Volterra Modeling of the Human Smooth Pursuit System in Health and Disease

Viktor Bro
Volterra Modeling of the Human Smooth Pursuit System in Health and Disease

Viktor Bro
viktor.bro@it.uu.se

May 2019

Division of Systems and Control
Department of Information Technology
Uppsala University
Box 337
SE-751 05 Uppsala
Sweden

http://www.it.uu.se/

Dissertation for the degree of Licentiate of Philosophy in Electrical Engineering with specialization in Automatic Control

© Viktor Bro 2019
ISSN 1404-5117
Printed by the Department of Information Technology, Uppsala University, Sweden
Abstract

This thesis treats the identification of Volterra models of the human smooth pursuit system from eye-tracking data. Smooth pursuit movements are gaze movements used in tracking of moving targets and controlled by a complex biological network involving the eyes and brain. Because of the neural control of smooth pursuit, these movements are affected by a number of neurological and mental conditions, such as Parkinson’s disease. Therefore, by constructing mathematical models of the smooth pursuit system from eye-tracking data of the patient, it may be possible to identify symptoms of the disease and quantify them. While the smooth pursuit dynamics are typically linear in healthy subjects, this is not necessarily true in disease or under influence of drugs. The Volterra model is a classical black-box model for dynamical systems with smooth nonlinearities that does not require much a priori information about the plant and thus suitable for modeling the smooth pursuit system.

The contribution of this thesis is mainly covered by the four appended papers. Papers I-III treat the problem of reducing the number of parameters in Volterra models with the kernels parameterized in Laguerre functional basis (Volterra-Laguerre models), when utilizing them to capture the signal form of smooth pursuit movements. Specifically, a Volterra-Laguerre model is obtained by means of sparse estimation and principal component analysis in Paper I, and a Wiener model approach is used in Paper II. In Paper III, the same model as in Paper I is considered to examine the feasibility of smooth pursuit eye tracking for biometric purposes. Paper IV is concerned with a Volterra-Laguerre model that includes an explicit time delay. An approach to the joint estimation of the time delay and the finite-dimensional part of the Volterra model is proposed and applied to time-delay compensation in eye-tracking data.
Acknowledgments

I would like to thank my supervisor, Professor Alexander Medvedev for his guidance and support through the last three years. I also want to thank my second supervisor, Dr. Marcus Björk for all his help and for the rewarding discussions.

A big thanks to all of my colleagues at SysCon for teaching me how everything is done around here, from printing to control engineering.

I want to express my gratefulness for everything my family has done for me, from the day I was born until today. Of course, I want to thank the Jansson family as well.

Lastly, I must thank Jessica for the neverending support, encouragement and inspiration. Without you I couldn’t have done any of this.
List of Papers

This thesis is based on the following papers


# Contents

1 **Introduction** 3  
1.1 Overview 3  
1.2 Contributions and outline of papers 4  
1.2.1 Paper I 4  
1.2.2 Paper II 4  
1.2.3 Paper III 5  
1.2.4 Paper IV 5  
1.3 Summary and future research 5  

2 **The human smooth pursuit system** 7  
2.1 The human oculomotor system 7  
2.2 Eye movements and their modeling 7  
2.3 Modeling of smooth pursuit 8  
2.4 Eye movements in neurological diseases and treatments 10  
2.5 Eye trackers 10  
2.6 Visual stimuli 11  
2.7 Experiments and setup 12  

3 **Volterra models** 15  
3.1 Volterra models in continuous time 15  
3.2 Volterra models in discrete time 15  
3.3 Continuous time Volterra-Laguerre models 16  
3.4 Discrete time Volterra-Laguerre models 17  
3.5 Estimation of Volterra models 18  
3.5.1 Least squares estimation 19  
3.5.2 Sparse estimation 20  

Bibliography 23  

**Paper I – Constrained SPICE in Volterra-Laguerre Modeling of Human Smooth Pursuit** 29  
4.1 Introduction 31
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Introduction</td>
<td>51</td>
</tr>
<tr>
<td>5.2 Preliminaries</td>
<td>52</td>
</tr>
<tr>
<td>5.2.1 The human smooth pursuit system</td>
<td>52</td>
</tr>
<tr>
<td>5.2.2 The Volterra model</td>
<td>53</td>
</tr>
<tr>
<td>5.2.3 Laguerre series representation of the Volterra Kernels</td>
<td>53</td>
</tr>
<tr>
<td>5.2.4 The Wiener model</td>
<td>55</td>
</tr>
<tr>
<td>5.2.5 The Particle Filter</td>
<td>56</td>
</tr>
<tr>
<td>5.3 Experiments and data collection</td>
<td>56</td>
</tr>
<tr>
<td>5.4 Wiener modeling of smooth pursuit</td>
<td>57</td>
</tr>
<tr>
<td>5.4.1 Identification of SPS</td>
<td>58</td>
</tr>
<tr>
<td>5.5 Wiener estimation using the particle filter</td>
<td>60</td>
</tr>
<tr>
<td>5.5.1 Synthetic data</td>
<td>60</td>
</tr>
<tr>
<td>5.5.2 Experiments with eye-tracking data</td>
<td>61</td>
</tr>
<tr>
<td>5.6 Conclusions</td>
<td>63</td>
</tr>
<tr>
<td>References</td>
<td>66</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Paper III – Modeling of Human Smooth Pursuit by Sparse Volterra Models with Functionally Dependent Parameters</th>
<th>69</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Introduction</td>
<td>71</td>
</tr>
<tr>
<td>6.2 Theoretical background</td>
<td>74</td>
</tr>
<tr>
<td>6.2.1 The Volterra model</td>
<td>74</td>
</tr>
<tr>
<td>6.2.2 Laguerre functions</td>
<td>75</td>
</tr>
<tr>
<td>6.2.3 The Volterra-Laguerre model</td>
<td>76</td>
</tr>
<tr>
<td>6.2.4 Tools for model sparsity: SPICE and PCA</td>
<td>77</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>6.3</td>
<td>Experiments &amp; data collection</td>
</tr>
<tr>
<td>6.4</td>
<td>Results</td>
</tr>
<tr>
<td>6.4.1</td>
<td>Sparse VL model of smooth pursuit</td>
</tr>
<tr>
<td>6.4.2</td>
<td>PD symptoms in a sparse VL model</td>
</tr>
<tr>
<td>6.4.3</td>
<td>Distinguishing between healthy individuals using a linear VL model</td>
</tr>
<tr>
<td>6.5</td>
<td>Discussion</td>
</tr>
<tr>
<td>6.6</td>
<td>Conclusions</td>
</tr>
<tr>
<td></td>
<td>References</td>
</tr>
<tr>
<td>Paper IV</td>
<td>Identification of Continuous Volterra Models with Explicit Time Delay</td>
</tr>
<tr>
<td>7.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>7.2</td>
<td>Problem formulation</td>
</tr>
<tr>
<td>7.3</td>
<td>The continuous Laguerre functions and polynomials</td>
</tr>
<tr>
<td>7.4</td>
<td>Volterra model with time delay</td>
</tr>
<tr>
<td>7.5</td>
<td>Volterra-Laguerre model</td>
</tr>
<tr>
<td>7.5.1</td>
<td>Time-delay VL model</td>
</tr>
<tr>
<td>7.6</td>
<td>Estimation of VL models</td>
</tr>
<tr>
<td>7.6.1</td>
<td>Second-order VL model with delay</td>
</tr>
<tr>
<td>7.7</td>
<td>The Smooth Pursuit System</td>
</tr>
<tr>
<td>7.7.1</td>
<td>Simulation data</td>
</tr>
<tr>
<td>7.7.2</td>
<td>Experimental data</td>
</tr>
<tr>
<td>7.8</td>
<td>Conclusions</td>
</tr>
<tr>
<td></td>
<td>References</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Overview

This thesis is, on the one hand, concerned with Volterra series and the identification of these. On the other hand, it is also about finding adequate models for ocular movements using data collected by means of an eye tracking camera. The synthesis of these two themes is what connects all work presented henceforth, and it is not an arbitrary choice.

The Volterra series can be compared to the Taylor series, in that we can use it to model nonlinearity. However, while the Taylor series can only handle static functions, the Volterra series is equipped with a memory allowing us to model dynamic behaviour. Thus, the Volterra series is a very flexible model, allowing us to describe nonlinear dynamical input-output systems. The model stems from the work by Italian mathematician Vito Volterra during the late 19th century, and it was formalized by Norbert Wiener in the 1940s [46]. Since then, it has found use in e.g. modeling of biological systems [27], heartbeat dynamics [43], and telecommunications [13].

Eye tracking means recording, directly or indirectly, the position of someones gaze for a period of time. Since the early 20th century, eye tracking has been used as a tool in e.g. reading and psychology research, as well as for clinical assessments. More recently, eye trackers have also been used in areas like computer games [9] and marketing research [44]. Eye tracking can be performed in a number of ways, each with its corresponding strengths and drawbacks. One highly accurate method is to use tight-fitting contact lenses and record the movements of these using magnetic fields, as in [32]. This may however be uncomfortable for the test subject and requires the machinery to generate and measure the surrounding magnetic field. A less intrusive method is so-called electrooculography (EOG), where electrodes
are placed around the eye, measuring the potential difference generated by the retina. With correct calibration, it is then possible to calculate the movement of the eyes. This approach is used clinically [7], although due to drift it is mainly suited to measure blinks and the quick shifting movements known as saccades. Finally, there is the optical technique of eye tracking. This is a non-intrusive method involving video recording of the eye, and extrapolating the gaze position from these recordings. While early experiments used glass screens and video cameras to produce these recordings [12], it is possible today to achieve highly accurate tracking using infrared cameras detecting reflections in the cornea and lens [8]. This makes the approach suitable for tracking of the continuous movements of the gaze called smooth pursuit, the main focus of this thesis.

The smooth pursuit system is complex, and involves not only the eyes, but also the extraocular muscles and several parts of the brain [40]. This of course makes the system vulnerable to neurological impairments due to drugs or alcohol [47], or due to diseases such as schizophrenia [30, 33] or Parkinson’s disease (PD) [10, 11, 16, 17, 18, 25]. It is therefore motivated to research the possibilities of modeling the smooth pursuit systems. By quantifying the symptoms of PD from eye-tracking data, one obtains an aid in diagnosis of the disease and evaluation of the effect of therapy.

1.2 Contributions and outline of papers

1.2.1 Paper I

In this paper, we present an approach to reducing the number of parameters in a Volterra-Laguerre (VL) model. As Volterra models are often overparameterized, this is a way to ensure identifiability and reduce the parameter estimate variance. The Volterra kernels are parameterized in the orthonormal basis of Laguerre functions, and the parameters of the model are estimated using an algorithm for sparse estimation with constraints. Then, functional dependencies are found in the resulting parameter estimates using principal component analysis. The approach is applied to identification of the human smooth pursuit system and yields a reduced model structure with only a small increase in output error.

1.2.2 Paper II

Here, the reduced VL model from Paper I is once again applied to identification of the smooth pursuit system. A Wiener model with a cubic polynomial nonlinearity is then obtained from the Volterra kernels, and a particle filter is used to infer probability distributions of the polynomial coefficients. It is
then shown that this modeling approach is capable of capturing the changes in smooth pursuit dynamics due to PD.

1.2.3 Paper III

In this paper, eye-tracking data are studied with the reduced VL model. The study concerns with whether or not it is feasible to employ smooth pursuit movements for biometric purposes and uses measurements from four healthy test subjects. It seems that while the distributions of the VL parameters differ between individuals, they are heavily overlapping. It is therefore not likely that a single smooth pursuit test can identify an individual.

1.2.4 Paper IV

This paper considers time delay in continuous-time Volterra-Laguerre models. While the VL models are readily capable of representing systems with time delay, the delay is implicit in the kernels. Here, we present an approach to joint estimation of the delay and the coefficients of a delay-free VL model, thus making the time delay explicit. The approach is applied to the identification of smooth pursuit dynamics from eye-tracking data and captures the processing delay in the device for subsequent compensation.

1.3 Summary and future research

This thesis is concerned with identification of the human smooth pursuit system from eye-tracking data using Volterra models. The main objectives have been to find sparse models that accurately describe the dynamics of SP in patients with PD and investigate whether these models can be used to quantify the symptoms of the disease. The number of model parameters to estimate has to be minimized to reduce the estimate variance, the duration of the eye-tracking test, and the excitation degree of the visual stimuli, because challenging stimuli can be perceived as uncomfortable by the patient.

In the future, there are several compelling research topics to address. While we have performed studies relating the effect of Levodopa treatment for PD to the SP movements, we have not yet systematically studied the influence of deep brain stimulation (DBS) treatment. The effect of DBS on the SPS is yet unclear [34], and thus makes an interesting subject for research given the parametric modelling tools developed in the present work. The eye-tracking experiments themselves can be improved, in terms of ease of use and mobility. While it takes now an expert in eye tracking to administer such a test, a user-friendly platform could enable hospital personnel or even the
patients themselves to perform the experiments, for symptom quantification or treatment evaluation.

Furthermore, the influence of other neurological conditions than PD on SP movements is relevant to consider. One hand, it may be possible to discover whether they affect the SPS at all, and if they do, to discern whether the influence is responsive to treatment.
Chapter 2

The human smooth pursuit system

2.1 The human oculomotor system

Three pairs of muscles control the movement of the human eye. These are referred to as the extraocular muscles, and a diagram showing how they connect to the eye is found in Fig. 2.1. The superior and inferior rectus muscles control the up- and downward movement of the eye, and the medial and lateral rectus muscles control the movement toward and away from the midline of the body. The third pair, consisting of the superior and inferior oblique muscles, controls the torsional, or rotational movement of the eye.

2.2 Eye movements and their modeling

One may talk about three qualitatively different types of eye movements. The first type, called fixation, occurs when maintaining the gaze on a single location. Fixations are, contrary to their name, not completely stationary. They are mainly associated with small scale jittering around the fixation point, similar to Brownian motion [23]. A second type of eye movement is the saccade. These are quick movements, getting their name from the French word for “jerk”, used when shifting focus from one point in space to another.

The third type of movement, which is also the focus of this thesis, is the smooth pursuit (SP). While the saccadic motion is ballistic in nature, the SP movement tracks a moving object by matching the angular velocity of the eye with that of the object being tracked [31]. The smooth pursuit must therefore be initiated by some external moving target, and is continuously driven by this movement. The SP movement is thought to be controlled
by a negative feedback loop, acting as a proportional-differential controller \cite{24}, which can explain the angular stationary error with respect to the target being tracked. The maximum angular velocity for the eye during SP movement is around $80 - 100^\circ/s$ \cite{28}. If the target being tracked moves faster than this, the saccadic eye movements take over, in order to catch up with the target.

\section{2.3 Modeling of smooth pursuit}

A simple way of classifying dynamical models is to use three categories referred to as \textit{white box}, \textit{grey box}, and \textit{black box} models \cite{37}.

White box models are based on first principles derived from e.g. physical laws and thus require a great amount of knowledge of the system being modeled. This makes white box models highly reliable and useful, as well as easy to interpret since all parameters have physical (or biological) meaning. However, it is hard to derive them for more complex or vaguely known systems.

Grey box modeling refers to describing systems only partly using physical laws and first principles. These models may also be referred to as \textit{semi-physical}, acknowledging the fact that a subset of the parameters can be assigned some direct physical meaning.

Lastly, black box models employ a general model structure, where the parameters have no physically interpretable meaning. Then the parameter values are then chosen so that the model describes the system as well as possible.

The SP system (SPS) is a complex feedback system comprised of the
Chapter 2. The human smooth pursuit system

eyes, the extraocular muscles, and regions of the brain [40]. Due to this complexity, white-box modeling is infeasible. While biomechanical models of the human eye [45] have made grey-box models of the SPS possible [19], no adequate mathematical description of the neural feedback exists at the moment. The preferred approach is therefore black-box modeling when it comes to quantification of the SPS. Furthermore, while the SP dynamics are typically linear in healthy subjects [19], nonlinear modeling may be necessary when the SPS does not operate under normal conditions, e.g. under the influence of drugs or disease.

Wiener models

The Wiener model is a block-structured black-box model consisting of a linear block, whose output acts as the input of a static nonlinearity. A general MIMO Wiener model with input \( u(t) \) and output \( y(t) \) can then be written in a state space form as

\[
\begin{align*}
\dot{x}(t) &= Ax(t) + Bu(t), \\
y_l(t) &= Cx(t), \\
y(t) &= f(y_l(t)),
\end{align*}
\]

where \( A, B, C \) are matrices of appropriate dimensions, and \( f \) is the static nonlinearity. Note also that if \( y_l \) is a scalar, the nonlinearity can be sometimes inverted. This is not true for a vector-valued \( y_l \).

Wiener models have been used for portraying SP earlier, based on both ARX [16] and Volterra [15] approaches for the linear block. In this thesis, Paper II treats a variant of the Wiener model where the linear part is given by a Volterra model and the nonlinearity is a third order polynomial.

Volterra models

The main focus of this thesis is the modeling of the SPS using Volterra models. The latter is a black-box structure for nonlinear modeling that has been used for a long time in biological applications, see e.g. [26, 27]. Volterra models approximate nonlinear functions in a similar way to the Taylor series, by summing powers of the input. In contrast to the Taylor series, the Volterra model is however capable of describing dynamical systems, and is formed by convoluting kernel functions with powers of the input signal. For example, a continuous time Volterra model can be written as

\[
y(t) = y_0 + \int k_1(\theta_1)u(t - \theta_1) \, d\theta_1 \\
+ \int \int k_2(\theta_1, \theta_2)u(t - \theta_1)u(t - \theta_2) \, d\theta_1 \, d\theta_2 + \ldots,
\]

(2.2)
where $k_1(\cdot), k_2(\cdot), \ldots$ are the Volterra kernels. Evidently, the Volterra model is very flexible and apt for capturing nonlinear behaviour. This comes at the price of overparameterization, resulting in uncertain parameter estimates and poor prediction performance due to overfitting. The latter problem is best alleviated by sparse estimation [22], which effectively eliminates the terms that do not contribute much to the model quality.

### 2.4 Eye movements in neurological diseases and treatments

Since the SPS constitutes a feedback loop involving of the eyes themselves, the extraocular muscles, the optic nerves, and multiple regions of the brain [40], one easily understands that there are numerous conditions that can impair it. Intoxication resulting from consuming of alcohol and other drugs is one cause of such conditions. Other include schizophrenia [30-33], and Parkinson’s disease [10, 14, 16, 17, 18, 25]. These facts motivate research on modeling of the SP movements, as it may contribute to, for example, tools for diagnosis or assessment of treatments. For example, Levodopa is a chemical precursor to the neurotransmitter dopamine, and is used to treat Parkinson’s disease. In Paper II, we provide some evidence that Levodopa treatment for Parkinson’s disease also reverses the smooth pursuit impairments.

### 2.5 Eye trackers

Eye tracking is the practice of measuring the point, direction, or motion of the gaze. The first reported eye tracker consisted of a tight fitting lens attached to a stylus [11], allowing for tracking of the gaze. A similar method involves generating an external magnetic field and using magnetic lenses, resulting in a highly accurate tracking of the eyes movement in three dimensions [32]. These types of eye trackers are highly intrusive and can be uncomfortable for the test subject, which arguably places some strict limits to the task he or she can perform while under observation.

