UNIVERSITY OF TWENTE.

Signals & Systems

Phenomenological modeling of the human tongue and lips

M.Sc. Thesis

B.K. Julsing

University of Twente
Department of Electrical Engineering,
Mathematics & Computer Science (EEMCS)
Signals & Systems Group (SAS)
P.O. Box 217
7500 AE Enschede
The Netherlands

Report Number:	SAS 16-09
Report Date:	December 4, 2009
Period of Work:	19/01/2009 – 10/12/2009
Thesis Committee:	Dr. ir. F.van der. Heijden
	drs. A. Kreeft
	Prof. Dr. ir. C.H. Slump

Abstract

This report describes a M.Sc. thesis project in which an exploration study has been performed to the development of a dynamic model of the human tongue and lips. This thesis project was part of a larger project in which a team of specialists in several fields works together to find a solution that enables pre-surgical assessment of function losses after surgical treatment of oral cancers. The ultimate goal is the development of a virtual environment in which a functional three-dimensional model of the oral cavity and pharynx can be used to predict patient specifically the consequences of surgical interventions on the post-operative functioning of the involved organs. Because of the complicated anatomical and muscular structure of organs like the tongue and lips, the project is focused on the development of a so-called *phenomenological black box-model*, instead of a complicated, physiological model of the underlying structures. The principle working of a phenomenological model relies on the hypothesis that an explicit causal relation can be established between groups of muscular activation signals and dynamic model variables describing the shape and motion of the tongue and the lips.

In this thesis project two of the main aspects in the development of such a phenomenological model are investigated. These aspects are methods for capturing and describing tongue and lip movements, and mathematical/statistical techniques for modeling dynamic systems. For the former an algorithm is developed that is able to automatically detect and track the tongue contour in (sequences of) magnetic resonance images. For a description of the dynamic behavior of the tongue and lips, linear state space models are investigated as possible frameworks. Although the current research was hampered by a lack of EMG data, in the near future this data does become available. The objective here was to already develop a possible dynamic model, which can be coupled to actual muscle activation signals in a later stage. Therefore, mathematical algorithms are derived and implemented for the estimation of input signals and system parameters from measured output variables. Performance of these models is evaluated by using data of lip movements. Although still a lot needs to be done to make the models empirically adequate, they at least show a proof of concept regarding the control of dynamic movements. Furthermore, a simple graphical user interface has been designed for the visualization and simulation of static and dynamic tongue and lip movements.

Acknowledgments

The present report is the result of my master thesis project, which I carried out at the chair Signals and System, a research group at the University of Twente (Enschede, The Netherlands) and in collaboration with the Netherlands Cancer Institute (NCI)/ Antoni van Leeuwenhoek Hospital (Amsterdam, the Netherlands). With this thesis project I concluded my studies Electrical Engineering and my life as a student, which I have been for almost six and a half years.

For about a small year I worked on the project in which it is aimed to develop a system that can assist surgeons with the removal of oral cancers. It was (and still is) a very challenging project, but hopefully with in the end a huge practical use in the medical world. Overall, I had a good time and I learned some new and interesting things in several fields. I also experienced the side activities that were involved in this thesis project very positive and useful. I would like to mention the visits to the NCI, Roessingh Research and Development and especially the visit to the scientific conference for surgeons and physicians in Nieuwegein, where I had the honor to present a poster about the project.

Therefore, I pay gratitude to some people for their support, aid and shared knowledge during the past time. First of all, I thank my supervisor Ferdi van der Heijden, who also introduced me to the project. Thanks for the useful discussions, conversations and advises on several fronts. Secondly, a lot of thanks to the people from the NCI, especially Fons Balm, Annemarijn Kreeft and Saar Muller. They provided the input for the project and made time to do actual experiments with the MRI scanner. Furthermore, I also thank my fellow students at the research group for their nice company. Although they did not always have a positive influence on my progress, working at the university would have been much more boring without them! Last but not least, I thank my parents for their continuous support and their sincerely inquisitiveness to all my activities during my study period.

Enschede, November 2009

Bram Julsing

Contents

Glossary

AAM Active Appearance Model

ASM Active Shape Model

BW Bandwidth

EMA Electromagnetic Articulography

EMG Electromyography

FA Flip Angle

FEM Finite Element Method

FFT Fast Fourier Transform

FOV Field Of View

GUI Graphical User Interface

HARP Harmonic Phase

MRI Magnetic Resonance Imaging

NIS Normalized Innovations Squared

NSA Number of Signal Averages

PCA Principal Component Analysis

PDE Partial Differential Equation

PDM Point Distribution Model

RF Radio Frequency

TE Echo Time

TR Repetition Time

TSE Turbo-Spin-Echo

1

Introduction

This M.Sc. thesis is concerned with the exploration of a dynamic functional model of the human tongue and lips. The thesis project is part of a larger project in which a team of specialists works together to find a solution that enables pre-surgical assessment of function losses after surgical treatment of oral cancers. The project team consists of specialist in the field of surgical oncology, surface electromyography, imaging, image analysis, and signal processing. The thesis project is executed at the research group Signals and Systems at the University of Twente (Enschede, the Netherlands) and in collaboration with the Netherlands Cancer Institute / Antoni van Leeuwenhoek Hospital (Amsterdam, the Netherlands).

This introduction chapter start with a short introduction to oral cancer and the problem with the current treatment possibilities. Next a description of the ultimate project goal will be given, followed by a formulation of the scope of this specific thesis project. The introduction concludes with an outline of the content of the report.

1.1 Oral cancer and treatment

Oral or mouth cancer represents about 3% of all cancers [1]. It can occur anywhere in the mouth (oral cavity) or pharynx (the part of the throat at the back of the mouth) which work together to allow breathing, talking, eating, chewing and swallowing. Oral cancer most commonly involves the tissue of the tongue and lips. A tongue or lip tumor can be very painful and awkward and can - in the worst case - even lead to death. Annual rates for oral cavity cancer deaths in the Netherlands are about 1.5 men and 0.8 women per 100,000 population [1]. Although the exact cause of oral cancer remains unknown, it most often occurs to people who use tobacco products.

Treatments for oral cancer are based on the stage (extent of spread) of the disease and may involve radiation therapy, chemotherapy and surgery. If the

[1] Source: http://www.wrongdiagnosis.com/o/oral_cancer/stats.htm

cancers are still small, they can quickly and successfully be treated by surgical removal, leaving hardly no cosmetic or functional changes behind. However, patients with a large tumor may suffer function losses after surgical removal, resulting in serious difficulties with speech and swallowing. The anatomical complexity of the tongue and the great variability of individual tumor extensions, which significantly differ among patients, makes it very difficult to predict the exact consequences of surgical interventions on the post-operative functioning of the tongue. The decision, concerning an individual patient, to whether or not remove such a tumor can therefore be very difficult.

1.2 Ultimate goal

Objective determination whether surgical treatment of oral cancer is a suitable choice for an individual patient, requires pre-surgical assessment of expected post-operative functioning. The ultimate goal of the project is therefore to develop a virtual environment in which a functional three-dimensional model of the patient's mouth and tongue can be used to predict the post-operative functioning which remains after resection of a part of the oral organs. Such a model should be based on patient specific parameters (e.g. geometric tongue and lip parameters), obtained by some kind of scan (e.g. MRI or ultrasone). The model is then formed by using these parameters as input for mathematical algorithms that describe the model. These algorithms will be the basis for an interactive visualization tool that enables virtual surgery.

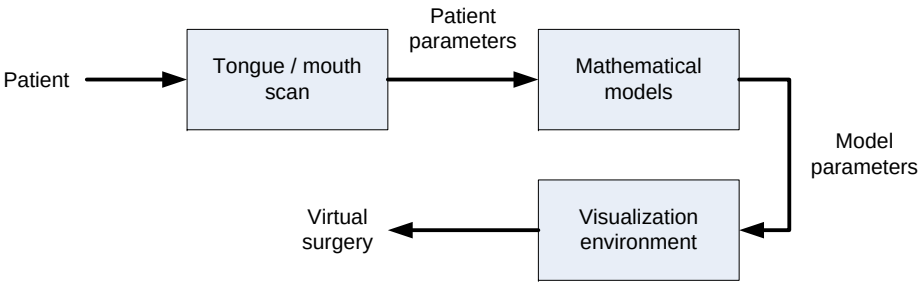


Figure 1.1: *Envisaged procedure for the creation of a patient specific tongue or lip model which can be used for virtual surgery.*

Because of the complicated anatomical and muscular structure of organs like the tongue and lips, the project team aims to develop a so-called *phenomenological black box-model*, rather than a complicated, detailed mechanical/biological model of the underlying structures. The principle working of a phenomenological model relies on the hypothesis that an explicit causal relation can be established between groups of muscular activation signals and dynamic model variables describing the shape and motion of the tongue and the lips. The availability of a model that describes this relation enables to predict which modes of motion and which shape deformations are still possible after resection of a part of the tongue or lips (see figure 1.2). This opens the door to the development of methods for the prediction of function loss. Initially the model will be confined

to the tongue and lips, ultimately it will be extended to the total oral cavity and the pharynx.

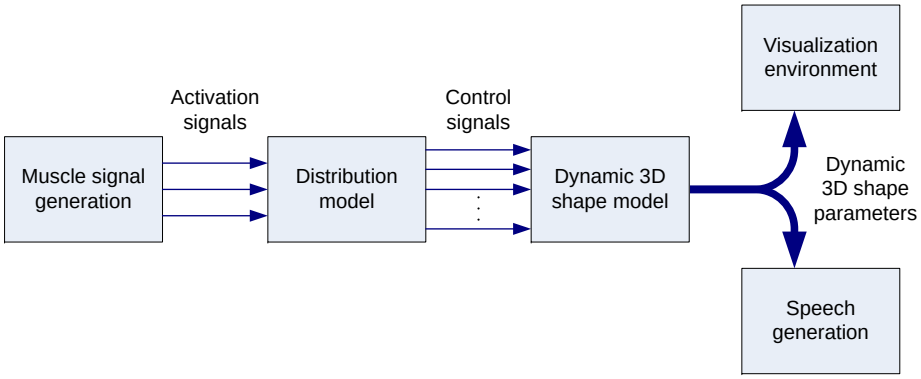


Figure 1.2: *Virtual surgery based on a phenomenological black box-model: signals can be generated, analog to muscle activation signals, and will be coupled to dynamic model variables according to a distribution model (not necessary one-to-one mapping).*

1.3 Scope of this thesis

The ultimate project goal is ambitious and it will take a lot of time and research in several fields before this goal is reached. The main research issues include the investigation of possible methods for obtaining patient specific tongue and lip parameters, investigation of techniques for measuring muscular activation signals (both for the lips and the tongue) and investigation and development of mathematical algorithms for modeling dynamical shapes. A big challenge will be the establishment of the distribution model (see figure 1.2), which should describe the causal relation between the actual muscular activation signals (e.g. measured with EMG) and the dynamic model variables.

This thesis can be seen as an initial exploration study regarding the development of a dynamic tongue and lip model and the involved aspects. The thesis includes the following topics:

- A literature survey to existing modeling techniques focused on the human tongue and lips.
- The development of an algorithm for automatic detection of the tongue contour in (sequences of) noisy magnetic resonance images.
- Derivation, implementation and evaluation of phenomenological dynamic modeling algorithms. This also includes the estimation of system parameters and input signals, given measured output variables (for example extracted tongue contours).
- The development of a simple graphical user interface to visualize and simulate static and dynamic tongue and lip movements (see appendix B).

A tongue contour detection algorithm is developed, since the initial idea was to use magnetic resonance imaging (MRI) for the acquisition of tongue data. However, the algorithm can, with some small adjustments, also be used for shape (e.g. lip) detection in normal images. The extracted data, either in MR or in optical images, is used for investigation and development of the modeling algorithms. These algorithms will be a starting point for building more extensive models that become feasible when the EMG data becomes available.

1.4 Report outline

In chapter 2 the different aspects that are part of the development of a tongue or lip model are discussed and explained. A general model structure for a system with input and output signals is presented and the involved variables and parameters are defined. The chapter also includes a literature survey to existing modeling techniques focused on tongue and lip modeling. Chapter 3 is about tongue contour detection in magnetic resonance images. This chapter first discusses possible methods for contour detection and clarifies the choice for the Active Shape Model. The rest of the chapter is mainly concerned with details and implementation issues of the ASM algorithm and concludes with a performance evaluation. In chapter 4 a discrete-time linear state space model as a possible framework for tongue and lip modeling is described. The chapter considers possible state vectors, discusses the involved matrices and system parameters, and motivates assumptions that have to be made in absence of actual input signals. Chapter 5 is focused on the actual identification of the linear state space model and is therefore mainly concerned with the derivation of algorithms for the estimation of states, input and system parameters from measured output variables. Finally, in chapter 6 conclusions are drawn about the executed research and recommendations are given for future work.

2

Tongue and lip modeling

2.1 Introduction

Human organs that are part of the oral cavity and the pharynx, are complicated biomechanical systems. This is especially true for the tongue. The development of a mathematical descriptive model for such a complicated physical system is an extremely complex and challenging task, without a straightforward approach. Over the years researchers have made several efforts to build a model that is empirically adequate. Such a model shows the same (outward) behavior as the system, regardless of whether the mathematical structure of the model corresponds to the internal structure of the actual system or not. However, a full three-dimensional model, that is able to simulate and predict realistic tongue movements, is not yet developed. Reasons for this are the complex muscular and neural structure of the tongue, the complicated shape, the interaction of different muscles, the limited visibility (inside the mouth) and the lack of sufficient anatomical data.

This chapter focuses on the aspects that are part of the development of a model for the tongue or lips. The chapter starts in section 2.2 with presenting a general structure for a model with input and output signals. In this section also the involved variables and parameters will be defined and some general characteristics to classify a model will be explained. For the development and testing of a model, measurements on the actual system are required. In section 2.3 several techniques to acquire data of real tongue and lip properties and movements will be discussed. Next, the commonly applied approach for modeling physical systems will be discussed in section 2.4. This is the so-called finite element approach. The resulting models are called physiological models. However, the finite element approach requires a lot of physiological information about the actual system. Therefore an introduction to phenomenological blackbox modeling, which requires less physiological information, will be given in section 2.5. In the last section (section 2.6) the different aspects and approaches for tongue and lip modeling will be summarized and their advantages and disadvantages will be discussed.

2.2 Model structuring

A mathematical model usually describes a system by a set of variables and parameters and a set of equations that establish relationships between these variables and parameters. The variables represent properties of the system. They are physical quantities that often change in time. Examples of variables are input signals, output signals and system state variables. Parameters (approximately) don't change in time. Examples of parameters are the mass and elasticity of a material. Furthermore, there are the running variables. These are time and position variables. The actual model is the set of functions that describes the relations between the different variables and parameters. In this section it will be discussed what these different variables, parameters and functions can be in case of a model for the tongue or lips.

2.2.1 Model overview

On the highest level of consideration, a human organ like the tongue or lips can be considered as a system with input and output variables (see figure 2.1). Input variables, indicated by $\mathbf{u}(t)$, are in this case muscular activation signals. Output variables, indicated by $\mathbf{z}(t)$, are for examples parameters that describe (dynamical) shapes of the tongue or lips. Between input and output, mathematical operations take place. The functions inside the model describe how a certain set of input signals at time t leads to an output at time $t + \Delta t$. (In case of a causal system Δt is equal to or greater than zero.)

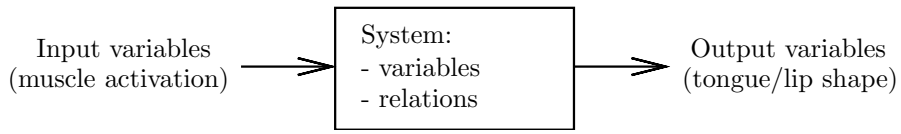


Figure 2.1: *General model structure.*

2.2.2 Feature vector

Features are variables of the system that represent specific properties of the system. These features together form the feature vector, indicated by $\mathbf{x}(t)$. The feature vector is based on the state vector, which is the minimum set of variables to describe the dynamics of the system, and that summarizes the system's past. The features depend on the state variables. Examples of basic features in case of a model for the tongue or lips are the positions of landmarks on the tongue or lip contour and the velocity and acceleration vectors of these landmarks. Other examples of features that could be included in the feature vector are the vertical distances between the upper and lower boundaries and the two angles of the mouth corners. Figure 2.2 shows in an image of the mouth with some possible lip features. But all in all, such a feature vector can become quite large, which can be a disadvantage for the computational performance of the model. However, there might be a lot of correlation between the different features. Therefore

a mathematical technique, called principal component analysis, can be applied to transform the original feature vectors to new features vector with a smaller number of uncorrelated variables. This technique will be further discussed in chapter 4.

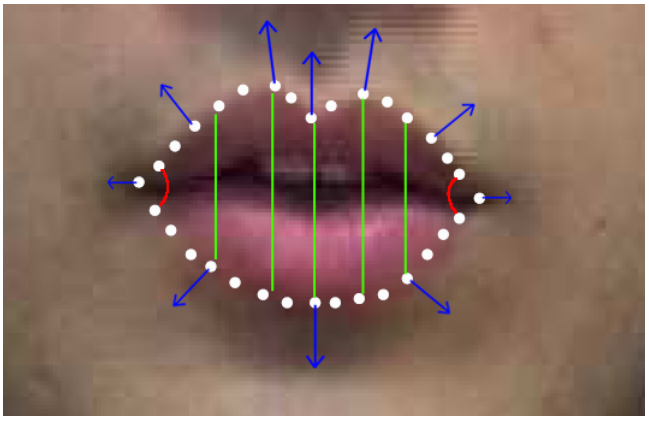


Figure 2.2: *Examples of lip features (white dots: landmarks, blue arrows: velocity vectors, green lines: lip distances, red archs: lip angles).*

2.2.3 System parameters

Examples of system parameters are the volume of the tongue or lips, the mass density and viscosity of the soft tissue and the damping and elasticity of muscles. These parameters are patient specific. However, it is not unrealistic to assume that most of the system parameters are time-invariant, i.e. the system characteristics do not change over time. Otherwise it would also make prediction more difficult. When all the physical variables and parameters are punctually identified and the relations are implemented according to the correct physical laws, the model is called a *white-box* model. On the other hand, when the model is only based on a description of the behavior between input and output variables, the model is called a *black-box* model. This type of modeling can be used when there is no a priori information about the system available or when it is difficult to identify the physical structure and parameters of the system. Usually it is preferable to use as much a priori information as possible to make the model more accurate. If there is not enough a priori information available, the system parameters have to be estimated from measured input and/or output data. When only a part of the model is constructed according to physical laws, the model is called a *gray-box* model.

2.2.4 Model characteristics

A model can be classified based on some general characteristics. The most important characteristics will be mentioned here.

Continuous vs. discrete time

The behavior of a system can be described with a model in the continuous-time domain or in the discrete-time domain. In case of a continuous-time model, state and output variables can be calculated at every time moment t. In case of a discrete-time model, this can only be done at discrete-time moments i, where i is an integer time index. Usually a model is time-discrete, since input or output variables are sampled signals from the actual system and thus time-discrete.

Static vs. dynamic

A model can be static or dynamic. In case of a static model, the variables are only a function of the current input signals. So, actually a static model does not account for the element of time. In case of a dynamic model, some variables depend on their past, i.e. on previous values. These are the state variables. Dynamic models typically are represented by differential equations when the model is time-continue and by difference equations when the model is time-discrete. Table 2.1 shows the form of the state vector function for the different type of models. The vector $\dot{\mathbf{x}}$ is the time-derivative of the state vector. In the table it is assumed that the system parameters are constant in time. In that case, the system is time-invariant. If the system parameters are time-dependent, the system is time-variant and the system function depends explicitly on time.

	Continuous-time	Discrete-time
Static	$\mathbf{x}(t) = f(\mathbf{u}(t))$	$\mathbf{x}(i) = f(\mathbf{u}(i))$
Dynamic	$\dot{\mathbf{x}}(t) = f\left(\mathbf{x}(t), \mathbf{u}(t)\right)$	$\mathbf{x}(i+1) = f\left(\mathbf{x}(i), \mathbf{u}(i)\right)$

Table 2.1: *Function form of state vector for different type of models.*

Linear vs. nonlinear

The state vector is a function of the input, the system parameters and of previous state variables. The output is a function of the state vector. When these functions are linear (i.e. there are no second or higher order terms involved), the model is defined as linear. Otherwise, the model is considered to be nonlinear.

Deterministic vs. probabilistic

A model can also be deterministic or probabilistic. A deterministic model is one in which every set of state and output variables is uniquely determined by the system parameters, input signals and previous states. A deterministic model always performs the same way for a given set of initial conditions. However, when there is randomness present, caused by process and/or measurement noise, the variables are not described by unique values, but rather by probability distributions. In that case the model is called probabilistic or stochastic.

Distributed vs. lumped

Furthermore, a difference can be made between distributed and lumped models. A distributed model is one in which all state variables are functions of time and one or more spatial variables. A lumped model is one in which the variables of interest are a function of time alone. A distributed model is usually described with a partial differential equation and a lumped model with an ordinary differential equation. A distributed model is more accurate and more complex than a lumped model. A lumped model can be seen as a simplification of its distributed version. (More details will follow in section 2.4.)

2.3 Data acquisition techniques

For the development of a dynamic model of the oral cavity and the pharynx, data about real tongue and lip movements is required. In case of a black-box model, this data consists of sequences of measured features that describe the evolution of shapes belonging to realistic movements. Realistic movements are assumed to be movements belonging to, for example, swallowing and the pronouncement of phonemes. Ideally, the measurements of these features are linked to measured muscle activation signals, such that also the corresponding input variables are available. Data of lip movements can relatively simple be obtained with (a high speed) video camera. However, tracking tongue movements is more difficult, especially in three dimensions. This section shortly reviews a few possible techniques for the acquisition of especially tongue data. (A detailed report about acquisition techniques for tongue data is recently presented by another student, see [2].)

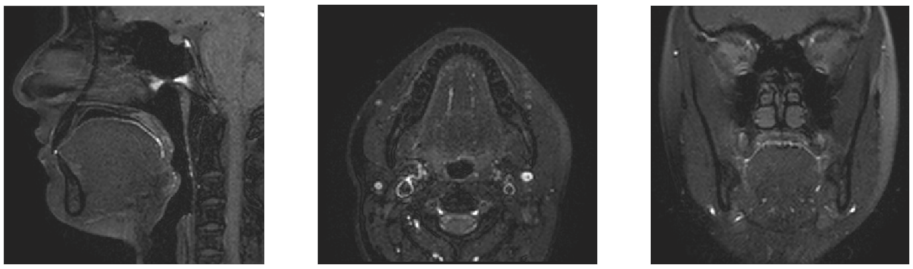
2.3.1 Magnetic resonance imaging

Magnetic resonance imaging (MRI) [3] is a medical imaging technique to visualize the internal structure of a body. It uses a powerful magnetic field (typically 2 to 3 tesla) to align the nuclear magnetization of hydrogen atoms in the body. Radio frequency fields are applied to systematically alter the alignment of this magnetization. When the fields are turned off, protons return to their original magnetization alignment. Thereby they create a signal which can be detected by the scanner (receiver coils). Additional magnetic fields are used to manipulate the signal, such that information can be obtained to construct an image of the body.

MRI has been used in many researches to extract information about (dynamic) tongue shapes. In [4] a three-dimensional static tongue model is developed by manually extracting tongue contours from MR images in several planes. However, most of the research is focused on (automatic) tracking of tongue motion. In [5] the motion of the internal tongue is modeled from tagged MR images. In tagged-MRI a grid is created on a cross-section of the tongue by temporarily terminating certain magnetic spins. In the meantime a short sequence of MR images can be created during a simple tongue movement. Afterward, positions in the different images can easily be linked thanks to the

grid-tags. Unfortunately, the termination of the magnetic spins on the grid is only very temporarily, such that only a few low-resolution images with tags on the tongue can be recorded. However, a lot of research is still going on to extent and improve the principle working of tagged-MRI. For example, in a quite recently publication [6] a certain sequence - called zHARP - of RF-pulses and magnetic field gradients is described to record a simple three-dimensional tongue motion from three orthogonal tag orientations (sagittal, coronal and transversal).

Summarized, MRI is a safe technique to create images of a cross-section inside the mouth. In these images, the tongue contour can be detected manually or automatically. From images in several planes it is possible to construct the three-dimensional shape. However, the quality of the MR images depends on the acquisition speed, i.e. the resolution is inversely proportional to the speed. For now, the acquisition speed is too low for tracking the tongue during realistic movements, especially in three dimensions. In the future MRI might be an option for the acquisition of proper tongue data.



(a) *Sagittal* **(b)** *Transverse* **(c)** *Coronal*

Figure 2.3: *Examples of MR images of the tongue in the different planes.*

2.3.2 Ultrasonic imaging

Also Ultrasonic imaging [3] is a safe and non-invasive medical imaging technique that enables visualization of the tongue inside the mouth without placing any obstructions on the tongue. The basic principle of ultrasonic imaging is simple. A propagating wave partially reflects at the interface between tissues with different densities. If these reflections are measured as a function of time, information is obtained on the position of the tissue. This way the tongue tissue can be distinguished from other tissue and air in the mouth.

In [7] ultrasonic images are recorded by placing a probe, mounted on a special helmet, under the test person's chin. The probe emits ultrasonic waves which are reflected at a boundary between different types of tissue. It appeared to be possible to record tongue images with a frame rate of 30 fps. A disadvantage is that a raised tongue tip with an air pocket below it cannot be imaged, since the reflection at the air boundary is almost 100%.

2.3.3 Radiography

Radiography is an imaging technique that uses electromagnetic radiation. The most useful type of radiation for imaging purposes is X-rays [3], because of the relative high energy of the electromagnetic waves. X-rays consist of photons that can interact with matter and tissue in three different ways. When a photon hits an atom it can lead to *photoelectric absorption*, electron scattering or electron-positron pair production. The way of interaction depends on the density and composition of the material. An image is formed by a detector, behind the object, that projects the not-absorbed X-rays on a radiation-sensitive film.

Although X-rays can be used to create very clear images of organs, there is always a small risk on radiation damage. Another disadvantage is that teeth in the mouth make the detection of the tongue more difficult, because the difference between different types of tissue is very small compared to the difference between tissue and teeth.

2.3.4 Accelerometers

A different way to track tongue motion might be accomplished with small acceleration sensors on the tongue. The main advantages are the high sample rate and the accuracy. Main disadvantages include the weight of the sensors, the required electric cords that have to go into the mouth and the low resolution (probably just a few sensors can be 'mounted' on the tongue). The sensor weights and the cords will probably influence and limit the tongue movements.

2.4 Physiological modeling

Most of the developed models of the tongue or lips so far, are so-called physiological models. A physiological model describes the (dynamical) behavior of a physical system by analyzing and modeling the content of the physical system. In case of developing a physiological model of the tongue or lips, information about the internal and external structure of these organs is required, like the extrinsic and intrinsic musculature, the shape and tissue properties (e.g. mass and stiffness). The required physiological information is generally obtained from anatomical and physiological studies, X-ray images and MRI scans.

