Graz University of Technology

Master Thesis

# A hands-free electro-larynx device controlled by a Myo Gesture Control Armband

conducted at the
Signal Processing and Speech Communication Laboratory
Graz University of Technology, Austria

by
Klaus Huber, 0831542

Supervisors:
Dipl.-Ing. Dr.techn. Anna Katharina Fuchs
Dipl.-Ing. Dr.techn. Martin Hagmüller

Assessors/Examiners:
Dipl.-Ing. Dr.techn. Martin Hagmüller

Graz, May 24, 2016

# Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present Master's thesis.

date

(signature)

# Abstract

People who have undergone a laryngectomy, often due to cancer, need an alternative way to speak. One possibility is the so-called electro-larynx. The electro-larynx is a small hand-held, battery-driven device which is held against the neck and its vibrations are modulated with the vocal tract. Electro-larynx speech sounds robotic, and the users always need one hand for holding the device to the neck. This restricts people in their daily life.

In this work I want to address these major drawbacks by developing a hands-free ON/OFF-controlled device with the added possibility to vary the fundamental frequency. In previous works a hands-free electro-larynx device which is controlled using surface electromyography was developed. In the past these surface electromyographic signals were detected via electrodes placed on the neck. In this work surface electromyographic signals are detected by the commercially available Myo Gesture Control Armband from Thalmic Labs. In addition to the eight electrodes, the Myo Gesture Control Armband has a nine-axis inertial measurement unit which gives access to raw accelerometer data, raw gyroscope data and orientation data.

The electromyographic signals are classified into gestures which are fist and garbage. Then they are mapped to ON/OFF-control of the device in a way that fist turns the device on and garbage turns the device off. Additionally, orientation data of the Myo Armband is used for the variation of the fundamental frequency which means that changing the vertical angle of the arm results in a changing fundamental frequency. Processing is done on a Raspberry Pi, which is a small, low-cost, embedded computer. On this small computer the surface electromyographic data gets classified and the device produces an excitation signal, when a fist gesture is made. In order to produce a more natural sounding speech, pitch modification is evaluated. Although the results do not give information regarding relevance in practical use, it is possible to control fundamental frequency via changing the angle of the arm.

The electro-larynx-control system developed in this work consists of the hands-free electro-larynx device, a pre-amplifier, the Myo Gesture Control Armband and the Raspberry Pi. The major goal of this thesis lies in achieving a hands-free device and in the evaluation of gesture recognition to perform this task. Therefore, different classifiers and the influence of the user, regarding training data for the classifiers, are investigated. A general usage test with 11 subjects evaluates the performance of the ON/OFF-control and fundamental frequency variation. The results of this work suggest that ON/OFF-control of electro-larynx device by Thalmic Labs Myo Armband is feasible.

# Kurzfassung

In der menschlichen Sprache ist der Kehlkopf von fundamentaler Bedeutung. Aufgrund von Kehlkopfkrebs oder ähnlichen Erkrankungen, muss dieser oft entfernt werden, was den Patienten und Patientinnen die Fähigkeit zum Sprechen nimmt. Diese Patienten und Patientinnen sind dadurch auf alternative Methoden zur Spracherzeugung angewiesen. Eine solche Alternative stellt der Elektro-Larynx, eine elektronische Sprachhilfe, dar. Dieses kleine tragbare Gerät wird mit einer Hand am Hals positioniert und ersetzt die Funktion der fehlenden Stimmlippen indem es das, für Sprache notwendige, Anregungssignal in den Vokaltrakt einprägt. Die so erzeugte, roboter ähnlich klingende Stimme und die unpraktische Handhabung sind die Hauptnachteile eines solchen handelsüblichen Elektro-Larynx.
In dieser Arbeit sollen diese Nachteile durch einen freihändig kontrollierbaren Elektro-Larynx bewältigt werden. In vorangegangenen Arbeiten wurde ein Elektro-Larynx entwickelt, welcher direkt am Hals befestigt werden kann und durch elektromyographische Signale gesteuert wird. Früher wurden solche Signale durch Elektroden am Hals detektiert. In dieser Arbeit wird nun das Myo Gesture Control Armband von Thalmic Labs für diesen Zweck verwendet. Zusätzlich zu den acht Elektroden hat das Myo Armband eine interne Messeinheit eingebaut, welche Beschleunigs-, Gyroscope- und Orientierungsinformationen liefert. Die detektierten elektromyographischen Signale werden klassifiziert und die resultierende Geste, welche entweder "Faust" oder "alles Andere" sein kann, wird zur EIN/AUS-Steuerung des Freihand-Elektro-Larynx verwendet. Die Orientierungsinformationen des Myo Armbandes werden zur Variation der Grundfrequenz herangezogen, das heißt, wenn der vertikale Winkel des Armes verändert wird, ändert sich auch die Grundfrequenz. Alle anfallenden Berechnungsschritte werden auf einem Raspberry Pi, einem kleinen und preiswerten Computer, durchgeführt. Auf diesem Minicomputer werden die elektromyographischen Daten klassifiziert. Wird eine Faust erkannt, wird vom Elektro-Larynx ein Anregungssignal abgegeben. Zusätzlich wird die Variation der Grundfrequenz, um natürlicher klingende Sprache zu erzeugen, evaluiert. Obwohl die Ergebnisse keine Information hinsichtlich der praktischen Anwendung liefern, kann darauf geschlossen werden, dass es möglich ist, die Grundfrequenz durch Verändern des Armwinkels zu variieren.
Das in dieser Masterarbeit vorgestellte System besteht aus dem Freihand- Elektro-Larynx, einem Vorverstärker, dem Myo Gesture Control Armband und einem Raspberry Pi. Das Hauptaugenmerk dieser Arbeit liegt auf der EIN/AUS-Steuerung des Freihand-Elektro-Larynx durch Gestenerkennung. Zu diesem Zweck wurden verschiedene Klassifizierer und der Einfluss des Benutzers hinsichtlich Trainingsdaten der Klassifizierer untersucht. Mithilfe eines speziellen Nutzertests, der von 11 Testpersonen durchgeführt wurde, konnten die EIN/AUS-Steuerung und Grundfrequenz Variation evaluiert werden. Aus den Ergebnissen dieser Arbeit kann geschlossen werden, dass eine EIN/AUS-Steuerung des Elektro-Larynx mittels Thalmic Labs Myo Armband durchaus praktikabel ist.