A less intrusive method of eye tracking is electrooculography (EOG). Here, electrodes are placed around the eyes (as shown in Fig. 2.2), so that the potential difference produced by the retina as it turns can be measured. This method thus measures the angular position of the eye and, since it only relies on electric potentials, it can be used even when the eyes are closed, such as when the test subject is asleep. EOG is thus used clinically [7] although, due to drift, it is more suited to measure blinks and saccades, than small fixational movements or smooth pursuit.
Figure 2.2: Electrooculography setup.

Video-based eye tracking is a non-intrusive method of tracking the gaze using camera recordings of the eyes. By tracking the pupil and using reflections in the cornea and lens, it is possible to calculate a gaze direction. With proper calibration, one can then compute the position of the gaze on e.g. a computer screen. This type of eye tracking has been developing quickly during the last decade and, due to its relative ease of use, video-based eye tracking has found application in, for example, usability evaluations of software [35], marketing research [44], and gaming [9]. Recent technological development has furthermore allowed for mobile eye tracking that can even be used in smartphones.

2.6 Visual stimuli

Since the SPS requires a moving target for initiation and sustained activation, one must use some visual stimulus in order to study this type of eye movements. In medical settings, visual stimuli are often simple: the test subject may be asked to track a target with constant or sinusoidally varying velocity [20]. From a viewpoint of dynamical systems theory, this is however unsatisfactory, since it is equivalent to measuring only one point in the frequency response. Moreover, when using nonlinear models such as the Volterra models dealt with in this thesis, the model gain depends on the input amplitude as well as on the frequency, placing further requirements on the excitation degree of the stimuli. Because the SPS is only activated in a bounded intervals of frequency and amplitude, it is however possible to compute visual stimuli that are sufficiently exciting for the nonlinear dynamics in question. Throughout the appended papers, the same two-dimensional visual stimuli have been used. These were designed by solving an optimization problem presented in [16] with an example shown in Fig. 2.3.
2.7 Experiments and setup

In the papers included in this thesis, two video-based eye trackers were used. The first is a SmartEye Pro system [1]. This stationary setup, shown in Fig. 2.4, consists of two IR cameras connected to a desktop computer. The SmartEye tracker is capable of recording at a frequency of 60Hz, and uses a head model to keep track of the eyes during recording. Because the head model must be calibrated with several photos of the face from several angles, in addition to the camera calibration to map gaze direction to screen position, setting this system up may take some time. However, the resulting tracking is robust, and the output file contains information on blinking and tracking quality, apart from the gaze position. A screenshot of the SmartEye interface is shown in Fig. 2.5.

The second eye-tracking setup consists of a portable tracker from The
Chapter 2. The human smooth pursuit system

Figure 2.5: Screenshot from the SmartEye interface showing features of the face model.

Figure 2.6: Portable eye tracker connected to tablet computer.

Eye Tribe [42] connected to a Microsoft Surface tablet computer and is shown in Fig. 2.6. This setup also features two cameras recording at 60Hz, however of a smaller size and included in a single casing. Furthermore, no head modeling is performed thus enabling quick camera calibration done in the provided Eye Tribe interface. This tracker only outputs gaze position and timestamp, and thus cannot produce extra information regarding e.g. blinks or tracking quality. For recording of eye-tracking data using setup, an interface developed at Uppsala University was used [38].
Chapter 3

Volterra models

3.1 Volterra models in continuous time

A smooth, nonlinear, dynamical, multi-input multi-output system with the input \( u(t) \) and output \( y(t) \) can be described by the continuous time Volterra series

\[
y(t) = y_0 + \int k_1(\theta_1)u(t - \theta_1) \, d\theta_1 \\
+ \int\int k_2(\theta_1, \theta_2)u(t - \theta_1)u(t - \theta_2) \, d\theta_1 \, d\theta_2 + \ldots
\]

given by the kernels \( \{k_1(\cdot), k_2(\cdot), \ldots\} \). Defining the continuous Volterra functional of degree \( n \) as

\[
(K_n u)(t) = \int k_n(\theta_1, \ldots, \theta_n)u(t - \theta_1)\cdots u(t - \theta_n) \, d\theta_1 \cdots d\theta_n
\]

(3.1) can be written in the compact form of

\[
y(t) = y_0 + \sum_{n=1}^{\infty} (K_n u)(t).
\]

3.2 Volterra models in discrete time

The Volterra series described by (3.1) can also be formulated in discrete time. Consider input and output signals \( u(k) \) and \( y(k) \) with discrete time indices \( k = 0, 1, \ldots, K \). Then, the discrete Volterra series is

\[
y(k) = y_0 + \sum_{i_1} h_1(i_1)u(k - i_1) + \\
+ \sum_{i_1} \sum_{i_2} h_2(i_1, i_2)u(k - i_1)u(t - i_2) + \ldots
\]

(3.5)
Similar to the continuous case above, we can define the discrete Volterra functionals

\[
(H_n u)(k) = \sum_{i_1, \ldots, i_n} h_n(i_1, \ldots, i_n) u(k - i_1) \cdots u(k - i_n)
\]  

(3.6)

and write the discrete model as

\[
y(k) = y_0 + \sum_{n=1}^{\infty} (H_n u)(k).
\]  

(3.7)

3.3 Continuous time Volterra-Laguerre models

The Volterra kernels \( \{k_1(\cdot), k_2(\cdot), \ldots\} \) are often cumbersome to calculate and, for this reason, they are often parameterized in terms of a suitable functional basis. For kernels \( k_n \in L_2(\mathbb{R}^N_{\geq 0}) \) that are not too oscillative, the multidimensional Laguerre functions provide such a basis \[29\]. Furthermore, if we assume the kernels to be separable, then the conventional Laguerre functions can be used as a basis for expansion of the kernels.

In the Laplace domain, we can express the Laguerre functions as

\[
\mathcal{L}\{l_k(t)\} = \ell_k(s) = \frac{\sqrt{2p}}{s + p} \left( \frac{s - p}{s + p} \right)^k,
\]  

(3.8)

where the constant \( p > 0 \) is the Laguerre parameter. The functions \( \ell_k(s), k = \{0, \infty\} \) constitute a complete orthonormal basis in \( \mathbb{H}_2 \) with respect to the inner product

\[
\langle W, V \rangle = \frac{1}{2\pi i} \int_{-\infty}^{\infty} W(s) V(-s) \, ds.
\]  

(3.9)

The \( k \)-th Laguerre coefficient of \( W(s) \in \mathbb{H}_2 \) is evaluated as a projection of \( W(s) \) onto \( \ell_k(s) \)

\[
w_k = \langle W, \ell_k \rangle,
\]

where the set \( \{w_k\}_{k=0,1,\ldots} \) is referred to as the Laguerre spectrum of \( W(s) \). In the time domain, the Laguerre functions \( l_k(t) = \mathcal{L}^{-1}\{\ell_k(s)\}, k = \{0, \infty\} \) yield an orthonormal basis in \( L_2[0, \infty) \).

Expressing the \( n \)-th kernel function \( k_n \) in terms of Laguerre functions, we get

\[
k_n(\theta_1, \ldots, \theta_n) = \sum_{j_1=0}^{\infty} \cdots \sum_{j_n=0}^{\infty} \gamma_n(j_1, \ldots, j_n) l_{j_1}(\theta_1) \cdots l_{j_n}(\theta_n),
\]  

(3.10)
where we will refer to the coefficients $\gamma_n(j_1, \ldots, j_n)$ as the Volterra-Laguerre (VL) coefficients. Now consider the convolution integral between the input signal $u(t)$ and the $j$-th Laguerre function

$$
\psi_j(t) = \int_0^t l_j(\theta_n)u(t - \theta) \, d\theta.
$$

Using this integral, the Laguerre filter output, combined with (3.3), we obtain an expression for the Volterra functionals in the Laguerre basis

$$
(K_n u)(t) = \sum_{j_1=0}^{\infty} \cdots \sum_{j_n=0}^{\infty} \gamma_n(j_1, \ldots, j_n) \psi_{j_1}(t) \cdots \psi_{j_n}(t).
$$

The full Volterra-Laguerre model corresponding to (3.1) is then

$$
y(t) = y_0 + \sum_{n=1}^{\infty} \sum_{j_1=0}^{\infty} \cdots \sum_{j_n=0}^{\infty} \gamma_n(j_1, \ldots, j_n) \psi_{j_1}(t) \cdots \psi_{j_n}(t).
$$

### 3.4 Discrete time Volterra-Laguerre models

Analogously to the continuous case, we can parameterize the Volterra kernels $h_n$ in an orthogonal basis. For kernels $h_n \in \ell^2[0, \infty)$ that are not excessively oscillative, we can use the discrete Laguerre functions. These functions are given in the $Z$-domain as

$$
\Phi_j(z) = \frac{\sqrt{1 - \alpha z}}{z - \sqrt{\alpha}} \left( \frac{1 - \sqrt{\alpha} z}{z - \sqrt{\alpha}} \right)^j,
$$

where $\alpha \in (0, 1)$ is the discrete Laguerre parameter. The inverse $Z$-transform yields the corresponding time-domain functions

$$
\phi_j(k) = Z^{-1} \left\{ \Phi_j(z) \right\}
$$

$$
= \alpha^{\frac{k-j}{2}} \sqrt{1 - \alpha} \sum_{l=0}^{j} (-1)^l \binom{k}{l} \binom{j}{l} \alpha^{j-l}(1 - \alpha)^l.
$$

Examples of the discrete Laguerre functions are provided in Fig. 3.1. These functions constitute an orthonormal basis in $\ell^2[0, \infty)$ with respect to the inner product

$$
\langle f, g \rangle = \sum_{k=0}^{\infty} f(k)g(k),
$$

and we can acquire the discrete Laguerre spectrum $\{w_j\}_{j=0,1,\ldots}$ of $f$ by projection onto the Laguerre functions $\{\phi_j\}_{j=0,1,\ldots}$ as

$$
w_j = \langle f, \phi_j \rangle.
$$
3.5 Estimation of Volterra models

As in the continuous case of (3.10), we can now write the $n$-th discrete kernel in terms of Laguerre functions

$$h_n(i_1, \ldots, i_n) = \sum_{j_1=0}^{\infty} \cdots \sum_{j_n=0}^{\infty} \gamma_n(j_1, \ldots, j_n) \psi_{j_1}(i_1) \cdots \psi_{j_n}(i_n). \quad (3.19)$$

Denoting the convolution between the input $u(k)$ and the $j$-th discrete Laguerre function, i.e. the discrete Laguerre filter output, as

$$\psi_j(k) = \sum_{k=0}^{i} \phi_j(i) u(k - i), \quad (3.20)$$

the $n$-th discrete Volterra functional can be written as

$$(H_n u)(t) = \sum_{j_1=0}^{\infty} \cdots \sum_{j_n=0}^{\infty} \gamma_n(j_1, \ldots, j_n) \psi_{j_1}(k) \cdots \psi_{j_n}(k). \quad (3.21)$$

The full discrete VL model corresponding to (3.4) is then

$$y(k) = y_0 + \sum_{n=1}^{\infty} \sum_{j_1=0}^{\infty} \cdots \sum_{j_n=0}^{\infty} \gamma_n(j_1, \ldots, j_n) \psi_{j_1}(k) \cdots \psi_{j_n}(k). \quad (3.22)$$

3.5 Estimation of Volterra models

In system identification, the problem at hand is typically estimating the system dynamics from input and output data. In the case with VL models, this amounts to computing the VL coefficients $\gamma$ given $u$ and $y$.

Since infinitely many kernels cannot be computed in practice, (3.3) must be truncated. We define the Volterra degree $N \in \mathbb{N}$ as the highest degree of
Chapter 3. Volterra models

19

the functional present in the series, and write the truncated Volterra series as

\[ y(t) = y_0 + \sum_{n=1}^{N} (K_n u)(t) + \xi(t), \]  

(3.23)

where \( \xi(t) \in \mathbb{R} \) is the error accounting for the information lost in truncation.

For the same practical reason, the number of functions used in the basis expansion must also be limited. We call the highest order of the Laguerre function present in the basis expansion of a kernel the \textit{Laguerre order} and denote it by \( L \). The doubly finite VL model in continuous time is then

\[ y(t) = y_0 + \sum_{n=1}^{N} \sum_{j_1=0}^{L} \cdots \sum_{j_n=0}^{L} \gamma_n(j_1, \ldots, j_n) \psi_{j_1}(t) \cdots \psi_{j_n}(t) + e(t), \]  

(3.24)

where \( e(t) \in \mathbb{R} \) contains both the truncation error \( \xi(t) \) and the error related to the truncation of the Laguerre series expansion.

Note now that the Volterra functionals are symmetric with respect to the kernel indices \( j \), since the multiplication of Laguerre filter outputs is commutative. For VL models of Volterra degree \( N > 1 \), this poses a problem of identifiability. Take for example the coefficients \( \gamma_2(j_1, j_2) \) and \( \gamma_2(j_2, j_1) \). These will correspond to the same product of Laguerre filter outputs, namely \( \psi_{j_1}(t)\psi_{j_2}(t) \). To overcome this problem, a triangular form of the Volterra functionals can then be enforced, where redundant coefficients are set to zero. The VL model then becomes

\[ y(t) = y_0 + \sum_{n=1}^{N} \sum_{j_1=0}^{L} \cdots \sum_{j_n=0}^{L} \gamma_n(j_1, \ldots, j_n) \psi_{j_1}(t) \cdots \psi_{j_n}(t) + e(t). \]  

(3.25)

The triangular form gets its name from the fact that the indices \( j_1, j_2 \) of the coefficients in the second order Volterra kernel correspond to the coefficients of the nonzero elements of a triangular matrix. This is then generalized to the higher order kernels. The number of coefficients in the \( n \):th Volterra functional is then \( \binom{L+n}{n} \) and the total number of coefficients is

\[ N_\Gamma = \binom{L+N+1}{N}. \]

The discrete case here is completely analogous to the continuous case above. This is also true for the following sections on identification methods and, unless otherwise explicitly noted, only the continuous case will be dealt with.

3.5.1 Least squares estimation

Since the VL model is linear in the coefficients \( \gamma_n(\cdot) \), it is possible to estimate these using the ordinary least squares (LS) method. Consider a vector of
\( N_T \) point values of the output, \( Y = [y(0) \ldots y(T)]^T \), and vectors of point values of the Laguerre filter outputs \( \Psi_j = [\psi_j(0) \ldots \psi_j(T)]^T, j = 0, \ldots, L \).

For a VL model of order \( N \), construct the following matrix using the Laguerre convolution products as entries

\[
\Psi_{0:T} = \begin{bmatrix}
1 & \psi_0(0) & \cdots & \psi_0(0) & \psi_0(0)^N & \cdots & \psi_L(0)^N \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
1 & \psi_0(T) & \cdots & \psi_0(T) & \psi_0(T)^N & \cdots & \psi_L(T)^N 
\end{bmatrix}.
\]

Then the parameter vector

\[
\Gamma = [y_0 \gamma_1(0) \ldots \gamma_2(0, 0) \ldots \gamma_N(0, \ldots, 0) \ldots \gamma_N(L, \ldots, L)]^T.
\]

satisfies

\[
Y = \Psi_{0:T} \Gamma + E, \tag{3.26}
\]

where \( E = [e(0) \ldots e(T)]^T \). When \( \Psi_{0:T} \) is of full rank, i.e. \( \text{rank}(\Psi_{0:T}) = (L + N + 1) \), the parameter vector can be recovered as

\[
\hat{\Gamma} = (\Psi_{0:T}^T \Psi_{0:T})^{-1} \Psi_{0:T}^T Y. \tag{3.27}
\]

### 3.5.2 Sparse estimation

As the Laguerre order of a kernel increases, the number of parameters can become quite large, as the number of parameters of the \( n \)-th kernel is of order \( O(L^n) \). However, for many systems, we can assume that their Volterra kernels are defined mainly by some dominating terms in the Laguerre spectra. Thus, one may want to decrease the number of estimated parameters without necessarily decreasing the Laguerre order, to reduce the variance in the parameter estimates without sacrificing model quality. In such cases, sparse estimation [21, 22, 36] may prove to be a practical tool for exclusion of parameters that do not contribute much to the model quality. One particular method for sparse estimation is the SParse Iterative Covariance-based Estimation (SPICE) algorithm [39]. This algorithm does not require any tuning of hyperparameters, unlike the LASSO [41]. However, the LASSO may be found as a special case of the SPICE algorithm [2].

In short, the SPICE algorithm is formulated as follows. First, we reformulate (3.26) as

\[
Y = \begin{bmatrix}
\Psi_{0:T}^T & I_{N_T}
\end{bmatrix} \begin{bmatrix}
\Gamma \\
E
\end{bmatrix} = B \beta, \tag{3.28}
\]

where \( I_{N_T} \) is a \( N_T \times N_T \) identity matrix. Now define weights \( w_k \) as

\[
w_k = \frac{\|b_k\|_2}{\|Y\|_2}, \tag{3.29}
\]
where $b_k$ denotes the $k$:th column of $B$ in (3.28) and $\| \cdot \|_2$ is the Euclidean vector norm. The SPICE estimate of the coefficients $\Gamma$ is then found by solving the linear program

$$\min_{\alpha, \beta} \sum_{i=1}^{N_T+N_T} w_i \alpha_i$$

s.t. $-\alpha_i \leq \beta_i \leq \alpha_i,$

$\alpha_i \geq 0, \ i = 1, \ldots, N_T+N_T,$

$Y = B\beta.$
Bibliography


Title
Constrained SPICE in Volterra-Laguerre Modeling of Human Smooth Pursuit

Authors
Viktor Bro and Alexander Medvedev

Edited version of
Constrained SPICE in Volterra-Laguerre Modeling of Human Smooth Pursuit

Abstract

The Volterra model is a well-established option in nonlinear black-box system identification. However, the estimated model is often over-parametrized. This paper presents an approach to reducing the number of parameters of a Volterra model with the kernels parametrized in the orthonormal basis of Laguerre functions by estimating it with a sparse estimation algorithm subject to constraints. The resulting parameter estimates are scrutinized for parameter redundancy and functional dependence by principal component analysis. The benefits of this approach are illustrated by identifying the human smooth pursuit system. Previous studies have suggested that the Volterra model structure is suitable for modeling the human smooth pursuit system both in health and disease. The data sets are obtained by eye tracking in a study performed on 7 test subjects diagnosed with Parkinson’s disease and 22 healthy control subjects. In terms of output error, the reduced model has similar performance to that of the full model.

4.1 Introduction

Nonlinear system identification has been an actively developing research field for at least two decades, see e.g. [1, 15, 16, 17]. Modeling of nonlinear dynamical systems from data is necessary in design of control and estimation algorithms that operate under significant changes in the plant and the environment and thus clearly reveal the nonlinear nature of real-life processes and systems. Further, nonlinear identification is gaining momentum as a tool of quantifying properties and discerning regulation mechanisms of living organisms. The latter function becomes increasingly important
due to the role that systems biology and computational medicine play in revolutionizing health care.