2.4.1 Continuous description

Initially, physical systems like the tongue and lips are considered as distributed systems. This means that different physical quantities interact (e.g. force and velocity) and that different dynamic laws are needed to describe the dynamical behavior. The most relevant laws, in case of a dynamical system in the (bio)mechanical domain, are Newton's second law, Hooke's law and the friction law. Newton's second law, $F = ma$, describes how an applied force F on a mass m results in an acceleration a of that mass. Hooke's law, $F = kx$, is an elasticity law and describes the relation between an applied force on a

spring and its stretching. The constant k represents the stiffness of the spring. Furthermore, in most mechanical systems, friction is involved. The friction or damping law, $F = dv$, describes how a friction force F influences the velocity v of a moving object. The constant d represents the damping of the material. Here, the friction is assumed to be viscous, i.e. linear. But often friction forces are nonlinear.

A simple physiological model of for example the tongue, consists of mass points connected to each other by springs and dashpots (dampers) in three dimensions. When, in case of a one-dimensional system, the number of points (or elements) is approximately infinite and the distance between two points approaches zero, it can be derived that the continuous dynamical behavior, in terms of force and velocity, can be described with the following two differential equations:

$$\frac{\partial v(x,t)}{\partial x} = -\frac{1}{k}\frac{\partial F(x,t)}{\partial t} - \frac{1}{d}F(x,t) \tag{2.1a}$$

$$\frac{\partial F(x,t)}{\partial x} = -m\frac{\partial v(x,t)}{\partial t} \tag{2.1b}$$

In these equations, $\partial v(x,t)$ and $\partial F(x,t)$ are respectively the differential velocity and the differential force of a particle at position x and time t. The constants m, k and d represent respectively the mass density, the stiffness and damping of the material. By differentiating equation (2.1a) with respect to x and equation (2.1b) with respect to t, the resulting equations can be combined to the following partial differential equation:

$$\frac{\partial^2}{\partial x^2}F(x,t) = \frac{m}{k}\frac{\partial^2}{\partial t^2}F(x,t) + \frac{m}{d}\frac{\partial}{\partial t}F(x,t) \tag{2.2}$$

In case of a three-dimensional dynamical system, the partial differential equation also contains the second derivatives of the force with respect to y and z. This is the divergence of the gradient of F, also called the Laplacian (∇^2) of F:

$$\nabla^2 F(x,y,z,t) = \frac{m}{k}\frac{\partial^2}{\partial t^2}F(x,y,z,t) + \frac{m}{d}\frac{\partial}{\partial t}F(x,y,z,t) \tag{2.3}$$

Together with some boundary conditions (e.g. $v(x,y,0,t) = 0$ and $F(x,y,z,0) = 0$), equation (2.3) can be used to derive the velocity and force of certain point on the material at a certain time moment.

2.4.2 Discretization

Solving partial differential equations (PDE) like the one of equation (2.2) is a complex task; especially in case of PDE's that describe constructions or systems in three dimensions this is practically impossible. A commonly used approach for finding approximate solutions of PDE's is the finite element method (FEM). The basic idea of the FEM is to completely eliminate the PDE's and to render them into an approximating system of ordinary differential equations. This is done by dividing the construction into a finite number of elements, which are connected to each other by nodes. The configuration of these nodes defines the finite element mesh. A finite element model is also called a lumped model.

Figure 2.4 shows a lumped model of a one-dimensional dynamical system (e.g. an elastic cord). It consists of a finite number of mass points connected to each other by springs and dashpots.

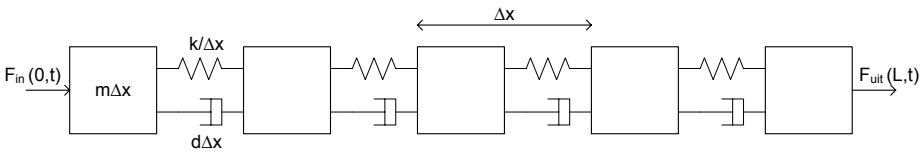
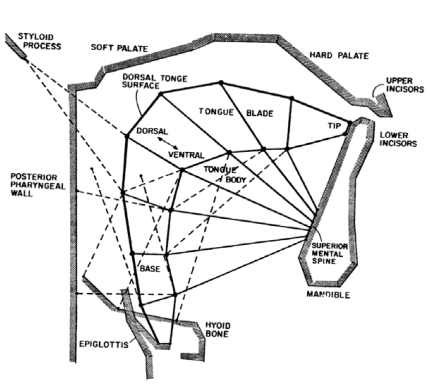


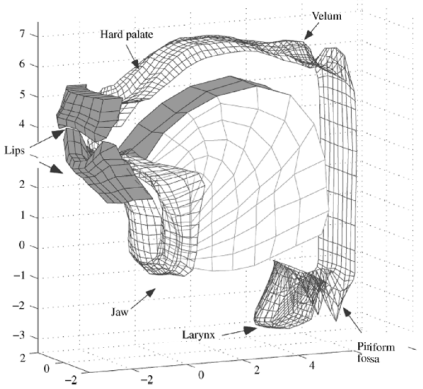
Figure 2.4: *Example of a lumped model of a one-dimensional dynamical system.*

Tongue and lip modeling using FEM

Over the years, there have been several efforts to model the tongue and lips by using the FEM approach. The developed models can be divided into two-dimensional, 'two-and-a-half' dimensional and three-dimensional models. One of the first physiological model of the tongue is presented by Perkell [8] in 1974. This is a two-dimensional model in the mid-sagittal plane, consisting of sixteen elements (see figure 2.5a). Muscles are modeled as linear elastic material with springs and-dashpots. Perkell based his model on information from anatomical studies. A more advanced two-dimensional finite-element model of the tongue is developed by Payan and Perrier [9] in 1997 and consists of 48 elements. The model geometry is based on X-ray images.



(a) *2D-model Perkell [8].*



(b) *2.5D-model Dang and Honda [10].*

Figure 2.5: *Two (and-a-half) dimensional finite element models of the tongue.*

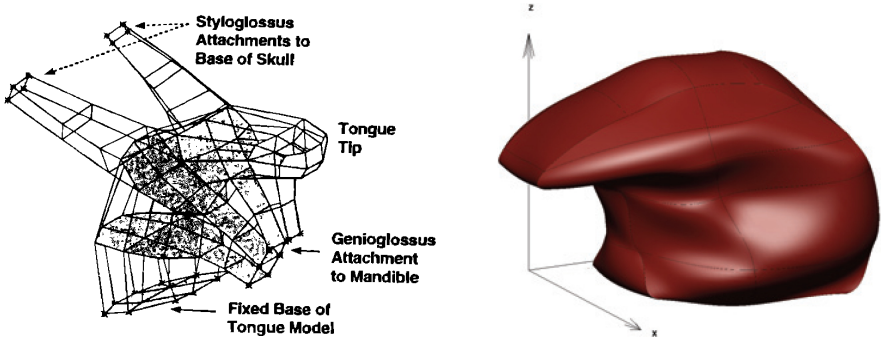
Dang and Honda presented several versions of a 'two-and-a-half' dimensional tongue model [10, 11]. Such a model does not cover the whole tongue, but has a thickness (2 cm) in the sagittal plane (see figue 2.5b). The 'two-and-a-half' dimensional model of Dang and Honda consists of 120 elements. The geometry of the model is based on three-dimensional anatomical data, consisting of 15 sagittal slices of MR images. The developed lumped model can be considered as

a network of mass points connected by spring-and-dashpot elements. The corresponding motion equation is described as a second order differential equation:

$$\mathbf{M}\ddot{\mathbf{x}}(t) + \mathbf{D}\dot{\mathbf{x}}(t) + \mathbf{K}\mathbf{x}(t) = \mathbf{F}(t) \tag{2.4}$$

In this equation $\mathbf{M}$ is a diagonal matrix consisting of the masses of all the mass points within the model. $\mathbf{D}$ and $\mathbf{K}$ are the damping and stiffness matrices and $\mathbf{x}$, $\dot{\mathbf{x}}(i)$ and $\ddot{\mathbf{x}}(i)$ are respectively the displacement, velocity and acceleration state vectors of the finite element assemblage at time t. $\mathbf{F}(t)$ denotes the external forces applied on the nodal points. Using a backward-difference method, it is relative simple to obtain the solution of $\mathbf{x}(t)$. However, the small number of elements constrains the number of possible shapes and movements in the sagittal plane.

Full three-dimensional tongue models that incorporate the complex muscle structure and biomechanical properties are rare. One of the most advanced and sophisticated model was introduced by Wilhelms-Tricarico [12] in 1995. He was the first to model passive stress using hyperelastic material. In previous models, material was assumed to be linear elastic. The finite element mesh also shows an increase of precision, compared to previous models. It consists of 740 elements and the node locations are based on data from the Visible Human Project[1]. The mesh proposed by Wilhelms-Tricarico was the basis for further FEM tongue models. However, most of the presented FEM tongue models are focused on the investigation of speech production and are therefore symmetric in the sagittal plane. Fujita [13] constructed a three-dimensional physiological tongue model focused on clinical applications and also included asymmetric postures. Estimated muscle activation patterns belonging to basic movements are incorporated in this model. Simulations were compared to actual tongue movements and demonstrated that the model is able to reproduce these basic movements. A quite recent (2006) three-dimensional finite element model of the tongue is presented by Wu and Han [14]. The volume mesh and fiber directions are derived by an iterative optimization procedure that fits mesh to data set obtained from the female Visible Human.



(a) *Model Wilhelms-Tricarico [12].* **(b)** *Model Wu and Han [14].*

Figure 2.6: *Three-dimensional finite element models of the tongue.*

[1] Website VHP: http://www.nlm.nih.gov/research/visible/visible_human.html

From the considered literature the general procedure for the creation of a three-dimensional finite element model of the tongue can be derived. This procedure consists of the following basic steps:

1. Based on geometric descriptions from anatomical studies and MR or X-ray images, a volumetric representation is produced. This representation is discrete: the volume of the tongue is represented by a collection of voxels. The geometric representation is smoothed by lofting between calculated splines.

2. The geometric representation is divided into simple volumetric elements (e.g. tetrahedrals), forming the finite element mesh. This division is based on muscle and fiber orientations.

3. A mathematical description of the behavior of the involved materials (e.g. soft tissue and muscles) is formulated. This description contains information about the deformation of the materials in response to applied external loads and the stresses generated by the material itself. It also involves a kinematic model of muscles.

4. Boundary conditions are assigned. This means that nodes in positions belonging to external attachment sites of the tongue are determined to be fixed. These nodes are based on anatomical criteria.

5. In the last step the applied loads (input signals) are described. In case of the tongue, the loads are provided mainly by muscle contraction. The muscle activation scheme can be specified by the user, generally in pressure units.

After these parts, the FEM model is defined and it is possible to calculate how the model will deform, given a certain set of inputs signals (innovated muscles). This deformation is generally calculated with an ordinary differential equation, similar to (2.4). Such a differential equation can relatively easy be solved, in contrary to the partial differential equation of (2.3).

2.5 Phenomenological modeling

In case of phenomenological or *black-box* modeling one tries to build a model of a system without looking at its internal structure, but only by considering the observable behavior of the system. Knowledge about the exact system parameters and state variables is not required. System identification by means of phenomenological modeling is especially useful for modeling systems that cannot easily be represented in terms of first principles or known physical laws. The challenge in phenomenological modeling is to estimate the system parameters, the state variables and possibly even the input signals from measured data. The system parameters of a phenomenological model do not need to have a physical interpretation.

The observable phenomena of a biomechanical system like the tongue and lips are the dynamical shapes of for example movements belonging to swallowing

and the pronouncement of phonemes. These dynamical shapes can be considered as the output variables of the general system model in figure 2.1. A few possible techniques for measuring data that describe dynamical tongue shapes are already reviewed in section 2.3. Once there is proper measurement data available, the model can be identified from these measurements by using statistical techniques. The main idea is to find a relative small number of control variables that can explain the most important (dynamical) shapes. The ultimate goal is to relate these control signals to measured activating EMG signals. Once this connection is established, it will be possible to calculate to dynamical deformation as a function of muscle activation signals.

So far, not many tongue or lip models have been developed by means of phenomenological modeling. In [15] the temporal evolution of lip features (landmarks on the lip contour) during the pronouncement of simple visemes is modeled as a linear dynamical system. The system parameters are estimated from the measurements by using system identification techniques. However, since this research was focused on lip articulation classification, input signals are not estimated. In [16] a phenomenological three-dimensional static model of the tongue is presented, based on (manually) extracted tongue contours from MR images. The used data contained 44 sets of MR images for different tongue shapes. Each set consisted of 54 MRI slices in different planes. The total acquisition time per set was 43 seconds and during this time the tongue had to be sustained at the same position. The slices were placed on an in advance determined grid for the three-dimensional construction, see figure 2.7. A statistical technique, called linear component analysis, was used to derive six static control parameters, representing tongue parameters like the jaw height, the tongue width and the tongue tip. Another measurement technique, called Electromagnetic articulography (EMA), was applied to measure the actual values of these parameters in time. The combination of the MRI and EMA was used to make animated sequences of tongue shapes as a function of these parameters.

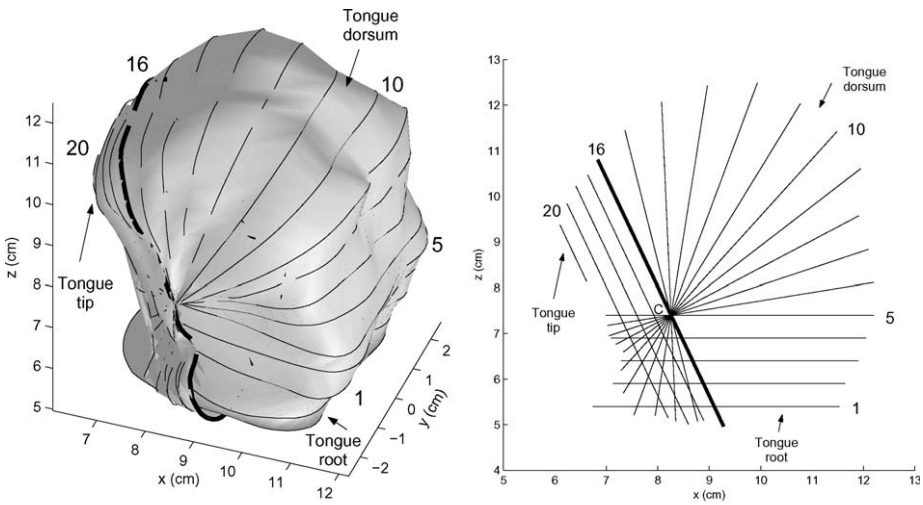


Figure 2.7: *Three-dimensional phenomenological tongue model from Engwall [16]. The right image shows the grid for the 3D construction from MRI slices.*

2.6 Summary and discussion

Modeling the human lips and especially the tongue is a difficult task, due to the complex muscular and neural structure, the complicated shape, the interaction of different muscles, the limited visibility (inside the mouth) and the lack of enough anatomical data. Over the years researchers have already made several efforts to arrive at a working model. The main distinctions concerning the different type of models are between physiological and phenomenological (or statistical) models, between two- and three-dimensional models and between static and dynamic models.

The physiological modeling approach aims at the understanding and modeling of the muscular structure and functions of the system and the biomechanical constraints involved, such as volume conservation and tissue deformation. However, physiological modeling has some big disadvantages and difficulties. The method requires detailed information and understanding of the actual system, like the direction and location of different muscles and neurons and values of physiological and mechanical parameters. Dynamical physiological models are generally constructed by using the finite element method. Although FEM is a relative simple method for solving complex differential equations, it is very computationally intensive and requires advanced software tools.

A different approach for the development of a tongue or lip model is phenomenological modeling. A phenomenological model is constructed based on observed or measured phenomena, i.e. the outside behavior of the system. So, the main advantage of phenomenological modeling is that it does not require knowledge about the exact anatomical structure of the system. Another advantage of dynamic phenomenological models is that they are less computational intensive and simple enough to be incorporated in a real-time system. However, this approach has also some disadvantages and difficulties. Because of the limited visibility of the tongue, it is difficult to obtain proper measurement data. Tracking (three-dimensional) tongue movements inside the mouth requires advanced measurement techniques. A few of those techniques have been discussed in section 2.3. Furthermore, other challenges in case of phenomenological modeling involve the estimation of system parameters and setting up the relation between derived control parameters and actual muscle activation signals.

3

Tongue contour detection

3.1 Introduction

In case MRI is used as the technique for the acquisition of tongue data or patient-specific parameters, the first step is the detection of the tongue contour in the MR images. The objective of the project part, described in this chapter, was therefore the development of an algorithm for automatic tongue contour detection in (sequences of) MR images. In such a MR image, the tongue cross-section (e.g. in a sagittal, coronal or transversal plane, see figure 2.3) usually covers only a small part of the image. Because of MRI technical reasons, it is more efficient and faster to make images of the whole head. Taking MR images involves making a trade-off between image quality (in terms of resolution and noise) and acquisition speed. Especially in capturing a sequence of MR images during a tongue movement, the quality suffers. The detection method should therefore be robust against a significant amount of noise in the image.

For the detection it was decided to implement an Active Shape Model (ASM) algorithm. The main reasons for choosing this algorithm include its performance in noisy images, its relative large feature detection range and its matching speed. The choice will be further motivated in section 3.2, where a comparison will be made with other methods for contour detection. In section 3.3 details and implementation issues of the ASM algorithm, focused on tongue contour detection in MR images, will be described. In section 3.4 the performance of the implemented algorithm, in terms of detection results, will be discussed. For this performance evaluation, sequences of captured MR images during simple tongue movements in the sagittal and transverse plane are used. This chapter concludes in section 3.5 with some critical remarks concerning the tongue contour detection algorithm.

3.2 Methods for contour detection

Most of the existing methods for finding a shape or contour in an image use flexible models or deformable templates that are build based on training images containing an example of the concerning object. Such models usually have a number of parameters to control the shape and pose of all parts of the model. During shape search in a new image, these parameters are adjusted in an iterative process based on object features - such as edges - in the image. Three of the most significant methods for shape or contour detection are Active Contours, Active Shape Models and Active Appearance Models. In this section a short review of these methods will be given.

3.2.1 Active Contours

The basic concept of contour detection algorithms was introduced in 1988 and is called Active Contours [17] or *snakes*. A *snake* is placed on an image and moves toward an optimal position and shape. Fitting active contours to shapes in images is an iterative process. The operator must suggest an initial contour, which is quite close to the intended shape. The contour will then be attracted to features in the image. This happens by minimizing an energy function, which consists of a sum of external and internal energy. The external energy is supposed to be minimal when the snake is at the boundary of an object. The internal energy is related to applied constraints. These constraints ensure that the contour remains smooth and limit the freedom of bending and deformation.

3.2.2 Active Shape Models

Although the deformation of active contours can be limited by applying some constraints, active contours are usually free to take almost any smooth shape and easily snap to wrong boundaries. Cootes introduced in 1995 [18] a method to effectively limit the deformation of contours. From a training set of shapes, a *point distribution model* is inferred that represents the mean geometry of the shapes and statistical modes of geometric variation. The point distribution model leads to an Active Shape Model (ASM), which can only deform to fit objects in ways consistent with the training set.

The ASM describes a shape with a set of points. The contour is created by interpolation between the points. During each iteration, a search is made around the current position of each point, along a profile normal to the contour, to find a point nearby which best matches the model of the texture expected at the landmark. The parameters of the shape model controlling the point positions are then updated to move the model points closer to the points found in the image. Because the shapes are constrained to be similar to those in the training set, the method is able to automatically locate structures in complex, noisy, and cluttered images. The ASM algorithm can easily be extended to the three-dimensional case [19]. An object in a three-dimensional space is than searched by taking samples along profiles normal to the object surface.

3.2.3 Active Appearance Models

The Active Appearance Model (AAM) [20] is closely related to the Active Shape Model. The AAM is generated by combining a model of shape variation with a model of texture variation. From the training set, a mean shape and modes of variation are inferred that represent both shape and texture. Given a new image, labeled with a set of landmarks, an approximation with the model can be generated in iterative process. In each iteration, the AAM only samples the image under the current position of the model. The model parameters are then updated based on these sample results. Figure 3.1 shows an example of applying this method to face images.

Figure 3.1: *Example of applying the AAM on face images.*

The AAM is able to give a better match with the image texture than the ASM. But since the AAM only examines the image directly under its current area, this method has a smaller capture range (feature detection range) than the ASM, which searches around the current location, along profiles. The smaller the capture range, the higher the demands on the initial position of the model on the new image and the slower the convergence speed. Also according to experimental results, described in [20], the ASM is faster and has a larger feature detection range than the AAM, especially in medical MR images.

3.2.4 Conclusion

Based on the reviews in the above subsections, it can be concluded that the Active Shape Model would be the most appropriate method for the detection of tongue contours in MR images. Simple Active Contours are not based on a trained model and can therefore deform into invalid shapes during search. The Active Appearance Model is focused on synthesizing a complete image of an object and might therefore be a bit overkill for this application. The Active Shape Model is fast, accurate, appropriate for noisy images and able to search for shape features in a wide range. The latter property is desirable since tongue shapes can have a large deviation from the mean shape (e.g. in case of an image with a protruded tongue). Furthermore, the ASM algorithm can easily be extended to the three-dimensional case. This is also desirable, since ultimately the envisaged system should enable virtual surgery in the three spatial dimensions.

3.3 ASM for tongue contour detection

Based on the papers [18, 20, 21], an ASM algorithm for the detection of tongue contours in MR images is implemented in MATLAB. Some small modifications and adjustments, compared to the basic version of the ASM, have been made to make the algorithm especially suitable for this application. In this section, implementation issues will be described and design issues will be motivated. In the upcoming subsection it will first be explained how tongue contours can actually be represented. The next two subsection describe the steps to be executed in the training and application stage.

3.3.1 Representation of tongue contours

The model of an object shape can be represented by a set of points (landmarks). In case of representing the contour of an object, the landmarks have to be placed at the object's boundary. For good performance, the locations of these landmarks should be places of interest where there is the most information. Excellent locations are for example corners and 'T'-junctions. Intermediate points can be used to define the boundary more precisely.

If a shape is described by l points in d dimensions, the shape can be represented by an element vector $\mathbf{x}$ of length $p = ld$, formed by concatenating the elements of the individual point position vectors. In case of representing the l landmark points, (x_i, y_i), of a shape in a 2D image, the shape vector becomes a $2l$ element vector:

$$\mathbf{x} = [x_1, x_2, \ldots, x_l, y_1, y_2, \ldots, y_l]^T \tag{3.1}$$

Next, a curve through the landmarks can be drawn by using a spline interpolation method. Beside doing this for visualization purposes, samples in the image will be taken at landmarks along a profile perpendicular to the contour. Several algorithms for calculating splines exist. One of the commonly used algorithms is cubic spline interpolation. Since MATLAB is provided with a ready-made function for calculating cubic spline curves, it was decided to use this one. The cubic spline between two points is of the form:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \tag{3.2}$$

The algorithm calculates the coefficients a_i, b_i, c_i, d_i such that the values of two spline functions are equal at landmark positions, as well as the derivatives and second derivatives of the functions at that position:

$$\begin{aligned} S_i(x_i) &= S_{i-1}(x_i) \\ S_i^{'}(x_i) &= S_{i-1}^{'}(x_i) \\ S_i^{''}(x_i) &= S_{i-1}^{''}(x_i) \end{aligned} \tag{3.3}$$

Since the tongue contours are closed contours, the curve of the last landmark should properly be connected to the first landmark. This is accomplished by including the following constraints: $S_1(x_1) = S_l(x_1)$, $S_1^{'}(x_1) = S_l^{'}(x_1)$ and $S_1^{''}(x_1) = S_l^{''}(x_1)$. Figure 3.2 shows three examples of MR images with assigned tongue landmarks and calculated spline curves.

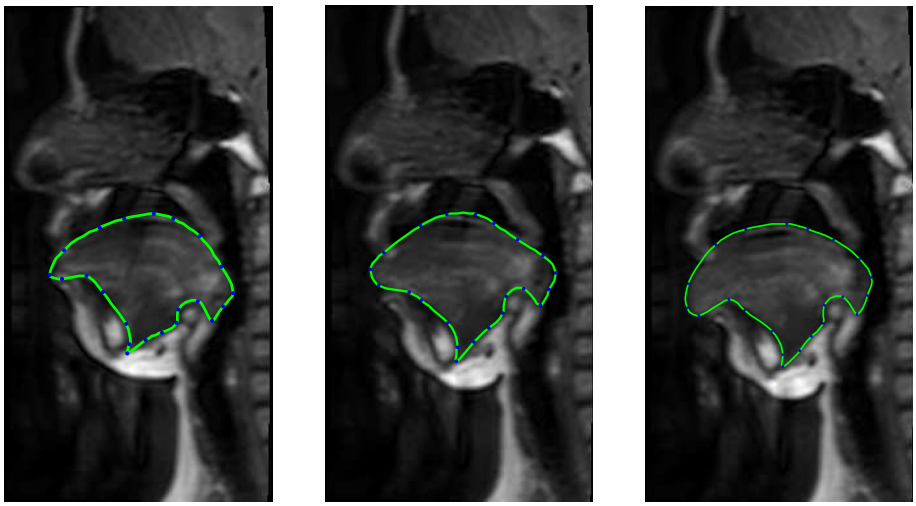


Figure 3.2: *Examples of MR images in the mid-sagittal plane with assigned tongue landmarks and calculated spline curves.*

3.3.2 Training stage

In the training stage data is generated that specifies the active shape model. The ASM-data can be used to find a shape, in a new image, that is similar to the shapes in the training set. During training, the following operations take place: generating profile statistics, aligning the training shapes, and extracting the modes of variation from the aligned training set.

Generating profile statistics

During each iteration in the application stage, a suggested movement for each shape point will be calculated by matching its local structure with a statistical model of the corresponding landmark. The model for a certain landmark is obtained by calculating its texture profile in each training image. So, suppose the training set consists of I images, with for each image a (manually) determined shape specified by l landmarks. The profile of the j^{th} landmark in the i^{th} image is then obtained by taking k samples at either side of the landmark (see figure 3.3).