# Acknowledgement

First I would like to thank my thesis advisors Martin Hagmüller and especially Anna Fuchs, her door was always open when I had a question or troubles.

Another huge thank-you belongs to all the participants, who patiently took part in the general usage test.

I would also like to thank my study colleagues Lukas Knöbl and Julia Ziegerhofer who had always an open ear for me. They enabled wonderful years of study and became really good friends of mine.

Finally, I must express my very profound gratitude to my parents and my family, without their help it would not have been possible to do this. And of course, thank you to all my friends, especially Michael, Christina, Tanja, Julia and Claudio, they know why...
This accomplishment would not have been possible without them.

Thank you.

# Contents

# Glossary

**EL** electro-larynx. 1–3

**ELCS** electro-larynx-control system. xi, 2, 4, 9, 13, 16–19, 21, 28–31, 39, 40, 44, 47, 48

**EM** expectation maximization. 6

**$f_0$** fundamental frequency. vii, xi, 1, 2, 9–12, 17, 18, 20, 21, 30, 31, 43, 48, 49, 57, 58

**GMM** Gaussian Mixture Models. 2, 5, 6, 17, 19, 26, 28

**IMU** inertial measurement unit. 13

**KNN** K-Nearest-Neighbors. 2, 5, 6, 18, 19, 26, 28–30, 50

**LF** Liljencrants-Fant. 8, 17

**LSVM** Support Vector Machines using linear kernel. 5, 7, 19, 26, 28

**MPA** MPV training data of all test persons. 22, 25, 26, 32, 35, 39, 47, 48

**MPC** Training data where MYO position was constant. 22, 24, 25, 28, 32, 35, 39, 43, 47, 50

**MPV** Training data where MYO position was varied. 22–26, 32, 35, 39, 47, 48, 50

**MPW** MPV training data of all test persons without the person used in test. 22, 25, 26, 32, 35, 39, 47, 48

**MYO** Myo Gesture Control Armband. 1–4, 13, 14, 17, 18, 20, 22, 24, 25, 41, 43

**OCSVM** One Class Support Vector Machines. 5, 7, 19, 25, 26, 28, 50

**RAFO** Random Forest. 2, 5, 7, 19, 26, 28

**rbf** radial basis function. 7

**RSVM** Support Vector Machines using rbf kernel. 5, 7, 19, 26, 28

**sEMG** surface electromyography. 1–4, 12, 14, 16, 18, 19, 22, 41, 47

**SVM** Support Vector Machines. 2, 7

**TD** Training data. 2, 4, 17–19, 22–26, 30, 32, 34–36, 39, 43, 47, 48, 50

**TP** Test person. 2, 17, 22, 24–26, 28–35, 39, 43, 44, 47–50, 55, 58

# List of Figures

# List of Tables

# 1 Introduction

The first chapter of this thesis outlines the motivation for this topic, and the major tasks of it. Additionally some necessary background information about speech production, alaryngeal speech, EMG-signals, classification and the LF-model is mentioned and briefly explained. Finally a short literature review on related work is given.

## 1.1 Motivation

Speech is the most important way for verbal communication in everyday life. So much the worse it is for people who have no larynx, to manage their normal course of life. A consequence for people who suffer from cancer in the vocaltract often is a laryngectomy, which is the surgical removal of the larynx. Therefore these people need alternative ways which enable them to speak again. The electro-larynx (EL) represents such an alternative. It is a small hand-held and battery-driven device. When it is hold against the neck, its vibrations are modulated with the vocal tract and the user is able to produce speech. Thus the EL restores the function of the vocal folds. The major drawbacks of these commercially available EL devices are the inconvenience of use because one always need a hand to hold the device to the neck, and the monotonic sounding excitation signal, which results in a robotic-like speech. Most commercially available devices provide the opportunity to change between two values for the fundamental frequency ($f_0$) of the excitation signal or to fluctuate the frequency of excitation signal using a single pressure sensitive button as it is the case for TruTone$^{TM}$ Artificial Larynx. The results of a survey under EL users by [Goldstein et al., 2004] prove this inconvenience of use and the monotonic nature of the speech. In former works, a hands-free EL device was developed, which can be mounted on the neck [Fuchs, 2015]. This device serves as basis for this thesis. In previous research such hands-free EL devices are controlled by surface electromyographic signals (sEMG-signals) detected through electrodes placed on the neck [Goldstein et al., 2004], [Fuchs, 2015]. In this work, the controlling of the prototype hands-free EL is obtained by using Thalmic Labs Myo Gesture Control Armband (MYO) which is worn on the forearm. The MYO Armband has eight sEMG sensors and additionally a nine-axis inertial measurement unit containing a three-axis gyroscope, a three-axis accelerometer, and a three-axis magnetometer. In this work, the Liljencrants-Fant model (LF-model) is used as excitation signal for the EL to provide a more natural sounding speech signal. It is also possible to vary the fundamental frequency ($f_0$) via processing MYOs orientation data.