There are two main approaches to nonlinear system identification with regard to model structure. One is block-oriented modeling, where the plant to be identified is stipulated to possess a certain internal structure, \cite{1}. Wiener and Hammerstein models, that are composed of linear dynamical blocks cascaded with a nonlinear static mapping of the output or input, correspondingly, are popular and practically useful examples of block-oriented modeling. In engineered systems, a block scheme of the system design is readily available and selecting an adequate model topology for system identification is typically a straightforward task. Of course, defining the orders of the linear dynamical block and parameterizing the nonlinearities can still be challenging.

Nonlinear black-box system identification is another approach to estimating models from data. In this case, also the internal topology of the systems is assumed unknown and has to be established from input-output data. Two mathematical paradigms appear to be useful in this respect: Volterra models \cite{13} and artificial neural networks \cite{21}. Both modeling vehicles introduce massive over-parametrization to capture the nonlinear dynamics that results in identifiability issues. Sparse estimation in Volterra models \cite{11,10,19} and pruning in neural networks \cite{20} are commonly enforced to keep model complexity reasonably low.

Applying nonlinear system identification to living organisms is not straightforward. Dynamical variability between subjects of the same species and in the same subject at different time instants is orders of magnitude higher than that one faces in technical systems, where most of the variability can be attributed to measurement noise. It is also often impossible to decouple a biological subsystem from the rest of the organism without losing essential functions, a property creating unavoidable cross-talk between the connected loops.

All biological systems operate in closed loop. The mechanisms underlying biological regulations are often unknown, as well as the topology of the system itself. In this sense, the situation with system uncertainty is reversed compared to that in engineered systems. While in engineered systems there is no uncertainty in control laws since they are readily defined by design, biological control is still not well understood, especially when it is implemented by neural circuits.

The present paper aims at demonstrating how sparse estimation can be exploited for elucidating the structure of a discrete Volterra model with kernels parametrized in the orthonormal basis of Laguerre functions. The human smooth pursuit system (SPS) is selected as a meaningful example of a biological plant to be modeled. The main contribution of the presented study
is twofold. On the one hand, it is suggested that the sparsity-promoting mechanism of a parameter estimation algorithm is to be used for constraining the structural degrees of freedom in Volterra models. On the other hand, the efficacy of the proposed approach is exemplified on a practically important application of SPS modeling that is involved in the areas of biometrics, medical diagnostics, and disability aids.

The paper is composed as follows. First, a brief summary of Volterra-Laguerre models and the sparse parameter estimation algorithm SPICE is provided. Next, the human SPS system and video-based eye tracking are described, followed up by relevant facts regarding the experimental setup. Further, model estimation results from eye tracking data are presented along with their principal component analysis. Finally, some conclusions are drawn.

4.2 Preliminaries

4.2.1 The human smooth pursuit system

There are two ways in which humans shift their gaze: saccades and smooth pursuit. While saccades are quick, episodic movements where the gaze shifts focus from one instant to another, the smooth pursuit system (SPS) allows the gaze to continuously track some object of interest, constantly keeping it within focus.

The SPS is a complex neurally controlled feedback system, involving the eyes, the extraocular muscles, and several parts of the brain [23]. Because of this, the SPS can be affected by for example alcohol and drugs [25], as well as by mental and neurological conditions such as schizophrenia [14] and Parkinson’s disease [3, 9, 5, 6, 8, 12].

4.2.2 The Volterra Model

The Volterra series is a functional expansion of a dynamical, nonlinear, time-invariant system. A discrete system with input $u(k) \in \mathbb{R}$ and output $y(k) \in \mathbb{R}$, $k = 0, \ldots , K - 1$ may be approximated by the truncated Volterra series

$$y(k) = y_0 + \sum_{n=1}^{N} H_n u(k) + e(k),$$

where $e(k) \in \mathbb{R}$ is a noise term, $N \in \mathbb{N}$ is the Volterra order, and

$$H_n u(k) = \sum_{i_1=0}^{\infty} \cdots \sum_{i_n=0}^{\infty} h_n(i_1, \ldots , i_n) u(k - i_1) \cdots u(k - i_n)$$
are the Volterra functionals. The functions $h_n$ are Volterra kernels. In most practical cases, the Volterra kernels are cumbersome to calculate explicitly. Therefore, the kernel functions are usually expanded in terms of some orthogonal basis. For the kernels that are in $\ell^2[0,\infty)$ and not excessively oscillative, the discrete Laguerre functions present a popular choice.

4.2.3 Laguerre series representation of the Volterra Kernels

The $j$:th Laguerre function is defined in the $\mathbb{Z}$-domain as

$$\Phi_j(z) = \frac{\sqrt{1 - \alpha z}}{z - \sqrt{\alpha}} \left( \frac{1 - \sqrt{\alpha} z}{z - \sqrt{\alpha}} \right)^j,$$  \hspace{1cm} (4.3)

where $0 < \alpha < 1$ is the Laguerre parameter. Denote the corresponding functions in the time domain through the inverse $\mathcal{Z}$-transform by $\phi_j(k) = \mathcal{Z}^{-1}\{\Phi_j(z)\}$. The properties of the Laguerre functions are described in detail in e.g. [13]. Furthermore, these functions form an orthonormal basis in $\ell^2[0,\infty)$ so that a function $h(\cdot) \in \ell^2[0,\infty)$ can be unambiguously written as

$$h(k) = \sum_{j=0}^{\infty} \gamma_j \phi_j(k). \hspace{1cm} (4.4)$$

The Volterra functionals can then be parametrized using the Laguerre functions as

$$h_n(i_1, \ldots, i_n) = \sum_{j_1=0}^{\infty} \cdots \sum_{j_n=0}^{\infty} \gamma_n(j_1, \ldots, j_n) \phi_{j_1}(i_1) \cdots \phi_{j_n}(i_n). \hspace{1cm} (4.5)$$

Note that, in practice, only a finite number of basis functions can be reliably calculated. Thus, a truncated Laguerre expansion usually becomes an approximation of the true kernel.

Let $L$ denote the Laguerre order of a truncated series, i.e. the highest order of the included Laguerre functions. Then

$$h_n(i_1, \ldots, i_n) \approx \sum_{j_1=0}^{L} \cdots \sum_{j_n=0}^{L} \gamma_n(j_1, \ldots, j_n) \phi_{j_1}(i_1) \cdots \phi_{j_n}(i_n) \hspace{1cm} (4.6)$$

and the Volterra functionals become

$$H_n u(k) = \sum_{j_1=0}^{\infty} \cdots \sum_{j_n=0}^{\infty} \gamma_n(j_1, \ldots, j_n) \psi_{j_1}(k) \cdots \psi_{j_n}(k), \hspace{1cm} (4.7)$$
where the convolution $\psi_j(k) = (\phi_j * u)(k)$ denotes the Laguerre filter output. The Volterra-Laguerre (VL) model is finally written as

$$y(k) = y_0 + \sum_{n=1}^{N} \sum_{j_1=0}^{L} \cdots \sum_{j_n=0}^{L} \gamma_n(j_1, \ldots, j_n) \prod_{l=1}^{n} \psi_{j_l}(k) + e(k).$$ (4.8)

The kernel functions are symmetric with respect to index, since the Laguerre functions commute. Therefore, many VL coefficients are redundant and do not have to be estimated individually. This reduces the number of coefficients in a Volterra model of order $N$ parametrized in $L$ Laguerre functions from $(L^{N+1} - 1)/(L - 1)$ to

$$N_c = \binom{L+N+1}{N}.$$

### 4.2.4 VL models in linear regression form

Let $\mathbf{c}$ be a vector of all $N_c$ Volterra-Laguerre coefficients, $\mathbf{y} = [y(0), \ldots, y(K-1)]^T$ be the vector of measurements, and $\mathbf{e} = [e(0), \ldots, e(K-1)]^T$ be the corresponding vector of noise terms. Then, a model for the data can be formulated as

$$\mathbf{y} = [\mathbf{\Psi} \ \mathbf{I}_K] \begin{bmatrix} \mathbf{c} \\ \mathbf{e} \end{bmatrix} = \mathbf{B}\beta,$$ (4.9)

where $\mathbf{\Psi} = [\psi_0 \ \psi_1 \ \ldots \ \psi_0^2 \ \psi_0\psi_1 \ \cdots]$ is the regression matrix constructed from the Laguerre filter outputs $\psi_j = [\psi_j(0) \ \ldots \ \psi_j(K-1)]$, and $\beta = [\beta_1 \ \ldots \ \beta_{N_c+K}]^T$. The parameters of the VL model may be estimated using ordinary least squares (LS). However, with a higher Laguerre order follows a higher number of estimands, which in turn results in a higher variance of the parameter estimates. This, logically, demands a higher degree of excitation in the system input $u(t)$, ultimately making it white noise. While being convenient in theoretical development, white noise inputs are infeasible in most practical system identification problems. For instance, in the identification of SPS, white visual stimuli would fail to invoke smooth pursuit in the test subject. Thus, it is of interest to reduce the number of model parameters as much as possible, e.g. by using sparse estimation.

### 4.2.5 SPICE

One method for sparse parameter estimation is SParse Iterative Covariance-based Estimation (SPICE) [22]. While other methods for sparse estimation, e.g. LASSO [24], require proper tuning of some hyperparameters, the SPICE algorithm does not.
The SPICE algorithm can be formulated as follows. Introduce the quantities
\[ w_k = \frac{\|b_k\|_2}{\|y\|_2}, \]
where \( b_k \) denotes the \( k \)th column of \( B \) in (4.9) and \( \| \cdot \|_2 \) is the Euclidean vector norm. The SPICE estimate of the coefficients is then found by solving the linear program

\[
\begin{align*}
\min_{\alpha, \beta} & \sum_{i=1}^{N_c+K} w_i \alpha_i \\
\text{s.t.} & -\alpha_i \leq \beta_i \leq \alpha_i, \\
& \alpha_i \geq 0, \ i = 1, \ldots, N_c + K \\
& y = B\beta.
\end{align*}
\] (4.10)

**Weighted SPICE estimation** Since the Volterra series converge, the coefficient values of higher order kernels decrease in magnitude. When relevant system information is communicated by higher-order terms, it becomes necessary to weigh the higher order coefficients when using the SPICE algorithm, so that these are not "lost" in pursuit of sparsity. The weight \( \omega_i \) for the individual coefficient \( \beta_i \) is introduced in the constraint \( -\alpha_i \leq \omega_i \beta_i \leq \alpha_i \). Thus, to weigh only the quadratic part with some uniform value \( \omega \), let \( \omega_i = \omega \) for coefficients corresponding to the quadratic kernels, and \( \omega_i = 1 \) otherwise.

**Constrained SPICE** If some function is known beforehand to be absent from a Volterra kernel, a constraint may be introduced in the SPICE algorithm, so that the corresponding coefficient is set to zero. Similarly, additional linear constraints in the estimated coefficients, such as positivity of certain terms in the Volterra series, can be added to the linear program to be solved.

### 4.2.6 Principal Component Analysis

Consider \( M \) measurements \( \{x_i, i = 1, \ldots, M\} \in \mathbb{R}^{N_c} \). Principal component analysis (PCA) can be thought of as fitting an ellipsoid to the measurements, and projecting these onto the axes of the ellipsoid, i.e. the principal components. If the data variance along some principal component is small, this component may be omitted, thus reducing the dimensionality of the problem. The PCA can also be used as an exploratory tool for finding functional relationships between data components.

In practice, the principal components are found as follows. Let \( \overline{x}_i \) denote the mean of \( x_i \), and let

\[ \tilde{X} = [x_1 - \overline{x}_1 \ 1 \ \ldots \ x_M - \overline{x}_M \ 1] \] (4.11)
be the mean subtracted data matrix. Consider now the singular value decomposition of \( \tilde{X} \): 
\[
\tilde{X} = U \Sigma W^T,
\]
where \( \Sigma \) contains the singular values of \( \tilde{X} \) and \( W \) contains the eigenvectors of \( \tilde{X}^T \tilde{X} \). The transformation
\[
T = \tilde{X}W = U \Sigma W^T W = U \Sigma
\]
then gives the projection onto the principal components. By deleting rows and columns from \( U \) and \( \Sigma \) and keeping only the ones corresponding to the \( d \) largest singular values, the dimensionality of the data set is reduced. At the same time, the error \( \| \tilde{X} - \tilde{X}_d \|_2^2 \) is minimized. Here, \( \tilde{X}_d \) denotes the reduced dimension data matrix.

### 4.3 Experiments and Data Collection

The experimental setup consisted of a computer screen and a video-based eye tracker from SmartEye AB, Sweden. The eye tracker records the gaze position of the test subject, i.e. the point on the screen where the subject is looking. Test subjects were placed about 50 cm from the computer screen, and a stimulus was displayed. The stimulus consisted of a colored dot moving on a black background. The stimuli, acting as input signals to the SPS, were generated using the method described in [6], and the gaze position was sampled at 60 Hz. The measured gaze position is then considered as the system output. Notice that as nonlinear identification is performed, the stimuli have to provide excitation both in amplitude and frequency. For this application, it implies that the dot has to visit all the areas within the stimulus window and move within a suitable for tracking range of acceleration.

Two different groups of test subjects are considered. One control group consisting of 22 healthy individuals between 50 and 76 years old, and one group of 7 individuals diagnosed with Parkinson’s disease, between 61 and 80 years old. In total 499 measurements from the control group and 144 measurements from the patient group have been collected and used in the present study.

The tests were conducted at CTC (Clinical Trials Consultants AB) Center at the University Hospital in Uppsala, Sweden, between May and August 2015. The study was performed as part of the project "MuSyQ: Multimodal motor symptoms quantification platform for individualized Parkinson’s disease treatment" and is described in detail in [18].

It is important to note that, prior to the first eye-tracking test, each patient received a dose of an anti-PD drug, after an 8 hours long washout. The administered dose was set to 150% of the patient’s usual morning dose of medication. This experimental protocol allows to track the patient’s symptoms, as they transition from off-state, to normal mobility and/or dyskinesia, and back. Thus, the measurements from the patient group exhibit
4.4 VL model reduction using SPICE and PCA

It has been shown that a VL model of Volterra order 2, with Volterra kernels parametrized in the three first Laguerre functions is a feasible model structure for modeling the SPS dynamics \cite{7}. This model may be written as

\[ y(k) = y_0 + \sum_{j=0}^{2} \gamma_1(j) \psi_j(k) + \]
\[ + \sum_{j_1=0}^{2} \sum_{j_2=0}^{2} \gamma_2(j_1, j_2) \psi_{j_1}(k) \psi_{j_2}(k), \]

and is parametrized by altogether 10 coefficients. The output \( y(k) \) is in this case the gaze position and \( \psi(k) \) are the Laguerre filter outputs, using the stimulus position as input signal. The model thus predicts gaze position from a known visual stimuli. As shown in \cite{7}, model (4.13) is significantly overparametrized and the SPICE algorithm can be used to reduce the number of nonzero parameters. Model (4.13) is therefore chosen as a starting point for describing the SPS dynamics of the test subjects described above. The steps outlined next describe how to systematically reduce the number of parameters until a minimal model is acquired.

4.4.1 Finding redundant parameters

Using the model structure described above and estimating the parameters for test subjects by SPICE gives a sparse representation of the dynamics. Yet, due to inter- and intra-individual variability, different sparse representations are obtained for different experimental datasets. To enable effective comparison between healthy controls and PD patients, and to quantify the treatment effect in the latter, a parsimonious model representation that captures the dynamics of all the data sets is sought.

To further investigate the sparse structure of the models, a PCA was performed. For the linear part, 96% of the data variance is described by the two largest principal components. Similarly, for the quadratic part, 93% of the data variance is described by the three largest principal components.
Figure 4.1: Linear coefficients of VL model, estimated by SPICE, projected onto the two largest principal components. There is a pronounced symmetric, triangular structure in the data, indicating redundancy in the parametrization.

**Linear part**  
Fig. 4.1 shows the linear coefficients projected onto the two largest principal components. A symmetric structure is evident in the data, suggesting redundancy in the model parameters. This redundancy can be removed by setting one coefficient in the linear part to zero. Table 4.1 presents the median normalized (with respect to signal norm) mean square error (NMSE) of the model output, in the case when all linear coefficients are estimated, as well as in the three different cases when one linear coefficient in set to zero. The results are similar in all three cases, although for the control group, the NMSE remains approximately unchanged when the coefficient $\gamma_1(2)$ is excluded. Therefore, the third coefficient was decided to be set to zero. Note however, that for the patient group, $\gamma_1(1)$ seems redundant as removing this gives the smallest increase in error. This may stem from differences in the dynamics between healthy individuals and individuals with Parkinson’s disease. Fig. 4.2 shows the coefficient values of the linear part, with this constraint, and demonstrates that the symmetric structure has disappeared with the reduced kernel representation.

**Nonlinear part**  
In the quadratic part, symmetries similar to those in the linear part were not observed. However, only three functions occur often. These functions correspond to the components of the quadratic kernel functions $\phi_0^2$, $\phi_0\phi_2$ and $\phi_2^2$. Three stimuli were used in the eye tracking experiments. With dissimilar excitation properties of the stimuli, they may excite different dynamics of the SPS and in turn result in different sets of
4.4. VL model reduction using SPICE and PCA

Table 4.1: Median error for different model structures.

<table>
<thead>
<tr>
<th>No constraints</th>
<th>Controls</th>
<th>Patients</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\gamma_1(0) = 0$</td>
<td>0.454</td>
<td>0.886</td>
</tr>
<tr>
<td>$\gamma_1(1) = 0$</td>
<td>0.458</td>
<td>0.890</td>
</tr>
<tr>
<td>$\gamma_1(2) = 0$</td>
<td>0.454</td>
<td>0.920</td>
</tr>
<tr>
<td>$\gamma_1(3) = 0$</td>
<td>0.462</td>
<td>0.887</td>
</tr>
</tbody>
</table>

Figure 4.2: Linear coefficients of VL model, estimated by SPICE, with the constraint that $\gamma_1(2) = 0$. The symmetric structure from before has disappeared. Because of the sparsity-promoting nature of the SPICE algorithm, the first linear coefficient, $\gamma_1(0)$, has been set to zero a number of times, which is visible in the data.
Table 4.2: Frequency with which a kernel function is present in the quadratic kernel for a certain stimulus.

<table>
<thead>
<tr>
<th>Function</th>
<th>Stimulus #</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>$\phi_0^2$</td>
<td>0.53</td>
</tr>
<tr>
<td>$\phi_0 \phi_1$</td>
<td>0</td>
</tr>
<tr>
<td>$\phi_0 \phi_2$</td>
<td>0.15</td>
</tr>
<tr>
<td>$\phi_2^2$</td>
<td>0.15</td>
</tr>
<tr>
<td>$\phi_1 \phi_2$</td>
<td>0.08</td>
</tr>
<tr>
<td>$\phi_2^2$</td>
<td>0.69</td>
</tr>
</tbody>
</table>

Laguerre functions in the estimated kernels of the sparse model. However, as the data in Table 4.2 indicate, the choice of stimuli set does not seem to have had any impact on the structure of the quadratic term of the VL model. That is, there is no significant difference in excitation ability between the different stimuli.