Since the sample points are most of the times not exactly located in the middle of a pixel, it was decided to apply bilinear interpolation:

$$g_{s_{ij}} = g_a(1-\alpha)(1-\beta) + g_b(\alpha)(1-\beta) + g_c(1-\alpha)(\beta) + g_d(\alpha)(\beta) \qquad (3.4)$$

In this equation g_a, g_b, g_c and g_d are the values of the four nearest pixels around the sample point s_{ij} and α and β are respectively the horizontal and vertical distance from the sample point to the centers of pixel a. The $2k+1$ samples are put in a vector $\mathbf{g}_{ij}$. To reduce the effects of global intensity changes (i.e. offset differences), the sampled profile is differentiated and then normalized by

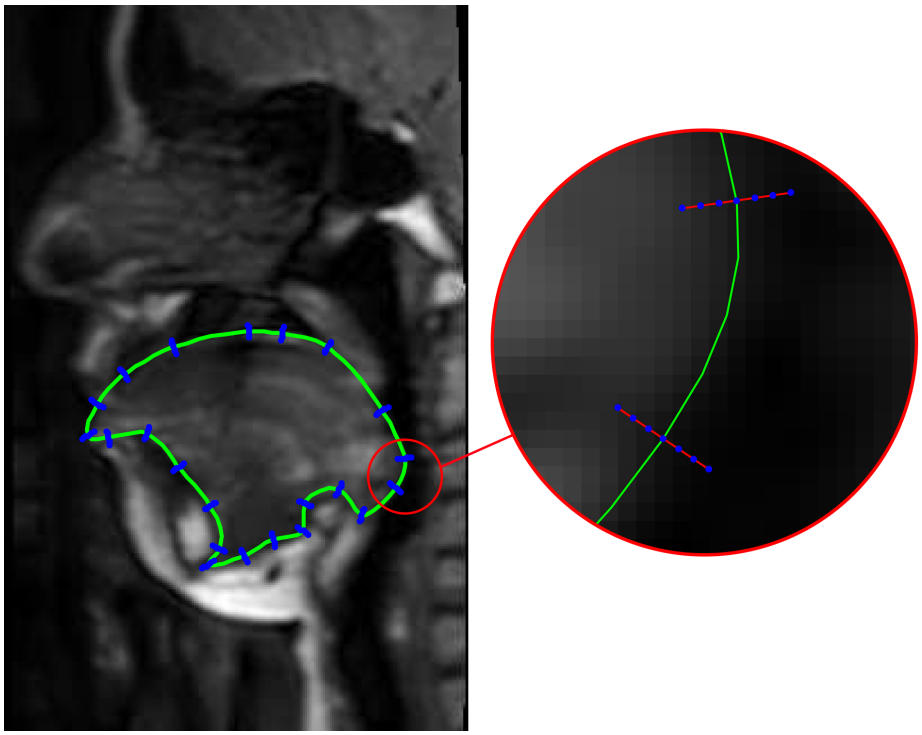


Figure 3.3: *For each landmark, samples are taken along a profile perpendicular to the contour. Sampling is done by using bilinear interpolation.*

dividing by the Euclidean distance of the differentiated vector $d\mathbf{g}_{ij}$. This results in a profile vector of length $2k$:

$$\mathbf{g}_{ij} \rightarrow \frac{d\mathbf{g}_{ij}}{\sqrt{\sum_{s=1}^{2k} dg_{s_{ij}}^2}} \tag{3.5}$$

The procedure is repeated for each training image and results in a set of I normalized profile vectors for each landmark point. Assuming that these vectors are distributed as a multivariate Gaussian, the mean profile vector $\bar{\mathbf{g}}_j$ and covariance matrix $\mathbf{Sg}_j$ of the j^{th} landmark can be calculated as follows:

$$\bar{\mathbf{g}}_j = \frac{1}{I}\sum_{i=1}^{I} \mathbf{g}_{ij} \tag{3.6}$$

$$\mathbf{Sg}_j = \frac{1}{I-1}\sum_{i=1}^{I} \left(\mathbf{g}_{ij} - \bar{\mathbf{g}}_j\right)\left(\mathbf{g}_{ij} - \bar{\mathbf{g}}_j\right)^T \tag{3.7}$$

Aligning the training set

During the acquisition of the MRI data, the head might have moved a bit. This kind of small movements results in small pose differences - between the

shapes - that are not caused by actual tongue movements. For the extraction of the statistical shape parameters, it is important that these pose differences are filtered out. Therefore the shapes are first aligned to each other by applying a transformation T_i on the landmarks of each shape $\mathbf{x}_i$, consisting of a translation $(X_t, Y_t)_i$, a rotation θ_i and a scaling s_i. For instance, if applied on a single landmark (x, y):

$$T_{X_t,Y_t,s,\theta} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} s\cos\theta & s\sin\theta \\ -s\sin\theta & s\cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} X_t \\ Y_t \end{bmatrix} \tag{3.8}$$

Aligning the shapes is an iterative process. First, all the shapes are translated such that their centers of gravity are at the origin. In each iteration the shapes are aligned, one by one, to the current estimate of the mean shape. Initially, the first shape in the training set is chosen as the mean shape and after each iteration the mean shape is re-estimated from the aligned set. The process continues until the mean shape does not change significantly after one iteration. The pseudo code of the alignment process is as follows:

1. Translate each shape such that its center of gravity is at the origin.
2. Choose first shape in set as initial estimate of mean shape: $\bar{\mathbf{x}} = \mathbf{x}_1$.
3. Start iterative alignment:
   (a) Align shapes one by one to the estimated mean shape.
   (b) Re-estimate mean shape from aligned shapes.
   (c) Return to 3(a) unless convergence or a maximum number of iterations is reached.

So, each iteration consists of the alignments of two shapes (shape i to the current estimate). However, there is no unique solution for the alignment of two shapes. The shapes are namely specified by a whole set of landmarks, while there are only four transformation parameters. Therefore, the transformation parameters (X_t, Y_t, s, θ) for the alignment of shape $\mathbf{x}_i$ onto the mean shape $\bar{\mathbf{x}}$ are calculated by minimizing the following quadratic criterion:

$$E = (\bar{\mathbf{x}} - T(X_t, Y_t, s, \theta)\mathbf{x}_i)^T \mathbf{W} (\bar{\mathbf{x}} - T(X_t, Y_t, s, \theta)\mathbf{x}_i) \tag{3.9}$$

In this equation $\mathbf{W}$ is a diagonal matrix of weights for each landmark. These weights are based on the variance of each landmark in the training set. The weight w_j for the j^{th} landmark is calculated as follows:

$$w_j = \left(\sum_{k=1}^{l} V_{R_{jk}} \right)^{-1}, \tag{3.10}$$

where R_{jk} represents the distance between landmarks j and k in a shape and $V_{R_{jk}}$ the variance in this distance over the set of shapes:

$$V_{R_{jk}} = \mathrm{Var} \left\{ \sqrt{(x_{i,j} - x_{i,k})^2 + (y_{i,j} - y_{i,k})^2} \right\}, i = 1, \ldots, l \tag{3.11}$$

So, stable landmarks have a low variance and are assigned with a higher weight than unstable landmarks, which have a high variance. Minimizing criterion (3.9) is done by differentiating the equation with respect to each of the four variables X_t, Y_t, s, θ and equating the resulting equations to zero. The alignment of shape $\mathbf{x}_i$ to the mean shape $\bar{\mathbf{x}}$ is then accomplished by carrying out the transformation of equation (3.8) on $\mathbf{x}_i$ with the calculated transformation parameters. Figure 3.4 shows the result of the alignment of ten tongue shapes.[1]

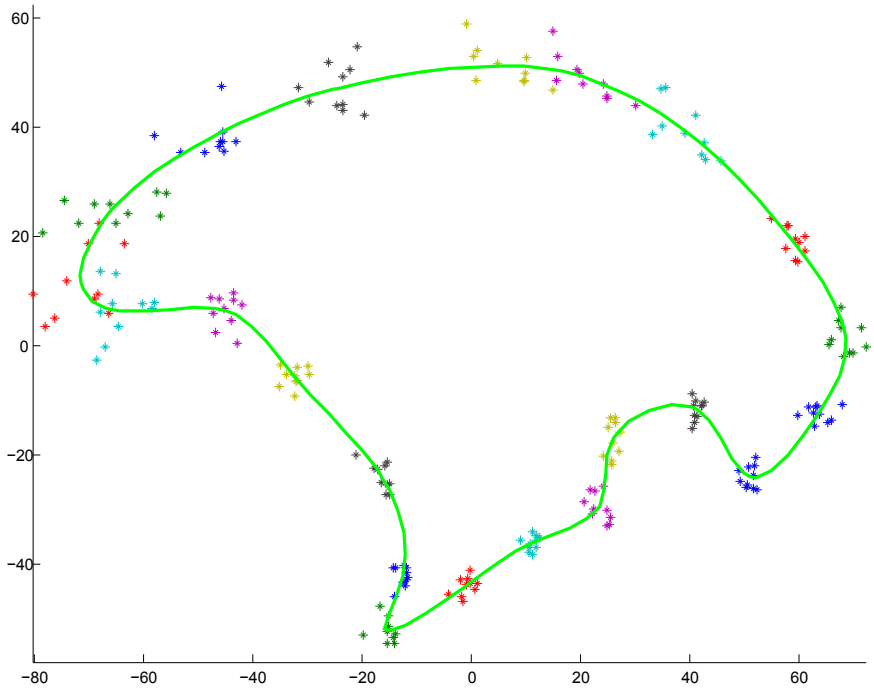


Figure 3.4: *Aligned training shapes (scatter plots) and mean shape (green curve).*

Modeling shape variation

From the aligned set of training shapes, a Point Distribution Model (PDM) can be derived, consisting of the basic modes of shape variation. With these modes, new examples of shapes can be generated that are similar to those in the training set. As can be seen in figure 3.4, some landmarks show little variability over the training set, while others form more diffuse clouds. The PDM seeks to model the variation of the coordinates within these clouds, but the PDM also takes into account that landmarks do not move independently - their positions are partially correlated. For the generation of the shape modes, Principal Component Analysis (PCA) is applied on the training set. PCA can transform the original data to a new coordinate system with less dimensions. This is done by calculating the eigenvalues λ_k $(k = 1, \ldots, 2l)$ and eigenvectors

[1] The difference with the unaligned set is in this case not very large, since the head was kept quite stable during the MRI-scans.

$\mathbf{p}_k$ of the covariance matrix $\mathbf{S}$ of the original data set:

$$\bar{\mathbf{x}} = \frac{1}{I}\sum_{i=1}^{I}\mathbf{x}_i \tag{3.12}$$

$$\mathbf{S} = \frac{1}{I}\sum_{i=1}^{I}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \tag{3.13}$$

$$\mathbf{S}\mathbf{p}_k = \lambda_k \mathbf{p}_k \tag{3.14}$$

The covariance matrix indicates how much the dimensions vary from the mean with respect to each other. The eigenvectors with the highest eigenvalues contain the most information and are the principal components of the data set. A new shape example can now be generated by taking the mean shape, $\bar{\mathbf{x}}$, and a weighted sum of the modes of variation:

$$\mathbf{x}_{new} = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b} \tag{3.15}$$

In this equation, $\mathbf{P}$ is a matrix consisting of t eigenvectors corresponding to the first t highest eigenvalues of the covariance matrix and $\mathbf{b}$ is a vector of weights (shape parameters):

$$\mathbf{P} = [\mathbf{p}_1\ \mathbf{p}_2\ \ldots\ \mathbf{p}_t] \tag{3.16}$$

$$\mathbf{b} = [b_1\ b_2\ \ldots\ b_t]^T \tag{3.17}$$

The number of modes t can be chosen so that it explains a certain proportion (e.g. 95 %) of the total variance in the training set (which is the sum of all the eigenvalues). To generate only plausible shapes, the values of b_k have to be limited. Since most of the population lies within three standard deviations of the mean, suitable limits are:

$$-3\sqrt{\lambda_k} \leq b_k \leq 3\sqrt{\lambda_k} \tag{3.18}$$

Figure 3.5 shows examples of tongue shapes by varying the weights of the first, second and third eigenvectors (shapes modes) within the allowed limits. As can be seen in this figure, these parameters mainly control the tip of the tongue. This corresponds with figure 3.4, since most of the variation occurs at this part of the tongue.

3.3.3 Application stage

In the application stage the generated ASM data is used to find a shape, similar to the shapes in the training set, in a new image. This is done in an iterative process. In each iteration a suggested movement for the current shape is calculated, based on the detection of model features. The shape is then transformed (scaled, rotated and translated) and deformed (in allowed deformation modes) to best match the shape to the new points. Since tongue shapes can have a large deviation from the mean shape, it was decided to further improve the detection range. This was accomplished by the implementation of multi-resolution ASM, which implies feature search on different resolution levels.

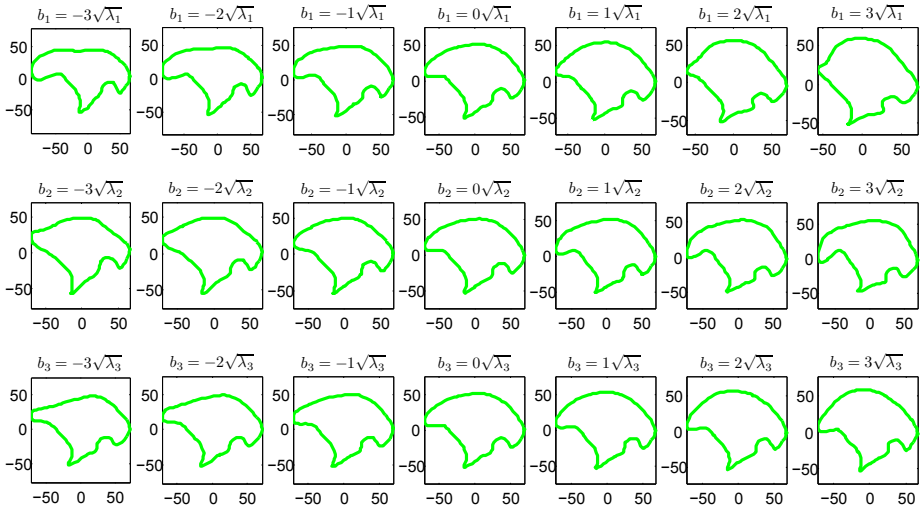


Figure 3.5: *Shape effects of parameter variation.*

Calculating a suggested movement

Given a certain shape, consisting of a set of model points in the image frame, the algorithm should determine a set of adjustments which will move each point toward a better position. Since the model points represent the boundary of the object, this involves moving them toward image edges. For the calculation of these suggested movements, the local texture profile of each shape point in the image is calculated. This is done in the same way the texture profiles are calculated during training, but now with a larger number of sample points along a longer profile normal.

The sample profile g_s of a certain shape point consists of m sample points either side of the shape point, where $m > k$ (k is the number of sample points either of the shape points during training). Next, the quality of fit with the training model is determined for each of the $2(m-k)+1$ possible positions, along the profile, by calculating the Mahalonobis distances (3.19) or the Euclidean distances (3.20):

$$D_M(\mathbf{g}_s) = (\mathbf{g}_s - \bar{\mathbf{g}})^T \mathbf{S}_g^{-1} (\mathbf{g}_s - \bar{\mathbf{g}}) \tag{3.19}$$

$$D_E(\mathbf{g}_s) = (\mathbf{g}_s - \bar{\mathbf{g}})^T \mathbf{I} (\mathbf{g}_s - \bar{\mathbf{g}}) \tag{3.20}$$

The Mahalonobis is scale-invariant and takes into account the correlations in the data set by using the covariance matrix S_g, see equation (3.7). However, in case of a small training set with not too much variation, the covariance matrix might become ill-conditioned. In such a case, calculating the matrix inverse can be a problem. This problem can be avoided by using the Euclidean distance or by applying matrix regularization (see appendix D). The best profile match is than calculated as follows:

$$D_R(\mathbf{g}_s) = (\mathbf{g}_s - \bar{\mathbf{g}})^T (\mathbf{S}_g + \gamma \mathbf{I})^{-1} (\mathbf{g}_s - \bar{\mathbf{g}}) \tag{3.21}$$

In this equation γ has a small value (e.g. 0.1), but big enough to make the matrix $\mathbf{S}_g + \gamma\mathbf{I}$ invertible. The position which gives the best profile match with the model (lowest value of $D(\mathbf{g}_s)$) is chosen as the suggested movement position for the concerning shape point, see figure 3.6.

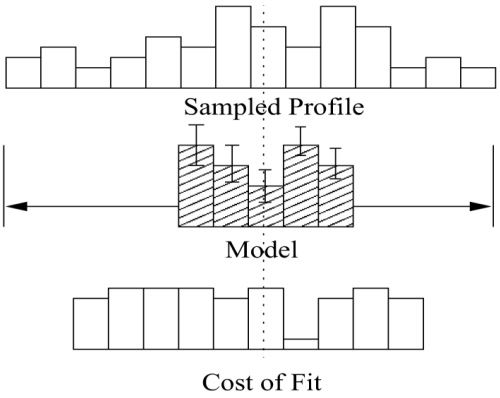


Figure 3.6: *Search along sample profile to find best match with training model.*

Updating pose and shape parameters

After a new position is calculated for each point, the current pose and shape parameters should be updated to best match the current shape positions $\mathbf{x}$ to the set of suggested positions $\mathbf{x}_{sug}$. The parameter are updated as follows:

$$\begin{aligned} X_t &\rightarrow X_t + dX_t \\ Y_t &\rightarrow Y_t + dY_t \\ \theta &\rightarrow \theta + d\theta \\ s &\rightarrow s(1 + ds) \\ \mathbf{b} &\rightarrow \mathbf{b} + d\mathbf{b} \end{aligned} \tag{3.22}$$

The translation (dX_c, dY_c), rotation $d\theta$, scaling factor $1 + ds$, and shape $d\mathbf{b}$ are calculated such that the following expression is minimized:

$$|\mathbf{x}_{sug} - T_{X_t,Y_t,s,\theta}(\bar{\mathbf{x}} + \mathbf{P}\mathbf{b})|^2 \tag{3.23}$$

Calculating (dX_c, dY_c), $d\theta$ and $1 + ds$ is done by using the same function (and weight matrix) used during training for the alignment of two shapes. The update, $d\mathbf{b}$, for the shape parameters is based on the residual adjustments dx (after applying the transformation with the pose parameters). $d\mathbf{b}$ is calculated such that:

$$\mathbf{x} + d\mathbf{x} \approx \bar{\mathbf{x}} + \mathbf{P}(\mathbf{b} + d\mathbf{b}) \tag{3.24}$$

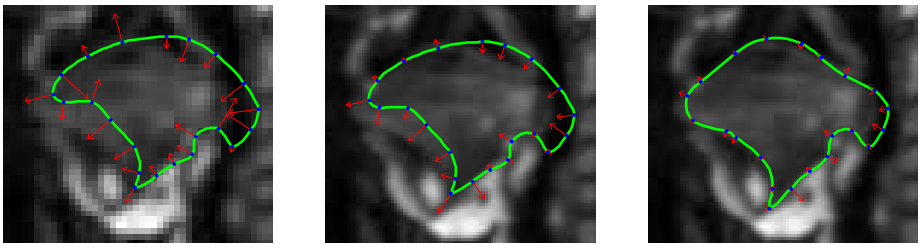
Since there are only t modes of variation available and $d\mathbf{x}$ can move the points in $2l$ different degrees of freedom, only an approximation to the required deformation can be achieved. To maintain only plausible shapes, the shape parameters will be limited according to equation (3.18). If the initial value of an updated parameter exceeds its limit, than the value becomes that limit.

Multi-resolution search

To improve the efficiency and the detection range of the ASM algorithm, it is implemented in a multi-resolution framework. This involves first searching in a low-resolution image, to find the location of the object on a coarse scale. Then searching is performed in a series of higher resolution images, to refine the location of the object. This extension leads to a faster algorithm, and one which is less likely to get stuck on wrong image structures.

For each training and application image, a set of different resolution images is created. The base image (level 0) is the original image. The next image (level 1) is formed by smoothing and subsampling the original to obtain an image with half the number of pixels in each dimension. Images on subsequent levels are formed by further smoothing and subsampling. During training, a PDM is build for each resolution level. The same number of sample points on the profile normals is used, regardless of the level. Since pixels of images at level L are 2^L times the size of those in the original image, the models at coarser levels represent more of the image. During search in the application stage, this will allow quite large movements. At finer resolution levels, the feature detection is more precise and this leads to smaller movements.

During search, the algorithm needs to decide when to switch to the next (higher) resolution level or to stop searching. This is done by recording the number of times that the best found position, along a search profile, is within a certain percentage (e.g. 50 %) of the profile length. When a sufficient number (e.g. 90 %) of these positions are found, the algorithm is declared to have converged at that resolution. The current shape model is then projected into the image at the next level and searching is performed again untill convergence is reached. When convergence is reached on the finest resolution level (level 0), the search is stopped. Figure 3.7 shows three images with the shape location during the search process at different resolution levels. In this case confergence was reached in nine iterations (two iterations at level 2, three iterations at level 1 and four iterations at level 0).



(a) *Level 2* **(b)** *Level 1* **(c)** *Level 0*

Figure 3.7: *Object search at different resolution levels. The red arrows indicate the suggested movements before pose and shape transformation.*

The multi-resolution version of the ASM-algorithm can be summarized as follows:

1. Set $L = L_{max}$.
2. While $L \geq 0$
   (a) Compute model point positions in image at level L.
   (b) Calculate a suggested movement for each model point.
   (c) Update pose and shape parameters to fit model to suggested points.
   (d) Return to 2(a) unless convergence or a maximum number of iterations is reached at this resolution level.
   (e) If $L > 0$ than $L \rightarrow L - 1$.
3. Final shape is described by the parameters after convergence at level 0.

Object tracking in sequences

In image sequences that describe the evolution of an object in time, like a tongue movement, the difference between object shapes in two adjacent images is small. This fact can be used to improve the contour detection and tracking of the object, by using the parameters $\mathbf{p}_{i-1}$, that describe the final shape in image $i-1$, as initial parameters $\mathbf{q}_i$ for the shape in image i. However, to prevent that a possible detection error in a certain image fully propagates into the next images, the mean parameter vector $\bar{\mathbf{p}}$ should be included as well:

$$\mathbf{q}_i = \alpha \mathbf{p}_{i-1} + (1-\alpha)\bar{\mathbf{p}}_{i-1} \tag{3.25}$$

In this equation, α $(0 \leq \alpha \leq 1)$ indicates the ratio between the preceding parameter vector and the mean parameter vector. The parameter vector contains both the pose and shape parameters: $\mathbf{p}_i = \left[X_{t,i}\ Y_{t,i}\ \theta_i\ s_i\ \mathbf{b}_i^T\right]^T$. The mean parameter vector is updated after each search:

$$\bar{\mathbf{p}}_i = \frac{1}{i}\sum_{j=1}^{i} \mathbf{p}_i \tag{3.26}$$

The initial parameters $\mathbf{q}_1$ for the shape in the first image have to be properly determined by the user. The values of these parameter should be chosen such that the initial shape is relative close to the object contour.

3.4 Performance evaluation

The in section 3.3 described ASM algorithm is implemented in MATLAB. The performance of the algorithm is tested by using MRI data. This MRI data consists of image sequences in the sagittal and transverse plane during simple and slow tongue movements. Objective was to detect and track the tongue contour in these MRI sequences.

3.4.1 MRI data

The used MRI data is obtained by using the MRI scanner at the department of clinical physics at the Netherlands Cancer Institute / Antoni van Leeuwenhoek Hospital, Amsterdam, the Netherlands. The MRI scanner is a Philips Achieva 3-Tesla scanner[2], see figure 3.8. A female test person was asked to perform the tongue movements, with her head positioned in the scanner.

Parameters MRI scanner

The sequences are captured with the following MRI parameters:

- 2D TSE single shot FA 90, TA = TR = 0.794 s (NSA = 1), TE = 44.
- FOV 230x122 mm acq pixel 1.8x2 mm recon pixel 0.6 mm S=L 5mm.
- Water fat shift 1.5 pixel BW 295 Hz.
- TSE factor 37.

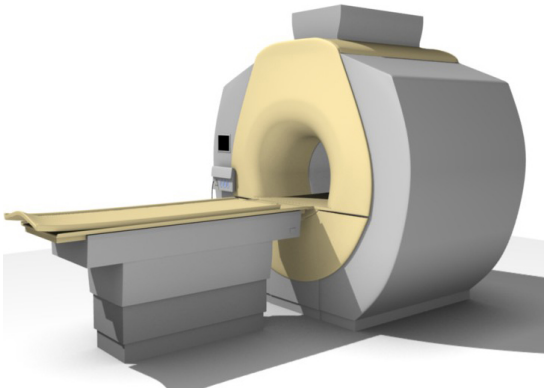


Figure 3.8: *Philips Achieva 3-Tesla MRI scanner.*

Tongue movements

MR Image sequences are captured in the sagittal and transverse plane (in the coronal plane the visible displacement is not very large) during some simple tongue movements: in and out sticking and movement from left to right. Since the acquisition time per MR image was 0.8 seconds, the tongue movements had to be carried out slowly. The MRI sequences consist of 50 images with a resolution of about 200 × 384 pixels (sagittal) and 200 × 200 pixels (transverse).

[2]For more specifications concerning the MRI scanner, see: http://www.medical.philips.com/us/products/mri/systems/achievatx/index.wpd

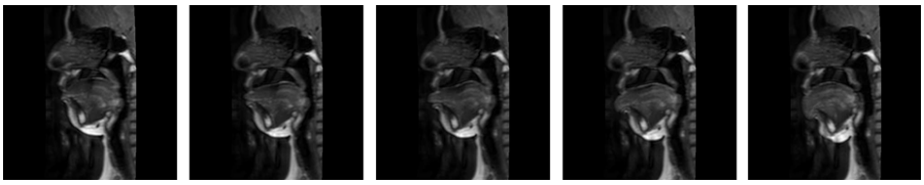


Figure 3.9: *Examples of sagittal MR images during protruding the tongue.*

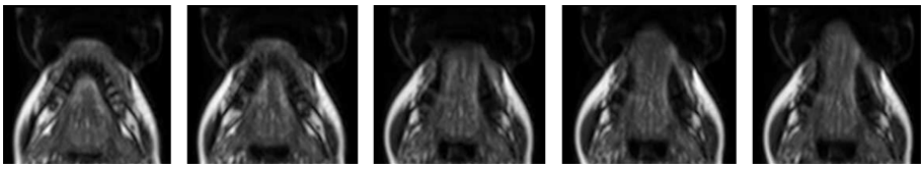


Figure 3.10: *Examples of transverse MR images during protruding the tongue.*

3.4.2 ASM parameters

The construction of the ASM training model requires the input of the following input arguments and parameters:

- A set of training images (representative for the whole data set);
- A matrix with landmark coordinates for each training image;
- k: the number of profile samples at either side of the landmarks during training;
- m: the number of profile samples at either side of the landmarks during application ($m > k$);
- L_{max}: the number of resolution levels (on which search is performed);
- t: the number of shape modes;
- $\mathbf{q}_1$: The initial shape and pose parameters for the shape in in the first image.

For obtaining a reliable ASM model, as much training images as possible should be used. These training images should be representative for the images in the application set (in terms of shape variation and texture profiles). However, in our case, the captured sequence during the tongue movement consists of just 50 images. Therefore, it was decided to use ten from the fifty sequence images for training: the fifth, the tenth, the fifteenth, etc. These images were assigned with 22 landmarks, most of them at places of interest (e.g. corners, see figure 3.2).