As the major task of this thesis lays on the ON- and OFF-controlling of the prototype EL

which is accomplished via gesture recognition, the sEMG data delivered by the MYO has to be classified. For this reason six different classifiers (K-Nearest-Neighbors (KNN), Support Vector Machines (SVM) with different kernels (e.g. linear or radial basis function (rbf)), One Class Support Vector Machine (OCSVM), Gaussian Mixture Models (GMM) and a Random Forest classifier (RAFO)) were built. Therefore a few considerations about training data (TD) for the classifiers in terms of their length and composition were necessary. Afterwards the classifiers were analyzed in terms of TD (i.e. length and composition of TD) and the results were evaluated.
Another question was, if it is in general possible to vary the fundamental frequency ($f_0$) of the excitation signal with the MYO Armband. There are two possibilities for managing this problem, either using sEMG data or the orientation data, which is also delivered by the MYO Armband. After trying different ways it was decided to use orientation data for variation of $f_0$ in a way that the vertical angle of the MYO and the forearm, respectively, was mapped to a frequency range which is different for males and females. Afterwards the performance of this feature was tested in general.
For the evaluation of the whole EL-control system (ELCS) the best working classifier was trained with four different sets of TD for each test person (TP). Additionally a general usage test, which should give insight in time-lags and robustness in terms of occurring errors for the individual training data (TD) sets, was conducted.

## 1.2 Background

### 1.2.1 Speech production

For the production of speech, the manipulation of an airstream is necessary. The representation of speech in an acoustical way is a sound pressure wave that originates from the physiological speech production system as it is stated in [Vary and Martin, 2006]. The main components of speech production can be seen in Figure 1.1 and their functions are:

| | |
|---|---|
| • lungs: | produce the energy, |
| • trachea: | transports the energy, |
| • larynx with vocal folds: | works as signal generator, and the |
| • vocal tract:<br>(pharynx, oral and nasal cavities) | which serves as the acoustic filter. |

In speech production, the lips and nostrils radiate an airflow modulated by the larynx and processed by the vocal tract. This airflow is produced by the lungs via contraction. The larynx provides several biological and sound production functions and in the case of speech it controls the stream of air that enters the vocal tract via the vocal folds [Vary and Martin, 2006].

### 1.2.2 Alaryngeal speech

The larynx is amongst others, which are all illustrated in Figure 1.1, the main organ of human speech production. In case of its absence there must be an alternative way that enables people to communicate again. One such alternative possibility for communication is alaryngeal speech.

As it can be read in [Goldstein et al., 2004], there are three major possibilities to produce alaryngeal speech:

- esophageal
- tracheo-esophageal

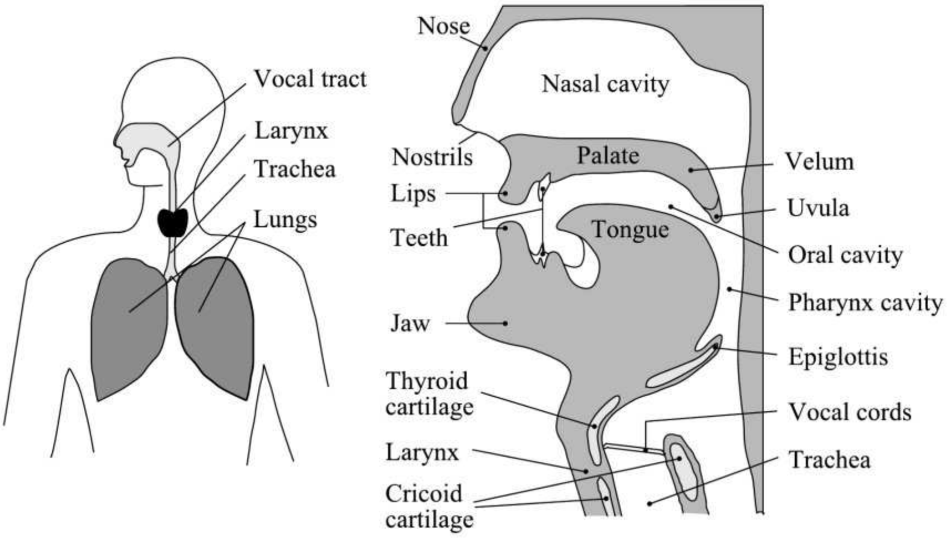


*Figure 1.1: The organs of human speech production [Vary and Martin, 2006].*

- electro-laryngeal (with electro-larynx (EL))

For producing esophageal speech, air has to be swallowed for inflating the esophagus. Then next the air is expelled to make the pharyngoesophageal sphincter vibrate. Esophageal speech sounds a bit like belching.

Tracheo-esophageal speech is produced via a one-way valve which has to be surgically implanted between trachea and esophagus. This valve drives the pharyngo-esophageal tissue for phonation.

The third method is the EL device which produces a buzzing sound that is injected through the neck tissue or the oral cavity into the vocal tract.

All of these alternatives generate acoustic energy that replaces the voice or, better expressed, the duty of the vocal folds and excite the vocal tract to produce speech.

### 1.2.3 sEMG-signals

> „EMG is a technique, which investigates the muscular contractions [...].“ [Kumar et al., 2004]
>
> „As a nerve impulse from an alpha motor neuron reaches the motor end plates of muscle fibers comprising a motor unit, the fibers innervated by that neuron discharge nearly synchronously. The electric potential field generated by the depolarization of the outer muscle- fiber membranes essentially reflects the alpha motor neuron activity; the electromyogram (EMG) is a representation of this ”myoelectricity” as summed over a number of motor units and measured at some distance. Tissues separating the EMG signal sources (depolarized zones of the muscle fibers) act like spatial low-pass filters on the potential distribution, and constitute a volume conductor. Therefore, the EMG may be measured intramuscularly or at the surface of the skin, yielding different information based on the distance of the observation site from the muscle fibers. For surface detection particularly, the effect of the separating tissues becomes significant.“ [Stepp, 2008]

sEMG-signals are detected through electrodes placed on the skin. As explained above, these electrodes measure the electrical activity produced by the muscles. In Figure 1.2 the sEMG signals detected by the MYO Armband can be seen. The numeration of the electrodes is depicted in Figure 2.1. The MYO Armband has eight electrodes which measure the muscle activity at eight different places around the forearm.