Furthermore, Fig. 4.3 shows that when a weight $\omega$ is introduced on the quadratic kernel in the SPICE algorithm, the probability of a function being present in the quadratic kernel increases as $\omega$ decreases. The weight acts as a threshold on the quadratic kernel functions and, when the threshold is lowered, the relation between the functions is intact. When the weight is lowered even further, to $\omega = 10^{-4}$, all functions are present in the quadratic kernel, every time. This strategy of using weights in the SPICE algorithm may be applied to get an idea of which functions are significant to the model. In the present case, the three kernel functions $\phi_0^2$, $\phi_0 \phi_2$, and $\phi_2^2$ are the most significant, followed by $\phi_1^2$. This is in line with the results presented in Table 4.2. Consequently, the model can be reduced further, to contain only the functions $\phi_0^2$, $\phi_0 \phi_2$ and $\phi_2^2$ in the quadratic kernel.

### 4.4.2 Finding and exploiting relationships between coefficients

While only three functions are present in the quadratic part, there is a linear relationship between the coefficient values, as seen in Fig. 4.4. This relation is estimated using linear regression to be

$$
\gamma_2(0, 2) = -1.81468 \cdot \gamma(0, 0),
$$

$$
\gamma_2(2, 2) = 0.938454 \cdot \gamma(0, 0),
$$

which can also be seen in the figure. Using this relation further reduces the number of coefficients that need to be estimated, while not introducing much modeling error.
Figure 4.3: Occurrence of functions in the quadratic VL kernel, using different weights in SPICE. The same functions are present in the control group and in the patient group, and the relations between functions are intact across weights. The most commonly occurring functions are $\phi_0^2$, $\phi_0 \phi_2$ and $\phi_2^2$.

Figure 4.4: Coefficients of the quadratic part of the reduced VL model. There is a clear linear relationship between the coefficient values, indicated by the dotted black line.
4.4.3 A minimal model

The final minimal model contains only four coefficients to be estimated, \( \{y_0, \gamma_1(0), \gamma_1(1), \gamma_2(0, 0)\} \), and can be written as

\[
y(k) = y_0 + \gamma_1(0)\psi_0(k) + \gamma_1(1)\psi_1(k) + \gamma_2(0, 0) \left( \psi_0^2(k) + \theta_{02}\psi_0(k)\psi_2(k) + \theta_{22}\psi_2^2(k) \right),
\]

where \( \theta_{02} = -1.81468 \) and \( \theta_{22} = 0.938454 \), for both the patient and control group.

Figure 4.5: Left: Distribution of coefficient values for the linear part of the final model, for both the control and patient group. Right: Distribution of the quadratic part coefficient for both control and patient group.

The distributions of the estimated coefficients of the linear part, \( \gamma_1(0) \) and \( \gamma_1(1) \), as well as the distribution of the parameter in the quadratic part, \( \gamma_2(0, 0) \), are shown in Fig. 4.5. The distributions for the control and patient groups have approximately the same mode, but the patient group distribution is wider. This could be due to the wider range of smooth pursuit dynamics being present in the patient group because of the design of the study. In a similar way, the distributions of the linear parameters are different, though, as expected, largely overlapping.

The output NMSE, for the final model as well as for the full model with parameters estimated using ordinary LS and SPICE, are approximately log-normal distributed. This can be seen in the normal probability plots for the logarithm of the errors shown in Fig. 4.6. A comparison of the fitted log-normal error distributions is provided in Fig. 4.7. For the control group, the difference in error is very small; the mode of the fitted distributions for the full, LS estimated, model and the reduced model are 0.26 and 0.29. For
the patient group the difference is slightly larger, with corresponding modes at 0.53 and 0.75.

4.5 Conclusions

While the black-box Volterra-Laguerre model structure is known to be suitable for modeling of nonlinear systems, it suffers from over-parametrization. Sparse estimation introduces a way of reducing the number of model parameters allowing at the same time for weights and constraints on the parameters. The resulting model structure can be scrutinized for redundancy and functional dependance between the parameter estimates by e.g. principal component analysis.

The efficacy of the approach is illustrated on a real world example, the human smooth pursuit system, where significant simplifications have been made in terms of model complexity, reducing the number of parameters from 10 to 4. Meanwhile, the increase in error is small, indicating that the informative parts of the model are intact and only superfluous terms due to overparametrization are eliminated. A log-normal distribution of the modelling error indicates that the model quality cannot be improved given the data.
Figure 4.7: Distribution of errors using three different model structures and estimation schemes. *LS, no constraints* indicates the case where all 10 parameters were estimated using least squares; *SPICE* indicates the case where the SPICE algorithm was used to get a sparse estimate of the parameters, without constraints, and *LS, reduced model* indicates the case where least squares was used to estimate only 4 parameters of the model.
References


Title
Nonlinear Dynamics of the Human Smooth Pursuit System in Health and Disease: Model Structure and Parameter Estimation

Authors
Viktor Bro and Alexander Medvedev

Edited version of
Abstract

Oculomotor tests (OMT) are administered to quantify symptoms in neurological and mental diseases. Eye movements in response to displayed visual stimuli are registered by a digital video-based eye tracker and processed. Stimuli of simple signal form, e.g. sine waves, are traditionally used in medical practice to test the performance of the oculomotor system in smooth pursuit (SP). The calculated SP gain and the phase shift at the frequency in question are then presented as the test outcome. This paper revisits the problem of quantifying the SP dynamics from eye-tracking data by means of nonlinear system identification. First, a sparse Volterra-Laguerre (VL) model is estimated from an OMT with sufficiently exciting (in frequency and amplitude) stimuli. Then the structure and initial parameter estimates of a polynomial Wiener model (WM) are obtained from the kernel estimates of the VL model. Finally, the parameter distributions of the WM are inferred by a particle filter (PF). In the proposed approach, the performance of the PF is improved by the individualized sparse model structure. Experimental data show that the latter captures the alternations in the SP dynamics due to aging and in Parkinson’s disease.

5.1 Introduction

The particle filter (PF) algorithms have become a state-of-the-art technology in nonlinear estimation and, in particular, system identification [17]. Compared to the Extended Kalman Filter (see e.g. [12]), a PF does not approximate the system equations, but rather utilizes a particle set to capture the conditional distribution functions involved in the nonlinear state estimation problem. Further, the PF is not limited to stochastic variables with single-mode distribution since the particle set estimates the distribution and there is no need in reducing it to a point estimate, i.e. the expectation.

Despite the methodological benefits, the PF poses at least two implementation problems: First, the algorithm is computationally demanding and does not scale well when the number of estimated states grows. This shortcoming can be somehow alleviated by parallelization of the algorithm. Linear speed-up in the number of parallel cores can be achieved in several PF algorithms executed on a standard multicore platform, [16]. Second, the PF relies on a perfect knowledge of the process model and attempts to capture model uncertainty as an effect of noise. Notice also that the standard PF algorithms lack explicit feedback and substitute it by resampling procedures. In e.g. the EKF, model uncertainty can be attenuated by the output error feedback.

A model structure is typically obtained in nonlinear identification from
5.2 Preliminaries

5.2.1 The human smooth pursuit system

Smooth pursuit and saccades are the two main ways that humans shift their gaze in. Saccades are quick, episodic movements, while the SPS allows the gaze to continuously track some object of interest and keep it within focus. The SPS is a complex neurally controlled feedback system involving the eyes, the extraocular muscles, and several parts of the brain. The
SPS performance may thus be affected if any of these parts are compromised. Common examples include alcohol and drugs [21], as well as mental conditions, such as schizophrenia [15], and neurological conditions, e.g. Parkinson’s disease [4, 8, 9, 11, 10, 6, 13].

5.2.2 The Volterra model

The Volterra series is a functional expansion of a dynamical, nonlinear, time-invariant system. A discrete system with input \( u(k) \in \mathbb{R} \) and output \( y(k) \in \mathbb{R} \), \( k = 0, \cdots, K - 1 \) may be approximated by the truncated Volterra series

\[
y(k) = y_0 + \sum_{n=1}^{N} H_n u(k) + e(k),
\]

(5.1)

where \( e(k) \in \mathbb{R} \) is the noise (or approximation error) term, \( N \in \mathbb{N} \) is the Volterra order, and

\[
H_n u(k) = \sum_{i_1=0}^{\infty} \cdots \sum_{i_n=0}^{\infty} h_n(i_1, \cdots, i_n) u(k - i_1) \cdots u(k - i_n)
\]

(5.2)

are the Volterra functionals. The functions \( h_n \) are called Volterra kernels and cumbersome to calculate explicitly. Therefore, the kernels are usually expanded in a functional basis. If the kernels \( h_n \in \ell^2[0, \infty) \) and are not excessively oscillative, the discrete Laguerre functions can be used.

5.2.3 Laguerre series representation of the Volterra Kernels

The \( j \):th Laguerre function is defined in the \( \mathbb{Z} \)-domain as

\[
\Phi_j(z) = \frac{\sqrt{1 - \alpha z}}{z - \sqrt{\alpha}} \left( \frac{1 - \sqrt{\alpha z}}{z - \sqrt{\alpha}} \right)^j,
\]

(5.3)

where \( 0 < \alpha < 1 \) is the Laguerre parameter. The inverse \( \mathbb{Z} \)-transform yields the corresponding time-domain functions \( \phi_j(k) = \mathcal{Z}^{-1} \{ \Phi_j(z) \} \). These functions form an orthonormal basis in \( \ell^2[0, \infty) \) so that any function \( h(k) \in \ell^2[0, \infty) \) can be unambiguously written as a Laguerre series

\[
h(k) = \sum_{j=0}^{\infty} \gamma_j \phi_j(k).
\]

(5.4)

The Volterra functionals can then be parametrized as

\[
h_n(i_1, \cdots, i_n) = \sum_{j_1=0}^{\infty} \cdots \sum_{j_n=0}^{\infty} \gamma_n(j_1, \cdots, j_n) \phi_{j_1}(i_1) \cdots \phi_{j_n}(i_n).
\]

(5.5)
Only a finite number of $\phi_j(k)$ can be reliably calculated and a truncated Laguerre series of the true kernel is employed instead. A truncated series of the Laguerre order $L$ is then

$$h_n(i_1, \ldots, i_n) \approx \sum_{j_1=0}^L \cdots \sum_{j_n=0}^L \gamma_n(j_1, \ldots, j_n) \phi_{j_1}(i_1) \cdots \phi_{j_n}(i_n), \quad (5.6)$$

and the Volterra functionals become

$$H_n u(k) = \sum_{j_1=0}^\infty \cdots \sum_{j_n=0}^\infty \gamma_n(j_1, \ldots, j_n) \psi_{j_1}(k) \cdots \psi_{j_n}(k), \quad (5.7)$$

where $\psi_j(k) = (\phi_j * u)(k) = \sum_{l=0}^k \phi_j(l) u(k - l)$ denotes the Laguerre filter output and $\cdot * \cdot$ is the convolution operator. The VL model is finally written as

$$y(k) = y_0 + \sum_{n=1}^N \sum_{j_1=0}^L \cdots \sum_{j_n=0}^L \gamma_n(j_1, \ldots, j_n) \prod_{l=1}^n \psi_{j_l} + e(k). \quad (5.8)$$

Note however that the kernel functions are symmetric with respect to index, since the Laguerre functions commute. Many VL coefficients are therefore redundant and cannot be estimated individually. This reduces the number of coefficients in a Volterra model of order $N$ parametrized in $L$ Laguerre functions from $(L^{N+1} - 1)/(L - 1)$ to

$$N_c = \binom{L + N + 1}{N}.\quad$$

Now, a linear VL model (i.e. $N = 1$) in state space is

$$\begin{align*}
\psi(k + 1) &= F \psi(k) + G u(k), \\
y(k) &= c^T \psi(k) + e(k),
\end{align*} \quad (5.9)$$

where $c = [\gamma_1(0) \quad \gamma_1(1) \quad \ldots]$ is a vector of VL parameters, $\psi(k) = [\psi_0(k) \quad \psi_1(k) \quad \ldots]$ and

$$F = \begin{bmatrix}
\sqrt{\alpha} & 0 & 0 \\
1 - \alpha & \sqrt{\alpha} & 0 \\
-\sqrt{\alpha}(1 - \alpha) & 1 - \alpha & \sqrt{\alpha} \\
\sqrt{\alpha^2(1 - \alpha)} & -\sqrt{\alpha}(1 - \alpha) & 1 - \alpha & \sqrt{\alpha} \\
\vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix}, \quad (5.10)$$

$$G = \sqrt{1 - \alpha} \begin{bmatrix}
1 & -\sqrt{\alpha} & \sqrt{\alpha^2} & -\sqrt{\alpha^3} & \ldots
\end{bmatrix}^T. \quad (5.11)$$
5.2.4 The Wiener model

The Wiener model (WM) is a block-structured model comprising a linear dynamic part cascaded with a static output nonlinearity. Consider a WM with the output polynomial nonlinearity

\[ f_m(x) = x + d_2 x^2 + \cdots + d_m x^m \]

\[ \psi_L(k+1) = F_L \psi_L(k) + G_L u(k), \]
\[ y_l(k) = c^T \psi_L(k) + \varepsilon(k), \]
\[ y(k) = f_m(y_l(k)), \tag{5.12} \]

where \( \psi_L(k) \), \( F_L \) and \( G_L \) are truncated versions of \( \psi(k) \), \( F \) and \( G \), and \( y_l \) is the output from the linear part of the model. The method of [7] for estimating the VL coefficients as well as the coefficients of (5.12) is recapitulated below.

The output in (5.12) can be expanded as

\[ y(k) = c^T \psi_L(k) + d_2 (c^T \psi_L(k))^2 + \cdots + d_m (c^T \psi_L(k))^m + g(\varepsilon(k)). \tag{5.13} \]

This recasts in regressor form as

\[ y = \phi^T \gamma + g(\varepsilon), \]

where

\[ \phi = \begin{bmatrix} 1 \\ \varphi_0 \\ \varphi_1 \\ \varphi_2 \\ \cdots \\ \varphi_N \end{bmatrix}, \]
\[ \gamma = \begin{bmatrix} c^T \\ d_2 (c \otimes c)^T \\ d_3 ((c \otimes c) \otimes c)^T \cdots \end{bmatrix}, \tag{5.14} \]

\( \otimes \) denotes the Kronecker product and \( \psi_L = [\psi_L(1) \ \psi_L(2) \ \cdots] \). Observing that \( \varphi_0^T \gamma_0 = d_n (\varphi_1^T \gamma_1)^n \), a scheme for estimating the linear coefficients \( c \) as well as the polynomial coefficients \( d = [d_2 \ d_3 \ \cdots]^T \) is constructed from two sequential least squares estimation steps.

First, find an estimate \( \hat{\gamma} \) of the VL coefficients by solving

\[ \hat{\gamma} = \arg \min_{\gamma} \sum_{k=1}^{K} |y(k) - \varphi^T(k) \gamma|^2. \tag{5.15} \]

An estimate of the linear part is then \( \hat{y}_l = \varphi_1^T(k) \hat{\gamma}_1 \), and an estimate of the output is \( \hat{y} = \varphi^T(k) \hat{\gamma} \). Second, form

\[ z(k) = \hat{y}(k) - \hat{y}_l(k) - \hat{\gamma}_0, \]
\[ \phi^T(k) = [(\varphi_1^T \gamma_1)^2 \ \cdots (\varphi_1^T \gamma_1)^m]^T. \tag{5.16} \]

The polynomial coefficients are then found as

\[ \hat{d} = \arg \min_{d} \sum_{k=1}^{K} |z(k) - \phi^T(k) d|^2. \tag{5.17} \]
Note that when the underlying system is not within the class of polynomial Wiener models but rather a general Wiener model parameterized in the form of a standard VL model as (5.8), then applying the procedure described above can be interpreted as projecting the nonlinearity onto a polynomial basis. This results in an additional approximation error due to the model mismatch. In [7], an iterative scheme for reducing the bias induced by said mismatch is presented.

5.2.5 The Particle Filter

The particle filter (PF), aka Sequential Monte Carlo, is a nonlinear Bayesian state estimation method [1]. Monte-Carlo simulation is used in the PF to approximate the posterior filtering distribution $p(x_k|y_{1:k}, u_{1:k})$ of the state-space model

$$
\begin{align*}
    x(k+1) &= f(x(k), u(k)) + v(k), \\
    y(k) &= h(x(k), u(k)) + e(k).
\end{align*}
$$

The approximation can be made arbitrarily accurate by increasing the number of particles. This, however, comes with an additional computational cost, limiting the number of states that can efficiently be estimated, as the number of particles needed to achieve a given resolution in the state space increases with the dimension.

One popular implementation of the PF is the Sequential Importance Resampling (SIR) algorithm. Let $x^{(i)}$ denote particle $i$, $w^{(i)}$ its corresponding weight, and $M$ the number of particles. Following [5], the estimation algorithm is

$$
\begin{align*}
    \tilde{x}^{(i)}_{k+1} &= f(x^{(i)}_k, u_k) + v^{(i)}_k, \\
    \tilde{w}^{(i)}_{k+1} &= w^{(i)}_k p_e(y_k - h(\tilde{x}^{(i)}_k, u_k)), \\
    w^{(i)}_{k+1} &= \frac{\tilde{w}^{(i)}_{k+1}}{\sum_{j=1}^{M} \tilde{w}^{(j)}_{k+1}}, \\
\end{align*}
$$

for $i = 1, 2, \ldots, N$. The particles are then resampled by drawing $M$ new particles $\{x^{(i)}_{k+1}\}_{i=1}^{M}$ with replacement, so that $\Pr(x^{(i)}_{k+1} = \tilde{x}^{(i)}_{k+1}) = w^{(i)}_{k+1}$. When the new particles have been resampled, all weights are set to $w^{(i)}_{k+1} = 1/M$.

5.3 Experiments and data collection

The experimental setup consisted of a computer screen and a video-based eye tracker from SmartEye AB, Sweden. The eye tracker records the gaze
position of the test subject, i.e. the point on the screen where the subject is looking. Test subjects were placed about 50 cm from the computer screen, and a stimulus constituting of a colored dot moving on a black background was displayed. The stimuli, acting as input signals to the SPS, were generated using the method in [8] to provide excitation both in amplitude and frequency, and the gaze position was sampled at 60 Hz. For the application in hand, it implies that the dot has to visit all the areas within the stimuli window and move within a suitable acceleration range, with regard to the bandwidth of the SPS.

Two groups of test subjects are considered. One control group consisting of 22 healthy individuals from 50 to 76 years old, and one group of 7 individuals diagnosed with Parkinson’s disease (PD), from 61 to 80 years old. In this paper, however, data from one subject in each group are considered, with 38 measurements from the patient, and 24 measurements from the control subject. Furthermore, in the experiments with synthetic data, the same input signals (visual stimuli) are used as in the clinical trials.

Prior to the first eye-tracking test, each patient received a dose of an anti-PD drug, after an 8 hours long washout. The administered dose was set to 150% of the patient’s usual morning dose of medication. This experimental protocol allows to track the patient’s symptoms, as they transit from off-state to normal mobility and/or dyskinesia and back. Thus, the measurements from the patient group exhibit a range of behaviors, including lack of visible PD symptoms. Because of this, a larger domain of coefficients is expected to be needed to describe the SPS dynamics the patient group, compared to that in the control group. The distribution of model coefficients estimated for the patients is anticipated to partially overlap with those for the control group.

The tests were conducted at CTC (Clinical Trials Consultants AB) Center at the University Hospital in Uppsala, Sweden, between May and August 2015, see [18] for details.