The choice of the number of shape modes is commonly based on the relation between the number of features ($p = 2 \times 22 = 44$) and the number of training images. According to a rule of thumb, there about $5 \times p = 220$ images required to properly estimate the covariance matrix. However, since in this case just 10 training images are used, only the first few eigenvectors can be trusted. It

was decided to choose the number of shape modes such that 95% of the total variation is explained. For the used training set, this resulted in six modes for the sagittal images and four modes for the transverse images.

Unfortunately, in literature not much is written about how to choose the exact parameter values. The values of k, m and L_{max} mainly depend on the resolution of the images. k should be large enough to obtain a proper 'description' of the local texture profiles. However, k should be also not too large, to prevent that variating structures are taken into account. The exact values are determined empirically. Adequate choices appeared to be: $k = 4$, $m = 9$ and $L_{max} = 3$. The initial shape and pose parameters were chosen such that the first shape is at the center (X_c, Y_c) of the image ($X_{t,1} = X_c$, $Y_{t,1} = Y_c$, $\theta_1 = 0$, $s_1 = 1$, $\mathbf{b}_1 = \mathbf{0}$).

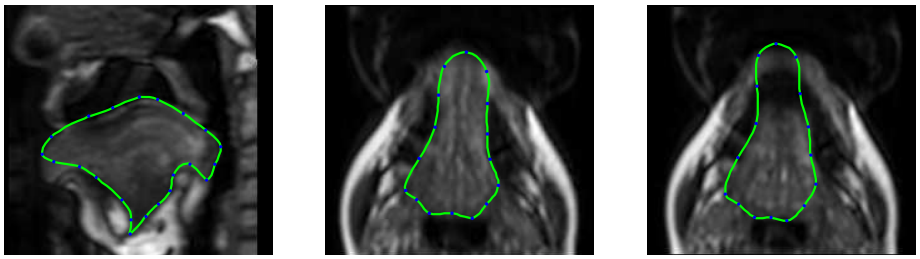
3.4.3 Experimental results

The Active Shape Model, with the parameters chosen as motivated in the previous subsection, has been applied on the MRI sequences of tongue movements in the sagittal plane and in the transverse plane. Especially in the sequence of sagittal images, the model performs well: in most images the tongue contour is detected correct. However, in 16 of the 50 images small detection errors are made. These detection errors can be caused by vague object edges or by inconsistency with the training model. Figure 3.11a shows an example of an image where such a detection error is made. An image of the tongue protruded this way is not included in the training set. So, the ASM model does not allow to deform in this way.

Detection and tracking of the tongue contour in the transverse images appeared to be more difficult. It happens frequently that the model snaps at wrong edges. Figure 3.11c shows an example of this kind of detection errors. The main cause is the big difference with the tongue shape in the previous image (figure 3.11b) of the sequence. In this image the tongue is protruded such that the tongue tip strikes the lip. And, as can be seen, the difference between the tongue tip (which is actually inside the mouth) and a part of the lip contour is very small. So, the algorithm actually considers the lip as part of the tongue in this case. Also the choice of the initial shape and pose parameters appeared to have much influence on the detection performance. When the initial shape in the first image is too far away from the actual tongue contour, the algorithm is not able to find the correct contour and the errors also propagate into the next images.

3.5 Conclusions

In this chapter the implementation and performance of an Active Shape Model algorithm are discussed. The ASM algorithm can be used to detect object contours with vague edges in images contaminated with noise. The basic idea is that from a set of training images a model is inferred that represents the mean geometry of the concerning shapes and statistical modes of geometric variation.



(a) *Sagittal* **(b)** *Transverse at* $i = 7$ **(c)** *Transverse at* $i = 8$

Figure 3.11: *Examples of tongue contour detection errors.*

This model can then be used to find similar shapes in new images. Some small extensions, compared to the basic ASM version, have been implemented to make the algorithm especially suitable for the application of contour detection of a moving object in an image sequence. The algorithm is therefore very useful for detection and tracking of tongue movements in a sequence of noisy magnetic resonance images. In case of such a sequence, tongue landmarks have to be assigned in only a small number of the images. The resulting model can then be used to detect the tongue contour in the other images. Although the ASM algorithm is only tested on MR images of the tongue, it is expected that the algorithm can also be used for lip contour detection in normal images.

An important issue in using ASM for object contour detection is training. The resulting model should be representative for the application images. Most of the detection errors are caused by deviations from the actual model (concerning both the allowed shapes and expected texture profiles). Although the ASM algorithm has a relative large detection range, it followed from the experiments that the choice of the initial shape parameters is also important. These parameters should be chosen such that the initial model shape is close to the object, otherwise detection errors are made easily. Furthermore, it can be concluded from the experiments that the algorithm performs well when the object in the images is clearly visible. In some images (e.g. figure 3.11c) the complete tongue is hardly visible with the bare eye. In such a case also the algorithm fails. Another disadvantage of the ASM algorithm is that landmarks in the training images have to be assigned manually. This can be a time consuming task. Ideally a fully automated system would be developed, that is able to automatically place landmarks on a presented set of training images. However, this is a difficult task, mainly because it is not clear what optimal landmark locations are.

4

Linear state space model

4.1 Introduction

This chapter is concerned with the setup and investigation of a possible model framework for the description of a physical system like the tongue and lips. It is expected that in the near future measured EMG data of muscle activation signals will become available, which can be used to derive the actual input signals. However, in the current research this input data is not yet available and this research therefore limited to measured output variables (lip/tongue movements). The objective here is to already perform an exploration study concerning the development of a phenomenological model of the tongue and lips. In a later stage, this model can be coupled to actual muscle activation signals.

The key challenges in this chapter and in the next chapter are the construction of an appropriate state vector and the estimation of system parameters and input variables. However, because of the limited available information it is necessary to make some assumptions about the actual system and input signals. These assumptions will be motivated in this chapter. For the derivation and estimation of the mentioned parameters and variables, the general framework of figure 4.1 will be considered:

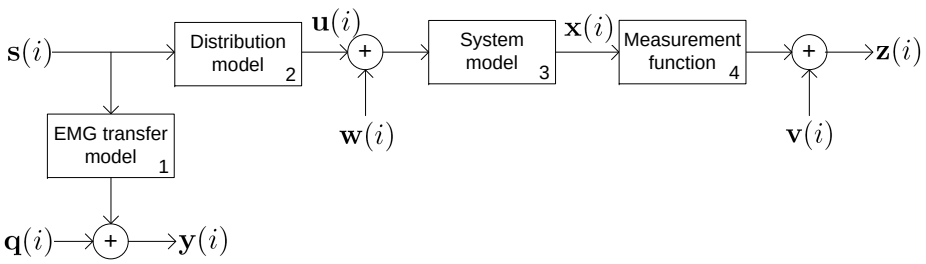


Figure 4.1: *General framework for a model of the tongue and lips.*

In this model framework the following variables are involved (i represents the discrete-time index, with $i = 0, 1, 2, \ldots, I-1$):

- $\mathbf{s}(i) \in \Re^k$: Muscle activation signals
- $\mathbf{y}(i) \in \Re^q$: EMG measurements
- $\mathbf{u}(i) \in \Re^m$: Model input variables
- $\mathbf{x}(i) \in \Re^n$: System states
- $\mathbf{z}(i) \in \Re^p$: Output variables (lip/tongue landmarks)
- $\mathbf{r}(i) \in \Re^q$: EMG measurement noise
- $\mathbf{v}(i) \in \Re^p$: Shape measurement noise (landmark detection noise)
- $\mathbf{w}(i) \in \Re^n$: Process noise

The relations between the variables are described by the four blocks:

1. EMG transfer model – Describes the transfer from the actual muscle activation signals $\mathbf{s}(i)$ to the measured EMG signals $\mathbf{y}(i)$. Note that the dimensions of these vector are not necessary equal to each other. It might for example be possible that less signals can be measured than actually are involved.
2. Distribution model – Describes the coupling between the actual muscle activation signals $\mathbf{s}(i)$ and the model input variables $\mathbf{u}(i)$.
3. System model – Describes how the system states $\mathbf{x}(i)$ depend on the input variables $\mathbf{u}(i)$ and possibly on previous states, e.g. $\mathbf{x}(i-1)$.
4. Measurement function – Describes the relation between system states $\mathbf{x}(i)$ and output variables $\mathbf{z}(i)$.

The ultimate objective is to identify in figure 4.1 the content of the blocks 2, 3 and 4, such that the causal relation between actual muscle activation signals and output (tongue and lip shapes and movements) can be established. Note that for the identification of block 2 (the distribution model) it is necessary to know the content of block 1 (the EMG transfer model) and to have the measured EMG signals available. The research for now is concerned with the parts between $\mathbf{u}(i)$ and $\mathbf{z}(i)$ in the figure and on the identification of blocks 3 and 4. Therefore, this chapter explores the setup of a possible dynamic system model. In section 4.2 this system model will be defined. Several possible state vectors will be discussed in section 4.3 and the construction or estimation of system matrices will be considered in section 4.4. Furthermore, in section 4.5 techniques will be explained that can be used for the evaluation and justification of the model in practice.

4.2 Model setup

The starting point is $\mathbf{z}(i)$, a sequence of measured output vectors (e.g. the detected tongue landmarks). The first step is to choose a possible system model (see section 2.2.4). In a simple case this is just a static model, which means that the state variables are a function of only the input variables, i.e. $\mathbf{x} = f(\mathbf{u}(i))$. An example of a static model is actually already considered in chapter 3, where static input variables are obtained by applying PCA. However, tongue and lip movements are probably dynamic processes.

In general, a dynamic process can be described in discrete-time with the equation $x(i+1) = f(x(i), u(i))$, which means that the next state is a function of the current state and the current (in our case unknown) input. However, in most situations, and especially in case of phenomenological modeling of the tongue and the lips, it is difficult to derive a non-linear function. Therefore the dynamical description of tongue or lip movements will (initially) be limited to a discrete-time, linear, time-invariant state space model. This is a restriction, of course. However, this type of model has already proved to approximate many real-world problems and physical processes accurately and it can be a good starting point for finding more elaborate, non-linear models. The state space model will therefore be described with the following set of difference equations:

$$\begin{aligned} \mathbf{x}(i+1) &= \mathbf{F}\left(\mathbf{x}(i) - \bar{\bar{\mathbf{x}}}\right) + \bar{\bar{\mathbf{x}}} + \mathbf{L}\mathbf{u}(i) + \mathbf{w}(i) \\ \mathbf{z}(i) &= \mathbf{H}\mathbf{x}(i) + \mathbf{v}(i) \end{aligned} \tag{4.1}$$

The system matrix $\mathbf{F} \in \Re^{n \times n}$ describes how the next state vector is linearly formed from the current state values. The matrix $\mathbf{L} \in \Re^{n \times m}$ represents the input matrix and describes the influence of the input signals on the next state vector. The matrix $\mathbf{H} \in \Re^{p \times n}$ is the measurement matrix and translates the state variables to the output variables. The system equations are linearized around an equilibrium state vector $\bar{\bar{\mathbf{x}}}$, such that the same equilibrium can be chosen for the sequences of all phonemes or visemes[1]. This equilibrium vector can be seen as the state vector corresponding to the tongue or lips in equilibrium or rest position. The process noise $\mathbf{w}(i)$ and measurement noise $\mathbf{v}(i)$ are modeled as zero mean, Gaussian white noise sequences and with covariance matrices $\mathbf{C}_w$ and $\mathbf{C}_v$. Process noise and measurement noise are assumed to be uncorrelated: $\mathbf{C}_{wv} = \mathbf{0}$. The elements of the process noise vector $\mathbf{w}(i)$ can sometimes also be considered as the unknown input signals at time i.

4.3 State vectors

The measurement vector consists of p elements, which are the measured coordinates of the $l = 0.5p$ landmarks (either on the tongue or lip contour):

$$\mathbf{z}(i) = \left[x_1^i, x_2^i, \ldots, x_l^i, y_1^i, y_2^i, \ldots, y_l^i\right]^T \tag{4.2}$$

[1] A visme is a basic unit of speech in the visual domain.

The first task is now to construct an appropriate state vector. In this section several possibilities concerning the representation of the state vector will be considered.

4.3.1 Position only

In the simplest case the state vector only consists of the actual landmark positions $\mathbf{x}_p$. The measurement matrix $\mathbf{H}$ (the third block in figure 4.1) is than just the identity matrix I and the state vectors are equal to the measured positions (including measurement noise). Since the measurement noise is assumed to have zero mean, the mean state vector $\bar{\bar{\mathbf{x}}}$ is equal to the mean $\bar{\bar{\mathbf{z}}}$ of the measurement vectors.

4.3.2 Position, velocity and acceleration

The set of landmark coordinates changes dynamically in time. Therefore a more advanced and dynamic representation of the state vector would also include the velocity ($\mathbf{x}_v$) and possibly even the acceleration ($\mathbf{x}_a$) components of the landmarks. The state vectors are in that case also provided with information about the transitions. The length of these state vectors becomes $3l$:

$$\mathbf{x}_{pva}(i) = \left[\mathbf{x}_p^T(i), \mathbf{x}_v^T(i), \mathbf{x}_a^T(i)\right]^T \tag{4.3}$$

where $\mathbf{x}_v(i)$ and $\mathbf{x}_a(i)$ are the velocity and acceleration vectors:

$$\begin{aligned} \mathbf{x}_v(i) &= \left[v^i_{x_1}, v^i_{x_2}, \ldots, v^i_{x_l}, v^i_{y_1}, v^i_{y_2}, \ldots, v^i_{y_l}\right]^T \\ \mathbf{x}_a(i) &= \left[a^i_{x_1}, a^i_{x_2}, \ldots, a^i_{x_l}, a^i_{y_1}, a^i_{y_2}, \ldots, a^i_{y_l}\right]^T \end{aligned} \tag{4.4}$$

The measurement matrix should only extract the position elements of the state vector. These are the first p elements of the vector. In equilibrium or rest position, it will be assumed that the velocity and acceleration are zero. So, the measurement matrix and the mean state vector are in case of including velocity and acceleration states given by the following equation:

$$\begin{aligned} \mathbf{H} &= [\mathbf{I}, \mathbf{0}, \mathbf{0}] \\ \bar{\bar{\mathbf{x}}}_{pva} &= [\bar{\bar{\mathbf{z}}}^T, \mathbf{0}^T, \mathbf{0}^T]^T \end{aligned} \tag{4.5}$$

The velocity and acceleration components of the landmarks during movement cannot easily be measured (unless the tags are provided with acceleration sensors or something similar). So, they have to be derived from the sequence of measured positions. The easiest way to find the velocity of a certain landmark (either the x- or y-component) at time i implies subtracting the previous position (at $i-1$) from the current position and dividing the difference by the sample time Δt, e.g. $v_{x_j}(i) = (x_j(i) - x_j(i-1))/\Delta t$. The acceleration can be found by applying the same operation on the calculated velocity sequence.

However, the just described way of calculating velocity and acceleration is not measurement noise robust and does not compensate for abrupt transitions

(no smoothing). A neater method for deriving the the velocity and acceleration components can be accomplished by using a Kalman filter. Details of the Kalman filter are described in section 4.5.1.

4.3.3 Dimension reduction using PCA

Principal Component Analysis (PCA) [22] can be be applied to reduce the size of the original measurement vector (e.g equation (4.3)) and to come close to the state vector that contains the minimum number of variables to describe the dynamic behavior. By choosing an appropriate number of eigenvectors of the covariance matrix of the original set of vectors, the dimensions can significantly be reduced, while nearly information is lost. The latter of course depends on the correlation between the vector elements. But as already shown in section 3.3.2, landmarks on the tongue during movement are certainly correlated.

The procedure for obtaining those eigenvectors is described in section 3.3.2 (equations (3.12 - 3.17)). The PCA-vector $\mathbf{x}_{PCA}(i)$ consists of the t weights for each eigenvector:

$$\begin{aligned} \mathbf{x}_{\mathrm{PCA}}(i) &= \mathbf{P}^T \left(\mathbf{x}_{original}(i) - \bar{\bar{\mathbf{x}}}\right) \\ &= \left[b_1^i, b_2^i, \ldots, b_t^i\right]^T \end{aligned} \tag{4.6}$$

In this equation, $\mathbf{P}$ is the matrix with the t eigenvectors of the covariance matrix of the of the training set (see equation (3.16)). The vector $\mathbf{x}_{original}(i)$ is the original measurement vector (e.g $\mathbf{x}_p(i)$ or $\mathbf{x}_{pva}(i)$) at time i and $\bar{\bar{\mathbf{x}}}$ is the mean feature vector. The measurement function has to transform the PCA-vector back to $\mathbf{z}$ and thus becomes:

$$\mathbf{h}_{PCA} = \mathbf{H}\mathbf{x}_{PCA} + \bar{\bar{\mathbf{x}}} \tag{4.7}$$

The measurement matrix $\mathbf{H}$ is equal to $\mathbf{P}$ in case of only position elements and is equal to $[\mathbf{I}, \mathbf{0}, \mathbf{0}]\mathbf{P}$ in case of position, velocity and acceleration elements.

4.4 System matrices and parameters

The matrices of the state space model (4.1) (block 3 in figure 4.1) that have to be identified are the system matrix $\mathbf{F}$, the input matrix $\mathbf{L}$, the covariance matrix $\mathbf{C}_w$ of the process noise and the covariance matrix $\mathbf{C}_v$ of the measurement noise. The latter depends on the measurement setup, see appendix A. For the identification of the other matrices it is necessary to make assumptions, since it is not possible to identify all these matrices based on measured output variables only. Two main options can be considered. The first one involves the full estimation of the system matrix $\mathbf{F}$ and the covariance matrix $\mathbf{C}_w$ from trajectories of state vectors. For this, it will be assumed that the states can directly be derived from the measurements and that the muscle activation signals can be considered as unknown process noise with zero-mean white Gaussian distribution. However, the latter is of course an unrealistic assumption (since muscle activation signals are probably correlated). Therefore, in the second option this

whiteness assumption will be relaxed. Instead, these input signals will be modeled explicitly and estimated simultaneously with the estimation of the states and system parameters. However, for this option it is necessary that the input matrix, the measurement matrix and the main structure of the system matrix are known or measured. Therefore, in the next subsection these matrices will be derived by making assumptions about the kinematic and dynamic properties of the system. The actual estimation possibilities are discussed in chapter 5.

4.4.1 Kinematic-dynamic assumptions

It will be assumed that a tongue or lip landmark can be modeled as a mass point that experiences muscle forces and behaves as a mass-spring-damper system, see figure 4.2. Without applied forces, the mass point is in equilibrium position (e.g. corresponding to a closed mouth). An applied muscle force F_{ext} causes an acceleration and thus a displacement of the mass point. However, the mass point also experiences an opposite force F_k, in the direction of its equilibrium position. This force increases as the distance from the equilibrium position increases. Furthermore, the mass point also experiences a kind of friction or damping force F_d, which limits its velocity.

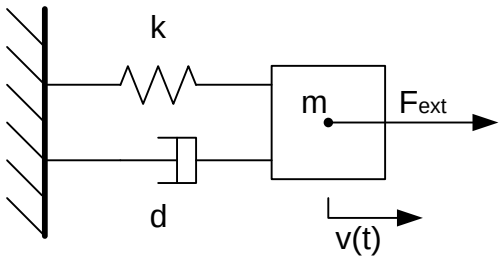


Figure 4.2: *Mass-spring-damper system.*

A mass-spring-damper system is a second order kinematic system. Such a system can be described with kinematic equations and dynamic laws. Kinematic equations (e.g. $v = \delta x/\delta t$ and $a = \delta v/\delta t$) describe the motion of a mass point without exact consideration of the causes leading to the motion. Dynamic laws describe the relationship between forces acting on a mass and the motion of that mass. These dynamic laws involve Newton's second law ($F_m = ma$), Hooke's law ($F_k = kx$) and the damping law ($F_d = dv$). The variables x, v and a represent respectively the position (or displacement), velocity and acceleration of the landmark with mass m. The elasticity constant k and the friction constant d are considered to be the system parameters.

System matrices

The system matrices can be derived from the force balance equation:

$$\begin{aligned} \Sigma F &= F_{\text{ext}} - F_k - F_d \\ ma(i) &= F_{\text{ext}}(i) - kx(i) - dv(i) \end{aligned} \tag{4.8}$$

The acceleration $a(i)$ can be approximated by $(v(i+i) - v(i))/\Delta t$, where Δt is the sampling interval. Substitution of the kinematic equations and dynamic laws in equation (4.8) results in the following difference equations for the position and velocity of the mass:

$$x(i+1) = x(i) + \Delta t\, v(i) \tag{4.9a}$$

$$v(i+1) = v(i) - \frac{d}{m}\Delta t\, v(i) - \frac{k}{m}\Delta t\, x(i) + \frac{F_{\text{ext}}(i)}{m}\Delta t \tag{4.9b}$$

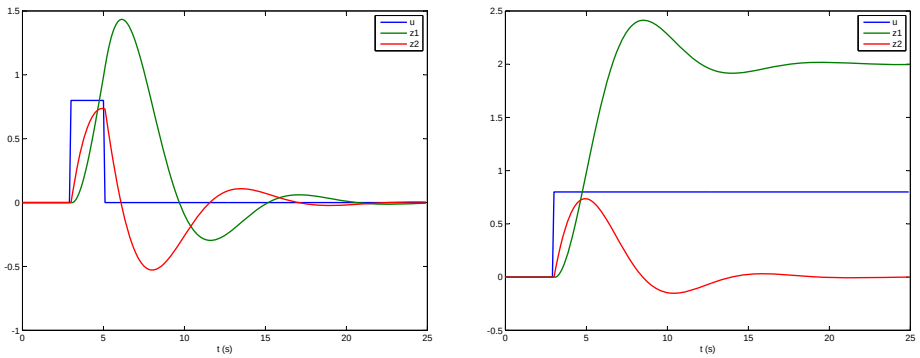
These two equations can be written in matrix form as follows:

$$\begin{bmatrix} x(i+1) \\ v(i+1) \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ -\frac{k}{m}\Delta t & 1 - \frac{d}{m}\Delta t \end{bmatrix} \begin{bmatrix} x(i) \\ v(i) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\Delta t}{m} \end{bmatrix} F_{\text{ext}}(i) \tag{4.10}$$

The system matrix $\mathbf{F}$ and the input matrix $\mathbf{L}$ follow by comparing this matrix equation with the state space model (5.5). Assuming that both the displacement and the velocity can be measured, the measurement matrix $\mathbf{H}$ becomes an identity matrix. The muscle force $F_{\text{ext}}(i)$ represents the input.

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta t \\ -\frac{k}{m}\Delta t & 1 - \frac{d}{m}\Delta t \end{bmatrix}; \quad \mathbf{L} = \begin{bmatrix} 0 \\ \frac{\Delta t}{m} \end{bmatrix}; \quad \mathbf{H} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.11}$$

Note that in case of multi-dimensional state vectors, the matrix elements have to be multiplied with the identity matrix. A one-dimensional test model (just one mass point) is implemented in MATLAB. Figure 4.3 shows the simulation results for two different input forces. These results are as expected; they comply with the kinematic equations and dynamic laws.



(a) *Short input force* **(b)** *Long input force*

Figure 4.3: *Simulations of the kinematic model (4.10): input force [N] (blue), velocity [m/s] (red) and displacement [m] (green) as a function of time [s]. (Parameter values: $m = 1$ kg, $k = 0.4$ N/m, $d = 0.6$ Ns/m, $\Delta t = 0.1$ s.)*

Process and measurement noise

It will be assumed that the process noise (now separated from input signals) and the measurement noise can be modeled as zero-mean, Gaussian white noise

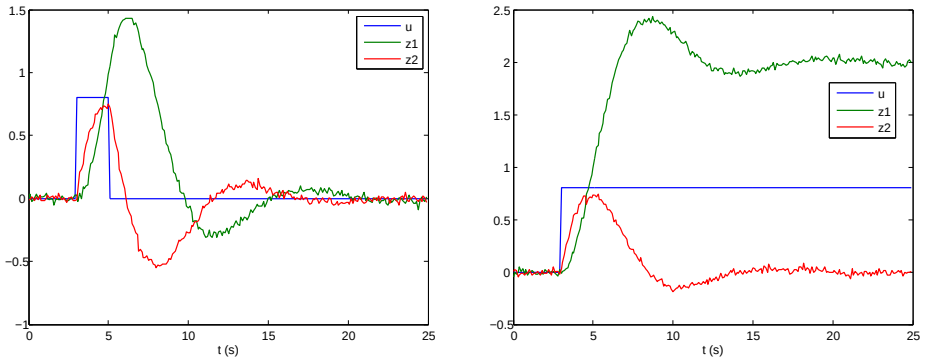
sequences, with covariance matrices $\mathbf{C}_w$ and $\mathbf{C}_v$. Suppose that in the actual system (continuous-time) the velocity undergoes process noise with a spectral density σ_w^2. In [23] (page 262-263) it is derived that in that case the covariance matrix of the discrete-time process noise is:

$$\mathbf{C}_w = E\left[\mathbf{w}(i)\mathbf{w}^T(i)\right] = \begin{bmatrix} \frac{1}{3}\Delta t^3 & \frac{1}{2}\Delta t^2 \\ \frac{1}{2}\Delta t^2 & \Delta t \end{bmatrix} \sigma_w^2 \tag{4.12}$$

Suppose that measurements can be taken with an accuracy of σ_v. In case both the position and the velocity can be measured, the covariance matrix of the measurement noise is:

$$\mathbf{C}_v = E\left[\mathbf{v}(i)\mathbf{v}^T(i)\right] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \sigma_v^2 \tag{4.13}$$

Figure 4.4 shows simulations of the model whereby the states are contaminated with process noise and the measurements with measurement noise.



(a) *Short input force* **(b)** *Long input force*

Figure 4.4: *Simulations of the kinematic model including process and measurement noise ($\sigma_w = 0.02$, $\sigma_v = 0.02$).*

4.5 Model evaluation techniques

Since it is assumed that the physical behavior of the tongue or the lips can be modeled as a linear state space model with Gaussian white noise processes, the model (4.1) can be tested and evaluated by using a Kalman filter. Furthermore, techniques will be considered which can be used to check whether the model behaves consistently. This mainly implies checking whether the model parameters contain as much information as provided by the available data and to determine if the Gaussian assumptions are actually justified.

4.5.1 Kalman filtering

A Kalman filter [22] is a recursive filter that estimates the states of a linear dynamic system from a series of (noisy) measurements. Each iteration cycles through a number of equations, which can be subdivided into an update part (4.14a) and a prediction part (4.14b).