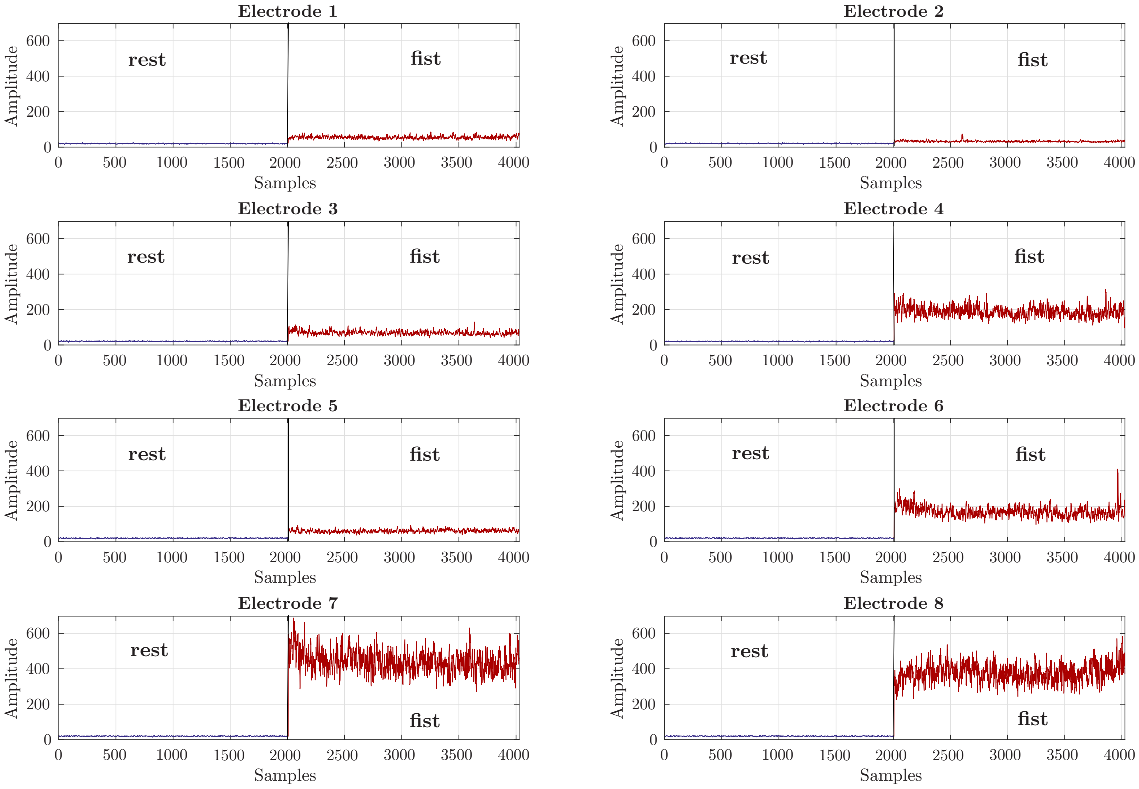


*Figure 1.2: sEMG-signals of each MYO electrode when doing a rest and a fist gesture (downward hanging arm); rest: no muscle activity measured, fist: muscle activity measured by some electrodes.*

### 1.2.4 Classification

Why is there a need for classification? The ON/OFF-control of EL-control system (ELCS) is managed through gesture recognition. It turned out that MYOs onboard gesture recognition is too slow for this application. Therefore it is necessary to classify the raw sEMG data from the MYO Armband. Using classifiers, the output of them should be the actual gesture. Two classes which are fist and garbage are defined. The fist class contains only sEMG samples for the fist gesture. The garbage class contains sEMG samples of every gesture except fist. For this, the subject has to make several gestures with the hand like spreading fingers, wave in, wave out, waving fingers and etc. during the recording of training data (TD) to cover all the different possibilities for gestures of the hand except the fist gesture. When the fist gesture is made, the ELCS is forced to turn on the excitation signal and play it as long as this gesture is made. For the whole time when garbage gestures are made (every gesture except fist), there must not be any output. We tested different classifiers and investigated their performance (see Table 1.1) for making a decision which one works best for this kind of application. Consecutive theory about classification and methods, which were evaluated in this work, will be briefly explained according to [Hastie et al., 2009], [sklearn, 2012], [Cunningham and Delany, 2007], [Statistics and Breiman, 2001], [Bishop, 2006] and [Haykin, 2009]. Concerning implementation of the different classifiers there is a *Python* package called *sklearn*[1] which has all these classifiers implemented as classes.