5.4 Wiener modeling of smooth pursuit

Two model structures are considered to capture the response of SPS to the visual stimuli. A VL model with $N = 2$ and the kernels parametrized in the three first Laguerre functions is referred to as the full model and is written
as
\[
y(k) = y_0 + \sum_{j=0}^{2} \gamma_1(j) \psi_j(k) + \\
+ \sum_{j_1=0}^{2} \sum_{j_2=j_1}^{2} \gamma_2(j_1, j_2) \psi_{j_1}(k) \psi_{j_2}(k).
\] (5.20)

A reduced model based on (5.20) is also considered
\[
y(k) = y_0 + \gamma_1(0) \psi_0(k) + \gamma_1(1) \psi_1(k) + \\
+ \gamma_2(0, 0) (\psi_0^2(k) + \theta_{02} \psi_0(k) \psi_2(k) + \theta_{22} \psi_2^2(k)),
\] (5.21)

where $\theta_{02} = -1.81468$ and $\theta_{22} = .938454$. Model structure (5.21) was produced by the SPICE algorithm for sparse estimation [19] combined with principal component analysis (PCA) to extract a parsimonious model structure from (5.20). The data sets described in Section 5.3 were used, and sparse VL models with $N = 2$ were estimated. PCA was used to identify possible functional relations between the coefficients, and a reduced model structure was selected. The VL-model reduction procedure is detailed in [2].

While (5.20) is parametrized by ten coefficients altogether, (5.21) has only four coefficients. Yet, as shown further, the modeling quality in terms of the output error is preserved with the use of the reduced model.

In view of these models, the nonlinear part is approximated by a third-order polynomial $f(x) = x + d_2 x^2 + d_3 x^3$ to align with the structure of the Wiener model in (5.12).

### 5.4.1 Identification of SPS

For one patient diagnosed with PD and one healthy control subject, the coefficients of the Wiener model were estimated using the two-step LS approach described in Section 5.2.4. In Fig. 5.1 the distributions of the normalized MSE (NMSE) for the patient and control subject are shown, in the full model case and reduced model case, respectively. Evidently, the output error does not increase due to approximating the nonlinearity by a third-order polynomial. The error distributions are almost identical in the reduced model case. As the distributions of the polynomial coefficients in Fig. 5.2 show, the coefficient variances for the patient is much larger than those for the control subject. The WM residuals are distributed with heavier tails than a normal distribution, see Fig. 5.3.
Figure 5.1: Distributions of NMSE using the full and reduced VL models and WM. The error distributions are almost identical in both cases.

Figure 5.2: Distributions of $d_2$ and $d_3$, using the full and reduced WM.
5.5 Wiener estimation using the particle filter

To estimate the parameters of system (5.12) by a PF, an augmented state-space model is formulated as

\[
\begin{bmatrix}
\psi(k+1) \\
\mathbf{c}(k+1) \\
\mathbf{d}(k+1)
\end{bmatrix} =
\begin{bmatrix}
F & 0 \\
0 & I
\end{bmatrix}
\begin{bmatrix}
\psi(k) \\
\mathbf{c}(k) \\
\mathbf{d}(k)
\end{bmatrix} +
\begin{bmatrix}
G \\
0
\end{bmatrix} u(k) + v(k)
\]

\[y_l(k) = \psi^T(k)\mathbf{c}
\]

\[y(k) = y_l(k) + d_2 y_l(k)^2 +
+ d_3 y_l(k)^3 + y_0 + \varepsilon(k),\]

where \(v_k \in \mathbb{R}^T\) and \(e_k \in \mathbb{R}\) are white Gaussian noise sequences. Thus, the parameter states modeled as random walk are estimated simultaneously with the dynamical states, i.e. the Laguerre filter outputs.

5.5.1 Synthetic data

To test the feasibility of the concept, synthetic data were generated by simulating output of WM (5.12), with the parameters \(\mathbf{c} = [-0.3, 0.6]^T\), \(\mathbf{d} = [d_2, d_3]^T = [5 \cdot 10^{-4}, 5 \cdot 10^{-7}]^T\), and \(y_0 = -9\). These parameter values are chosen to render dynamics similar to those actually observed in the SPS. Gaussian white noise was added to the simulated output resulting in a SNR of 20 dB. The same input signals were used as in the clinical experiments.

Three scenarios were examined: estimating all states, estimating only the linear VL part, and estimating only the Laguerre filter outputs (i.e.
Table 5.1: NMSE for PF estimation

<table>
<thead>
<tr>
<th>Estimated states</th>
<th>Unscaled</th>
<th>Scaled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input signal 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>All</td>
<td>$2.43 \times 10^{-3}$</td>
<td>$1.03 \times 10^{-3}$</td>
</tr>
<tr>
<td>Linear part &amp; output</td>
<td>$0.60 \times 10^{-3}$</td>
<td>$0.34 \times 10^{-3}$</td>
</tr>
<tr>
<td>Output only</td>
<td>$8.22 \times 10^{-8}$</td>
<td>$0.96 \times 10^{-8}$</td>
</tr>
<tr>
<td>Input signal 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>All</td>
<td>$1.75 \times 10^{-3}$</td>
<td>$0.57 \times 10^{-3}$</td>
</tr>
<tr>
<td>Linear part &amp; output</td>
<td>$0.76 \times 10^{-3}$</td>
<td>$0.37 \times 10^{-3}$</td>
</tr>
<tr>
<td>Output only</td>
<td>$1.05 \times 10^{-7}$</td>
<td>$0.15 \times 10^{-7}$</td>
</tr>
</tbody>
</table>

tracking the output, given the true model). The computations were performed directly on the simulated data, as well as with the input and output signals scaled to constrain the output values to $y \in [-1, 1]$. This scaling keeps the linear parameters intact but yields larger polynomial coefficients. Thus, numerical issues with estimating very small values may be avoided. Furthermore, three different input signals (visual stimuli sets) were used in order to examine whether the differences in excitation matter.

The parameter estimates at each PF iteration when using the original signals are shown in Fig. 5.4, while Fig. 5.5 depicts the parameter estimates with signal scaling. Note that the parameter estimates have been re-calculated to the original scale, to facilitate comparison. In general, the linear parameter estimates are close to the true values, both when using the scaled and unscaled data. These estimates are also very similar both in value and in the trajectory of the estimate over time, between the two cases. The true polynomial coefficients are not found, indicating that the contribution from the nonlinear part is small.

The output, however, is tracked very well, even when the parameter estimates are off. Table 5.1 shows the NMSE in the different estimation cases, where the scaled signals were scaled back to allow for comparison. There is only little difference between the three estimation cases, while the output errors differ approximately by a factor 2–5 between the scaled and unscaled case. Note here, that the errors are small, compared to the estimates obtained from real data (see Fig. 5.1). Furthermore, there is no significant difference in the output error between the different input signals.

5.5.2 Experiments with eye-tracking data

A PF with $M = 10^6$ was used to estimate the parameters of a WM for three test subjects: two controls and one patient. Three and two data sets from the control subjects were used, respectively, and three data sets from
5.5. Wiener estimation using the particle filter

Figure 5.4: Parameter estimates for unscaled data when all parameters and states are estimated (All), only the linear part and the Laguerre filter outputs are estimated (Linear), together with the true values. The estimates of the linear coefficients are close to the true values, while the polynomial coefficients are not estimated as well. There is a slight bias in the estimate of the constant.

Figure 5.5: Parameter estimates for scaled data when all parameters and states are estimated (All), only the linear part and the Laguerre filter outputs are estimated (Linear), together with the true values. The linear coefficients’ estimates are close to the true values. The polynomial coefficients are estimated with small errors and the estimate of the constant is biased.
the patient. Two of the patient data sets were recorded in an unmedicated state (after overnight wash-out, and after the medication effect had ceased, respectively), and one corresponds to the time instant when the anti-PD medication had full effect.

The estimates of the linear coefficients over time are shown in Fig. 5.6. There are large variations between the test subjects, while the intra-subject variations are significantly smaller. Here, no effect of disease or medication is seen in the linear part. In Fig. 5.7, the estimates of the Wiener polynomial coefficients are depicted. The estimates of the coefficient $d_2$ for the Parkinson’s patient are lower than those for the controls, both in the medicated and unmedicated state. However, the estimates of $d_3$ differ significantly from those of the control subjects only in the unmedicated cases. The particle distributions of the PF in the last iteration are shown in Fig. 5.8. These distributions are instrumental in assessing the uncertainty of the parameter estimates. Evidently, the distributions overlap, yet there is a visible difference between the subjects. Also noteworthy is the fact that the uncertainty seems to be larger in the unmedicated state than under medication, as the distributions in the unmedicated state are wider and in some cases multimodal.

![Figure 5.6: Estimate trajectories of linear coefficients. Inter-patient variability is large in comparison to intra-patient variability. The linear coefficients do not seem to capture the effect of PD on SPS.](image)

5.6 Conclusions

The PF approach has been applied for estimation of the parameters in a polynomial Wiener model of the human SPS. Experiments on synthetic data show the potential performance of the approach. Experiments on eye-tracking data suggest that the uncertainty of the parameter estimates in-
5.6. Conclusions

Figure 5.7: Estimate trajectories of Wiener polynomial coefficients. The values for the quadratic coefficient, $d_2$, differ between the patient and the controls. The controls together with the medicated patient form one group with higher $d_3$ values, and the unmedicated patient forms another, with lower $d_3$ values. This indicates that the nonlinear part may capture the effects of PD on SPS.

crease with symptoms of Parkinson’s disease. There is also some indication that symptoms of Parkinson’s disease can be captured by the nonlinear part of the Wiener model.

The results presented should be viewed as indications, and future work needs to be done on two fronts. First, larger datasets must be considered, to be able to generalize the results further. Second, the variance of the driving noise in the parameter proposals may be tuned further, and better parameter estimates could possibly be found.
Figure 5.8: Distributions of particles for the Wiener polynomial coefficients in the last iteration of PF. While there is overlap between the distributions, the two groups can be seen, with the exception of one data set from a control subject. Note that the width of the distribution is considerably reduced in the patient’s medicated state, compared to the unmedicated state.
References


Title
Modeling of Human Smooth Pursuit by Sparse Volterra Models with Functionally Dependent Parameters

Authors
Viktor Bro and Alexander Medvedev

Edited version of
Abstract

This paper deals with the identification of Volterra models that capture the dynamics of human eye movements in smooth pursuit. Eye movements in response to specially designed visual stimuli are recorded by an eye tracker. The modeling framework is primarily motivated by medical applications, namely quantification of oculomotor dynamics in neurology, but can also be useful in biometrics. In health, ocular dynamics are known to be predominantly linear, while disruptions in the neural control due to neurological conditions inflict nonlinearity on smooth pursuit eye movements. Besides the well-known issue of overparameterization, the estimated Volterra models also exhibit functional dependence among the model coefficients. A combination of sparse estimation and Principle Component Analysis is shown to be instrumental in estimating parsimonious Volterra models from eye-tracking data. The efficacy of the approach is demonstrated on experimental data collected from patients diagnosed with Parkinson’s disease in different states of medication as well as healthy controls.

6.1 Introduction

Eye tracking technology has been developing at high pace during the last decade and is now widespread in certain application areas such as usability evaluations of software [28], optimization of visual interfaces [35], and gaming [6]. The role of an eye tracker is to provide an accurate estimate of where on the computer screen the user gaze is directed. The eye move-
ment in question is then fixation and can be reliably characterized with or without additional infrared illumination of the iris and with one or two cameras \[23\]. The gaze trajectory in between the fixations is less important in this context, compared with the information on how often and for how long a certain position on the screen is inspected, i.e. characterization of areas of interest.

The human eyeball is actuated by three pairs of muscles (see Fig. \[6.1\]), whose actions are coordinated by the central nervous system. Different types of eye movements are controlled by different neural centra and the exact information flow in the control loops is not known \[25\]. Medical applications are often concerned with other types of eye movements than fixations, in particular smooth pursuit (SP) and saccades.

In SP, the gaze is shifted when the person continuously follows the movements of a target. The smooth pursuit system (SPS) cannot be activated voluntarily and needs a target to follow. The SPS operates in a feedback loop involving the eyes and the extraocular muscles, as well as several parts of the brain \[33\]. Injury or an alteration caused by a disease to any of these parts may thus result in impaired SPS performance. Examples of this include mental and neurological conditions such as schizophrenia \[24\], Parkinson’s disease (PD) \[7, 9, 11, 13, 14, 20\] or the use of alcohol and narcotics \[38\]. SP tests are popular in neurology as they allow for examining the integrity of combined visual and motor feedback loops, as well as the impairment of feedback control in oculodynamics.

Saccades are used to swiftly shift gaze from one point in space to another. While the angular velocity of the eyeball in SP is matched with that of the target, saccades result in significantly faster ballistic eye movements generated in order to cover as much of the visual field as possible in short time.

Regarding SP, there is an overall agreement that this type of eye movements is controlled by a negative feedback that follows a proportional-differential control law \[19\]. The latter fact explains an angular stationary error with respect to the tracked target.

The type of mathematical model to capture the oculomotor dynamics naturally depends on the end use of the model. Parametric models are suitable when a certain feature of the oculodynamics has to be quantified, as often required in medical applications. Parametric models are typically designed to be parsimonious and express the observed dynamics in terms of a few parameters. When a comparison of a gaze trajectory to a population distribution or another trajectory is sought, nonparametric models come in handy, \[8\]. Instead of characterizing the whole data set as parametric models do, nonparametric ones can highlight the important (deviating) parts in data, thus yielding higher specificity \[8\].
The neural feedback control mechanism of SP renders oculodynamics linear in healthy subjects [15]. Therefore, nonlinear modeling, although biologically motivated, is meaningful only when the normal mechanisms of SPS are disrupted, e.g. in disease or under influence of drugs. Biomechanical studies of the human eye [36] open up for grey-box modelling of eye movements [14]. A major complication with the latter in portraying SP is lack of adequate mathematical description for the neurally implemented feedback. Therefore, black-box models are still preferable in the SPS performance quantification.

Volterra models (VM) [26] are black-box nonlinear models and traditionally more used in some research areas than other. Biological applications, especially in neuroscience, have been addressed with VM for a long time, e.g. [21]. A notorious issue with VMs is their overparametrization, a property which gives them high flexibility in capturing nonlinear phenomena in data but also makes them less accurate in prediction tasks. This shortcoming is aptly met by sparse estimation [17]. Another issue with VMs pertaining to nonlinearity is that a high degree of excitation (white noise) is necessary to reveal the output dependence on both frequency and amplitude of the input in absence of a priori information. Fortunately, in SPS modeling, the relevant frequency and amplitude range of visual stimuli are known and the identification input can be designed by solving an optimization problem [11]. Further, VMs are seldom used in control and filtering applications since they are not readily expressed in terms of a differential or difference equations. Yet, methods for estimating a polynomial Wiener model from the kernels of a VM have been developed in [10] and applied to the problem of nonlinearity estimation in SPS.

There is no doubt that practically useful models need to be sparse, i.e. involve as few free parameters as possible. In real-life applications, sparse models exhibit functional dependence of the coefficients on each other. In contrast to model sparsity, this matter is seldom addressed in system identification since a model is the final product of an identification algorithm. In e.g. symptom quantification, a mathematical model is just a tool to extract relevant information from data and the model coefficients have to be related to clinically relevant phenomena. When functional dependencies between the model parameters can be approximated as linear, Principle Component Analysis (PCA) is shown to be useful in discovering and parametrizing them [2, 22, 31].

Alterations of SPS in neurological conditions, including PD, are well known and broadly studied, also by means of eye tracking. Most of the available publications restrict themselves to simple visual stimuli in order to apply the established SPS performance metrics such as SP gain, [9]. The dynamical and nonlinear nature of SPS cannot be properly revealed in those
experiments and the modeling framework of the present paper is proposed as an alternative.

The main contributions of this paper are:

• Sparse second-order VMs with functionally dependent parameters are demonstrated to yield suitable representation of human SP oculodynamics across two eye-tracking platforms and applications.

• Single eye-tracking data sets of SP experiments do not render sufficient information for distinguishing between healthy individuals. The use of probability distributions of the VM estimates increases the test specificity but is not yet reliable enough even for a small group of individuals.

• The parameter estimates in a sparse second-order VM of SP evaluated from eye-tracking data of PD patients are shown to respond to a dose of levodopa, the mainstay drug alleviating the symptoms of Parkinsonianism.

The rest of the paper is organized as follows. First, the mathematical tools and concepts utilized in the study are briefly summarized. Then, the data collection procedure and equipment are described. The main part of the paper is concerned with identifying sparse VMs and relating them to two practically important problems: motor symptoms quantification in PD and biometrics. The obtained results are evaluated and discussed in detail. Yet, patient data are only used to illustrate the feasibility of the modelling approach and not demonstrate its clinical efficacy. The latter would require an extensive clinical study. Finally, conclusions are drawn.

6.2 Theoretical background

This section summarizes the mathematical models and tools used for their estimation and evaluation further in the paper.

6.2.1 The Volterra model

The Volterra series is a functional expansion of a dynamic, nonlinear, time-invariant system, \[37\]. This can be compared to, for example, the Taylor series expansion of static functions.

A discrete dynamical system with input \( u(k) \in \mathbb{R} \) and output \( y(k) \in \mathbb{R} \), \( k = 0, \cdots, K - 1 \) can be approximated by a truncated Volterra series

\[
y(k) = y_0 + \sum_{n=1}^{N} (H_n u)(k) + e(k),
\]

(6.1)
where $e(k) \in \mathbb{R}$ is the noise (or approximation error) term, $N \in \mathbb{N}$ is the nonlinearity degree, and

$$
(H_n u)(k) = \sum_{i_1=0}^{\infty} \cdots \sum_{i_n=0}^{\infty} h_n(i_1, \cdots, i_n) u(k - i_1) \cdots u(k - i_n) \quad (6.2)
$$

are the Volterra functionals. The functions $h_n$ are called Volterra kernels and are often cumbersome to calculate explicitly. Therefore, the kernels are usually expanded in an orthogonal functional basis. If the kernels satisfy $h_n \in \ell^2[0, \infty)$ and are not excessively oscillative, the discrete Laguerre functions present a suitable choice.

### 6.2.2 Laguerre functions

The Laguerre functions are exponentially decreasing, oscillative functions that form an orthonormal basis in $\ell^2[0, \infty)$. Higher order functions oscillate more than those of lower order, as examples of the Laguerre functions in Fig. 6.2 demonstrate.