Update:

$$\begin{aligned}
\hat{\mathbf{z}}(i) &= \mathbf{H}\hat{\mathbf{x}}(i|i-1) && \text{(predicted measurement)}\\
\tilde{\mathbf{z}}(i) &= \mathbf{z}(i) - \hat{\mathbf{z}}(i) && \text{(innovations)}\\
\mathbf{S}(i) &= \mathbf{H}\mathbf{C}(i|i-1)\mathbf{H}^T + \mathbf{C}_v && \text{(innovation matrix)}\\
\mathbf{K}(i) &= \mathbf{C}(i|i-1)\mathbf{H}^T\mathbf{S}^{-1}(i) && \text{(Kalman gain matrix)}\\
\hat{\mathbf{x}}(i|i) &= \hat{\mathbf{x}}(i|i-1) + \mathbf{K}(i)\tilde{\mathbf{z}}(i) && \text{(updated estimate)}\\
\mathbf{C}(i|i) &= \mathbf{C}(i|i-1) - \mathbf{K}(i)\mathbf{S}(i)\mathbf{K}^T(i) && \text{(error covariance matrix)}
\end{aligned} \tag{4.14a}$$

Prediction:

$$\begin{aligned}
\hat{\mathbf{x}}(i+1|i) &= \mathbf{F}(i)\left(\hat{\mathbf{x}}(i|i) - \bar{\bar{\mathbf{x}}}\right) + \bar{\bar{\mathbf{x}}} && \text{(predicted state vector)}\\
\mathbf{C}(i+1|i) &= \mathbf{F}(i)\mathbf{C}(i|i)\mathbf{F}^T(i) + \mathbf{C}_w && \text{(predicted state covariance)}
\end{aligned} \tag{4.14b}$$

In each iteration i the estimate for the current states $\hat{\mathbf{x}}(i|i)$ is computed using only the estimated states from the previous time step $\hat{\mathbf{x}}(i|i-1)$ and the current measurements $\mathbf{z}(i)$. Beside the state estimates, an error covariance matrix $\mathbf{C}(i|i)$ is calculated. The values of this matrix are measures for the uncertainty of the state estimates. Both $\hat{\mathbf{x}}(i|i)$ and $\mathbf{C}(i|i)$ are based on their previous values ($\hat{\mathbf{x}}(i|i-1)$ and $\mathbf{C}(i|i-1)$), the innovation matrix $\mathbf{S}(i)$ and the Kalman gain matrix $\mathbf{K}(i)$. The innovation matrix represents the uncertainty of the predicted measurements and is determined by the uncertainty of the previous states, expressed by $\mathbf{C}(i|i-1)$, and the current measurement noise $v(i)$, expressed by $\mathbf{C}_v$. The Kalman gain matrix $\mathbf{K}(i)$ can be seen as the feedback matrix. This matrix has large values when the measurements are relative accurate.

If there is no measurement vector available, the values of the measurement covariance matrix $\mathbf{C}_v$ are infinitely large and thus the values of the Kalman gain matrix $\mathbf{K}(i)$ become zero. In that case, the next state estimate and error covariance matrix just become equal to their predictions: $\hat{\mathbf{x}}(i|i) = \hat{\mathbf{x}}(i|i-1)$; $\mathbf{C}(i|i) = \mathbf{C}(i|i-1)$. For evaluating the prediction performance of the system matrix $\mathbf{F}$, vectors can be eliminated from the measurement sequences. In this way it can be seen how the states evaluate without measurement input.

Summarized, the Kalman filter can be used for three things: filtering of noise source (both process noise and measurement noise), connecting the measurement data with the state variables (even the ones that cannot directly be measured) and prediction of next state values. Measurement errors can, in case of measuring landmark positions on the tongue or lips, be seen as small landmark detection errors. Suppose that all landmarks can be detected with an inaccuracy of σ_v (e.g. two pixels), than the error covariance matrix becomes: $\mathbf{C}_v = \sigma_v\mathbf{I}$. The state estimates can be seen as a kind of trade-off between prediction and innovations. In case of large (diagonal) values in $\mathbf{C}_w$, emphasis is put on the innovations. In case of small values, emphasis is put the predicted states. The Kalman filter can in this way also be used for estimation of velocity and acceleration components from position measurements.

4.5.2 Consistency checks

The purpose of consistency checks is to determine whether the model on which the Kalman filter is built is accurate enough. Therefore three different types of estimation error variances have to be considered. These are the minimal variances, $V_{\min}$, that would be obtained with the most appropriate model, the actual variances, V_{act}, in case of using a certain model and the estimated variances, V_{kf}, from the covariance matrix of the Kalman filter. In case of a correct model, these variances are equal: $V_{\min} = V_{\text{act}} = V_{\text{kf}}$. The error signals that can be considered are the innovation sequences and the state estimation error signals. However, for calculating the state estimation error signals, the exact state values have to be available. Since in the situation of measuring marker locations on the lips or tongue, this data won't be available, only the innovation signals will be considered. The innovation sequence $\tilde{\mathbf{z}}(i)$ is the predicted measurement sequence subtracted from the measurement sequence:

$$\tilde{\mathbf{z}}(i) = \mathbf{z}(i) - \hat{\mathbf{z}}(i) \tag{4.15}$$

In case of a correct model, these innovations should be white and normally distributed with zero mean and covariance similar to the covariance matrix $\mathbf{S}$, i.e.: $\tilde{\mathbf{z}}(i) \sim N(\mathbf{0}, \mathbf{S})$. To check if the distribution property is satisfied, the so-called NIS (Normalized Innovations Squared) can be investigated:

$$\text{NIS}(i) = \tilde{\mathbf{z}}^T(i)\mathbf{S}^{-1}(i)\tilde{\mathbf{z}}(i) \tag{4.16}$$

The NIS should comply with a χ^2_p distribution (with p the number of degrees of freedom, in this case the number of vector elements). This follows from the following proof:

Proof 4.1

A χ^2_N distributed signal can be constructed from the sum of the square of random and independent variables with zero mean and unit variance. So:

$$\begin{aligned} \tilde{\boldsymbol{z}}(i) &\sim N(\boldsymbol{0}, \boldsymbol{S}) \\ \boldsymbol{S}^{-\frac{1}{2}}\tilde{\boldsymbol{z}}(i) &\sim N(\boldsymbol{0}, \boldsymbol{I}) \\ \sum_{i=1}^{N} \left[\boldsymbol{S}^{-\frac{1}{2}}\tilde{\boldsymbol{z}}(i)\right]^2 &\sim \chi^2_N \end{aligned} \tag{4.17}$$

And since

$$\sum_{i=1}^{N} \left[\boldsymbol{S}^{-\frac{1}{2}}\tilde{\boldsymbol{z}}(i)\right]^2 = \left[\boldsymbol{S}^{-\frac{1}{2}}\tilde{\boldsymbol{z}}(i)\right]^T \left[\boldsymbol{S}^{-\frac{1}{2}}\tilde{\boldsymbol{z}}(i)\right] = \tilde{\boldsymbol{z}}^T(i)\boldsymbol{S}^{-1}\tilde{\boldsymbol{z}}(i) \tag{4.18}$$

$\tilde{\boldsymbol{z}}^T(i)\boldsymbol{S}^{-1}\tilde{\boldsymbol{z}}(i) \sim \chi^2_N$ when $\tilde{\boldsymbol{z}}(i) \sim N(\boldsymbol{0}, \boldsymbol{S})$.

To check the whiteness property of the innovations, the periodogram of the sequence can be considered. The periodogram is the normalized frequency power spectrum of the innovation sequence:

$$P_n(k) = \frac{|\tilde{Z}_n(k)|^2}{I} \tag{4.19}$$

where $\tilde{Z}_n(k)$ is the discrete Fourier transform of the nth innovation sequence:

$$\tilde{Z}_n(k) = \sum_{i=0}^{I-1} \tilde{z}_n(i) e^{-j2\pi ki/I} \tag{4.20}$$

The whiteness property implies that the power spectrum of any element of $\tilde{\mathbf{z}}(i)$ must be flat. In appendix C it is proven that the flatness of $P_n(k)$, and thus the whiteness of $\tilde{z}_n(i)$, can be tested by checking whether the sequence $2P_n(k)/\sigma_n^2$ is χ_2^2 distributed.

In order to check whether the NIS or the $2P_n(k)/\sigma_n^2$ sequence obey the desired distribution, a distribution test needs to be applied. A quick and commonly used test for chi-square distributions is *the one-sided 95% acceptance boundary test* [22]. For this test a boundary is defined, under which 95% of the distribution must be. The corresponding value also depends on the number of degrees of freedom and can be calculated with chi-square cumulative distribution function. If the number of samples below this boundary is about 95% of the full number of samples, the model predictions are considered to make sense.

5

System identification with unknown inputs

5.1 Introduction

This chapter is concerned with the system identification of the linear state space model, defined in chapter 4. Assuming this type of model, the objective is to already derive some possibilities for the identification of the blocks 3 and 4 in figure 4.1 and the estimation of states, input variables and system parameters. Once again, it is emphasized that the actual muscle activation signals are not yet available. The identification and estimation are thus only based on measured output variables and on the in chapter 4 motivated assumptions. The two possibilties to be considered are:

- Estimation of entire system matrix and covariance of process noise from state sequences. *Assumption:* Muscle activation signals can be considered as unknown process noise with zero-mean white Gaussian distribution.
- Estimation of states, input and system matrix parameters from measurement sequences. *Assumptions:* Input matrix, measurement matrix, main structure of system matrix and the noise distributions are known.

Section 5.2 considers the first possibility. However, it will be shown that the required assumption regarding the input is indeed unrealistic. The next two sections of this chapter are about the second possibility. These sections are mainly focused on the derivation of the estimation algorithms. In section 5.3 two algorithms will be derived. The first algorithm performs recursive state and input estimation and assumes unconstrained input. The second algorithm is based on closed-form estimation, which makes it possible to constrain the input. In section 5.4 the estimation problem problem will be extended to parameter estimation, by deriving a cost function that also includes the system parameters. The performance of the algorithms is tested on the kinematic-dynamic model defined in section 4.4.1.

5.2 Estimation of system matrix

Assuming that the measurement noise is small and that the states can easily be derived from the measurements, it is possible to estimate the system matrix of the linear state space model from the sequence of state vectors. This can be done by calculating the covariance matrices $E\left[\boldsymbol{\varepsilon}(i+1)\boldsymbol{\varepsilon}^T(i)\right]$ and $E\left[\boldsymbol{\varepsilon}(i)\boldsymbol{\varepsilon}^T(i)\right]$, where $\boldsymbol{\varepsilon}(i)$ is the deviation from the mean state vector: $\boldsymbol{\varepsilon}(i) = \mathbf{x}(i) - \bar{\mathbf{x}}$. The derivation of the system matrix on this way is as follows:

$$\begin{aligned} \boldsymbol{\varepsilon}(i+1)\boldsymbol{\varepsilon}^T(i) &= \mathbf{F}\boldsymbol{\varepsilon}(i)\boldsymbol{\varepsilon}^T(i) + \mathbf{w}(i)\boldsymbol{\varepsilon}^T(i) \\ E\left[\boldsymbol{\varepsilon}(i+1)\boldsymbol{\varepsilon}^T(i)\right] &= E\left[\mathbf{F}\boldsymbol{\varepsilon}(i)\boldsymbol{\varepsilon}^T(i)\right] + E\left[\mathbf{w}(i)\boldsymbol{\varepsilon}^T(i)\right] \\ E\left[\boldsymbol{\varepsilon}(i+1)\boldsymbol{\varepsilon}^T(i)\right] &= \mathbf{F}E\left[\boldsymbol{\varepsilon}(i)\boldsymbol{\varepsilon}^T(i)\right] \end{aligned} \tag{5.1}$$

Resulting in the following equation for the estimation of the system matrix:

$$\mathbf{F} = E\left[\boldsymbol{\varepsilon}(i+1)\boldsymbol{\varepsilon}^T(i)\right] E\left[\boldsymbol{\varepsilon}(i)\boldsymbol{\varepsilon}^T(i)\right]^{-1} \tag{5.2}$$

As can be seen in the derivation (5.1), the process noise (including the unknown input $\mathbf{Lu}(i)$) is crossed out. This can be done because of the assumption that the process noise has zero mean, which means that the expectation $E\left[\mathbf{w}(i)\boldsymbol{\varepsilon}^T(i)\right]$ is zero as well.

An estimate of the covariance matrix of the process noise can now be obtained from the sequence of residuals $\mathbf{P}_w$.

$$\mathbf{P}_w = [\boldsymbol{\varepsilon}(2), \boldsymbol{\varepsilon}(3), \ldots, \boldsymbol{\varepsilon}(I)] - \mathbf{F} \times [\boldsymbol{\varepsilon}(1), \boldsymbol{\varepsilon}(2), \ldots, \boldsymbol{\varepsilon}(I-1)] \tag{5.3}$$

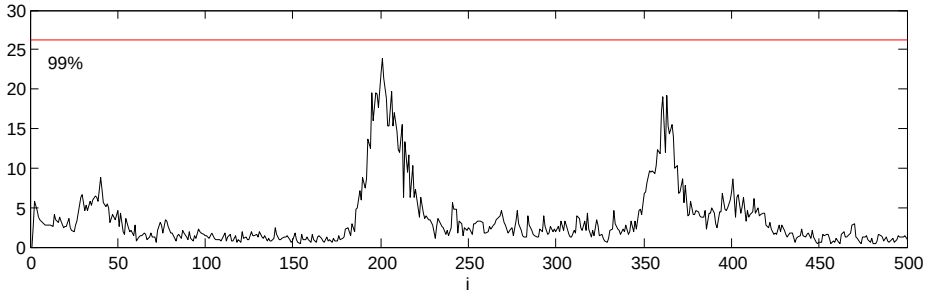
So, the covariance matrix of the process noise can be estimated as follows:

$$\mathbf{C}_w = \frac{\mathbf{P}_w \mathbf{P}_w^T}{I-1} \tag{5.4}$$

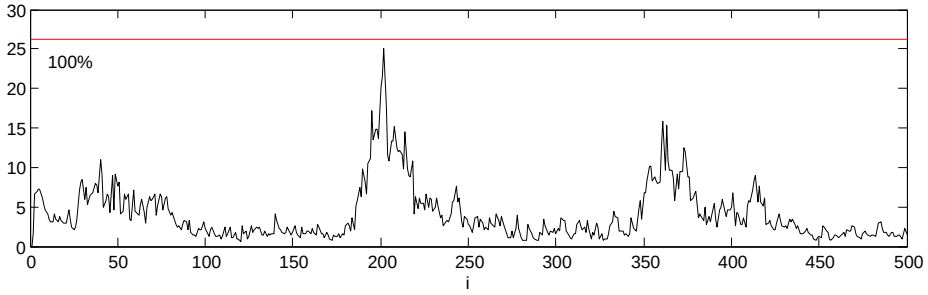
In these equations, I is the number of measurement samples.

Consistency checks

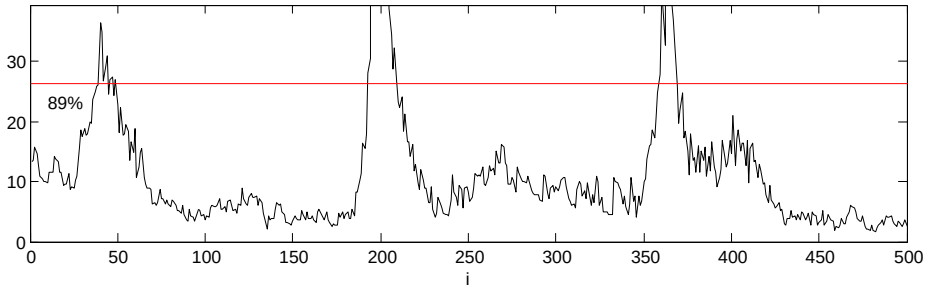
The system matrix and covariance matrix of the process noise are estimated from state vectors derived from measured landmark trajectories on the lips, see appendix A. The considered state vectors are: position only ($n = 16$), position + velocity ($n = 32$), position - PCA-reduced ($n = 8$) and position + velocity - PCA-reduced ($n = 12$). The NIS and innovation sequences are calculated for the resulting models. Figure 5.1 shows the NIS of the four different model configurations. In all four cases, only a small number of samples is near the acceptance boundary. This means that the NIS sequences are not χ^2_{16} distributed. Figure 5.2 and 5.3 show respectively the innovations sequence $\tilde{z}_1(i)$ and the corresponding periodogram. As can be seen, the sequence $2P_n(k)/\sigma_n^2$ does not obey the χ^2_2 distribution and thus the power spectrum is not flat. The periodograms of the other innovation sequences look similar. From these consistency checks it can be concluded that the innovations are not white. This means that the state estimator is not optimal or that the data is not Gaussian distributed. Therefore it can be concluded that the assumption about the input distribution (white Gaussian zero-mean) is indeed invalid.



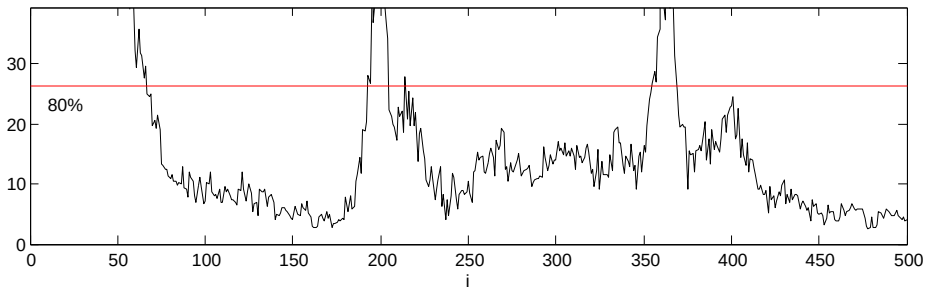
(a) *State vector: positions only*



(b) *State vector: positions and velocity*



(c) *State vector: PCA-reduced positions*



(d) *State vector: PCA-reduced positions and velocity*

Figure 5.1: *NIS of four different model configurations, applied on the sequence of random lip movements. The red lines indicate the one-sided 95% acceptance boundary of the χ^2_n cumulative distribution, which is in case of $n = 16$ about 26.*

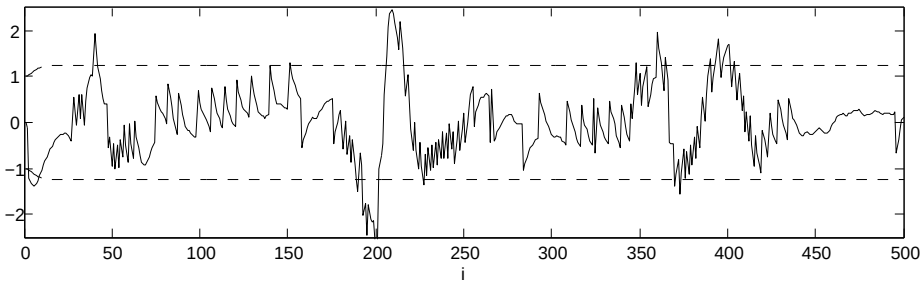


Figure 5.2: *Innovation sequence $\tilde{z}_1(i)$ and corresponding variance σ_1^2 (striped lines) of the model based on position and velocity features.*

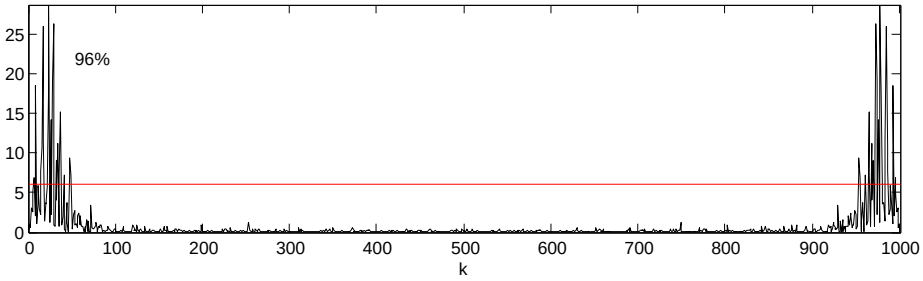


Figure 5.3: *Periodogram $2P_1(k)/\sigma_1^2$ of the innovation sequence $\tilde{z}_1(i)$. The red line indicates the χ_2^2 cumulative distribution level.*

5.3 State and input estimation

This section is focused on the derivation of mathematical algorithms for state and input estimation from a measurement sequence. Two different possibilities are considered: recursive state and input estimation and closed-form state and input estimation. The recursive algorithm is more appropriate for a system with vectors and matrices of high dimensions and a large number of measurement samples. The closed-form algorithm stacks the vectors, which enables input constrainment. This can be done by including possible a priori information concerning the correlation between input signals.

The discrete-time linear state space model, on which the derivations are based, is repeated here (see section 4.2 for the definitions of the involved matrices and vector):

$$\begin{aligned} \mathbf{x}(i+1) &= \mathbf{F}\mathbf{x}(i) + \mathbf{L}\mathbf{u}(i) + \mathbf{w}(i) \\ \mathbf{z}(i) &= \mathbf{H}\mathbf{x}(i) + \mathbf{v}(i) \end{aligned} \tag{5.5}$$

Once again, it is assumed that the model matrices are known and that the process and measurement noise are mutually uncorrelated, zero-mean, white random signals with known covariance matrices ($\mathbf{C}_w$ and $\mathbf{C}_v$). To make the derivations more orderly, it will be assumed that the state vectors are already normalized to zero, i.e. $\bar{\mathbf{x}} = \mathbf{0}$. For convenience, in this section the discrete-time indices of vectors are given in subscript, instead of between brackets.

5.3.1 Recursive

Objective formulation

Objective is to obtain an optimal estimate - in terms of minimum variance - of the state vector $\mathbf{x}_{i+1} \in \Re^n$ and the input vector $\mathbf{u}_i \in \Re^m$ given the measurement $\mathbf{z}_{i+1} \in \Re^p$ and the estimate of the current state vector $\hat{\mathbf{x}}_i$. Solving this problem comes down to maximizing the probability density function $p(\mathbf{x}_{i+1}, \mathbf{u}_i | \mathbf{z}_{i+1}, \hat{\mathbf{x}}_i)$:

$$(\hat{\mathbf{x}}_{i+1}, \hat{\mathbf{u}}_i) = \underset{\mathbf{x}_{i+1}, \mathbf{u}_i}{\operatorname{argmax}} \{p(\mathbf{x}_{i+1}, \mathbf{u}_i | \mathbf{z}_{i+1}, \hat{\mathbf{x}}_i)\} \tag{5.6}$$

Solution derivation

By using Bayes' theorem[1], equation (5.6) can be written into probability density functions with known distributions:

$$\begin{aligned} p(\mathbf{x}_{i+1}, \mathbf{u}_i | \mathbf{z}_{i+1}, \hat{\mathbf{x}}_i) &= p(\mathbf{x}_{i+1} | \mathbf{u}_i, \mathbf{z}_{i+1}, \hat{\mathbf{x}}_i) p(\mathbf{u}_i | \mathbf{z}_{i+1}, \hat{\mathbf{x}}_i) \\ &= \frac{p(\mathbf{x}_{i+1} | \mathbf{u}_i, \hat{\mathbf{x}}_i) p(\mathbf{z}_{i+1} | \mathbf{x}_{i+1})}{p(\mathbf{z}_{i+1} | \mathbf{u}_i, \hat{\mathbf{x}}_i)} \frac{p(\mathbf{u}_i | \hat{\mathbf{x}}_i) p(\mathbf{z}_{i+1} | \mathbf{u}_i, \hat{\mathbf{x}}_i)}{p(\mathbf{z}_{i+1} | \hat{\mathbf{x}}_i)} \\ &= \frac{p(\mathbf{x}_{i+1} | \mathbf{u}_i, \hat{\mathbf{x}}_i) p(\mathbf{z}_{i+1} | \mathbf{x}_{i+1}) p(\mathbf{u}_i | \hat{\mathbf{x}}_i)}{p(\mathbf{z}_{i+1} | \hat{\mathbf{x}}_i)} \end{aligned} \tag{5.7}$$

The function $p(\mathbf{z}_{i+1} | \hat{\mathbf{x}}_i)$ is independent of the arguments $\mathbf{x}_{i+1}$ and $\mathbf{u}_i$. Furthermore, $\mathbf{u}_i$ and $\hat{\mathbf{x}}_i$ are uncorrelated, so $p(\mathbf{u}_i | \hat{\mathbf{x}}_i)$ can also be disregarded. Therefore, the following holds:

$$p(\mathbf{x}_{i+1}, \mathbf{u}_i | \mathbf{z}_{i+1}, \hat{\mathbf{x}}_i) \propto p(\mathbf{x}_{i+1} | \mathbf{u}_i, \hat{\mathbf{x}}_i) p(\mathbf{z}_{i+1} | \mathbf{x}_{i+1}) \tag{5.8}$$

The remaining two probability density functions are known. Both $\mathrm{p}(\mathbf{x}_{i+1} | \mathbf{u}_i, \hat{\mathbf{x}}_i)$ and $\mathrm{p}(\mathbf{z}_{i+1} | \mathbf{x}_{i+1})$ are multivariate normally distributed. Their mean vector and covariance matrix can be derived from the state space model (5.5).

$$p(\mathbf{x}_{i+1} | \mathbf{u}_i, \hat{\mathbf{x}}_i) \sim N\left(\mathbf{F}\hat{\mathbf{x}}_i + \mathbf{L}\mathbf{u}_i, \mathbf{F}\mathbf{C}_{i|i}\mathbf{F}^T + \mathbf{C}_w\right) \tag{5.9}$$

$$p(\mathbf{z}_{i+1} | \mathbf{x}_{i+1}) \sim N\left(\mathbf{H}\mathbf{x}_{i+1}, \mathbf{C}_v\right) \tag{5.10}$$

In general, the probability density function of a multivariate normally distributed signal $\mathbf{x} \in \Re^N$, with mean vector μ and covariance matrix Σ, is as follows:

$$f_{\mathbf{x}} = \frac{1}{(2\pi)^{N/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)\right) \tag{5.11}$$

Since the term in front of the exponential does not include the input argument vector, the maximization problem of equation (5.6) can be written as:

$$\begin{aligned} &(\hat{\mathbf{x}}_{i+1}, \hat{\mathbf{u}}_i) = \underset{\mathbf{x}_{i+1}, \mathbf{u}_i}{\operatorname{argmax}} \\ &\exp\left(-\frac{1}{2}(\mathbf{x}_{i+1} - \mathbf{F}\hat{\mathbf{x}}_i - \mathbf{L}\mathbf{u}_i)^T \mathbf{P}_{1,i} (\mathbf{x}_{i+1} - \mathbf{F}\hat{\mathbf{x}}_i - \mathbf{L}\mathbf{u}_i)\right) \\ &\times \exp\left(-\frac{1}{2}(\mathbf{z}_{i+1} - \mathbf{H}\mathbf{x}_{i+1})^T \mathbf{P}_2 (\mathbf{z}_{i+1} - \mathbf{H}\mathbf{x}_{i+1})\right) \end{aligned} \tag{5.12}$$