[1] http://scikit-learn.org/stable/

| Overview of classifiers implemented and evaluated in this thesis | |
|---|---|
| K-Nearest-Neighbors | (KNN) |
| Gaussian Mixture Models | (GMM) |
| Support Vector Machines using rbf kernel | (RSVM) |
| Support Vector Machines using linear kernel | (LSVM) |
| One Class Support Vector Machines | (OCSVM) |
| Random Forest | (RAFO) |

*Table 1.1: Implemented and evaluated classifiers.*

### Supervised learning

In [Hastie et al., 2009] for simplicity the assumption of additive errors was made and the model $Y = f(X) + \varepsilon$ was chosen as reasonable. In supervised learning it is attempted to learn $f$ by example through a teacher. The system is observed by studying the inputs and outputs (target values) and assembling a training set of observations $\mathcal{T} = (x_1, y_i, i = 1, ..., N)$. The observed input values to the system $x_i$ are also fed into an artificial system, which is a learning algorithm that is usually a computer program. This artificial system produces the outputs $\hat{f}(x_i)$ in response to the inputs. The learning algorithm has the property that it can modify its input and output relationship $\hat{f}$ in response to the differences between the original outputs and the generated ones $(y_i - \hat{f}(x_i))$. This is called learning by example. The hope is that the artificial and real outputs are close enough to be useful for all sets of inputs.

### Unsupervised learning

> „In other pattern recognition problems, the training data consists of a set of input vectors x without any corresponding target values. The goal in such unsupervised learning problems may be to discover groups of similar examples within the data, where it is called clustering, or to determine the distribution of data within the input space, known as density estimation, or to project the data from a high-dimensional space down to two or three dimensions for the purpose of visualization.“ [Bishop, 2006]

Hence, unsupervised learning or ”learning without a teacher” deals with solving a problem without the help of a teacher or supervisor. In this case there is a set of $N$ observations $(x_1, x_2, ..., x_N)$ of a random $p$-vector $X$ which has the joint density $\Pr(X)$. Now the goal is to directly infer the properties of this probability density without the help of a supervisor which provides the correct answer or degree-of-error for each observation. Sometimes the dimension of $X$ can be much higher than in supervised learning, and the properties of interest are often more complicated than simple location estimates [Hastie et al., 2009].

### K-Nearest-Neighbors (KNN)

The following subsection is based on [Hastie et al., 2009], [Cunningham and Delany, 2007] and [sklearn, 2012].

The K-Nearest-Neighbors classifier is perhaps one of the most straightforward classifiers in machine learning techniques. In nearest neighbor classification, examples are classified based on the class of their nearest neighbor. When it is more useful to take more nearest neighbors into account, this method is more commonly named K-Nearest-Neighbors (KNN). Hence, for this method the training samples are needed at run-time which means, that they need to be in memory at run-time, it is also called a memory-based classification.

KNN classification in general consists of two stages, first the $k$-nearest neighbors have to be

found and at the second they are used to determine the class of the query point [Cunningham and Delany, 2007].
The euclidean distance is mostly used as distance measure between query point and training samples. For determination of the class a straightforward approach would be to assign the majority class among the nearest neighbors to the query, it is also possible to weight the distances to achieve a distance weighted voting.

Since this classification method is memory-based, there is no need to fit a model. If there is a query point $x_0$ given, the $k$ training points $x_{(r)}, r = 1, ..., k$ closest in their distance to $x_0$ are found and it is classified among the $k$ neighbors using majority vote [Hastie et al., 2009].

KNN belongs to the category of supervised learning [sklearn, 2012].

### Gaussian-Mixture-Models (GMM)

Gaussian-Mixture-Models (GMMs) represent another method for classification. They belong to unsupervised learning and a parametric probability density function is represented as a weighted sum of Gaussian component densities. The expectation-maximization (EM) algorithm is used for fitting a mixture-of-Gaussian models. Covariance matrix of GMM's can be either spherical, diagonal, tied or full. The number of components depicts another parameter of GMM's.

In [Bishop, 2006] it is said, that the Gaussian distribution has some important analytical properties, but it suffers from significant limitations when it comes to modelling real data sets, a linear superposition of more Gaussians gives a better characterization of these data sets.
By using a sufficient number of Gaussians, and by adjusting their means and covariances as well as the coefficients in their linear combination, almost any continuous density can be approximated to arbitrary accuracy. A superposition of $K$ Gaussian densities as can be seen in 1.1

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{1.1}$$

is called a mixture of Gaussians. Each Gaussian density $\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is called a component of the mixture with its own mean $\mu_k$ and covariance $\Sigma_k$. The parameters $\pi_k$ are called mixing coefficients.

These parameters $\boldsymbol{\pi}$, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ where the notation $\boldsymbol{\pi} \equiv \{\pi_1, ..., \pi_K\}$, $\boldsymbol{\mu} \equiv \{\boldsymbol{\mu}_1, ..., \boldsymbol{\pi}_K\}$ and $\boldsymbol{\Sigma} \equiv \{\boldsymbol{\Sigma}_1, ..., \boldsymbol{\Sigma}_K\}$ is used, govern the form of the Gaussian mixture distribution. One possible way to set the values of these parameters would be to use maximum likelihood. From Equation 1.1 the log of the likelihood function is given by 1.2 where $\boldsymbol{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$. Due to the presence of the summation over $k$ inside the logarithm the situation now is much more complex than with a single Gaussian. Hence, there is no closed-form analytical solution. One approach for maximizing the likelihood function is to use iterative numerical optimization techniques or the expectation-maximization (EM) algorithm. The EM algorithm is an powerful and elegant method for finding the maximum likelihood solutions for models with latent variables. Given a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters (comprising the means and covariances of the components and the mixing coefficients) [Bishop, 2006].

$$\ln p(\boldsymbol{X} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} \tag{1.2}$$

### Support Vector Machines SVM

As it is written in [Haykin, 2009] the support vector machine (SVM) is basically a binary learning machine with some highly elegant properties. The main idea behind this machine can be summed up the following way:

> „Given a training sample, the support vector machine constructs a hyperplane as the decision surface in such a way that the margin of separation between positive and negative examples is maximized.“ [Haykin, 2009]

SVMs can be used for both, classification and regression and form a part of supervised learning. The inner-product kernel between a support vector $\boldsymbol{x}_i$ and a vector $\boldsymbol{x}$ which is drawn from the input data space is central to the support vector learning algorithm. The support vectors consist of a small subset of data points which are extracted by the learning algorithm from the training sample itself.
SVMs are effective in high dimensional spaces. Different kernel functions can be used for the decision function. The two different kernel functions used in this work are a linear kernel where the classifier is referenced with LSVM and a radial basis function (rbf) as kernel where the classifier is referenced with RSVM.
A submethod of SVMs is the one class SVM (OCSVM) which is an unsupervised outlier detection. OCSVM can be used for novelty detection, that means that, given a set of samples, it detects the soft boundary of that set and classifies new points as belonging to that set or not. In this case it is a type of unsupervised learning as the fit method of *sklearn* only takes an array as input but no class labels [sklearn, 2012].