The $j$:th Laguerre function is defined in the $\mathcal{Z}$-domain as

$$
\Phi_j(z) = \sqrt{1 - \alpha z} \left( \frac{1 - \sqrt{\alpha} z}{z - \sqrt{\alpha}} \right)^j, \quad (6.3)
$$

where $0 < \alpha < 1$ is the Laguerre parameter, acting as a scaling coefficient. The inverse $\mathcal{Z}$-transform yields the corresponding time-domain functions $\phi_j(k) = \mathcal{Z}^{-1}\{\Phi_j(z)\}$. Thus, any function $h(k) \in \ell^2[0, \infty)$ can be uniquely
6.2. Theoretical background

Laguerre functions with $\alpha = 0.5$.

$$\phi_0$$

$$\phi_1$$

$$\phi_4$$

$Laguerre function$ $\phi_1$ with varying $\alpha$

$$\alpha = 0.2$$

$$\alpha = 0.5$$

$$\alpha = 0.8$$

Figure 6.2: Examples of the discrete Laguerre functions in the time domain.

expressed as

$$h(k) = \sum_{j=0}^{\infty} \gamma_j \phi_j(k). \quad (6.4)$$

6.2.3 The Volterra-Laguerre model

With the Laguerre functions used to parametrize the Volterra kernels $h_n$, one has

$$h_n(i_1, \ldots, i_n) = \sum_{j_1=0}^{\infty} \cdots \sum_{j_n=0}^{\infty} \gamma_n(j_1, \ldots, j_n) \phi_{j_1}(i_1) \cdots \phi_{j_n}(i_n). \quad (6.5)$$

However, since only a finite number of Laguerre functions can be computed, a truncated Laguerre series of $Laguerre$ $order$ $L$ is used instead:

$$h_n(i_1, \ldots, i_n) \approx \sum_{j_1=0}^{L} \cdots \sum_{j_n=0}^{L} \gamma_n(j_1, \ldots, j_n) \phi_{j_1}(i_1) \cdots \phi_{j_n}(i_n). \quad (6.6)$$

Then, assuming that the $L$ first Laguerre functions are sufficient to capture the kernels, the Volterra functionals are given by

$$H_n u(k) = \sum_{j_1=0}^{L} \cdots \sum_{j_n=0}^{L} \gamma_n(j_1, \ldots, j_n) \psi_{j_1}(k) \cdots \psi_{j_n}(k), \quad (6.7)$$

where $\psi_j(k) = (\phi_j \ast u)(k) = \sum_{l=0}^{k} \phi_j(l) u(k - l)$ is the $Laguerre$ $filter$ $output$ and $(\cdot \ast \cdot)$ denotes the convolution operator. The doubly finite Volterra-Laguerre model is then stated as

$$y(k) = y_0 + \sum_{n=1}^{N} \sum_{j_1=0}^{L} \cdots \sum_{j_n=0}^{L} \gamma_n(j_1, \ldots, j_n) \prod_{l=1}^{n} \psi_{j_l} + e(k), \quad (6.8)$$
where the coefficients $\gamma_n(j_1, \ldots, j_n)$ are referred to as the Volterra-Laguerre (VL) coefficients.

The number of coefficients in the model above depends both on the nonlinearity degree $N$, and on the Laguerre order $L$. The total number of coefficients capturing the system dynamics is $(L^{N+1} - 1)/(L - 1)$. However, since the Laguerre functions commute, the kernel functions are symmetric with respect to index. Many VL coefficients are therefore redundant as they correspond to the same product of Laguerre functions and thus cannot be estimated individually. The number of coefficients to estimate in model (6.8) can thus be reduced to

$$N_c = \binom{L + N + 1}{N},$$

The VL model is then

$$y(k) = y_0 + \sum_{n=1}^{N} \sum_{j_1=0}^{L} \cdots \sum_{j_n=j_{n-1}}^{L} \gamma_n(j_1, \ldots, j_n) \prod_{l=1}^{n} \psi_{j_l} + e(k), \quad (6.9)$$

where $e(k)$ is the modeling error.

**The VL model in linear regression form** While the Volterra model is nonlinear in the variables, it is linear in the parameters $\gamma_n(j_1, \ldots, j_n)$. Therefore, the model in (6.8) can be recast in regression form

$$y = \begin{bmatrix} \Psi & \mathbf{I}_K \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{e} \end{bmatrix} = \mathbf{B}\beta, \quad (6.10)$$

where $\Psi = [\psi_0 \ \psi_1 \ \ldots \ \psi_0^2 \ \psi_0\psi_1 \ \ldots ]$ is the regression matrix constructed from the Laguerre filter outputs $\psi_j = [\psi_j(0) \ \ldots \ \psi_j(K-1)]$, and $\mathbf{c} = [\gamma_1(0) \ \ldots \ \gamma_N(L, \ldots, L)]^T$. The number of samples is denoted by $K$, the errors $e(k)$ are collected in the vector $\mathbf{e}$ and $\mathbf{I}_K$ denotes an identity matrix of dimension $K \times K$.

### 6.2.4 Tools for model sparsity: SPICE and PCA

A high Laguerre order in (6.8) means that many parameters must be estimated. This, in turn, implies higher variance in the parameter estimates obtained from the data record. At the same time, the Laguerre order is basically defined by the nature of the dominating nonlinearities in the modelled system. It is therefore of interest to reduce the number of model parameters without reducing the Laguerre order since the coefficients corresponding to the terms of lower order than $L$ do not generally have to be large.
Linear sparse estimation \cite{16, 17, 29} constitutes an efficient tool of excluding parameters that do not contribute much to the model fidelity. However, model sparsity does not account for possible functional relationships between the significant parameters and, therefore, permits the number of independent model parameters to be reduced even further. Principal component analysis (PCA) \cite{39} is a standard tool to reveal linear dependence between model parameters but seldom highlighted in system identification. It has though been used to discern significant coefficients in VM \cite{22}.

**SPICE.** The *SParse Iterative Covariance-Based Estimation* (SPICE) algorithm, is a method for sparse estimation \cite{32}. This method, unlike the highly popular LASSO \cite{34}, does not require the tuning of any hyperparameters. At the same time, LASSO can be obtained as a special case of Square-Root SPICE \cite{1}.

The SPICE algorithm is formulated as follows. Let

\[ w_k = \frac{\|b_k\|_2}{\|y\|_2}, \]

where \(b_k\) denotes the \(k\):th column of \(B\) in (6.10) and \(\|\cdot\|_2\) is the Euclidean vector norm. The SPICE estimate of the coefficients is then found by solving the linear program

\[
\min_{\alpha, \beta} \sum_{i=1}^{N_c+K} w_i \alpha_i
\]

s.t. \(-\alpha_i \leq \beta_i \leq \alpha_i,
\]

\[ \alpha_i \geq 0, \quad i = 1, \ldots, N_c + K, \]

\[ y = B\beta. \]  

(6.11)

Note also that, when some Laguerre function is beforehand known to be absent in the series of a Volterra kernel, a constraint may be introduced in the SPICE algorithm, so that the corresponding coefficient is set to zero. Similarly, additional linear constraints in the estimated coefficients, such as positivity of certain terms in the Volterra series, can be added to the linear program to be solved.

**Principal component analysis** Consider \(M\) measurements \(\{x_i, i = 1, \ldots, M\} \in \mathbb{R}^{N_c}\). PCA can be thought of as fitting an ellipsoid to the measurements, and projecting these onto the axes of the ellipsoid, i.e. the principal components. If the data variance along some principal component is small, this component may be omitted, thus reducing the dimensionality of the problem. The PCA can also be used as an exploratory tool for finding functional relationships between data components, e.g. in linear regression.
6.3 Experiments & data collection

Two databases have been used in this paper to illustrate the feasibility of the modeling approach. The databases have been collected during two different studies and consist of data sets.

Repetitions of the same eye-tracking experiment are reflected in the data in both databases. A moving dot was presented on a computer screen, and the gaze position of the test subject (i.e. the coordinates of the point on the screen at which the gaze is directed) was recorded throughout the experiment. The visual stimulus was designed to possess not only enough excitation frequency-wise, but also amplitude-wise, to reveal possible non-linear dynamics. A description of the optimization-based stimulus design is provided in [11]. An example of the horizontal projection of a stimulus realization is shown in Fig. 6.3.

Each eye-tracking experiment yields an input-output pair of 2D signals, $u(k), y(k)$, where the stimulus $u(k)$ acts as the input to the SPS that, in response, outputs the gaze position $y(k)$. The stimulus duration is roughly 25 s long, and the eye trackers were run at a sampling frequency of 60 Hz. Thus, one data set contains about 1500 samples. Throughout the paper, only the horizontal parts of the involved signals are utilized. Yet, the vertical components can be processed in a similar way.

One database (DB1) was collected at Uppsala University hospital by Clinical Trial Consultants AB (CTC) in 2015. This database consists of a total number of 144 data sets from a group of seven patients diagnosed with PD, and 499 data sets from 22 healthy control subjects of matching age. A video-based desktop eye tracker from SmartEye AB was used for collecting experimental data.

In this study, response of the patients to pharmacotherapy was evaluated over a day as follows: Before the first test, each patient received a dose of anti-PD medication (levodopa) after an eight hour long washout period. The dose was set to 150% of the patient’s usual morning dose. The patients then performed eye tracking and motor tests with regular intervals...
6.4 Results

Figure 6.4: Mobile eye tracking setup with EyeTribe eye tracker and Microsoft Surface tablet computer.

of about 20 min. This experimental design allows for tracking of the patient’s symptoms as they progress from Parkinsonian symptoms (off state) to normal mobility (on state) or even dyskinesia (due to the exaggerated dose), and, finally, back to off state. The eye-tracking test data were used to estimate the coefficients of VL models describing the SPS dynamics at each test instant. This study was a part of the project ”MuSyQ: Multimodal motor symptoms quantification platform for individualized Parkinson’s disease treatment”, and a more detailed description of the experiments is found in [27].

The second database (DB2) was collected using a mobile eye tracking setup consisting of an eye tracker from The EyeTribe and a Microsoft Surface tablet computer (shown in Fig. 6.4). This solution has been developed at Information Technology, Uppsala University [30]. Between 27 and 30 data sets were collected from each of four healthy individuals. Four test subjects participated in the study:

S1: Woman, age 25
S2: Man, age 26
S3: Man, age 59
S4: Man, age 25.

6.4 Results

Two experimental model-based studies of SP are presented in this section: The first one, using the data in DB1, compares the SPS dynamics in a healthy control group to those of a group of patients diagnosed with PD.
In the second study, based on DB2, it is demonstrated that the linear SPS dynamics differ between healthy individuals, and this can be made evident even by means of a simple model.

### 6.4.1 Sparse VL model of smooth pursuit

As shown in [2], a VL model with $N = 2$ and $L = 2$\
\[
y(k) = y_0 + \sum_{j=0}^{2} \gamma_1(j) \psi_j(k) + \sum_{j_1=0}^{2} \sum_{j_2=0}^{2} \gamma_2(j_1, j_2) \psi_{j_1}(k) \psi_{j_2}(k),
\] (6.12)
is a suitable structure for modeling the SPS dynamics. However, (6.12) has 10 parameters and is overparametrized for the purpose of SPS modeling, a shortcoming which can be recuperated by means of the SPICE algorithm [12]. To further reduce the number of free parameters by finding functional relations between them, an approach combining SPICE and PCA is proposed in [2]. Making use of the data collected from patients and controls in DB1, the following reduced model was obtained\
\[
y(k) = y_0 + \gamma_1(0) \psi_0(k) + \gamma_1(1) \psi_1(k) + \gamma_2(0, 0) (\psi_0^2(k) + \theta_{02} \psi_0(k) \psi_2(k) + \theta_{22} \psi_2^2(k)),
\] (6.13)
where $\theta_{02} = -1.81468$ and $\theta_{22} = .938454$. To estimate the coefficients of it from eye-tracking experiment data, the procedure below comprising the steps Step 1–Step 3 is followed.

**Step 1:** Collect data by recording gaze position while presenting a stimulus on the computer screen. For horizontal/vertical movement, an input vector $u$, with stimulus position, and output vector $y$, with gaze position are obtained. Only horizontal movements are covered in this paper.

**Step 2:** Compute the Laguerre filter outputs, $\psi_j(k) = (\phi_j \ast u)(k), j = 0, 1, 2$.

**Step 3:** Estimate the coefficients $\gamma_1(0), \gamma_1(1), \gamma_2(0, 0)$.

### 6.4.2 PD symptoms in a sparse VL model

Probability distributions of the estimated model parameters, for all controls and patients data sets in DB1, are depicted in Fig. 6.7 The distributions for both the linear and quadratic part are largely overlapping. This can be explained by the study design. Since the distributions include all measurements, different kinds of oculodynamics are represented, corresponding to
6.4. Results

Figure 6.5: Normal probability plot of the linear coefficients from controls and patients.

Figure 6.6: Normal probability plot of the quadratic coefficients from controls and patients.

Parkinsonian and dyskinetic states, as well as symptom-free SPS dynamics. In Fig. 6.5 and Fig. 6.6, normal probability plots for the coefficient of the linear and quadratic parts of the model are provided. The evaluated distributions are fairly close to normal, except for the linear coefficients of the patient models. This can be also clearly seen in Fig. 6.7 and explained by the wide range of SPS dynamics exhibited by the patients in the experiments. The distribution of the quadratic coefficients for the controls is more peaked than a normal distribution. This agrees with the fact that the nonlinear SPS dynamics in healthy controls are expected to be nonexistent.

6.4.3 Distinguishing between healthy individuals using a linear VL model

An application of VL modeling of SPS to biometrics is considered in this part and illustrated by the data in DB2. The research question is whether
or not it is possible to distinguish between healthy individuals on the basis of their SP movements.

To start with, a linear VL model ($N = 1$), parametrized in three Laguerre functions ($L = 2$) was used. As shown in Fig. 6.8 there is a clear linear relation between the parameters. This relation was then exploited to minimize the number of free model parameters by estimating the straight lines

\begin{align}
\gamma_1(1) &= \theta_{1,1} \gamma_1(0) + \theta_{0,1} \\
\gamma_1(2) &= \theta_{1,2} \gamma_1(0) + \theta_{0,2}
\end{align}

(6.14)
depicted in Fig. 6.8. The parameters of the affine map were evaluated to

\begin{align}
\theta_{0,1} &= 0.147714 & \theta_{1,1} &= -1.95084, \\
\theta_{0,2} &= 0.317524 & \theta_{1,2} &= 1.00591,
\end{align}

(6.15)

thus yielding the reduced model

\begin{align}
y(k) &= y_0 + \sum_{j=0}^{2} \tilde{\gamma}_1(j) \psi_j(k),
\end{align}

(6.16)
\begin{align}
\tilde{\gamma}_1(1) &= \theta_{1,1} \gamma_1(0) + \theta_{0,1}, \\
\tilde{\gamma}_1(2) &= \theta_{1,2} \gamma_1(0) + \theta_{0,2}.
\end{align}

The model above provides a highly parsimonious representation of the SPS dynamics without introducing much additional error, as is illustrated in Fig. 6.9.
Figure 6.8: Scatter plot of the estimated parameters for all measurements using the full model (blue) and the model with linearly dependent parameters (orange). The red line is the estimated linear relation in the parameters of the full model.

For a gaze trajectory $y(k)$ and an estimated gaze trajectory $\tilde{y}(k)$, $k = 1, \ldots, K$, the loss $L$ is computed as

$$L = \frac{1}{K \|y\|} \sum_{k=1}^{K} (\tilde{y}(k) - y(k))^2,$$

where $\|y\| = \sqrt{\sum_{k=1}^{K} y(k)^2}$. The relative change in model performance due to the reduction from the full model $L_{\text{full}}$ to the one with less free parameters $L_{\text{reduced}}$ is

$$L_\Delta = (L_{\text{reduced}} - L_{\text{full}})/L_{\text{full}}.$$

Using reduced model (6.16), the distributions of the parameter $\gamma_1(0)$ have been computed by kernel density estimation, for the four test subjects, and are shown in Fig. 6.10. In order to compare the parameter distributions, the Kullback-Leibler divergence (KLD) was computed pairwise for all distributions [4, 5, 18]. The divergence values, normalized with respect to the largest divergence value, are shown in Table 6.1 and suggest, e.g. that subjects S1 and S2 are more similar to each other than to the other subjects. S3 is the most dissimilar subject in terms of pairwise distribution divergence,
Relative increase in loss, $L_\Delta = (L_{\text{reduced}} - L_{\text{full}})/L_{\text{full}}$

Figure 6.9: Relative increase in loss due to model reduction $L_\Delta$. For a majority of the measurements, the increased error falls below 10%, and for almost all measurements, the increase is below 20%.

Table 6.1: Normalized Kullback-Leibler divergence values for VL parameter distributions. Element $i, j$ shows $D_{KL}(P_i \parallel P_j)$ where $P_i$ is the parameter distribution for test subject $i$.

<table>
<thead>
<tr>
<th>Test subject</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>0.14</td>
<td>0.61</td>
<td>0.57</td>
<td></td>
</tr>
<tr>
<td>S2</td>
<td>0.06</td>
<td>0.48</td>
<td>0.54</td>
<td></td>
</tr>
<tr>
<td>S3</td>
<td>1.0</td>
<td>0.59</td>
<td>0.86</td>
<td></td>
</tr>
<tr>
<td>S4</td>
<td>0.34</td>
<td>0.38</td>
<td>0.31</td>
<td></td>
</tr>
</tbody>
</table>

which can be expected given the fact that SPS performance decreases with age. This is also confirmed by studying the Mahalanobis distance between the individual parameter estimates of S3, and the four parameter distributions. Histograms of these distributions are presented in Fig. 6.11 and confirm that the distributions differ. Yet, subject age is apparently not the only factor behind SPS dissimilarity. There is a clear distinction in the distributions between S4 and S1, S2. Sex seems not to be important as the distributions for S1 and S2 are much alike.

6.5 Discussion

A study on four healthy subjects presented in the paper was concerned with the question whether or not it is possible to use SP movements for subject identification, i.e. for biometrics. By means of a parsimonious linear VL model, parameter distributions were estimated from the experimental data in DB2. The KLD values calculated for the estimates pairwise show that
6.5. Discussion

Figure 6.10: Distribution of the Linear VL parameter $\gamma_1(0)$ for the four different test subjects.

Figure 6.11: Mahalanobis distances between parameter estimates of individual data sets from S3, and the four parameter distributions. The distance to the S3 distribution is often shorter than to the other distributions.
the distributions are quite different from each other. While direct subject identification using eye-tracking measurements from a single SP experiment seems infeasible, one can still conclude that there are individual differences captured by the VL model on a distribution level. This study was intended to illustrate a potential use of the modeling paradigm at hand and therefore limited to a few subjects. It is interesting to consider the implications of a larger set of participants. What are the limits for similarity and dissimilarity between people, and how common is a given degree of similarity? More concretely: with what certainty one can claim that a distribution of parameters belongs to a certain person? Furthermore, it is also interesting to consider the dependence among the parameters here. While the model is parsimonious and its free parameters have been made fewer by introducing linear functional relations, the relations themselves are in turn calculated from the parameter estimates on a population level. While reducing the variance in the parameter estimates through more parsimonious model, there still variance in the functional relation estimates.

The visual stimuli used in the experiments were designed to possess sufficient excitation in both frequency and amplitude content, so that the SPS dynamics can be properly examined. The stimuli design is especially important in disease when nonlinearities are expected, and the present method differs from numerous medical studies of the SPS, where much simpler signal shapes, such as sine waves, ramps, and constants, are considered and presented to the patient at a constant velocity. Neither of the standard medical test signals satisfies the excitation conditions normally required in nonlinear system identification.

6.6 Conclusions

A complete approach to Volterra modeling of the SPS is presented. The utility of sparse VL models with functional relations between the coefficients and capturing the SPS dynamics is demonstrated in two experimental settings.

