal of Applied Physiology*, 26 (5):548, 1969. → pages 6

[90] E. Schneider, S. Kohlbecher, K. Bartl, F. Wallhoff, and T. Brandt. Experimental platform for wizard-of-oz evaluations of biomimetic active vision in robots. In *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*, pages 1484–1489. IEEE, 2009. → pages 39

[91] E. Schneider, T. Villgrattner, J. Vockeroth, K. Bartl, S. Kohlbecher, S. Bardins, H. Ulbrich, and T. Brandt. Eyeseecam: An eye movement–driven head camera for the examination of natural visual exploration. *Annals of the New York Academy of Sciences*, 1164(1): 461–467, 2009. ISSN 1749-6632. → pages 39

[92] P. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31:1–10, 1966. ISSN 0033-3123. URL http://dx.doi.org/10.1007/BF02289451. 10.1007/BF02289451. → pages 48

[93] F. Schumann, J. Conradt, J. Vockeroth, K. Bartl, E. Schneider, and P. Konigc. Eye–head coordination during free exploration in human and cat. 2008. → pages 39

[94] C. Scudder. A new local feedback model of the saccadic burst generator. *Journal of Neurophysiology*, 59(5):1455, 1988. → pages 13

[95] J. Shi and C. Tomasi. Good features to track. In *Proceedings 1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, 1994. → pages 45

[96] T. Shibata, S. Vijayakumar, J. Conradt, and S. Schaal. Biomimetic oculomotor control. *Adaptive Behaviour*, 9(3-4):189–207, 2001. → pages 38, 39, 45

[97] T. Shibata, S. Vijayakumar, J. Conradt, and S. Schaal. Biomimetic oculomotor control. *Adaptive Behavior*, 9(3-4):189, 2001. ISSN 1059-7123. → pages 12

[98] S. Sklavos, J. Porrill, C. R. S. Kaneko, and P. Dean. Evidence for wide range of time scales in oculomotor plant dynamics: implications for models of eye-movement control. *Vision research*, 45(12):1525–42, June 2005. ISSN 0042-6989. doi:10.1016/j.visres.2005.01.005. → pages 6

[99] R. Snell and M. Lemp. *Clinical anatomy of the eye*. Wiley-Blackwell, 1998. → pages 5

[100] S. Song, J. Lim, S. Baek, and K. Sung. Gauss Newton variable forgetting factor recursive least squares for time varying parameter tracking. *Electronics letters*, 36:988, 2000. → pages 7

[101] P. Tomei. Adaptive PD controller for robot manipulators. *IEEE transactions on robotics and automation*, 7(4):565–570, 1991. ISSN 1042-296X. → pages 12

[102] G. Turk and J. OBrien. Variational implicit surfaces. *Techical Reports GIT-GVU-99-15*, 1999. → pages 18

[103] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings 2001 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages I–511 – I–518 vol.1, Kauai, Hawaii, 2001. doi:10.1109/CVPR.2001.990517. → pages 55

[104] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 2001. → pages 55

[105] C. Weir. Proprioception in extraocular muscles. *Journal of neuro-ophthalmology*, 26(2):123, 2006. ISSN 1070-8022. → pages 11

[106] K. Welke, M. Przybylski, T. Asfour, and R. Dillmann. Kinematic calibration for saccadic eye movements. *Institute for Anthropomatics, University of Karlsruhe (TH), Tech. Rep.*, 2008. → pages 12

[107] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *Proceedings 2010 IEEE Conference on Computer*

Vision and Pattern Recognition (CVPR), San Francisco, USA, June 2010. IEEE Computer Society, IEEE. → pages 38, 44

[108] D. Wolpert, R. Miall, and M. Kawato. Internal models in the cerebellum. *Trends in Cognitive Sciences*, 2(9):338–347, 1998. → pages 36

[109] Y. Yildiz, A. Sabanovic, and K. Abidi. Sliding-mode neuro-controller for uncertain systems. *Industrial Electronics, IEEE Transactions on*, 54(3): 1676–1685, 2007. ISSN 0278-0046. → pages 12

[110] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11): 1330–1334, 2000. → pages 47

[111] Y. Zhu and X. Li. Recursive least squares with linear constraints. In *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, volume 3, pages 2414–2419. IEEE, 1999. → pages 23

Appendix A

Basis Function for Approximating Physically Plausible Models

We want to utilize a basis function that ensures physical plausibility in the learned model. This requires a monotonically increasing basis function as shown in Section 3.4.1. A derivation for one such basis function for the spring forces from a hat RBF is presented here. The derivation for damping forces follow similarly.