### Random Forest (RAFO)

Random forests are part of supervised learning. As written in [Statistics and Breiman, 2001] Random forests are a combination of tree predictors. Each tree is dependent on the values of a random vector which is independently sampled and has the same distribution for all trees in the forest. The nature and dimensionality of $\Theta$ depends on its use in tree construction.

> „For instance, in bagging the random vector $\Theta$ is generated as the counts in $N$ boxes resulting from $N$ darts thrown at random at the boxes, where $N$ is number of examples in the training set.“ [Statistics and Breiman, 2001]

The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them and it converges almost surely to a limit as the number of trees in the forest becomes large.
A random vector $\Theta_k$, which is independent of the past random vectors $\Theta_1, ..., \Theta_{k-1}$ but with the same distribution, is generated for the $k$th tree. A tree is grown using the training set and $\Theta_k$ which results in a classifier $h(\boldsymbol{x}, \Theta_k)$ where $\boldsymbol{x}$ is an input vector.
In order to reduce the correlation between the trees without increasing the variance too much, a random selection of the input variables is used in the tree-growing process. Specifically when growing a tree on a bootstrapped dataset, before each split $m \leq p$ input variables, where $p$ is the number of input variables, are selected at random as candidates for splitting [Hastie et al., 2009].

Definition: A random forest is a classifier consisting of a collection of tree-structured classifiers $\{h(\boldsymbol{x}, \Theta_k), k = 1, ...\}$ where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input $\boldsymbol{x}$ [Statistics and Breiman, 2001].

### 1.2.5 Liljencrants-Fant-model (LF-model)

The motivation for using the LF-model was to gain a more natural sounding excitation signal by modelling the glottal flow. In the work of [Fuchs, 2015] this model was also recommended, as it outperforms other excitation signals in terms of signal to noise ratio. In 1985 Gunnar Fant, Johan Liljencrants and Qi-guang Lin developed the LF-model [Fant et al., 1985] out of Liljencrants' L-model and Fants' F-model. The LF-model describes the glottal flow and is optimal for non-interactive flow parameterization. In Figure 1.3 and 1.4 one can see the first derivative of the LF-model from the glottal flow (see Formula 1.3).

$$\dot{U}_g(t) = \begin{cases} E_0 e^{\alpha t} \sin(\omega_g t) & \text{for } 0 \leq t \leq t_e \\ \frac{-E_e}{\varepsilon t_a} \cdot [e^{-\varepsilon(t-t_e)} - e^{-\varepsilon(T_0-t_e)}] & \text{for } t_e < t < T_0 \end{cases} \tag{1.3}$$

with

| | |
|---|---|
| $U_g(t)$ | glottal flow |
| $\dot{U}_g(t)$ | differentiated glottal flow |
| $t_p$ | rise-time (from glottal opening to maximum volume velocity) |
| $t_e$ | time instant when vocal folds close |
| $t_a$ | time instant of the exponential return phase<br>extrapolation of the derivative's tangent onto the time-axis at time instant $t_e$ |
| $E_e$ | maximum negative amplitude in the differentiated glottal flow |
| $T_0$ | Fundamental period of LF-model |
| $E_0$ | scale factor |
| $\alpha = B\pi$ | |
| $B$ | bandwidth of the exponentially growing amplitude |
| $\omega_g = 2\pi F_g$ | |
| $F_g = 1/(2t_p)$ | |
| $\varepsilon \approx 1/t_a$ | for small $t_a$ |
| when $t = t_e$ | $\varepsilon t_a = 1 - e^{\varepsilon(T_0 - t_e)}$ |

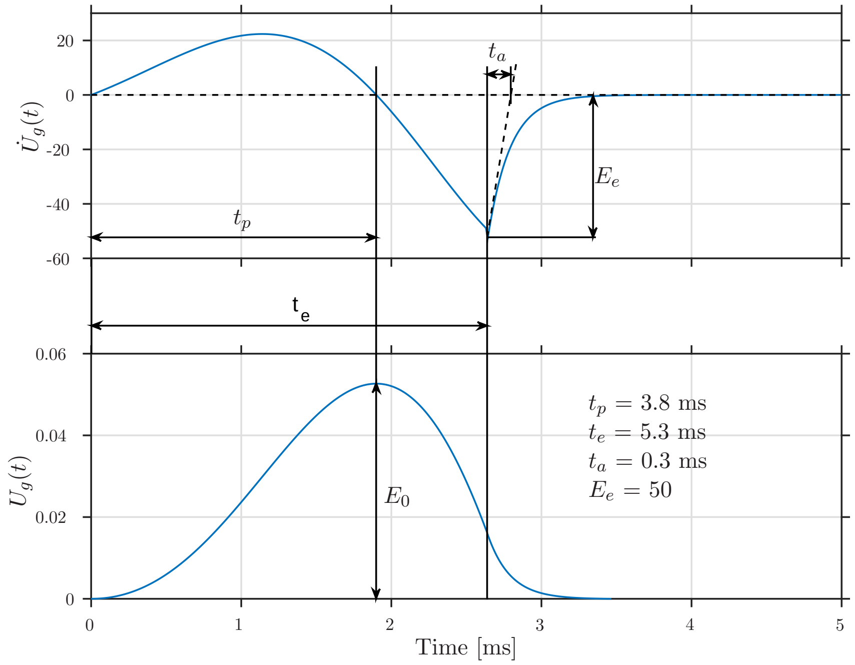


*Figure 1.3: Waveforms generated by the Liljencrants-Fant model, with the four wave-shape parameters: $t_p$ instant of maximum glottal flow, $t_e$ instant of flow derivative negative peak, $t_a$ time of exponential closure, and $E_e$ absolute value of glottal flow derivative at $t_e$ (based on [Veldhuis, 1998]).*