[1] Bayes' theorem: $p(A|B)p(B) = p(B|A)p(A)$

where

$$\mathbf{P}_{1,i} = \left(\mathbf{F}\mathbf{C}_{i|i}\mathbf{F}^T + \mathbf{C}_w\right)^{-1} \tag{5.13}$$

and

$$\mathbf{P}_2 = \mathbf{C}_v^{-1} \tag{5.14}$$

By applying the *ln*-function (natural logarithm), equation (5.12) can be further simplified and comes now down to a minimization problem:

$$\begin{aligned}&(\hat{\mathbf{x}}_{i+1}, \hat{\mathbf{u}}_i) = \underset{\mathbf{x}_{i+1},\mathbf{u}_i}{\operatorname{argmin}} \\ &\frac{1}{2}\left(\begin{bmatrix}\mathbf{I}_n & -\mathbf{L}\end{bmatrix}\begin{bmatrix}\mathbf{x}_{i+1}\\ \mathbf{u}_i\end{bmatrix} - \mathbf{F}\hat{\mathbf{x}}_i\right)^T \mathbf{P}_{1,i}\left(\begin{bmatrix}\mathbf{I}_n & -\mathbf{L}\end{bmatrix}\begin{bmatrix}\mathbf{x}_{i+1}\\ \mathbf{u}_i\end{bmatrix} - \mathbf{F}\hat{\mathbf{x}}_i\right) + \\ &\frac{1}{2}\left(\begin{bmatrix}-\mathbf{H} & \mathbf{0}_{p\times m}\end{bmatrix}\begin{bmatrix}\mathbf{x}_{i+1}\\ \mathbf{u}_i\end{bmatrix} + \mathbf{z}_{i+1}\right)^T \mathbf{P}_2\left(\begin{bmatrix}-\mathbf{H} & \mathbf{0}_{p\times m}\end{bmatrix}\begin{bmatrix}\mathbf{x}_{i+1}\\ \mathbf{u}_i\end{bmatrix} + \mathbf{z}_{i+1}\right)\end{aligned} \tag{5.15}$$

Differentiation with respect to the vector $\begin{bmatrix}\mathbf{x}_{i+1}\\ \mathbf{u}_i\end{bmatrix}$ and equating to zero results in the following two equations:

$$\begin{bmatrix}\mathbf{I}_n & -\mathbf{L}\end{bmatrix}^T \mathbf{P}_{1,i}\left(\begin{bmatrix}\mathbf{I}_n & -\mathbf{L}\end{bmatrix}\begin{bmatrix}\mathbf{x}_{i+1}\\ \mathbf{u}_i\end{bmatrix} - \mathbf{F}\hat{\mathbf{x}}_i\right) = 0 \tag{5.16a}$$

$$\begin{bmatrix}-\mathbf{H} & \mathbf{0}_{p\times m}\end{bmatrix}^T \mathbf{P}_2\left(\begin{bmatrix}-\mathbf{H} & \mathbf{0}_{p\times m}\end{bmatrix}\begin{bmatrix}\mathbf{x}_{i+1}\\ \mathbf{u}_i\end{bmatrix} + \mathbf{z}_{i+1}\right) = 0 \tag{5.16b}$$

The equations (5.16) can be written in the form:

$$\mathbf{A}_i\begin{bmatrix}\mathbf{x}_{i+1}\\ \mathbf{u}_i\end{bmatrix} = \mathbf{B}_i\begin{bmatrix}\hat{\mathbf{x}}_i\\ \mathbf{z}_{i+1}\end{bmatrix} \tag{5.17}$$

where

$$\mathbf{A}_i = \begin{bmatrix}\begin{bmatrix}\mathbf{I}_n & -\mathbf{L}\end{bmatrix}^T \mathbf{P}_{1,i}\begin{bmatrix}\mathbf{I}_n & -\mathbf{L}\end{bmatrix} \\ \begin{bmatrix}-\mathbf{H} & \mathbf{0}_{p\times m}\end{bmatrix}^T \mathbf{P}_2\begin{bmatrix}-\mathbf{H} & \mathbf{0}_{p\times m}\end{bmatrix}\end{bmatrix} \tag{5.18}$$

$$\mathbf{B}_i = \begin{bmatrix}\begin{bmatrix}\mathbf{I}_n & -\mathbf{L}\end{bmatrix}^T \mathbf{P}_{1,i}\mathbf{F} & \mathbf{0}_{(n+m)\times p} \\ \mathbf{0}_{(n+m)\times n} & \begin{bmatrix}\mathbf{H} & \mathbf{0}_{p\times m}\end{bmatrix}^T \mathbf{P}_2\end{bmatrix} \tag{5.19}$$

Since the matrix $\mathbf{A}_i$ is not square (and thus not invertible), there is no unique solution for $\mathbf{x}_{i+1}$ and $\mathbf{u}_i$. A least squares ('best fit') solution can be calculated by using the pseudo inverse of $\mathbf{A}_i$: $\mathbf{A}_i^+ = (\mathbf{A}_i^T\mathbf{A}_i)^{-1}\mathbf{A}_i^T$. So, the solution of equation (5.6) and thus the optimal estimate of $\mathbf{x}_{i+1}$ and $\mathbf{u}_i$ is given by equation (5.20). The corresponding minimum covariance matrices by are given by equation (5.21).

$$\begin{bmatrix}\hat{\mathbf{x}}_{i+1}\\ \hat{\mathbf{u}}_i\end{bmatrix} = \mathbf{A}_i^+\mathbf{B}_i\begin{bmatrix}\hat{\mathbf{x}}_i\\ \mathbf{z}_{i+1}\end{bmatrix} \tag{5.20}$$

$$\begin{bmatrix}\mathbf{C}_{i+1|i} & \mathbf{0}_{n\times p}\\ \mathbf{0}_{p\times n} & \mathbf{U}_i\end{bmatrix} = \mathbf{A}_i^+\mathbf{B}_i\begin{bmatrix}\mathbf{C}_{i|i} & \mathbf{0}_{n\times p}\\ \mathbf{0}_{p\times n} & \mathbf{C}_v\end{bmatrix}\mathbf{B}_i^T(\mathbf{A}_i^+)^T \tag{5.21}$$

Algorithm summary

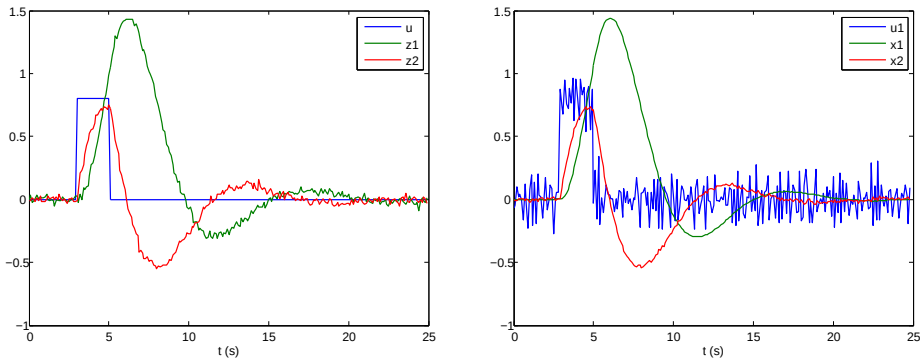
Summarized, each iteration of the derived algorithm for recursive state and input estimation consists of the following steps:

1. Calculate the inverse matrices $\mathbf{P}_{1,i}$ (5.13) and $\mathbf{P}_2$ (5.14).
2. Calculate the matrices $\mathbf{A}_i$ (5.18) and $\mathbf{B}_i$ (5.19).
3. Estimate the states $\hat{\mathbf{x}}_{i+1}$ and input $\hat{\mathbf{u}}_i$ (5.20).
4. Update the covariance matrices $\mathbf{C}_{i+1|i}$ and $\mathbf{U}_i$ (5.21).

The initial state $\mathbf{x}_0$ and initial state error covariance $\mathbf{C}_0$ are input arguments.

Simulation results

The algorithm is implemented in MATLAB and tested on the kinematic-dynamic model defined in section 4.4.1. The simulation results are shown in figure 5.4. As can be seen, the process noise on the first state is filtered. All the noise is shifted to (explained by) the input, since the input is unconstrained.



(a) *Measurements and actual input* **(b)** *Estimated states and estimated input*

Figure 5.4: *Simulation results of recursive state and input estimation from position and velocity measurements.*

5.3.2 Closed form

Objective formulation

The estimated input without enforced constraints is contaminated with a high amount of (white) noise, as can be seen in figure 5.4. The input value at a certain discrete-time moment can be totally different from neighbor values, i.e. there is no correlation between input signals at different time moments. To improve the estimation of the input and the states, two assumptions regarding the input will be made:

- The chance on high input signals is at the beginning of a sequence larger than at the end of the sequence.
- Within a certain time-window, input signals are correlated. This means it will be assumed that input signals can not change very quickly, i.e. a kind of bandwidth limitation.

To accomplish the enforcement of these constraints, the recursive state space model of equation 5.5 has to be translated to a model in closed-form by stacking the vector variables:

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_0 \\ \vdots \\ \mathbf{u}_{I-1} \end{bmatrix}; \quad \mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_I \end{bmatrix}; \quad \mathbf{Z} = \begin{bmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_I \end{bmatrix} \tag{5.22}$$

The closed-form model (non-iterative) is described by the following two equations:

$$\begin{aligned} \mathbf{X} &= \mathbf{G}\mathbf{U} + \mathbf{F}_s\mathbf{x}_0 + \mathbf{W} \\ \mathbf{Z} &= \mathbf{H}_s\mathbf{X} + \mathbf{V} \end{aligned} \tag{5.23}$$

The vectors $\mathbf{W}$ and $\mathbf{V}$ are respectively the stacked process and measurement noise vectors. $\mathbf{G}$ and $\mathbf{H}_s$ are respectively the input matrix and measurement matrix for the stacked case. Furthermore, $\mathbf{F}_s$ is a matrix that describes the influence of the initial state $\mathbf{x}_0$ on the stacked state vector.

Now a covariance matrix $\mathbf{C}_U$ for the stacked input vector $\mathbf{U}$ can be constructed to include the two assumptions. The objective is then to estimate the stacked state and input vectors $(\mathbf{X}, \mathbf{U})$ given the stacked measurement vector $\mathbf{Z}$ and the initial state vector $\mathbf{x}_0$:

$$\left(\hat{\mathbf{X}}, \hat{\mathbf{U}}\right) = \underset{\mathbf{X},\mathbf{U}}{\operatorname{argmax}} \left\{p\left(\mathbf{X}, \mathbf{U}|\mathbf{Z}, \mathbf{x}_0\right)\right\} \tag{5.24}$$

Solution derivation

Again by using Bayes theorem, equation (5.24) can be written into (conditional) probability density functions with known distributions:

$$\begin{aligned} p\left(\mathbf{X}, \mathbf{U}|\mathbf{Z}, \mathbf{x}_0\right) &= \frac{p\left(\mathbf{Z}|\mathbf{X}, \mathbf{U}, \mathbf{x}_0\right) p\left(\mathbf{X}, \mathbf{U}, \mathbf{x}_0\right)}{p\left(\mathbf{Z}, \mathbf{x}_0\right)} \\ &= \frac{p\left(\mathbf{Z}|\mathbf{X}, \mathbf{U}, \mathbf{x}_0\right) p\left(\mathbf{X}|\mathbf{U}, \mathbf{x}_0\right) p\left(\mathbf{U}|\mathbf{x}_0\right) p\left(\mathbf{x}_0\right)}{p\left(\mathbf{Z}, \mathbf{x}_0\right)} \\ &\propto p\left(\mathbf{Z}|\mathbf{X}\right) p\left(\mathbf{X}|\mathbf{U}, \mathbf{x}_0\right) p\left(\mathbf{U}\right) \end{aligned} \tag{5.25}$$

Note that the probability density functions $p\left(\mathbf{Z}, \mathbf{x}_0\right)$ and $p\left(\mathbf{x}_0\right)$ are disregarded because they are independent of the argument vectors $\mathbf{X}$ and $\mathbf{U}$. Furthermore, $p\left(\mathbf{Z}|\mathbf{X}, \mathbf{U}, \mathbf{x}_0\right) = p\left(\mathbf{Z}|\mathbf{X}\right)$ and $p\left(\mathbf{U}|\mathbf{x}_0\right) = p\left(\mathbf{U}\right)$. These remaining probability density functions are multivariate normally distributed as follows:

$$p\left(\mathbf{Z}|\mathbf{X}\right) \sim N\left(\mathbf{H}_s\mathbf{X}, \mathbf{C}_V\right) \tag{5.26a}$$

$$p\left(\mathbf{X}|\mathbf{U},\mathbf{x}_0\right) \sim N\left(\mathbf{G}\mathbf{U}+\mathbf{F}_s\mathbf{x}_0,\mathbf{C}_X\right) \tag{5.26b}$$

$$p\left(\mathbf{U}\right) \sim N\left(\mathbf{0},\mathbf{C}_U\right) \tag{5.26c}$$

The matrices $\mathbf{H}_s$ and $\mathbf{C}_V$ (with respectively dimensions $pI \times nI$ and $pI \times pI$) are easily constructed. They just consist of $\mathbf{H}$ and $\mathbf{C}_v$ on the diagonal:

$$\mathbf{H}_s = \begin{bmatrix} \mathbf{H} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{H} \end{bmatrix} \tag{5.27}$$

$$\mathbf{C}_V = \begin{bmatrix} \mathbf{C}_v & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{C}_v \end{bmatrix} \tag{5.28}$$

The matrices $\mathbf{G}$ and $\mathbf{F}_s$ can be deduced from the following recursions:

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{F}\mathbf{x}_0 + \mathbf{L}\mathbf{u}_0 \\ \mathbf{x}_2 &= \mathbf{F}\mathbf{x}_1 + \mathbf{L}\mathbf{u}_1 = \mathbf{L}\left(\mathbf{F}\mathbf{u}_0 + \mathbf{u}_1\right) + \mathbf{F}^2\mathbf{x}_0 \\ \mathbf{x}_3 &= \mathbf{F}\mathbf{x}_2 + \mathbf{L}\mathbf{u}_2 = \mathbf{L}\left(\mathbf{F}^2\mathbf{u}_0 + \mathbf{F}\mathbf{u}_1 + \mathbf{u}_2\right) + \mathbf{F}^3\mathbf{x}_0 \\ &\text{etc.} \end{aligned} \tag{5.29}$$

In general, the state vector $\mathbf{x}_i$ can be calculated from the inputs $(u_0, \ldots, u_{i-1})$ and the initial state $\mathbf{x}_0$ as follows:

$$\mathbf{x}_i = \mathbf{L}\sum_{j=0}^{i-1}\mathbf{F}^{i-j-1}\mathbf{u}_j + \mathbf{F}^i\mathbf{x}_0 \tag{5.30}$$

So, when $\mathbf{X}$ and $\mathbf{U}$ are stacked as indicate by equation (5.24), equation (5.30) can be written as $\mathbf{X} = \mathbf{G}\mathbf{U} + \mathbf{F}_s\mathbf{x}_0$, where:

$$\mathbf{G} = \begin{bmatrix} \mathbf{L} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{F}\mathbf{L} & \mathbf{L} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{F}^2\mathbf{L} & \mathbf{F}\mathbf{L} & \mathbf{L} & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{F}^{I-1}\mathbf{L} & \mathbf{F}^{I-2}\mathbf{L} & \mathbf{F}^{I-3}\mathbf{L} & \ldots & \mathbf{L} \end{bmatrix} \tag{5.31}$$

$$\mathbf{F}_s = \begin{bmatrix} \left(\mathbf{F}^1\right)^T & \left(\mathbf{F}^2\right)^T & \ldots & \left(\mathbf{F}^I\right)^T \end{bmatrix}^T \tag{5.32}$$

The state error covariance matrix $\mathbf{C}_X$, belonging to the stacked state vectors $\mathbf{X}$, consists of the covariance matrices $E\left[\mathbf{x}_i\mathbf{x}_i^T\right]$ on the diagonal, the cross-covariance matrices $E\left[\mathbf{x}_{i+j}\mathbf{x}_i^T\right]$ left from the diagonal and $E\left[\mathbf{x}_i\mathbf{x}_{i+j}^T\right]$ right from the diagonal:

$$\mathbf{C}_X = E\left[\mathbf{X}\mathbf{X}^T\right] = \begin{bmatrix} E\left[\mathbf{x}_1\mathbf{x}_1^T\right] & E\left[\mathbf{x}_1\mathbf{x}_2^T\right] & \ldots & E\left[\mathbf{x}_1\mathbf{x}_I^T\right] \\ E\left[\mathbf{x}_2\mathbf{x}_1^T\right] & E\left[\mathbf{x}_2\mathbf{x}_2^T\right] & \ldots & E\left[\mathbf{x}_2\mathbf{x}_I^T\right] \\ \vdots & \vdots & \ddots & \vdots \\ E\left[\mathbf{x}_I\mathbf{x}_1^T\right] & E\left[\mathbf{x}_I\mathbf{x}_2^T\right] & \ldots & E\left[\mathbf{x}_I\mathbf{x}_I^T\right] \end{bmatrix} \tag{5.33}$$

The matrices $E\left[\mathbf{x}_i\mathbf{x}_i^T\right]$ can be expressed as a function of the system matrix $\mathbf{F}$ and the process noise covariance matrix $\mathbf{C}_w$:

$$\begin{aligned} E\left[\mathbf{x}_1\mathbf{x}_1^T\right] &= \mathbf{C}_w \\ E\left[\mathbf{x}_2\mathbf{x}_2^T\right] &= \mathbf{F}\mathbf{C}_w\mathbf{F}^T + \mathbf{C}_w \\ E\left[\mathbf{x}_3\mathbf{x}_3^T\right] &= \mathbf{F}^2\mathbf{C}_w\left(\mathbf{F}^2\right)^T + \mathbf{F}\mathbf{C}_w\mathbf{F}^T + \mathbf{C}_w \\ E\left[\mathbf{x}_i\mathbf{x}_i^T\right] &= \sum_{k=0}^{i-1}\mathbf{F}^k\mathbf{C}_w\left(\mathbf{F}^k\right)^T \end{aligned} \tag{5.34}$$

An equation for the error covariance $E\left[\mathbf{x}_{i+j}\mathbf{x}_i^T\right]$ can be derived by first considering the following state recursions:

$$\begin{aligned} \mathbf{x}_{i+1} &= \mathbf{F}\mathbf{x}_i + \mathbf{w}_i \\ \mathbf{x}_{i+2} &= \mathbf{F}^2\mathbf{x}_i + \mathbf{F}\mathbf{w}_i + \mathbf{w}_{i+1} \\ \mathbf{x}_{i+3} &= \mathbf{F}^3\mathbf{x}_i + \mathbf{F}^2\mathbf{w}_i + \mathbf{F}\mathbf{w}_{i+1} + \mathbf{w}_{i+2} \\ \mathbf{x}_{i+j} &= \mathbf{F}^j\mathbf{x}_i + \sum_{k=0}^{j-1}\mathbf{F}^{j-k-1}\mathbf{w}_{i+k} \end{aligned} \tag{5.35}$$

So, $E\left[\mathbf{x}_{i+j}\mathbf{x}_i^T\right]$ becomes:

$$\begin{aligned} E\left[\mathbf{x}_{i+j}\mathbf{x}_i^T\right] &= E\left[\left(\mathbf{F}^j\mathbf{x}_i + \sum_{k=0}^{j-1}\mathbf{F}^{j-k-1}\mathbf{w}_{i+k}\right)\mathbf{x}_i^T\right] \\ &= E\left[\mathbf{F}^j\mathbf{x}_i\mathbf{x}_i^T\right] + E\left[\sum_{k=0}^{j-1}\mathbf{F}^{j-k-1}\mathbf{w}_{i+k}\mathbf{x}_i^T\right] \\ &= \mathbf{F}^j E\left[\mathbf{x}_i\mathbf{x}_i^T\right] \end{aligned} \tag{5.36}$$

Note that $E\left[\sum_{k=0}^{j-1}\mathbf{F}^{j-k-1}\mathbf{w}_{i+k}\mathbf{x}_i^T\right]$ is always zero, because $\mathbf{w}_i$ is assumed to be zero-mean white noise. Thus the expectation of $\mathbf{w}_{i+k}$ times another vector is zero as well. Furthermore, it can be proven that $E\left[\mathbf{x}_i\mathbf{x}_{i+j}^T\right] = E\left[\mathbf{x}_{i+j}\mathbf{x}_i^T\right]^T$.

Now that all matrices in the equations (5.26) are defined, the maximization problem (5.24) can be further evaluated by using the general probability density function (5.11) for multivariate normally distributed signals. Again, since maximizing this equation is equal to maximizing the natural logarithm of this equation, the problem can be simplified to the following minimization problem:

$$\begin{aligned} \left(\hat{\mathbf{X}}, \hat{\mathbf{U}}\right) &= \underset{\mathbf{X},\mathbf{U}}{\operatorname{argmax}}\left\{p\left(\mathbf{Z}|\mathbf{X}\right)p\left(\mathbf{X}|\mathbf{U},\mathbf{x}_0\right)p\left(\mathbf{U}\right)\right\} \\ &= \underset{\mathbf{X},\mathbf{U}}{\operatorname{argmin}} \\ &\frac{1}{2}\left(\begin{bmatrix}\mathbf{H}_s & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{X}\\ \mathbf{U}\end{bmatrix} + \mathbf{Z}\right)^T\mathbf{C}_V^{-1}\left(\begin{bmatrix}\mathbf{H}_s & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{X}\\ \mathbf{U}\end{bmatrix} + \mathbf{Z}\right) + \\ &\frac{1}{2}\left(\begin{bmatrix}\mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{X}\\ \mathbf{U}\end{bmatrix}\right)^T\mathbf{C}_U^{-1}\left(\begin{bmatrix}\mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{X}\\ \mathbf{U}\end{bmatrix}\right) + \\ &\frac{1}{2}\left(\begin{bmatrix}\mathbf{I} & -\mathbf{G}\end{bmatrix}\begin{bmatrix}\mathbf{X}\\ \mathbf{U}\end{bmatrix} - \mathbf{F}_s\mathbf{x}_0\right)^T\mathbf{C}_X^{-1}\left(\begin{bmatrix}\mathbf{I} & -\mathbf{G}\end{bmatrix}\begin{bmatrix}\mathbf{X}\\ \mathbf{U}\end{bmatrix} - \mathbf{F}_s\mathbf{x}_0\right) \end{aligned} \tag{5.37}$$

Differentiation with respect to the vector $\begin{bmatrix} \mathbf{X} \\ \mathbf{U} \end{bmatrix}$ and equating to zero results in the following least squares solution:

$$\begin{bmatrix} \hat{\mathbf{X}} \\ \hat{\mathbf{U}} \end{bmatrix} = \mathbf{A}^{+}\mathbf{B} \tag{5.38}$$

where $\mathbf{A}^{+}$ is the pseudo inverse of $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} \begin{bmatrix} -\mathbf{H}_s & \mathbf{0} \end{bmatrix}^T \mathbf{C}_V^{-1} \begin{bmatrix} -\mathbf{H}_s & \mathbf{0} \end{bmatrix} \\ \begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix}^T \mathbf{C}_U^{-1} \begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix} \\ \begin{bmatrix} \mathbf{I} & -\mathbf{G} \end{bmatrix}^T \mathbf{C}_X^{-1} \begin{bmatrix} \mathbf{I} & -\mathbf{G} \end{bmatrix} \end{bmatrix} \tag{5.39}$$

and $\mathbf{B}$:

$$\mathbf{B} = \begin{bmatrix} \begin{bmatrix} \mathbf{H}_s & \mathbf{0} \end{bmatrix}^T \mathbf{C}_V^{-1}\mathbf{Z} \\ \mathbf{0} \\ \begin{bmatrix} \mathbf{I} & -\mathbf{G} \end{bmatrix}^T \mathbf{C}_X^{-1}\mathbf{F}_s\mathbf{x}_0 \end{bmatrix} \tag{5.40}$$

Algorithm summary

The algorithm for state and input estimation in closed-form consists of the following mains steps:

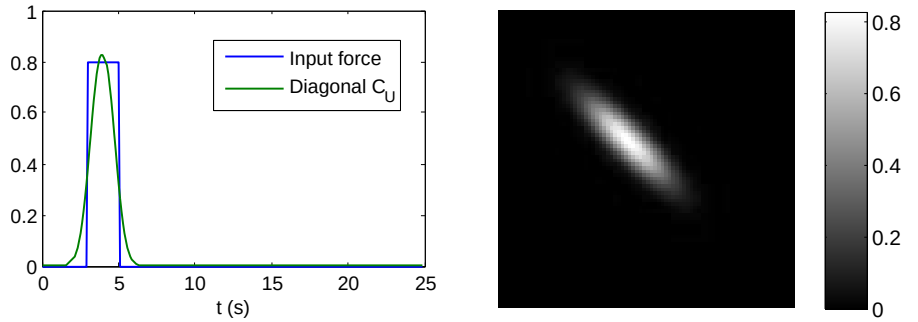
1. Stack measurement vectors.
2. Construct the model matrices $\mathbf{G}$ (5.31), $\mathbf{F}_s$ (5.32) and $\mathbf{H}_s$ (5.32).
3. Construct the covariance matrices $\mathbf{C}_X$ (5.33) and $\mathbf{C}_V$ (5.28).
4. Calculate matrices $\mathbf{A}$ (5.39) and $\mathbf{B}$ (5.40) and estimate $\mathbf{X}$ and $\mathbf{U}$ (5.38).
5. Unstack $\mathbf{X}$ and $\mathbf{U}$ into vectors $\mathbf{x}_{i+1}$ and $\mathbf{u}_i$.

The calculation of the covariance matrices $E\left[\mathbf{x}_i\mathbf{x}_i^T\right]$ and $E\left[\mathbf{x}_{i+j}\mathbf{x}_i^T\right]$ and the input matrix $\mathbf{G}$ can be done iteratively.

Simulation results

As already explained, constraints on the input estimation can now be enforced by means of a covariance matrix. The diagonal elements of this covariance matrix can be used to assign an estimation of the input profile in time. The cross elements can be used to limit the bandwidth and thus to reduce the noise. In case of the kinematic-dynamic test model, the diagonal elements of the input covariance matrix are calculated by convolving the original input sequence with a (one-dimensional) Gaussian window, see figure 5.5a. Furthermore, a small off-set is added to all diagonal elements, to make the matrix invertible (see appendix D for explanation). The cross elements are assigned by applying a two-dimensional Gaussian convolution, see figure 5.5b.