We first define the displacement variable r_k from c_k, the centre of the k'th RBF.

$$r_k = \theta - c_k \, .$$

Starting from a hat function for the RBF used to approximate the force gradient,

$$\tilde{B}_k(\theta) = \begin{cases} 1 - |r_k| & \text{if } |r_k| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

we can integrate this to obtain:

$$B_k(\theta) = \begin{cases} 0 & \text{if } r_k < -1 \\ r_k + \frac{r_k^2}{2} + \frac{1}{2} & \text{if } -1 \leq r_k < 0 \\ r_k - \frac{r_k^2}{2} + \frac{1}{2} & \text{if } 0 \leq r_k < 1 \\ 1 & \text{if } r_k \geq 1 \end{cases}$$

The resulting basis function is monotonically increasing and we can guarantee the approximation it generates, $F_{\text{spring}} = \sum_k^m w_k B_k$, to be also monotonically increasing by constraining the weights w_k to be positive for all k.

Although we have shown a derivation for a monotonically increasing basis function from a standard RBF to show a connection with using the classical RBF approach in the gradient, actually any monotonically increasing basis function could be used to the same end.

Appendix B

Online Constrained Least Squares Formulation With Exponential Forgetting Factor

To incorporate an exponential forgetting factor into the online constrained least squares algorithm, we first apply it to the standard constrained least squares form:

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} & \|\Lambda_{t+1}(A_{t+1}\mathbf{x} - \mathbf{b_{t+1}})\| \\ \text{subject to} & \mathbf{x} \geq 0\,. \end{array}$$

Here, the subscript indicates the measurement index such that

$$A_{t+1} = \left[\begin{array}{c} A_t \\ \hline \mathbf{a}_{t+1}^T \end{array}\right]$$

$$\mathbf{b}_{t+1}^T = \left[\mathbf{b}_t^T \;\middle|\; b_{t+1}\right]$$

and Λ_{t+1} is the diagonal matrix of exponential forgetting factor λ:

$$\Lambda_{t+1} = \begin{bmatrix} \lambda^t & & & & \\ & \lambda^{t-1} & & & \\ & & \ddots & & \\ & & & \lambda & \\ & & & & 1 \end{bmatrix} = \left[\begin{array}{c|c} \lambda\Lambda_t & 0 \\ \hline 0 & 1 \end{array}\right].$$

We can also define

$$\begin{aligned} \tilde{A}_{t+1} &= \Lambda_{t+1} A_{t+1} \\ \tilde{\mathbf{b}}_{t+1} &= \Lambda_{t+1} \mathbf{b}\,. \end{aligned}$$

Rewritting this as a QP,

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \frac{1}{2}\mathbf{x}^T H_{t+1}\mathbf{x} + \mathbf{c}_{t+1}^T \mathbf{x} \\ \text{subject to} \quad & \mathbf{x} \geq 0\,, \end{aligned}$$

we get, similar to Equation 3.11 ,

$$\begin{aligned} H_{t+1} &= 2\tilde{A}_{t+1}^T \tilde{A}_{t+1} \\ \mathbf{c}_{t+1}^T &= -2\mathbf{b}_{t+1}^T \tilde{A}_{t+1}\,. \end{aligned}$$

Expanding the above expressions, we get the appropriate update rules for the Hessian matrix and linear term:

$$\begin{aligned} H_{t+1} &= 2A_{t+1}^T \Lambda_{t+1}^T \Lambda_{t+1} A_{t+1} \\ &= 2\lambda^2 A_t^T \Lambda_t^2 A_t + 2\mathbf{a}_{t+1}\mathbf{a}_{t+1}^T \\ &= \lambda^2 H_t + 2\mathbf{a}_{t+1}\mathbf{a}_{t+1}^T \\ \mathbf{c}_{t+1}^T &= -2\tilde{\mathbf{b}}_{t+1}^T \tilde{A}_{t+1} \\ &= -2\lambda^2 \tilde{\mathbf{b}}_t^T \tilde{A}_t - 2b_{t+1}\mathbf{a}_{t+1}^T \\ &= \lambda^2 \mathbf{c}_t^T - 2b_{t+1}\mathbf{a}_{t+1}^T\,. \end{aligned}$$