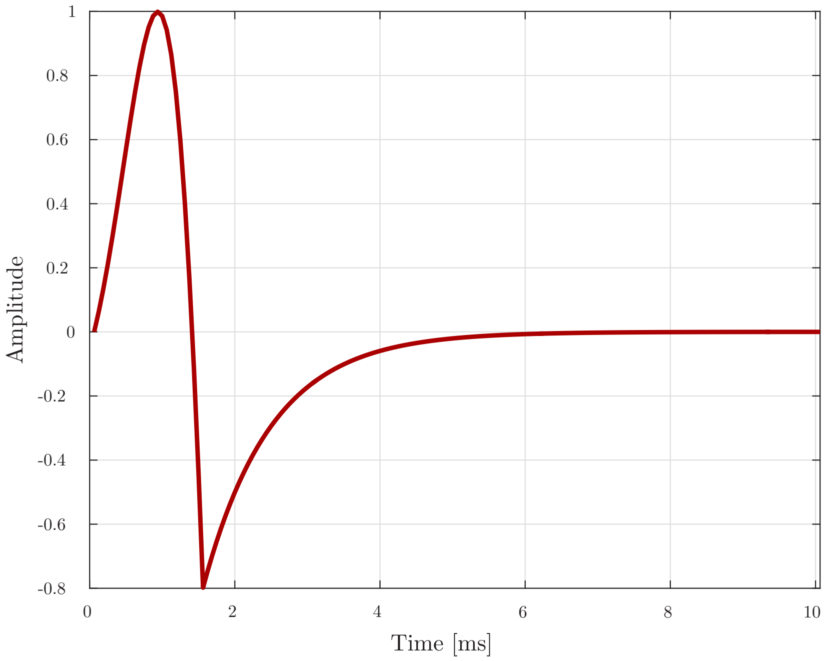


*Figure 1.4: First derivative of the LF-model for 100 Hz, used as one period in a 100 Hz excitation signal for the ELCS.*

## 1.3 Literature review

In this chapter relevant papers dealing with EL, ON/OFF control and $f_0$ variation will be be discussed and methods for controlling an EL will be explained. Some of these works are using a hands-free EL device.

In [Kikuchi and Kasuya, 2004] the developement and evaluation of a pitch adjustable EL is explained. The main goal was to control pitch and vibration ON/OFF with the thumb as it can move freely up and down or left and right. They developed two different mechanisms. In the first mechanism pitch period is controlled by the up-down displacement of the finger and vibration ON/OFF is controlled by the left-right movement. In the other method pitch period is controlled by the left-right displacement amount of the finger and vibration ON/OFF by the up-down movement. These adjustment mechanisms were connected to a personal computer via an interface and the pitch was controlled by a control program.

> „To investigate the ease of operation of the prototype electrolarynx, a learning test was performed by healthy subjects. The aim of the test was to copy the pitch variation of various utterances which were made previously. In the test, it was not compulsory to place the electrolarynx in contact with the throat. Measurements were performed after 30 minutes practice.“ [Kikuchi and Kasuya, 2004]

The subjects of their experiments were five female students which intended to become speech therapists, and although they had knowledge of speech production, they were not familiar with the operation of the EL. The test sentences were conversation sentences which are used in daily life. For investigating the features of this EL they used question sentences, where utterances could be started or finished with a desired pitch. In the results of this paper it was obvious, that the absolute magnitude and variation range of the pitch for the subjects was different from the utterances, but similar pitch variations could be produced. From the tests they performed it turned out that for four of these five subjects the left-right pitch control mechanism was easier to use. As a previous research reported that an ON/OFF-control of the vibration for consonant parts of an utterance improved naturalness of the utterance, they tested if a simultaneous ON/OFF-control of the vibration and pitch change could be attained with the proposed left-right pitch control mechanism. It came out that although the pitch adjustment operation was somewhat sluggish, the control went well.

> „It was found that operations relating to pitch control were learnt in a short time of about 30 minutes. By making pitch adjustment independent from vibration ON/OFF commands, the pitch of the last phrase could easily be made to rise, as in a question sentence. It was also found that four out of five subjects considered pitch adjustments in the left-right direction were easier to perform.“ [Kikuchi and Kasuya, 2004]

There are many publications on using surface electromyography signals for controlling the EL. One of these publications is [Heaton et al., 2011] which describes the developement of a wireless electromyographically controlled electrolarynx voice prosthesis. In their work they searched for the best position for the electrode, then the EMG activity was band-pass filtered between 10 Hz and 500 Hz, amplified, rectified and low-pass filtered along two parallel pathways. In each pathway they tracked the envelope, one for “slow” (1 Hz corner frequency) and one for “fast” (1-9 Hz corner frequency) time constraints as in [Goldstein et al., 2004]. These envelopes were used for controlling the fundamental pitch and ON/OFF of the EL device. As EL they used a device from TrueTone$^{TM}$ which has a microprocessor and a radio frequency transceiver. An important aspect of their work was that the device can be used in the “automatic” way and in the “manual” way (push-button). So they were able to make certain control combinations like for example ON/OFF with push-button control and $f_0$ modulation with EMG envelope and of course other combinations.