While single measurements might not be useful for differentiating between subjects, comparing probability distributions evaluated from many measurements offers a better alternative. However, although the distributions were found to be substantially different, the approach still cannot guarantee reliable subject identification.
References


Title
Identification of Continuous Volterra Models with Explicit Time Delay through Series of Laguerre Functions

Authors
Viktor Bro and Alexander Medvedev

Edited version of
Identification of Continuous Volterra Models with Explicit Time Delay through Series of Laguerre Functions

Abstract

The problem of estimating nonlinear time-delay dynamics captured by continuous Volterra models from input-output data is treated. The delayed Volterra kernels are seen as impulse responses of time-invariant systems with time delay. Analytical expressions for the Laguerre series, where the Laguerre coefficients of the finite-dimensional part are admixed with the terms due to the delay are provided. By utilizing the linearity of Volterra-Laguerre models in the unknown parameters, the model is estimated by a nonlinear least-squares method. An application of the proposed approach to the problem of Volterra modelling of the human smooth pursuit system from eye-tracking data is provided. The proposed approach demonstrates consistently accurate performance on both simulated and experimental data.

7.1 Introduction

Time delays appear often in many applications, where nonlinear models are required. Volterra models offer a black-box system identification framework for smooth nonlinear systems, see e.g. [16]. A classical way of parameterizing a Volterra model is to expand the kernels in an orthonormal basis. When the kernels are smooth and expected to vanish at infinity, Laguerre functions constitute a useful choice of basis. Volterra models with the kernels written in terms of Laguerre series are termed Volterra-Laguerre (VL) models. Since Volterra models exist in continuous and discrete time formulations, both continuous and discrete Laguerre functions come in handy. In practice,
only truncated Laguerre series are used, implying that the arbitrary kernels are assumed to be approximated by impulse responses of finite-dimensional linear time-invariant systems.

VL models \cite{14,16} are linear in the coefficients of the Laguerre series and, in principle, can be estimated by linear methods, e.g. least squares. Notice that the well-known shortcoming of VL models possessing a large number of model parameters can be overcome by sparse estimation \cite{10,11,17} and exploiting functional dependencies between the coefficients. Obtained in such way minimal VL models have been used, for instance, to describe the human smooth pursuit mechanism and capture the alternations in it due to Parkinson’s disease \cite{2}.

Volterra models with time delay are seldom addressed and only a few papers dealing with their estimation can be found. This is ostensibly because the general structure of the model readily includes time delays but in an implicit form. Indeed, the Volterra kernels are not restricted to a delay-free case. Naturally, when dealt with in discrete time, the delay in a Volterra model does not have to be explicitly parameterized but can be included in the kernel. In continuous time, a Volterra model with time delay is infinite-dimensional and, therefore, has to be properly handled in estimation.

The paper is organized as follows. First, the necessary background on Laguerre functions with focus on modelling time-delay systems is provided. Then, the VL model with time delay is introduced and its properties are highlighted in time and Laplace domain. Further, methods for identifying the VL model with delay are presented and their performance is illustrated by numerical experiments. Finally, an application of the proposed identification approach to the estimation of the human smooth pursuit system is described to illustrate the utility of the obtained results.

### 7.2 Problem formulation

The dynamics of a smooth finite-dimensional nonlinear dynamical multi-input multi-output system with the input $u(t)$ and output $y(t)$ can be captured by the Volterra model

\[
y(t) = y_0 + \int k_1(\theta_1)u(t - \theta_1) \, d\theta_1 \\
+ \int\int k_2(\theta_1, \theta_2)u(t - \theta_1)u(t - \theta_2) \, d\theta_1 \, d\theta_2 + \ldots
\]

completely defined by the kernels \{\(k_1(\cdot), k_2(\cdot), \ldots\)\}. Assume now, under zero initial conditions on the system, that $u(t)$ is subject to a constant delay $u(t) = H(t - \tau)v(t - \tau)$, where $H(\cdot)$ is the Heaviside function.
This paper deals with the estimation of the kernels \( \{k_1, k_2, \ldots \} \) and the delay value \( \tau \) from the input-output data \( y(t), v(t), t \in [0, \infty) \).

### 7.3 The continuous Laguerre functions and polynomials

Both Laguerre functions \( l_k(\cdot) \) and Laguerre polynomials \( L_n^{(\alpha)}(\cdot) \) are utilized in what follows. The notation for the Laguerre functions differs here from the standard one in order to avoid confusion with the Laguerre polynomials.

In the Laplace domain, the continuous Laguerre functions can be expressed as

\[
\mathcal{L}\{l_k(t)\} = \ell_k(s) = \frac{\sqrt{2p}}{s + p} \left( \frac{s - p}{s + p} \right)^k, \quad p > 0,
\]

where the constant \( p \) is called the Laguerre parameter. The functions \( \ell_k(s), k = \{0, \infty\} \) constitute an orthonormal complete basis in \( \mathbb{H}_2 \) with respect to the inner product

\[
\langle W, V \rangle = \frac{1}{2\pi i} \int_{-\infty}^{\infty} W(s)V(-s) \, ds. \tag{7.2}
\]

Further, \( k \)-th Laguerre coefficient of \( W(s) \in \mathbb{H}_2 \) is evaluated as a projection of \( W(s) \) onto \( \ell_k(s) \)

\[
w_k = \langle W, \ell_k \rangle,
\]

while the set \( w_k, k = \{0, \infty\} \) is referred to as the Laguerre spectrum of \( W(s) \). The time domain representations of the Laguerre functions \( l_k(t), k = \{0, \infty\} \) yield an orthonormal basis in \( \mathbb{L}_2[0, \infty) \).

The well-known family of associated Laguerre polynomials with \( \alpha \in \mathbb{R} \) is given explicitly by

\[
L_n^{(\alpha)}(x) = \sum_{i=0}^{n} \frac{1}{i!} \binom{\alpha + n}{n - i} (-x)^i, \quad n = 0, 1, 2, \ldots \tag{7.3}
\]

The first few associated Laguerre polynomials are

\[
L_0^{(\alpha)}(x) = 1,
L_1^{(\alpha)}(x) = -x + \alpha + 1,
L_2^{(\alpha)}(x) = \frac{1}{2} \left( x^2 - 2(\alpha + 2)x + (\alpha + 1)(\alpha + 2) \right)
\]

The associated Laguerre polynomials are orthogonal in the sense of

\[
\int_{0}^{\infty} x^\alpha e^{-x} L_n^{(\alpha)}(x) L_m^{(\alpha)}(x) \, dx = 0, \quad m \neq n, \alpha \geq -1. \tag{7.4}
\]
For $\alpha < -1$, Laguerre polynomials are no longer orthogonal on the positive real axis but a milder property applies.

### 7.4 Volterra model with time delay

For $u(t), y(t) \in [0, \infty)$, Volterra model (7.1) becomes

$$y(t) = y_0 + \int_0^t k_1(\theta_1)u(t - \theta_1) \, d\theta_1$$

$$+ \int_0^t \int_0^t k_2(\theta_1, \theta_2)u(t - \theta_1)u(t - \theta_2) \, d\theta_1 \, d\theta_2 + \ldots$$

(7.5)

Assuming non-anticipatory dynamics, i.e. the case when $y(t)$ is uniquely defined by $u(\theta) : \theta \in [0, t]$, this is equivalent to

$$y(t) = y_0 + \int_0^\infty k_1(\theta_1)u(t - \theta_1) \, d\theta_1$$

$$+ \int_0^\infty \int_0^\infty k_2(\theta_1, \theta_2)u(t - \theta_1)u(t - \theta_2) \, d\theta_1 \, d\theta_2 + \ldots$$

(7.6)

The one-sided formulation of (7.5) allows to consider initial conditions on the model dynamics. In what follows, it is assumed that $k_p \in L^2(R_\geq 0), p = 1, 2, \ldots$ for bounded-input bounded-output stability of the model.

**Proposition 1.** Consider a second-order Volterra model given by the three first terms of (7.6). Then, in Laplace domain, the model is given by

$$Y(s) = \frac{y_0}{s} + K_1(s)U(s) +$$

$$\frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} K_2(z, s - z)U(z)U(s - z) \, dz,$$  

where the constant $c$ is defined by the domain of convergence for $U(s)$.

**Proof.** The second-order term in Volterra model (7.6) reads

$$q_2(t) = \int_0^\infty k_2(\theta_1, \theta_2)u(t - \theta_1)u(t - \theta_2) \, d\theta_1 \, d\theta_2,$$  

(7.7)

where $k_2(\cdot, \cdot)$ is a symmetrical kernel, i.e. $k_2(t_1, t_2) = k_2(t_2, t_1)$. In Laplace domain

$$\mathcal{L}\{q_2(t)\} = Q_2(s)$$

$$= \int_0^\infty k_2(\theta_1, \theta_2)\mathcal{L}\{u(t - \theta_1)u(t - \theta_2)\} \, d\theta_1 \, d\theta_2.$$  

(7.8)
Using the property of Laplace transform for the product of two functions in time domain \([1, Table 14.1, p. 385]\), it follows that

\[
\mathcal{L}\{u(t - \theta_1)u(t - \theta_2)\} = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} U(z)e^{-z\theta_1}U(s-z)e^{-(s-z)\theta_2} \, dz,
\]

where the constant \(c\) is defined by the domain of convergence for \(U(s)\). Now, by substituting the relationship above into (7.8) and changing the order of integration, one obtains

\[
Q_2(s) = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} \int_0^{\infty} \int_0^{\infty} k_2(\theta_1, \theta_2)e^{-z\theta_1}e^{-(s-z)\theta_2} \, d\theta_1 \, d\theta_2 U(z)U(s-z) \, dz.
\]

Introduce the multi-dimensional Laplace transform as

\[
\mathcal{L}\{f(t_1, \ldots, t_k)\} = F(s_1, \ldots, s_k) = \int_0^{\infty} \cdots \int_0^{\infty} f(t_1, \ldots, t_k)e^{-s_1t_1} \cdots e^{-s_k t_k} \, dt_1 \cdots dt_k.
\]

Then, it turns out that

\[
\int_0^{\infty} \int_0^{\infty} k_2(\theta_1, \theta_2)e^{-z\theta_1}e^{-(s-z)\theta_2} \, d\theta_1 \, d\theta_2 = K_2(z, s-z),
\]

which fact leads to

\[
Q_2(s) = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} K_2(z, s-z)U(z)U(s-z) \, dz.
\]

The integral above represents a two-dimensional convolution in complex variables and immediately leads to the sought result.

It is worth noticing that the exact nature of the kernels \(k_p(\cdot), p = 1, 2, \ldots\) in the Volterra model is not specified and a kernel can be a solution of a finite- or infinite-dimensional system of differential equations. Naturally, a kernel does not have to satisfy a differential equation at all (e.g. being non-differentiable), but this scenario is not relevant to the identification problem at hand.

For the purpose of estimation, parameterize the delay explicitly by considering Volterra model \([7.5]\) fed with the delayed input \(u(t) = v(t - \tau)\). This restricts the model to cover the cases of input delay, output delay or the same delay value in all the kernels. Notice also that the constant component \(y_0\) of the output signal \(y(t)\) arises in response to the input \(v(t)\) and
is also delayed in the time domain. Using properties of the convolution operator, (7.5) can also be written as

\[ y(t) = y_0 + \int_0^t k_1(t - \theta_1)u(\theta_1)\,d\theta_1 \]

\[ + \int_0^t \int_0^t k_2(t - \theta_1, t - \theta_2)u(\theta_1)u(\theta_2)\,d\theta_1\,d\theta_2 + \ldots \]

Then, for the delayed input, it follows

\[ y(t) = y_0 H(t - \tau) + \int_{t-\tau}^t k_1(t - \tau - \theta_1)v(\theta_1)\,d\theta_1 + \]

\[ \int_0^{t-\tau} \int_{t-\tau}^t k_2(t - \tau - \theta_1, t - \tau - \theta_2)v(\theta_1)v(\theta_2)\,d\theta_1\,d\theta_2 + \ldots, \]

thus implying the result below.

**Corollary 1.** Consider the same Volterra model as in Proposition 7 but with the delayed input

\[ u(t) = v(t - \tau)H(t - \tau). \]

Then the input-output model description in Laplace domain becomes

\[ Y(s) = e^{-\tau s} \left( \frac{y_0}{s} + K_1(s)V(s) \right) \]

\[ + \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} K_2(z, s-z)V(z)V(s-z)\,dz. \]

An observation relevant to the rest of the paper following from Corollary 1 is that the delay can be equally attributed to the kernels of the model as to the input or output signal. Also, if \( L^{-1}\{K_p\} \in L_2(\mathbb{R}_{\geq 0}^p), p = 1, 2, \ldots \) then \( L^{-1}\{e^{-\tau s}K_p\} \in L_2(\mathbb{R}_{\geq 0}^p), p = 1, 2, \ldots \) and the Laguerre functions constitute a (complete) basis for the delayed kernels. This property is exploited in Laguerre approximation of linear time-invariant systems, e.g. [13].

### 7.5 Volterra-Laguerre model

Consider now the \( N \)-th term in Volterra series (7.9)

\[ q_N(t) = \int_0^t \cdots \int_0^t k_N(\theta_1, \ldots, \theta_N)u(t - \theta_1) \cdots u(t - \theta_N)\,d\theta_1 \cdots d\theta_N. \]
In general, \( k_N \in \mathbb{L}_2(\mathbb{R}_{\geq 0}^N) \) can be written as a multidimensional Laguerre series \[15\]. For a separable kernel, which property is further assumed to hold, it suffices to use the conventional Laguerre functions in each dimension, yielding

\[
k_N(\theta_1, \ldots, \theta_N) = 
\sum_{j_1=0}^{\infty} \cdots \sum_{j_N=0}^{\infty} \gamma_N(j_1, \ldots, j_N) l_{j_1}(\theta_1) \cdots l_{j_N}(\theta_N).
\]

Notice here that by approximating the Volterra kernels with truncated Laguerre series, it is implicitly assumed that the kernels are impulse responses of finite- or infinite-dimensional linear time-invariant systems. The dynamics underlying a kernel can be recovered (through a realization algorithm) from the Laguerre coefficients of the series in the framework of Laguerre-domain system identification, see \[4, 6\].

Denote the convolution integrals between the input signal \( u(t) \) and the Laguerre functions, also known as the outputs of Laguerre filters, by

\[
\psi_j(t) = \int_0^t l_j(\theta_n) u(t - \theta) \, d\theta.
\]

Then (7.11) can be expressed as

\[
q_N(t) = \sum_{j_1=0}^{\infty} \cdots \sum_{j_N=0}^{\infty} \gamma_N(j_1, \ldots, j_N) \psi_{j_1}(t) \cdots \psi_{j_N}(t),
\]

and Volterra series (7.9) is given by

\[
y(t) = y_0 + \sum_{N=1}^{\infty} \sum_{j_1=0}^{\infty} \cdots \sum_{j_N=0}^{\infty} \gamma_N(j_1, \ldots, j_N) \psi_{j_1}(t) \cdots \psi_{j_N}(t).
\]

In particular, \( k_2 \in \mathbb{L}_2(\mathbb{R}_{\geq 0}^2) \) can be written as

\[
k_2(\theta_1, \theta_2) = \sum_{j_1=0}^{\infty} \sum_{j_2=0, j_2= j_1}^{\infty} \gamma_2(j_1, j_2) l_{j_1}(\theta_1) l_{j_2}(\theta_2).
\]

Then, in view of (7.7), one has

\[
\mathcal{L}\{q_2(t - \tau)\} = \frac{1}{2\pi i} \sum_{j_1=0}^{\infty} \sum_{j_2=0, j_2= j_1}^{\infty} \gamma_2(j_1, j_2) \cdot
\int_{c-i\infty}^{c+i\infty} \ell_{j_1}(z) \ell_{j_2}(s - z) U(z) U(s - z) \, dz \cdot e^{-\tau s}.
\]
To recapitulate, a second-order VL model can thus be written as

\[ y(t) = y_0 + \sum_{j=0}^{\infty} \gamma_1(j) \psi_j(t) + \sum_{j_1=0}^{\infty} \sum_{j_2=j_1}^{\infty} \gamma_2(j_1, j_2) \psi_{j_1}(t) \psi_{j_2}(t) \]  

(7.15)

7.5.1 Time-delay VL model

Consider a linear VL model representing a linear time-invariant system

\[ y(t) = \sum_{j=0}^{\infty} \gamma_1(j) \psi_j(t) \]

\[ = \sum_{j=0}^{\infty} \gamma_1(j)(l_j * u)(t) \]

\[ = \left[ \left( \sum_{j=0}^{\infty} \gamma_1(j)l_j \right) * u \right](t). \]  

(7.16)

Evidently, the system output is given by a convolution between a Laguerre series representing the impulse response of the system and the input signal.

Lemma 1 ([7]). Consider linear VL model (7.16) with a delayed input \( u(t) = v(t - \tau) \). Then, the model output is given by

\[ y(t) = \sum_{j=0}^{\infty} \gamma_1(j) \psi_j^\tau(t) = \sum_{j=0}^{\infty} \gamma_1^\tau(j) \psi_j(t), \]  

(7.17)

where \( \kappa = 2p\tau \) and

\[ \gamma_1^\tau(j) = e^{-\kappa/2} \left( \gamma_1(j) + \sum_{m=1}^{j} L_m(-1)(\kappa) \gamma_1(j - m) \right), \]  

(7.18)

\[ \psi_j^\tau(t) = \int_0^{t-\tau} l_j(\theta_n)v(t - \tau - \theta) \, d\theta. \]

Proposition 2. The second-order time-delay Volterra model

\[ y(t) = y_0 H(t - \tau) + \int_0^{t-\tau} k_1(t - \tau - \theta_1)v(\theta_1) \, d\theta_1 + \int_0^{t-\tau} \int_0^{t-\tau} k_2(t - \tau - \theta_1, t - \tau - \theta_2)v(\theta_1)v(\theta_2) \, d\theta_1 \, d\theta_2, \]  

(7.19)
with a separable kernel $k_2(\cdot, \cdot) \in L_2(\mathbb{R}^2_{\geq 0})$ is equivalently given by

$$y(t) = \sum_{j=0}^{\infty} \gamma_1^\tau(j) \psi_j(t) + \sum_{j_1=0}^{\infty} \sum_{j_2=j_1}^{\infty} \gamma_2^\tau(j_1, j_2) \psi_{j_1}(t) \psi_{j_2}(t),$$

(7.20)

where $\gamma_1^\tau(j)$ is specified by (7.18) and

$$\gamma_2^\tau(j_1, j_2) = e^{-\kappa} \sum_{w=0}^{j_1} L^{(-1)}_w(\kappa) \sum_{m=0}^{j_2-j_1+w} L^{(-1)}_m(\kappa) \gamma_2(j_1 - w, j_2 - m).$$

(7.21)

**Proof.** Consider the quadratic term of the delay-free VL model

$$q_2(t) = \sum_{j_1=0}^{\infty} \sum_{j_2=0}^{\infty} \gamma_2(j_1, j_2) \psi_{j_1}(t) \psi_{j_2}(t).$$

(7.22)

By exploiting the separability, one obtains

$$q_2(t) = \sum_{j_1=0}^{\infty} \sum_{j_2=0}^{\infty} \lambda_{j_1} \mu_{j_2} \psi_{j_1}(t) \psi_{j_2}(t)$$

$$= \sum_{j_1=0}^{\infty} \lambda_{j_1} \psi_{j_1}(t) \sum_{j_2=0}^{\infty} \mu_{j_2} \psi_{j_2}(t).$$

(7.23)

Introducing the delay $\tau$ yields

$$q_2(t - \tau) = \sum_{j_1=0}^{\infty} \sum_{j_2=0}^{\infty} \lambda_{j_1} \mu_{j_2} \psi_{j_1}^\tau(t) \psi_{j_2}^\tau(t)$$

$$= \sum_{j_1=0}^{\infty} \lambda_{j_1} \psi_{j_1}^\tau(t) \sum_{j_2=0}^{\infty} \mu_{j_2} \psi_{j_2}^\tau(t)$$

$$= \sum_{j_1=0}^{\infty} \lambda_{j_1} \psi_{j_1}^\tau(t) \sum_{j_2=0}^{\infty} \mu_{j_2} \psi_{j_2}^\tau(t)$$

$$= \sum_{j_1=0}^{\infty} \sum_{j_2=0}^{\infty} \gamma_2^\tau(j_1, j_2) \psi_{j_1}(t) \psi_{j_2}(t).$$

(7.24)
Making use of the identity $\psi_{j_1}(t)\psi_{j_2}(t) = \psi_{j_2}(t)\psi_{j_1}(t)$, the relationship above can be simplified as

$$q_2(t - \tau) = \sum_{j_1=0}^{\infty} \lambda^\tau_{j_1} \psi_{j_1}(t) \sum_{j_2=0}^{\infty} \mu^\tau_{j_2} \psi_{j_2}(t)$$  \hfill (7.25)

$$= \sum_{j_1=0}^{\infty} \sum_{j_2=0}^{\infty} \gamma^\tau_{2}(j_1, j_2) \psi_{j_1}(t)\psi_{j_2}(t).$$

Now, taking (7.18) into account results in

$$\gamma^\tau_{2}(j_1, j_2) = \lambda^\tau_{j_1} \mu^\tau_{j_2}$$

$$= e^{-\kappa/2} \left( \lambda_{j_1} + \sum_{m=1}^{j_1} L^{(-1)}_{m(\kappa)} \lambda_{j_1-m} \right)$$

$$\times e^{-\kappa/2} \left( \mu_{j_2} + \sum_{m=1}^{j_2} L^{(-1)}_{m(\kappa)} \mu_{j_2-m} \right).$$  \hfill (7.26)

Performing the multiplications and applying the identity $\lambda_{j_1} \mu_{j_2} = \gamma_2(j_1, j_2)$ leads to (7.21). Together with the result of Lemma 1, this completes the proof. \hfill \square

### 7.6 Estimation of VL models

To estimate a general VL model given by (7.5) (without an explicit delay), consider two vectors: $Y = [y(0) \ldots y(T)]^T$ of point values of the output, and point values of the Laguerre filter outputs $\Psi_j = [\psi_j(0) \ldots \psi_j(T)]^T$, $j = 0, \ldots, L$.