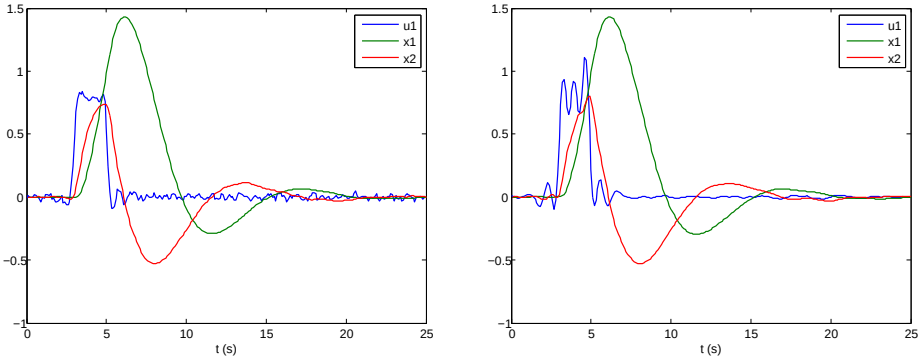
The closed-form state and input estimation is implemented and applied on the test model as well. Figure 5.6a shows the simulation results whereby the



(a) *Diagonal elements* **(b)** *Top view left upper part*

Figure 5.5: *Input covariance matrix $\boldsymbol{C}_U$ of the kinematic test model.*

estimation is based on measured position and velocity (i.e. $\mathbf{H}$ is the identity matrix). As can be seen the noise on the estimated input is significantly reduced. Furthermore, also a simulation is performed whereby only the position is measured (i.e. $\mathbf{H} = [1\ 0]$). These results are shown in figure 5.6b. As can be seen in this figure, the states and input are still estimated quite well. So, it can be concluded that the algorithm performs well even in the situation where not all states can be measured.



(a) *Measured position and velocity* **(b)** *Measured position only*

Figure 5.6: *Simulation results of closed-form state and input estimation from position and velocity measurements (a) and from position measurements only (b).*

5.4 Parameter estimation

In the previous section the system parameters were assumed to be known. However, in case of the development of a tongue or lip model, information about system parameters requires information about (possible unavailable) physical or biomechanical properties of the actual system. In case of a kinematic-dynamic model, these system parameters are for example mass, damping and stiffness coefficients of the soft tissue. In this section it will be discussed how to deal

with a system with known (kinematic) structure, but with unknown system parameters. The objective is to estimate, beside the states and input, also the system parameters from the measurement sequence.

Derivation of cost function

Suppose the main structure of the system matrix is known (or assumed to be known), but does contain some unknown parameters, gathered in the vector $\boldsymbol{\alpha}$. The system matrix is than a function of these unknow parameters: $\mathbf{F}(\boldsymbol{\alpha})$. In case of the test model, the unknow parameters are for example the spring coefficient k and the damping coefficient d, so $\boldsymbol{\alpha} = [k\ d]^T$. The objective is now to estimate, beside the states $\mathbf{X}$ and the input $\mathbf{U}$, also the system parameters $\boldsymbol{\alpha}$, given the measurements $\mathbf{Z}$:

$$\begin{aligned}\left(\hat{\mathbf{X}}, \hat{\mathbf{U}}, \hat{\boldsymbol{\alpha}}\right) &= \underset{\mathbf{X},\mathbf{U},\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{p\left(\mathbf{X}, \mathbf{U}, \boldsymbol{\alpha}|\mathbf{Z}\right)\right\} \\ &= \underset{\mathbf{X},\mathbf{U},\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{p\left(\mathbf{Z}|\mathbf{X}, \mathbf{U}, \boldsymbol{\alpha}\right) p\left(\mathbf{X}|\mathbf{U}, \boldsymbol{\alpha}\right) p\left(\mathbf{U}|\boldsymbol{\alpha}\right) p\left(\boldsymbol{\alpha}\right)\right\} \\ &= \underset{\mathbf{X},\mathbf{U},\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{p\left(\mathbf{Z}|\mathbf{X}, \mathbf{U}\right) p\left(\mathbf{X}|\mathbf{U}, \boldsymbol{\alpha}\right) p\left(\mathbf{U}\right)\right\}\end{aligned} \tag{5.41}$$

It will be assumed that $p\left(\boldsymbol{\alpha}\right)$ is uniformly distributed (possibly within a certain interval). This means that each possible value for the system parameters has an equal chance and that $p\left(\boldsymbol{\alpha}\right)$ has no maximum value. Furthermore, $p\left(\mathbf{U}|\boldsymbol{\alpha}\right) = p\left(\mathbf{U}\right)$ and $p\left(\mathbf{Z}|\mathbf{X}, \mathbf{U}, \boldsymbol{\alpha}\right) = p\left(\mathbf{Z}|\mathbf{X}, \mathbf{U}\right)$, since these probabilities are independent of the system parameters.

The distributions of $p\left(\mathbf{Z}|\mathbf{X}, \mathbf{U}\right)$ and $p\left(\mathbf{U}\right)$ are given by respectively the equations (5.26a) and (5.26c). The distribution of $p\left(\mathbf{X}|\mathbf{U}, \boldsymbol{\alpha}\right)$ is similar to equation (5.26b). However, the matrix $\mathbf{G}$ and the covariance matrix $\mathbf{C}_X$ are now a function of $\mathbf{F}$ and thus a function $\boldsymbol{\alpha}$:

$$p\left(\mathbf{X}|\mathbf{U}, \mathbf{x}_0\right) \sim N\left(\mathbf{G}(\boldsymbol{\alpha})\mathbf{U} + \mathbf{F}_s\mathbf{x}_0, \mathbf{C}_X(\boldsymbol{\alpha})\right) \tag{5.42}$$

Since the covariance matrix is thus also a function of the input arguments, the term in front of the exponential in the probability density function cannot simply be neglected anymore when maximizing this function. An expression for the maximization of $p\left(\mathbf{X}|\mathbf{U}, \boldsymbol{\alpha}\right)$ can is this case be derived as follows:

$$\begin{aligned}&\underset{\mathbf{X},\mathbf{U},\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{p\left(\mathbf{X}|\mathbf{U}, \boldsymbol{\alpha}\right)\right\} \\ &= \underset{\mathbf{X},\mathbf{U},\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{\frac{1}{(2\pi)^{nI/2}|\mathbf{C}_X|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{X} - \mathbf{G}\mathbf{U})^T\mathbf{C}_X^{-1}(\mathbf{X} - \mathbf{G}\mathbf{U})\right)\right\} \\ &= \underset{\mathbf{X},\mathbf{U},\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{\log\left(\frac{1}{|\mathbf{C}_X|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{X} - \mathbf{G}\mathbf{U})^T\mathbf{C}_X^{-1}(\mathbf{X} - \mathbf{G}\mathbf{U})\right)\right)\right\} \\ &= \underset{\mathbf{X},\mathbf{U},\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{-\frac{1}{2}(\mathbf{X} - \mathbf{G}\mathbf{U})^T\mathbf{C}_X^{-1}(\mathbf{X} - \mathbf{G}\mathbf{U}) - \frac{1}{2}\log|\mathbf{C}_X|\right\} \\ &= \underset{\mathbf{X},\mathbf{U},\boldsymbol{\alpha}}{\operatorname{argmin}} \left\{(\mathbf{X} - \mathbf{G}\mathbf{U})^T\mathbf{C}_X^{-1}(\mathbf{X} - \mathbf{G}\mathbf{U}) + \log|\mathbf{C}_X|\right\}\end{aligned} \tag{5.43}$$

Combining this equation with the other two probability density functions results in the following cost function:

$$J(\mathbf{X},\mathbf{U},\boldsymbol{\alpha}) = (\mathbf{Z}-\mathbf{H}_s\mathbf{X})^T\,\mathbf{C}_V^{-1}\,(\mathbf{Z}-\mathbf{H}_s\mathbf{X}) + \mathbf{U}^T\mathbf{C}_U^{-1}\mathbf{U} + (\mathbf{X}-\mathbf{G}\mathbf{U})^T\,\mathbf{C}_X(\boldsymbol{\alpha})^{-1}\,(\mathbf{X}-\mathbf{G}\mathbf{U}) + \log|\mathbf{C}_X(\boldsymbol{\alpha})| \tag{5.44}$$

As can be seen, this cost function also includes the determinant of the covariance matrix $|\mathbf{C}_X|$. In case of a large matrix with many values smaller than one on its diagonal, the determinant becomes very small. Mathematical software packages, like MATLAB, round this easily to zero. The determinant of a certain matrix can therefore also be calculated by calculating the product of all the eigenvalues λ_i of the matrix:

$$|\mathbf{C}_X| = \prod_{i=1}^{nI}\lambda_i \tag{5.45}$$

The natural logarithm of the matrix determinant can now be calculated without any problem by computing the sum of all the eigenvalue-logarithms:

$$\log|\mathbf{C}_X| = \log\left(\prod_{i=1}^{nI}\lambda_i\right) = \sum_{i=1}^{nI}\log(\lambda_i) \tag{5.46}$$

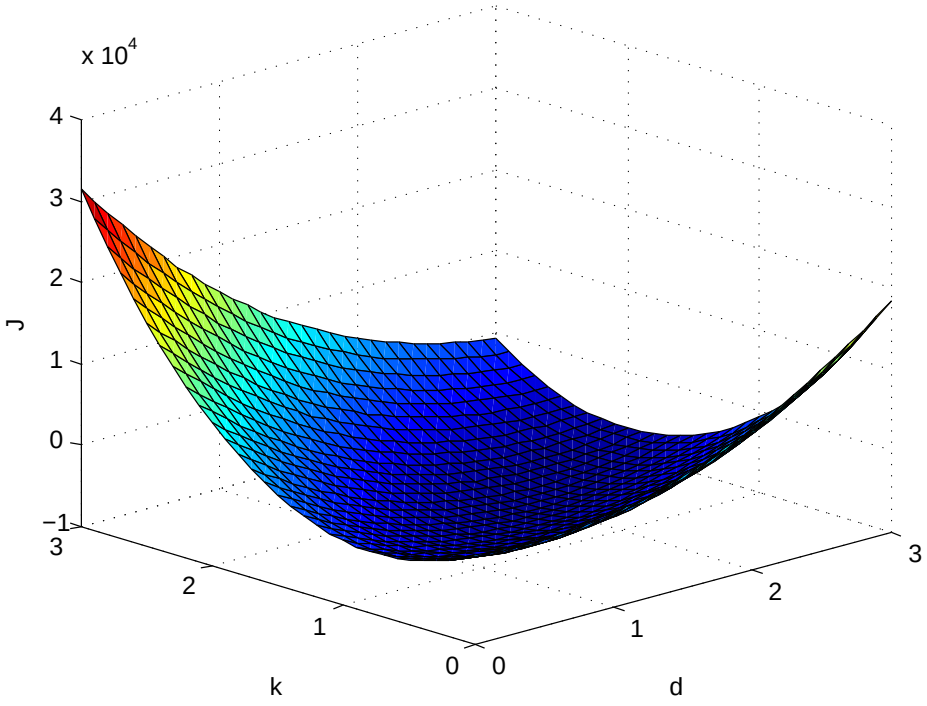
Minimization of cost function

An estimate of the stacked states $\mathbf{X}$, input $\mathbf{U}$ and system parameters $\boldsymbol{\alpha}$ can be obtained by minimizing the cost function of equation (5.44). However, this equation is not linear anymore and cannot easily be differentiated. Therefore an algorithm is implemented in MATLAB that finds the minimum value of J and the corresponding parameters numerically, by using the function `fminsearch`. This function uses the simplex search method described in [24].

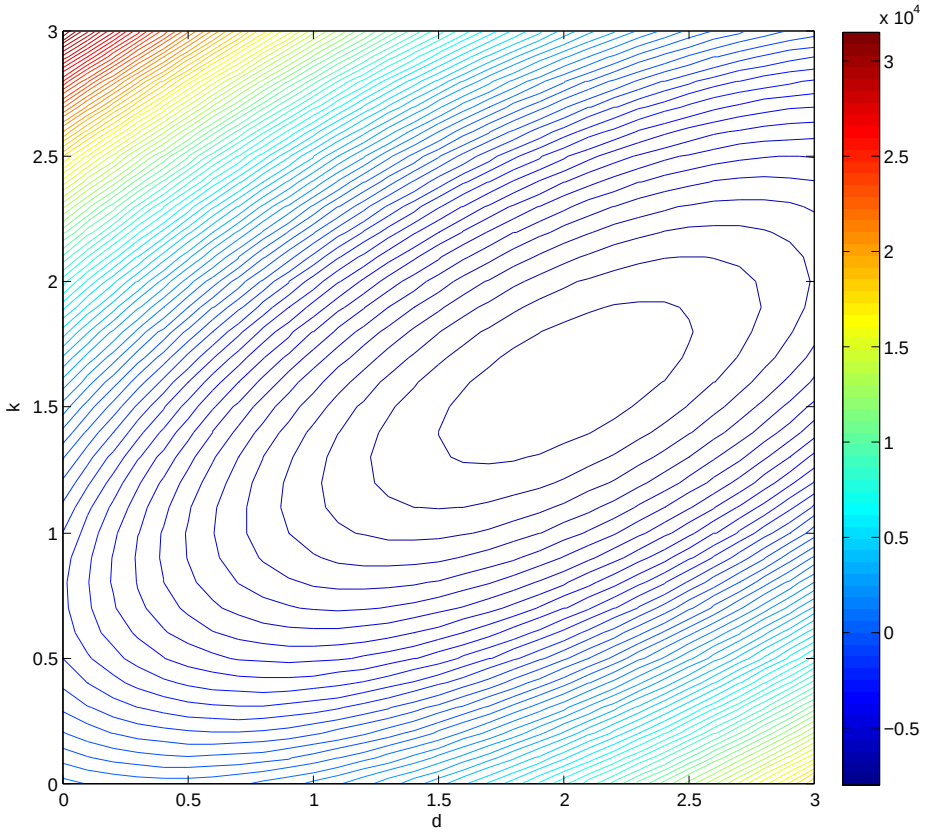
Simulation results

With the test model a measurement sequence is generated with system matrix parameters $k = 1.6$ and $d = 2$. This measurement sequence and the model structure (without the parameter values) are supplied to the state, input and parameter estimation algorithm. Figure 5.7 shows the calculated values of J for different values of the system parameters k and d. The function has a global minimum and the shape suggest that finding the minimum value can't be too difficult. This minimum value is numerically calculated and is indeed at the correct parameter values of k and d.

Furthermore, in the contour plot it can clearly be seen that there is a kind of trough. This trough goes in the direction of the origin. So, it looks like that there is a small linear relation between the two parameters and that scaling both parameters with the same (small) value does not have much influence on the minimum value of the cost function. This is an advantage in case the (patient specific) parameters can be measured and the focus is on input (and state) estimation, since it relaxes the accuracy of these measurements a bit. Most import is the linear relation between the two parameters.



(a) *Three-dimensional plot*



(b) *Contour plot*

Figure 5.7: *Value of J for different system parameter values.*

5.5 Conclusions

This chapter was concerned with the investigation of possibilities for system identification with unknown inputs. Therefore, the chapter was mainly focused on the derivation of algorithms for the estimation of states, input and system parameters of a linear discrete-time state space model. The developed algorithms assume that the main structure (e.g. kinematic-dynamic) of the system is known and require, beside a sequence of measurements, the input of covariance matrices for the measurement noise, the process noise and the input signals.

For the estimation of the states and input signals, two different algorithms have been derived and implemented. The first one is recursive and is therefore appropriate for a system with vectors and matrices of high dimensions and a large number of measurement samples. The second algorithm is in closed-form, which means that the states and input are estimated from the stacked measurement vectors and stacked model matrices. This enables the possibility to constrain the input by supplying a covariance matrix that describes the correlation between input signals (bandwidth limitation) and the chance on input signals at certain time moments. On this way, the input can be estimated more accurately. However, the disadvantage of the closed-form algorithm is that the construction of the closed-form matrices is time-consuming and that the matrices might become too large in case of high-dimensional vectors. Furthermore, a cost function (for the closed-form case) has been derived to include the estimation of system parameters as well. This cost function is minimized numerically. From experiments performed on the implemented test model, it can be concluded the algorithms work properly: the measurement noise and process noise are mainly filtered out, the input signals are estimated quite accurate and the system parameters are determined correctly (with a reasonable initial guess).

Summarized, with these algorithms a first step is made in the research to phenomenological identification possibilities of a physical system like the tongue and lips. The algorithms show that it is possible to estimate the model input variables from measured output variables only. However, to do so, a priori knowledge about the actual system is required, for example that the system can be modeled as a kinematic-dynamic system. Overall, it can be concluded that for the development of an adequate and complete model, at least information from EMG measurements is required and preferably also more information about the actual physical system (e.g. muscle behavior).

6

Conclusions and recommendations

6.1 Conclusions

This thesis project was concerned with the exploration of a dynamic model of the human tongue and lips. Because of the complicated anatomical and muscular structure of these organs, it was decided to aim at a phenomenological black-box model, rather than at a complicated, detailed physiological model. Since a phenomenological model is developed based on the outside behavior of a system (or organ), methods have been investigated to capture and describe tongue and lip shapes and movements. Furthermore, a large part of this research was focused on the derivation, investigation and implementation of mathematical algorithms for modeling dynamic systems. The purpose of this research part was not to directly postulate the ultimate solution, but rather to show a proof of concept of possible approaches.

Contour detection and tracking

Initially, the research was mainly focused on the tongue. Therefore, an algorithm is developed that is able to detect and to track the tongue contour in (sequences of) magnetic resonance images. This algorithm uses an Active Shape Model, which is formed by training. The algorithm performs well when the tongue is clearly visible and when the images are not contaminated with too much noise. Although the ASM algorithm is only tested on MR images of the tongue, it is expected that it can also be used for lip contour detection in normal images.

Phenomenological modeling

A discrete-time linear state space model is investigated as a possible framework for the description of the dynamic behavior of the tongue and lips. This relative

simple model framework was chosen because of the limited amount of knowledge and information about the actual 'system' (i.e. the tongue or lips). From measured output signals (trajectories of landmarks on the lips) it was tried to derive as much information as possible for the construction and estimation of the model matrices. Therefore, it was also necessary to make some assumptions. In a first attempt it was assumed that the input (muscle activation signals) can be approximated as white Gaussian distributed process noise with zero-mean. This hypothesis enabled the estimation of the system matrix and the covariance matrix for the process noise. However, model evaluation by applying consistency checks showed that this model in combination with the mentioned assumptions is not optimal.

To estimate input signal from measured output signals, it is necessary to make assumptions about the system matrices. Therefore, in a second consideration, it was assumed that the motion modes of the tongue and lip can be modeled as a damped harmonic oscillator. Based on this hypothesis, a kinematic-dynamic test has been defined with known system matrices and known input distribution. Algorithms have been derived and implemented to estimate the input signals and system parameters from a measurement sequence. The algorithms perform well on the test model. However, application on the actual lip data does not yet make much sense, since validation of the estimations is not possible with this limited amount of data and information. Therefore, the derived algorithms should be considered as a proof of concept, i.e. they proof that it is possible to estimate input signals and system parameters, but only when the system structure is known and statistical information about the distribution of input signals is available. Summarized, it can be concluded that for the development of an adequate model of the tongue and lips information from EMG measurements is required.

6.2 Recommendations

The main focus of future work, concerning this project, should be on measuring muscular activation signals and on the establishment of a distribution model. This distribution model should describe the relation between groups of muscular activation signals and the dynamic variables describing the shape and motion of the tongue and lips. It is easier to capture and track lip movements than tongue movements and it is probably also easier to measure muscular activation signals on the lips (by using EMG). Therefore it is recommended to derive a distribution model for the lips first. Once this distribution model is established for the lips, possibilities can be investigated to do something similar for the tongue as well.

For the development of an adequate and accurate model, it is expected that beside measured input variables (muscle activation signals) and measured output variables (tongue and lip movements), it would also be useful to include some more information about biomechanical properties of the tongue and lips. These properties are for example the mass and elasticity of tongue and lip tissue. The resulting biomechanical constraints and parameters can be used for better estimation of system parameters. More information also opens the door

to the investigation of more advanced models. As is shown in this research, a linear state space model is quite limited. This implies that it is actually recommended to focus not on pure phenomenological modeling, but also to include some physiological information.

Furthermore, there are some other issues that have to be investigated. One of them is the investigation of proper methods for the acquisition of fast tongue movements. The current acquisition speed of an MRI scanner appeared to be too low for capturing realistic (fast) tongue movements. Also a way has to be found to measure muscle activation signals of the tongue. In a later stage, when such a model is successfully derived for the tongue as well, the models can even be further extended to the total oral cavity and pharynx. In the end this should result in a complete system that enables virtual surgery for a specific patient.

A

Lip data

Because of a shortage of appropriate tongue data (i.e. trajectories of landmarks on the tongue contour during fast and realistic[1] tongue movements), developed models can initially be evaluated by using data of fast lip movements. This data consists of trajectories of landmarks on the outer lip contour. This appendix describes how this data is obtained. Furthermore, in this appendix also some small experimental results are presented to provide a general idea of lip movement experiments.

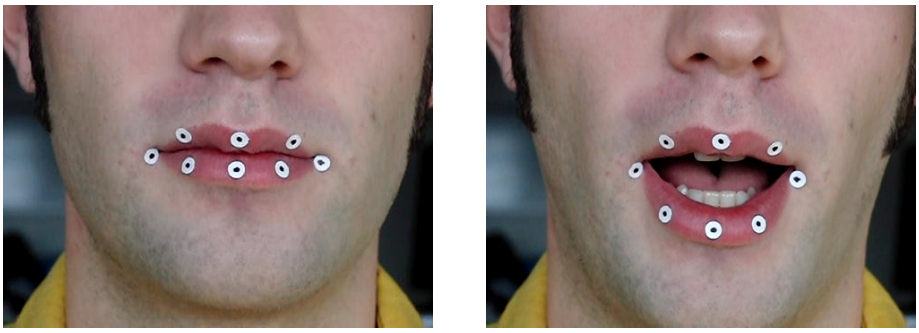
A.1 Acquisition method

Since the emphasis of a large part of the research project is on the development of a dynamic model and not on the way how the lip data is obtained, it was decided to design a simple, but robust, acquisition method. For easy automatic detection of lip features, the lips were provided with eight markers. The created markers were circular white stickers with a diameter of 5 mm. They were provided with a black dot - with a diameter of about 1.5 mm - in the middle of the sticker. This kind of markers was chosen since they can easily be detected by convolving the images with a Gaussian function. More details concerning the detection procedure will follow in subsection A.2.

For capturing the lip movements, a normal consumer camera (the *Casio Exilim EX-FC100*) was used. This camera can take color images with a resolution of 480 $\times$ 360 pixels at a frame rate of 210 images per second. The camera was installed in a room with sufficient daylight. The test person, provided with eight markers on his lips, had to take place on a chair with his face in front of the camera. Figure A.1 shows an example of the captured images.

To obtain proper data for the evaluation models, marker trajectories of different lip movements are captured: two sequences of 14 seconds (2940 images) during random movements and furthermore sequences during the pronounce-

[1] Realistic tongue movements are assumed to be movements belonging to for example swallowing and the pronouncement of visemes.



(a) *Mouth closed* **(b)** *Mouth open*

Figure A.1: *Examples of a captured image for the extraction of lip data.*

ment of several basic visemes. The random lip movements consisted of a few clear movements, like opening and closing of the mouth, compression and decompression of the lips in horizontal direction and the pronouncement of letters in English. This kind of data is expected to be appropriate for the extraction of principal lip shapes (by applying PCA). Furthermore, sequences are captured during the pronouncement of the words `/papa/`, `/mama/` and a selection of visemes from [25]: `/silent/`, `/boat/`, `/wet/`, `/size/`, `/eat/`, `/earth/` and `/if/`. It is expected that for example `/papa/` and `/mama/` results in similar feature trajectories and visemes like `/papa/` and `/boat/` in different trajectories.

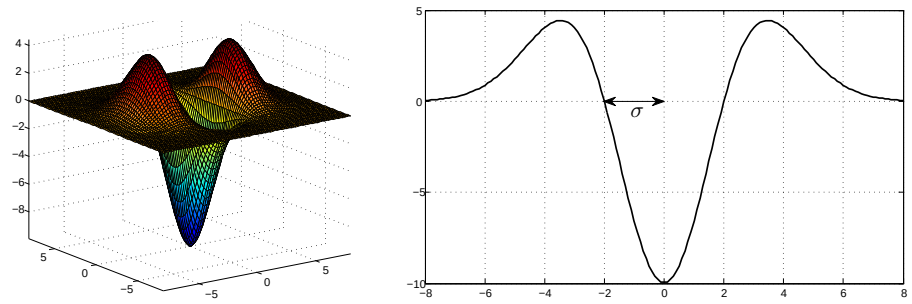
A.2 Detection and tracking of markers

Detection

As already described, the used lip markers are circular white stickers with a black dot in the middle. These simple textures do not occur in the rest of the face (and background) and can easily be detected by convolving the images with the second derivatives of a Gaussian function, one in x- and one in y-direction. The formula of the second derivative in x-direction is given by equation (A.1).

$$h_{xx} = \left(-\frac{1}{2\pi\sigma^4} + \frac{x^2}{2\pi\sigma^6}\right) \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{A.1}$$

The width of the function is determined by σ (see figure A.2) and its value is based on the width (in pixels) of the dots on the stickers in the images. The Gaussian functions also work as a kind of low pass filter. So, after the convolution ($h_{xx} * I + h_{yy} * I$) the image is smoothed and only the positions of the markers are amplified (figure A.3b). By finding the m highest regional maxima (m is the number of markers), the center locations of the markers are determined (figure A.3c). In this way the markers can be detected with an estimated accuracy of about two pixels.

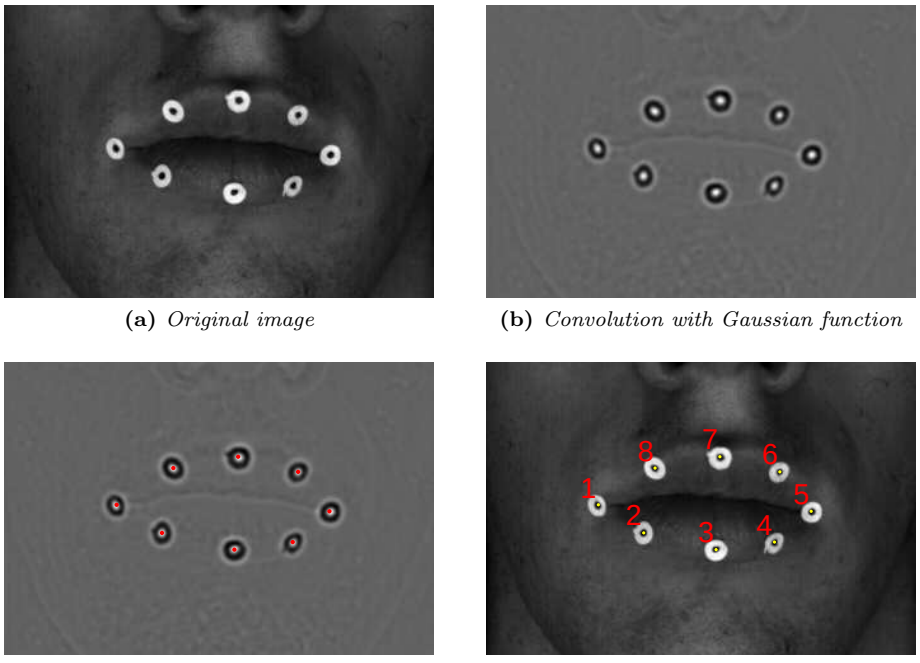


(a) *3D-plot* **(b)** *Cross section*

Figure A.2: *Second derivative of a Gaussian in x-direction.*

Tracking

For tracking landmarks in a sequence of lip images, the detected markers have to be sorted. In the first image of a sequence, the markers are sorted by angle from the center of the mouth, which is the mean of all markers in x- and y-direction (figure A.3d). Since the distances between corresponding markers in two successive images is small, tracking is done by finding for each marker in an image the nearest marker in the previous image. This is done by calculating the Euclidean distances.