For the second-order VL model ($N = 2$), construct the following matrix using the Laguerre convolution products as entries

$$\Psi_{0:T} = \begin{bmatrix} 1 & \psi_0(0) & \cdots & \psi_0(0)\psi_0(0) & \cdots & \psi_L(0)\psi_L(0) \\ \vdots & \ddots & \ddots & \vdots \\ 1 & \psi_0(T) & \cdots & \psi_0(T)\psi_0(T) & \cdots & \psi_L(T)\psi_L(T) \end{bmatrix}.$$  

Then the parameter vector

$$\Gamma = [y_0 \ \gamma_1(0) \ \cdots \ \gamma_2(0,0) \ \cdots \ \gamma_2(L,L)]^T.$$  

satisfies

$$y = \Psi_{0:T}\Gamma.$$  

When $\Psi_{0:T}$ is of full rank, i.e. $\text{rank}(\Psi_{0:T}) = \binom{L+N+1}{N}$, the parameter vector can be recovered as

$$\Gamma = (\Psi_{0:T}^T\Psi_{0:T})^{-1}\Psi_{0:T}^Ty.$$  \hfill (7.27)
7.6.1 Second-order VL model with delay

Let $\Gamma^\tau$ be the parameter vector characterizing second-order VL model (7.20),

$$\Gamma^\tau = \begin{bmatrix} y_0 & \gamma_1(0) & \ldots & \gamma_2(0,0) & \ldots & \gamma_2^\tau(L,L) \end{bmatrix}^T,$$  \hspace{1cm} (7.28)

with the elements given by the relationships (7.18) and (7.21). Denote with $\Gamma^0$ the parameter vector corresponding to the system without delay, i.e. (7.15)

$$\Gamma^0 = \begin{bmatrix} y_0 & \gamma_1(0) & \ldots & \gamma_2(0,0) & \ldots & \gamma_2(L,L) \end{bmatrix}^T.$$  \hspace{1cm} (7.29)

The following result specifies the effect of the time delay on the coefficients of the second-order VL model.

**Proposition 3.** Consider the block matrix

$$T(\kappa) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & T_1(\kappa) & 0 \\ 0 & 0 & T_2(\kappa) \end{bmatrix},$$

where $T_1(\kappa)$ is a lower-triangular Toeplitz matrix

$$T_1(\kappa) = e^{-\kappa/2} \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ L_1^{(-1)}(\kappa) & 1 & 0 & \ldots & 0 \\ L_2^{(-1)}(\kappa) & L_1^{(-1)}(\kappa) & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ L_L^{(-1)}(\kappa) & L_{L-1}^{(-1)}(\kappa) & \ldots & \ldots & 1 \end{bmatrix},$$

and the matrix $T_2(\kappa)$ is a block matrix structured as

$$T_2(\kappa) = e^{-\kappa} \begin{bmatrix} T_2^{(0,0)}(\kappa) & 0 & \ldots & 0 \\ T_2^{(1,0)}(\kappa) & T_2^{(1,1)}(\kappa) & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ T_2^{(L,0)}(\kappa) & T_2^{(L,1)}(\kappa) & \ldots & T_2^{(L,L)}(\kappa) \end{bmatrix},$$

with the Toeplitz matrix block $T_2^{(a,b)}(\kappa) \in \mathbb{R}^{(L+1-a) \times (L+1-b)}$, whose element $i,j$ is defined by

$$T_2^{(a,b)}(\kappa)\{i,j\} = \begin{cases} L_{a-b}^{(-1)}(\kappa)L_{i-j+a-b}^{(-1)}(\kappa), & i - j + a - b \geq 0 \\ 0, & a - b \geq 0 \end{cases}.$$
Then, for given $\tau$ and $p$, the mapping $\Gamma^0 \mapsto \Gamma^\tau$ is linear and parameterized by the matrix $T(\kappa)$, i.e

$$\Gamma^\tau = T(\kappa)\Gamma^0.$$ 

Proof. The structure of $T_1(\kappa)$ is specified by (7.18). Partition now $T_2(\kappa)$ as

$$T_2(\kappa) = e^{-\kappa} \begin{bmatrix}
T_2^{(0,0)}(\kappa) & T_2^{(0,1)}(\kappa) & \cdots & T_2^{(0,L)}(\kappa) \\
T_2^{(1,0)}(\kappa) & T_2^{(1,1)}(\kappa) & \cdots & T_2^{(1,L)}(\kappa) \\
& & \ddots & \\
T_2^{(L,0)}(\kappa) & T_2^{(L,1)}(\kappa) & \cdots & T_2^{(L,L)}(\kappa)
\end{bmatrix}.$$ 

It is seen from (7.21) that $T_2^{(a,b)}(\kappa) \in \mathbb{R}^{(L+1-a) \times (L+1-b)}$ and describes the mapping $\gamma_2(b, \cdot) \mapsto \gamma_2^{(a, \cdot)}$. From this relation one can also calculate the elements of each block as

$$T_2^{(a,b)}(\kappa)\{i,j\} = 
\begin{cases}
L_{i-j+a-b}(\kappa), & i-j+a-b \geq 0 \\
0, & a-b \geq 0,
\end{cases}$$

Thus, from the properties of the associated Laguerre polynomials, $T_2(\kappa)$ is a lower triangular matrix and each element on the main diagonal of $T_2(\kappa)$ is equal to 1. 

7.7 The Smooth Pursuit System

The developed estimation method is applied in this section to the modelling of the human smooth pursuit system (SPS). Smooth pursuit is an involuntary eye movement enabling tracking of a steadily moving target and controlled by complex neural circuitry. Mathematical modeling of SPS has important medical applications since SPS is compromised in many neurological conditions [5, 12] and can be used for symptom quantification in neurology [2, 8].

Time delay in eye-tracking experiments comes primarily from digital data processing and depends on the hardware and software of the tracking device. This circumstance complicates comparison of mathematical models obtained from data collected on different platforms. By estimating the time delay in the eye tracker, data sets can be properly aligned in time, and, therefore the impact of delay on the model coefficients can be alleviated.

This section presents a simulation example to illustrate the benefits and shortcomings of the possible approaches to the estimation of the considered VL model with time delay, as presented in Section 7.6. Then the approaches are applied to experimental data sets, where ground truth is not available.
7.7.1 Simulation data

To investigate the feasibility of the proposed time-delay estimation method, numerical experiments have been performed with data obtained by simulating for about 25 seconds a continuous VL systems with of Laguerre order $L = 3$ and nonlinearity degree $N = 2$, $\tau = 1$, and different Laguerre parameters $p$. The parameters of the model were sampled as random uniformly distributed numbers such that $\gamma_1(j_1) \sim U(-1,1)$ and $\gamma_2(j_1,j_2) \sim U(-0.01,0.01)$ for $j_1,j_2 = 0,\ldots,L$, a selection consistent with the parameter values estimated from smooth pursuit experiments with a delay-free model. In actual application, the system dynamics are only approximated by a VL model and $p$ is usually selected to obtain the best (output) data fit. Naturally, the choice of $p$ influences the accuracy of the identification result, similarly to the sampling time in sampled-data identification. Indeed, the continuous fast dynamics and the time delay become difficult to estimate with longer sampling times.

Analytical approach

The autoregressive form of (7.18) and (7.21) cause trouble when used in practice. When a delay is present in the kernel functions, coefficients of high order will have significant nonzero values even if this was not the case in the kernel without delay. In a practical application, the Laguerre expansions must be truncated and information will necessarily be lost. It turns out that even for small delays in the kernels, the amount of information lost is too large to reconstruct the kernel functions without delay in a satisfactory manner.

Gridding approach

In many applications of VL models, including eye tracking, the interval of values for the time delay is known beforehand. This allows to assume a certain delay $\tau$, shift the measured system output (or input) for this value backwards in time, and estimate the coefficients of the delay-free model by (7.27). The delay value within the assumed interval yielding the least value of the loss function is then taken as the delay estimate $\hat{\tau}$.

Assuming $p$ known, the Laguerre coefficients of the kernels were estimated by LS at each grid point for $\tau$. In Fig. 7.1 the output loss function is shown as function of $\hat{\tau}$, confirming that both the delay and the coefficients are estimated well when $p$ is known. The loss function used here is the mean squared error normalised with respect to the 2-norm of the measured signal. That is, for a measured signal $\{y_k\}_{k=1}^K$ of length $K$ and model output
7.7. The Smooth Pursuit System

Figure 7.1: Output loss for a VL model with $N = 2$ when the Laguerre parameter $p$ is known (right image shows a close-up around the true value $\tau = 1$). The delay as well as the parameters are estimated correctly. Notice flatter loss function character for higher values of $p$.

\[ \{\hat{y}\}_{k=1}^{K} \]

\[ \text{NMSE} = \frac{1}{K\|y\|} \sum_{k=1}^{K} (y_k - \hat{y}_k)^2. \]

It is also observed that a larger value of the Laguerre parameter produces a flatter output loss function thus potentially lowering the delay estimation accuracy in case of modeling uncertainty. Furthermore, Fig. 7.1 indicates that, while there are several local minima in the loss function, the problem is locally convex. This fact is encouraging with respect to the use of gradient-based optimization methods, provided proper initialization.

Further, the impact of unknown value of the Laguerre parameter on the delay estimation is investigated. Here, the same VL system as above was simulated with $p = 5$. The time delay and coefficients were then estimated assuming different values of $p$. An assumed incorrect value of the Laguerre parameter leads to estimation bias, see Fig. 7.2. Since the output loss function achieves the minimum at the true value of $p$, it can also be obtained by gridding.

**Nonlinear least squares**

A nonlinear least squares (NLS) method was also used to estimate the VL parameters and the time delay, using an initial guess of $\Gamma = 0$ and $\tau = 0$. Fig. 7.3 confirms that the NLS algorithm arrives to the same minimum as the gridding approach with respect to the delay-free model parameters as well as the delay, when $p$ is known. When $p$ is unknown neither the parameters nor the delay is estimated correctly, as is shown in Fig. 7.4. Here, the parameter loss is defined as the mean squared error over the parameters.
Figure 7.2: Output loss function with unknown values of the Laguerre parameter $p$. Left plot shows output loss with respect to the estimated delay $\hat{\tau}$, with minima marked with red dots. The true value is $p = 5$. An estimation bias in the delay estimation appears if a wrong $p$ is assumed. The true delay value is $\tau = 1$. The right plot shows the output loss with respect to the value of the Laguerre parameter $p$. There is a minimum at the true value of $p$.

the NLS has to be initialized in a proper way in order to avoid the local minima.

### 7.7.2 Experimental data

A moving dot was presented on a computer screen to the test subject, and the gaze position was recorded for the duration of the experiment. The gaze position is defined as the coordinates on the computer screen where the gaze is directed. The visual stimulus, i.e. the moving pattern of the dot, was designed to provide sufficient excitation both amplitude- and frequency-wise so that possible nonlinearities in the SPS dynamics would be revealed. A description of the stimulus design through optimization is found in [9].

Each eye-tracking experiment results in an input-output pair of discrete-time 2D signals, $u(k), y(k)$. The stimulus $u(k)$ then acts as the input to the SPS that outputs the gaze position $y(k)$. The experiment duration is approximately 25 s long, and the eye tracker has a sampling frequency of 60 Hz. One data set therefore contains about 1500 samples. Only the horizontal parts of the signals are considered in this paper. The vertical components may however be used in the same way. An example of the horizontal part of the stimulus is depicted in Fig. 7.5.

The time delay estimates summarized in Table I indicate a latency in the eye tracking data in the range $0 - 0.308s$, when estimated with NLS, and in the range $0 - 0.117s$, when estimated with gridding. The optimal
Figure 7.3: Output and parameter loss for NLS estimation of VL parameters and time delay $\tau$, compared to gridding over values of $\tau$ with $p = 5$. The correct parameter and delay values are found.

Figure 7.4: Output and parameter loss for NLS estimation of VL parameters and time delay $\tau$ when the Laguerre parameter $p$ is unknown. It is evident that an incorrect guess of the Laguerre parameter gives bad results.
Figure 7.5: Example stimulus, horizontal part.

Table 7.1: Time delay estimation results from experimental eye-tracking data

<table>
<thead>
<tr>
<th>Set</th>
<th>$\hat{\rho}$</th>
<th>$\hat{\tau}$</th>
<th>NMSE</th>
<th>$\hat{\rho}$</th>
<th>$\hat{\tau}$</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.5</td>
<td>0.117</td>
<td>0.21</td>
<td>6.5</td>
<td>0.115</td>
<td>0.03</td>
</tr>
<tr>
<td>2</td>
<td>2.5</td>
<td>0</td>
<td>2.00</td>
<td>12.5</td>
<td>0.308</td>
<td>0.35</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>2.34</td>
<td>1</td>
<td>0</td>
<td>0.46</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>0.017</td>
<td>0.20</td>
<td>13</td>
<td>0.001</td>
<td>0.03</td>
</tr>
<tr>
<td>5</td>
<td>2.5</td>
<td>0.050</td>
<td>0.26</td>
<td>2.5</td>
<td>0.055</td>
<td>0.04</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>0</td>
<td>0.23</td>
<td>10.5</td>
<td>0.025</td>
<td>0.03</td>
</tr>
<tr>
<td>7</td>
<td>5.5</td>
<td>0</td>
<td>1.18</td>
<td>6</td>
<td>0.088</td>
<td>0.19</td>
</tr>
<tr>
<td>8</td>
<td>13</td>
<td>0</td>
<td>0.20</td>
<td>15</td>
<td>0.093</td>
<td>0.03</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>0.117</td>
<td>0.20</td>
<td>4</td>
<td>0.122</td>
<td>0.03</td>
</tr>
</tbody>
</table>
7.8 Conclusions

The effect of a time delay on the kernel coefficients in the continuous Volterra-Laguerre (VL) model is investigated. Analytical expressions relating the coefficients of the delay-free model to those with time delay are derived. Both the time-delay value and the kernel coefficients of the VL model can therefore be estimated by means of standard linear regression techniques with suitable regularization. In the proposed time-delay estimation framework, the Laguerre parameter of the utilized functional basis constitutes a degree of freedom that directly impacts the estimates and can be optimized. The case of a second-order VL model with an explicit time delay that can be attributed to either the model input (output) or the kernels is considered in detail. A numerically sound procedure of estimating the VL model with an

Figure 7.6: Estimated values of the Laguerre parameter $p$ and delay $\tau$ for 9 datasets, for original and time-shifted signals.

loss function value is generally lesser for NLS. This is quite natural since the search for the minimum was for each point initiated with the estimates obtained by the gridding approach. It should be noted that the value of the loss function depends on the expansion coefficients, as well as the time delay. This can explain the difference in loss value between gridding and NLS when the time delay estimates in the two approaches are quite similar.

**Validation of experimental data** Since the time delay in eye-tracking data cannot be measured directly, there is no ground truth to compare the estimates with. In order to validate the estimation method, the gaze direction signals acquired from the eye tracker had been shifted in time to eliminate the estimated delay and the parameters were re-estimated. As Fig. 7.6 shows, all delay estimates obtained from of the shifted data sets are zero, while the optimal values of the Laguerre parameter $p$ remain unchanged. This validates the estimation procedure.
explicit time delay by means of Nonlinear Least Squares is proposed. The approach at hand is applied to the problem of estimating the time delay in the human smooth pursuit modeling from eye-tracking data. The estimation algorithm is shown to produce accurate results on simulated and experimental data.
References


Recent licentiate theses from the Department of Information Technology


2019-001  Diane Golay: An Invisible Burden: An Experience-Based Approach to Nurses’ Daily Work Life with Healthcare Information Technology

2018-004  Charalampos Orfanidis: Robustness in Low Power Wide Area Networks

2018-003  Fredrik Olsson: Modeling and Assessment of Human Balance and Movement Disorders Using Inertial Sensors

2018-002  Tatiana Chistiakova: Ammonium Based Aeration Control in Wastewater Treatment Plants - Modelling and Controller Design

2018-001  Kim-Anh Tran: Static Instruction Scheduling for High Performance on Energy-Efficient Processors


2017-002  Germán Ceballos: Modeling the Interactions Between Tasks and the Memory System

2017-001  Diana Yamalova: Hybrid Observers for Systems with Intrinsic Pulse-Modulated Feedback

2016-012  Peter Backeman: New Techniques for Handling Quantifiers in Boolean and First-Order Logic

2016-011  Andreas Svensson: Learning Probabilistic Models of Dynamical Phenomena Using Particle Filters

2016-010  Aleksandar Zeljić: Approximations and Abstractions for Reasoning about Machine Arithmetic

Department of Information Technology, Uppsala University, Sweden