(a) *Original image* **(b)** *Convolution with Gaussian function*

(c) *Detection of regional maxima* **(d)** *Arrangement of detected markers*

Figure A.3: *Detection procedure of markers on the lips.*

Alignment

During capturing the sequences of lip movements, the test person's head might have moved a bit, for example closer to or further away from the camera or a bit more to the left or right. So, the capturing method is not yet scale, rotation and translation invariant. For comparing different marker trajectories, these possible scaling, rotation and translation differences have to filtered out. This is accomplished by applying the same algorithm as described in section 3.3.2 for the alignment of training shapes.

A.3 Experiments

This section describes the results of some experiments performed on the measured lip landmark trajectories. These experiments include the estimation of velocity (and acceleration) components, feature reduction by using principal component analysis and the investigation of feature trajectories for different visemes.

Estimation of velocity and acceleration

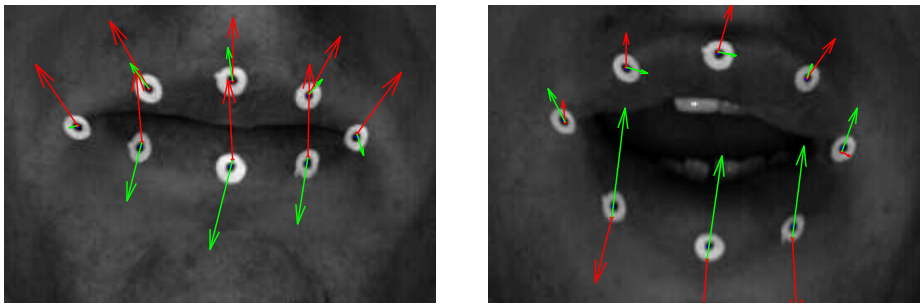
The velocity and acceleration components are estimated from the detected landmark position by using the implemented Kalman filter (see section 4.5.1). The used system matrix is based on kinematic equations:

$$\mathbf{F} = \begin{bmatrix} \mathbf{I} & \Delta t\mathbf{I} & \frac{1}{2}\Delta t^2\mathbf{I} \\ \mathbf{0} & \mathbf{I} & \Delta t\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{A.2}$$

The sample period Δt is 1/210 second. Since the markers are detected with an accuracy of about two pixels, the used covariance matrix of the measurement noise is chosen as: $\mathbf{C}_v = \sigma_v^2\mathbf{I}$, with $\sigma_v^2 = 2$. The covariance matrix of the process noise is determined empirically. Figure A.4 shows two examples of images with detected markers and estimated velocity and acceleration components. The results are as would be expected: the arrows point in the correct moving direction.

Principal lip shapes

PCA training is applied on the state vectors belonging to the sequence of random lip movements. In case of only landmark positions ($n = 16$), four principal components appeared to be sufficient to explain 99% of the total variation. In case of position and velocity components ($n = 32$), six components appeared to be sufficient to obtain this percentage. Figure A.5 shows the effects on lip shapes and velocity components by varying the first three principal shape parameters. As can been seen in this figure, the first parameter describes the dynamic opening and closing of the mouth in vertical direction, the second parameter the dynamic opening and closing in horizontal direction and the third parameter



(a) *Start of mouth opening* **(b)** *Start of mouth closing*

Figure A.4: *Captured images during the pronouncement of a viseme. The blue asterisks indicate the measured tag locations, the red asterisks the estimated locations, the red arrows the estimated velocity (magnitude and direction) and the green arrows the estimated acceleration (magnitude and direction).*

mainly the static vertical opening. The parameters are varied only within ± 1.5 times the standard deviation. By exceeding these limits, unrealistic shapes are generated. For example when the third parameter becomes smaller than -1.5 times the standard deviation, the lip landmarks cross each other in vertical direction. Apparently, the feature data it not fully Gaussian distributed.

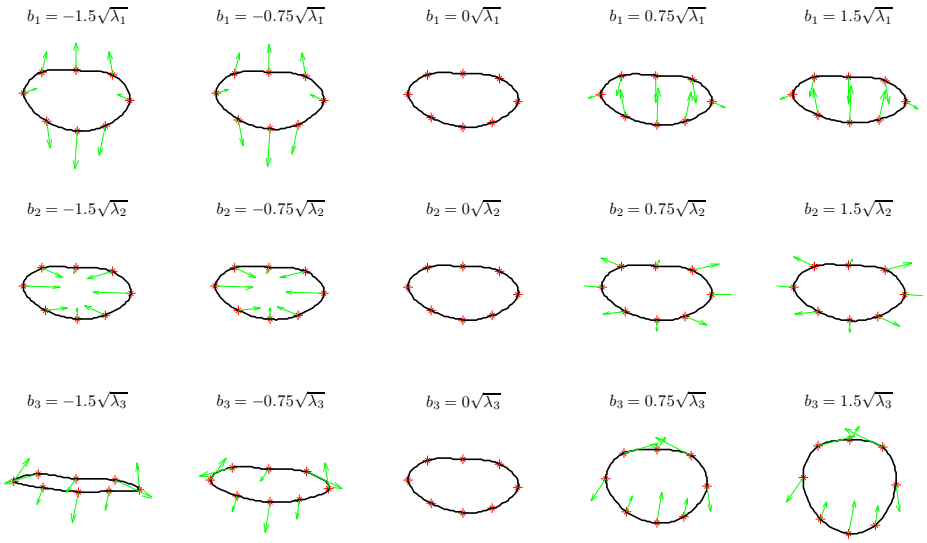


Figure A.5: *Effects on lip shapes and velocity components by varying the first three principal static control parameters (which explains 95% of the total variation).*

Trajectory investigation

The trajectories (evolution in time) of the first element in the PCA-reduced state vector have been investigated for the pronouncement of the different visemes. This vector element describes the opening and closing of the mouth. Figure A.6

shows the trajectories for several pronouncements of /papa/ and /mama/. For comparison purposes, these trajectories are aligned in time (horizontal translation) by using the convolution technique. As can been in the figure, the pattern of the trajectories is similar (as would be expected), but the pronouncement of /mama/ is shorter than the pronouncement of /papa/. Furthermore, the amplitudes are different: in case of /papa/ the mouth opening is larger and the lips are pushed closer to each other. Figure A.7 shows a comparison between the pronouncement of the visemes /papa/ and /boat/. As can be seen, the trajectories have different patterns in time and the amplitudes are different as well. So, these visemes can clearly distinguished by considering the evolution of their lip features in time.

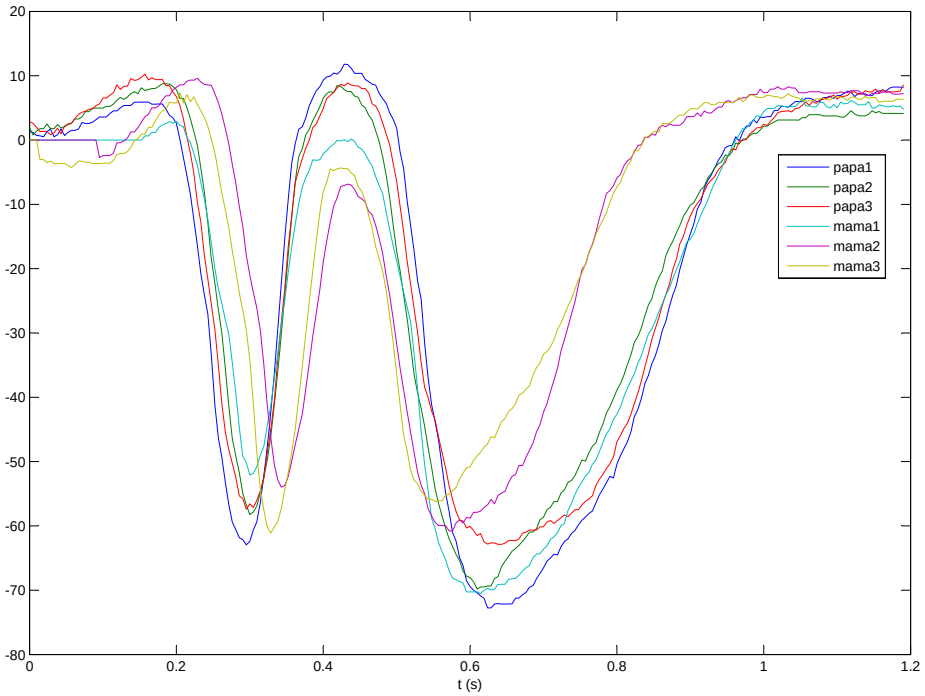


Figure A.6: *Trajectories of the first shape parameter for the pronouncement of /papa/ (3×) and /mama/ (3×). The value on the y-axis indicates the opening of the mouth (the lower the value, the larger the mouth opening).*

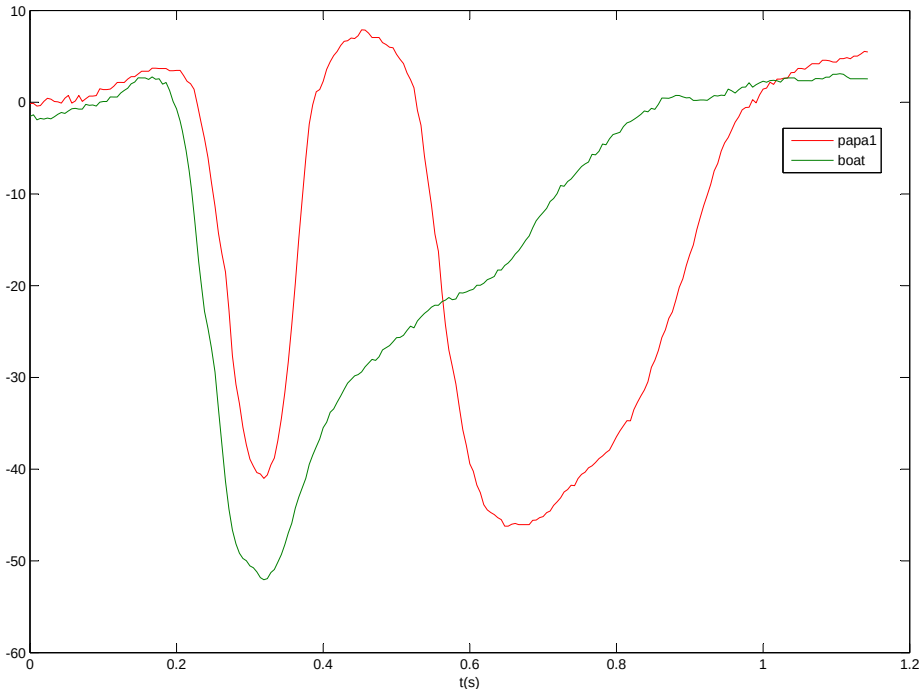


Figure A.7: *Trajectories of the first shape parameter for the pronouncement of* `/papa/` *and* `/boat/`.

B

GUI for tongue and lip simulations

To illustrate the idea of a static and dynamic model of the tongue and lips, a graphical user interface (see figure B.1) has been developed, which can be used to simulate static shapes and dynamic movements. The used lip data consists of aligned landmark trajectories that describe the random lip movements and the movements belonging to the pronouncement of the visemes. On this data PCA has been applied to extract the mean shape and the deformation modes. The dynamic behavior is described with the discrete-time linear state space model:

$$\begin{bmatrix} \mathbf{x}(i+1) \\ \mathbf{v}(i+1) \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \Delta t\mathbf{I} \\ -\frac{k}{m}\Delta t\mathbf{I} & \mathbf{I} - \frac{d}{m}\Delta t\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}(i) \\ \mathbf{v}(i) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \frac{\Delta t}{m}\mathbf{I} \end{bmatrix} \mathbf{c}(i) \tag{B.1}$$

$$\mathbf{z}(i) = \mathbf{P}\,[\mathbf{I}\ \mathbf{0}] \begin{bmatrix} \mathbf{x}(i) \\ \mathbf{v}(i) \end{bmatrix}$$

The states $\mathbf{x} \in \Re^t$ and $\mathbf{v} \in \Re^t$ represent respectively the PCA-transformed position and velocity of the landmarks. The matrix $\mathbf{P} \in \Re^{n \times t}$ consists of the eigenvectors corresponding to the first t highest eigenvalues of the covariance matrix of the training set. The elements of the vector $\mathbf{c} \in \Re^t$ represent the muscle activation signals. The mass m, the elasticity k and the damping d are the system parameters and Δt is the sample period.

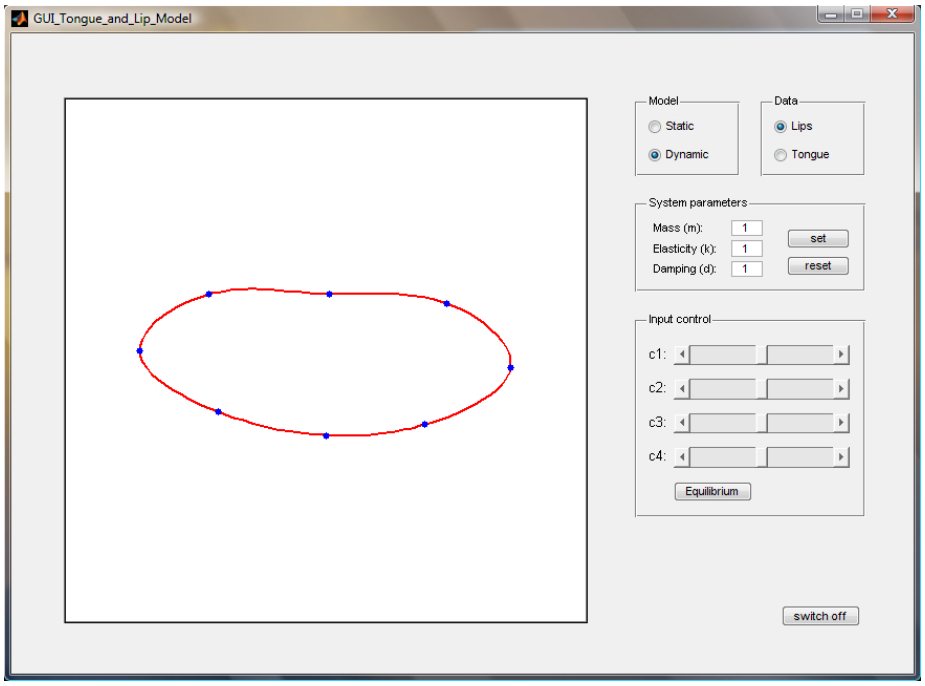


Figure B.1: *Print-screen of the GUI for lip (and tongue) shape and movement simulations, using a kinematic-dynamic model whereby* $t = 4$.

C

Distribution normalized periodogram

Suppose that $x(i)$, with $i = 0, \ldots, I-1$, is a normally distributed signal with zero mean and variance σ^2, i.e:

$$\begin{aligned} &x(i) \sim N(0, \sigma^2) \\ &x(i), x(u) \text{ uncorrelated for } i \neq u \end{aligned} \tag{C.1}$$

The discrete Fourier transform of $x(i)$, calculated over $i = 0, \ldots, I-1$, is:

$$\begin{aligned} X(k) &= \sum_{i=0}^{I-1} x(i) e^{-j2\pi ki/I} \\ &= \sum_{i=0}^{I-1} x(i) \left[\cos\left(\frac{2\pi ik}{I}\right) - j \sin\left(\frac{2\pi ik}{I}\right) \right] \end{aligned} \tag{C.2}$$

The normalized periodogram is defined as $P(k) = |X(k)|^2/I$. In this appendix it will be proven that the sequence $2P(k)/\sigma^2$ ($k = 1, \ldots, I-1$) is χ^2_2 distributed for all k.

Proof C.1

A random variable y is χ^2_N distributed when it can be constructed from the sum of the square of random and independent Gaussian random variables u_i with zero mean and unit variance:

$$y = \sum_{i=1}^{N} u_i^2 \tag{C.3}$$

To prove that $\frac{2|X(k)|^2}{\sigma^2 I} \sim \chi^2_2$, the signal $|X(k)|^2$ has to be elaborated. $X(k)$ can be split into a real part and an imaginary part:

$$X(k) = \Re(X(k)) - j\Im(X(k)) \tag{C.4}$$

with

$$\Re(X(k)) = \sum_{i=0}^{I-1} x(i) \cos\left(\frac{2\pi ik}{I}\right) \tag{C.5a}$$

$$\Im(X(k)) = \sum_{i=0}^{I-1} x(i) \sin\left(\frac{2\pi ik}{I}\right) \tag{C.5b}$$

The squared magnitude of X(k) can be written as:

$$|X(k)|^2 = Re(X(k))^2 + \Im(X(k))^2 \tag{C.6}$$

The above equation already explains the two degrees of freedom. Compare it with equation C.3, where $u_1 = \Re(X(k))$ *and* $u_2 = \Im(X(k))$*. Since* $x(i)$ *are random variables with zero mean, the expectation values of* $\Re(X(k))$ *and* $\Im(X(k))$ *are also zero. The variance of* $x(i)$ *is* σ^2*, so the variance of* $\Re(X(k))$ *is calculated as follows:*

$$\begin{aligned} Var\left(\Re(X(k))\right) &= Var\left(\sum_{i=0}^{I-1} x(i) \cos\left(\frac{2\pi ik}{I}\right)\right) \\ &= \text{Var}(x(i)) \sum_{i=0}^{I-1} \cos^2\left(\frac{2\pi ik}{I}\right) \\ &= \sigma^2 \sum_{i=0}^{I-1} \cos^2\left(\frac{2\pi ik}{I}\right) \\ &= \frac{\sigma^2 I}{2} \end{aligned} \tag{C.7}$$

The last step follows from the fact that the series $\sum_{i=0}^{I-1} \cos^2\left(\frac{2\pi ik}{I}\right)$ *is equal to* $I/2$ *for all* k*. Since* $Var\left(\Im(X(k))\right)$ *will give the same result, both* $\Re(X(k))$ *and* $\Im(X(k))$ *are normally distributed with zero mean and variance* $\frac{\sigma^2 I}{2}$*. So,* $|X(k)|^2 \sim \frac{\sigma^2 I}{2}\chi_2^2$ *and thus:*

$$\frac{2|X(k)|^2}{\sigma^2 I} \sim \chi_2^2 \tag{C.8}$$

D

Matrix regularization

In case of large covariance matrices, calculating the inverse can be a problem. This is the case when the difference between the maximum and minimum eigenvalues of the matrix is large. Such a matrix is called ill-conditioned. In this appendix it will be shown that the condition of a covariance matrix can be improved (regularized) by adding the identity matrix times a certain factor γ.

Consider the covariance matrix $\mathbf{A}$, which is a symmetric $n \times n$ matrix with eigenvector matrix $\mathbf{V} = [\mathbf{v}_1 \dots \mathbf{v}_n]$ and eigenvalue matrix $\mathbf{\Lambda}$, a diagonal matrix with diagonal entries $\lambda_1, \dots, \lambda_n$ ($\lambda_i \geq 0$). These matrices are related to each other as follows:

$$\mathbf{AV} = \mathbf{V\Lambda} \tag{D.1}$$

Since $\mathbf{A}$ is a symmetric matrix, its eigenvectors are orthogonal and thus the inverse of $\mathbf{V}$ is equal to its transpose. So, the inverse of $\mathbf{A}$ can be expressed in its eigenvector and eigenvalue matrices as follows:

$$\mathbf{A}^{-1} = \mathbf{V\Lambda}^{-1}\mathbf{V}^T = \sum_{i=1}^{n} \frac{1}{\lambda_i}\mathbf{v}_i\mathbf{v}_i^T \tag{D.2}$$

In this equation it can be seen that for a big difference between eigenvalue values, the inverse of the matrix becomes unstable. The condition of a symmetric matrix is defined as the absolute value of the ratio between the highest and smallest eigenvalue:

$$\text{Cond}(\mathbf{A}) = \left| \frac{\lambda_{\max}}{\lambda_{\min}} \right| \tag{D.3}$$

In case of a stable matrix, this value is close to one. In case of an unstable matrix the condition value is large, which means that the matrix is close to singularity. Now it will be proven that a matrix can be stabilized by adding the identity matrix times a factor γ: $\mathbf{A} + \gamma\mathbf{I}$.

Proof D.1

Suppose the eigenvector matrix of $\boldsymbol{A} + \gamma\boldsymbol{I}$ is $\boldsymbol{W}$ and its eigenvalues are α_i

($i = 1, \ldots, n$). These eigenvalues can be expressed in λ and γ as follows:

$$\begin{aligned} (\boldsymbol{A} + \gamma \boldsymbol{I})\, \boldsymbol{W} &= \boldsymbol{\alpha}\, \boldsymbol{W} \\ \boldsymbol{A}\, \boldsymbol{W} + \gamma\, \boldsymbol{V} &= \boldsymbol{\alpha}\, \boldsymbol{W} \\ \boldsymbol{A}\, \boldsymbol{W} &= (\boldsymbol{\alpha} - \gamma \boldsymbol{I})\, \boldsymbol{W} \end{aligned} \tag{D.4}$$

By comparing this result with equation (D.1), it can be seen that $\boldsymbol{W} = \boldsymbol{V}$ and $(\boldsymbol{\alpha} - \gamma \boldsymbol{I}) = \boldsymbol{\Lambda}$. The eigenvectors of $\boldsymbol{A} + \gamma \boldsymbol{I}$ are $\lambda_i = \alpha_i - \gamma_i$ and thus $\alpha_i = \lambda_i + \gamma$. So, the matrix $\boldsymbol{A} + \gamma \boldsymbol{I}$ has eigenvalues $\lambda + \gamma$ and the same eigenvector matrix $\boldsymbol{V}$ as $\boldsymbol{A}$. The condition of $\boldsymbol{A} + \gamma \boldsymbol{I}$ is:

$$Cond(\boldsymbol{A} + \gamma \boldsymbol{I}) = \left| \frac{\lambda_{max} + \gamma}{\lambda_{min} + \gamma} \right| \tag{D.5}$$

When γ is much larger than λ_{min}, λ_{min} can be neglected in relation to γ. So, on this way the condition of a covariance matrix can be improved, while keeping the same eigenvectors.

Bibliography

[1] L. Wertheimer-Hatch, G.F. Hatch, K.F. Hatch, G.B. Davis, D.K. Blanchard, R.S. Foster, and J.E. Skandalakis. Tumors of the oral cavity and pharynx. *World Journal of Surgery*, 24(4):395–400, April 2000.

[2] M.J.A. van Alphen. Investigation on possibilities for dynamic 3D visualization of the tongue. University of Twente.

[3] P. Suetens. *Fundamentals of Medical Imaging*. Cambridge University Press, 2002.

[4] O. Engwall. A 3D tongue model based on MRI data. *Proc of ICSLP 2000, 6th Intl Conf on Spoken Language Processing*, pages 901–904, 2000.

[5] M.S. Avila-Garcia, J.N. Carter, and R.I. Damper. Extracting tongue shape dynamics from magnetic resonance image sequences. *Proceedings of world academy of science, engineering and technology*, 2(1307-6884):121–124, January 2005.

[6] X. Liu, J. Zhuo, H. Agarwal, K.Z. Abd-Elmoniem, E. Murano, M. Stone, R. Gullapalli, and J.L. Prince. Quantification of three-dimensional tongue motion during speech using zHARP, February 2009.

[7] T. Kocjancic. Tongue movement and syllable onset complexity: ultrasound study. In *Proc. of ISCA Experimental Linguistics ExLing 2008*, 2008.

[8] J. Perkell. *A physiological-oriented model of tongue activity in speech production*. PhD thesis, MIT, 1974.

[9] Y. Payan and P. Perrier. Articulatory and acoustic simulations of VV transitions with a 2D biomechanical tongue model controlled by the equilibrium point hypothesis. *Speech Production Seminar: Models and Data*, pages 121–124, 1996.

[10] J. Dang and Honda K. A physiological articulatory model for simulating speech production. *Acoust. Sci. & Tech.*, 22(6):415–425, February 2001.

[11] J. Dang and Honda K. Speech production of vowel sequences using a physiological articulatory model. *Proceedings of the International Conference of Spoken Language Processing*, 5(2):1767–1770, 1998.

[12] R. Wilhelms-Tricarico. Physiological modeling of speech production: Methods for modeling soft-tissue articulators. *J. Acoust. Soc. Am.*, 97(5):3805–3098, May 1995.

[13] S Fujita, J. Dang, N Suzuki, and Honda K. A computational tongue model and its clinical applications. *Oral Science International*, 4(2):97–109, November 2007.

[14] M.C. Wu, J.C. Han, O. Rohrle, W. Thorpe, and P. Nielsen. Using Three-Dimensional Finite Element Models and Principles of Active Muscle Contraction to Analyse the Movement of the Tongue. *Proceedings of the 11th Australian International Conference on Speech Science & Technology*, December 6-8 2006.

[15] H.E. Çetingül, R.A. Chaudhry, and R. Vidal. A system theoretic approach to synthesis and classification of lip articulation. Int. Workshop on Dynamical Vision (WDV'07) at ICCV'07, October 2007.

[16] O. Engwall. Combining MRI, EMA and EPG measurements in a three-dimensional tongue model. *Speech Communication*, 41:303–329, October 2003.

[17] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, pages 321–331, 1988.

[18] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active Shape Models - Their Training and Application. *Computer Vision and Image Understanding*, 61(1):38–59, January 1995.

[19] M.M. Dickens, S.S. Gleason, and H. Sari-Sarraf. Volumetric segmentation via 3D active shape models. *Image Analysis and Interpretation, 2002. Proceedings. Fifth IEEE Southwest Symposium on*, pages 248–252.

[20] T.F. Cootes and C.J. Taylor. *Statistical Models of Appearance for Computer Vision*. PhD thesis, Image Science and Biomedical Engineering, University of Manchester, Manchester M13 9PT, U.K., March 2004.

[21] T.F. Cootes, C.J. Taylor, and A Lanitis. *Active Shape Models: Evaluation of a Multi-Resolution Method for Improving Image Search*. PhD thesis, Department of Medical Biophysics, University of Manchester, Manchester M13 9PT, U.K., March 2004.

[22] F. van der Heijden, R.P.W. Duin, D. de Ridder, and D.M.J. Tax. *Classification, Parameter Estimation and State Estimation - An Engineering Approach using Matlab*. John Wiley & Sons, Ltd, 2004.

[23] Y. Bar-Shalom and X Li. *Estimation and Tracking: Principles, Techniques, and Software*. Artech House, Inc., 1993.

[24] J.C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions.

[25] Catherine Pelachaud, Norman I. Badler, and Mark Steedman. Generating facial expressions for speech. *Cognitive Science*, 20:1–46, 1994.