> „In preliminary recordings of surface EMG signals in an anatomically intact adult male, placement of the sensor on the neck surface above the thyroid cartilage of the larynx or under the chin produced envelopes that closely corresponded to voice onset/offset (...), and were on average >360% of baseline for individual words, and >380% of baseline for running speech for both sensor locations. Vocal-related signals obtained with the sensor placed on the face superficial to peri-oral musculature (such as the depressor anguli oris and orbicularis oris) were substantially stronger (twice or more) than the neck or submental locations, but face placement was also prone to frequent non-speech-related signals associated with facial expression.“ [Heaton et al., 2011]

> „Upcoming experiments with laryngectomee participants will compare EL voice timing control, serial speech capability, speech naturalness, and intonational control using several different EMG-EL control modes ranging from completely manual to completely automatic. Naive listeners will then judge the naturalness and intelligibility of recorded EMG-EL speech samples produced under these different control modes. We will also obtain feedback from EMG-EL users with formal questionnaires concerning speech quality and device form/function to help guide the final steps toward commercial availability of the EMG-EL system.“ [Heaton et al., 2011]

The work of [Uemi et al., 1994] explains the design and evaluation of an EL having a pitch control function. To improve the naturalness of EL speech they designed a new method for controlling voice intonation by using expiration. First they investigated the ability of pitch control of 16 laryngectomees. Two out of these 16 paticipants were able to control the pitch frequency accurately and this was improved after one week of training. After that they designed a new EL were the pitch can be controlled over the expiration pressure. This new EL device consists of three parts.

> „The first part is a pressure sensor that can detect expiration pressure produced from a stoma made by a surgical incision into the neck for pulmonary respiration. The second part is an electrical circuit that can convert air pressure into a pitch frequency for voice. The third part is an electromechanical vibrator that can be attached to the neck.“ [Uemi et al., 1994]

In order to find the optimal function for calculating the frequency out of the expiration pressure they made some experiments and determined the optimal parameter.

> „The purpose of this paper is to design a practical electrolarynx that can produce the natural intonation. First, we investigated the ability of pitch control using 16 laryngectomees. From the results, two of the laryngectomees could control the pitch frequency. The reason why they succeeded might be that they have been using their expiration for substitute speech. However, the ability of pitch control using expiration improved after a short period of training. Next, we designed a new electrolarynx having a pitch control function based on the above experiment. From the view point of naturalness of voice and easiness to control of this electrolarynx, the optimal transform function was found to be F = 60 + 25 x (P-1) (where F means the pitch frequency in Hz and P means the expiration pressure in cmH2O).“ [Uemi et al., 1994]

[Hashiba et al., 2007] developed and evaluated a wearable EL for laryngectomees. This wearable EL consists of three parts: a thin-vibrator mounted to a thermo-plastic brace, a wireless ON/OFF switch and a pocked-sized controller. The ring-like thermo-plastic brace was used due to its properties to attach the thin-vibrator to the optimal place on the neck.
For the evaluation of the prototype wearable EL quality tests of larynx voices and usability tests were performed by a 72 years old subject who had undergone a laryngectomy and is an excellent electrolaryngeal speaker. The subject was asked to produce an utterance facing frontward and next the subject had to produce the same utterance facing leftward, rightward, upward and downward without moving shoulders. It came out that there are leakage noises when the subject spoke rightward, upward or downward. To eliminate this leakage noise they used stretchable fastener tape in addition to the thermo-plastic brace and it turned out that the slight leakage noise only was observed in the upward direction.
To investigate the advantages of such a wearable EL they did some usability test with the same subject. One task was a telephone conversation where the subject had to make handwritten notes. With a conventional EL the conversation was frequently interrupted because the subject had to put away the receiver when writing. With the prototype wearable EL there was no problem. Another task was a chalk talk lecture using a microphone, again same results.

> „Through these evaluations, it was confirmed that their daily activity significantly improves by making the electro-larynx wearable. According to the subject’s report, the thermo-plastic brace is sufficient for their daily communication, because they usually speak in the positions facing leftward or rightward, not upward or downward.“ [Hashiba et al., 2007]

The following paper [Matsui et al., 2014] explores the feasibility of using a motion sensor to replace the user interface of a conventional EL device. They used a mobile phone motion sensor with multi-agent platform to investigate ON/OFF and pitch frequency control capability. Further a small battery operated ARM-based control unit was developed to evaluate the motion sensor based user-interface. They placed the control unit on the wrist and the vibration device on the throat using support bandage. In this article two different methods for converting the forearm tilt angle to pitch frequency are explained: a linear mapping method and a $f_0$ template-based method. For the perceptual evaluation of this system they had two well-trained normal speakers and ten subjects. They let each speaker read some phonetically balanced test material (10 sentences) and used therefor one commercially available EL device, their linear mapping method and their $f_0$ template-based method. So they had 60 stimuli (2 speakers $\cdot$ 3 devices $\cdot$ 10 sentences) recorded. Then they prepared two sets of differently randomized stimuli where five subjects evaluated one set of stimuli and another five subjects rated the other set of stimuli. They presented each speech stimulus two times. The speech stimuli were rated in terms of “intelligibility (clarity)”, “naturalness of the prosody” and “stability of the prosody” by a five level scaling. After this subjective evaluation it turned out that both of their methods obtained higher naturalness scores than the conventional EL device. On the other hand there was no mentionable difference in intelligibility and stability among those three devices.

To our knowledge nobody ever used gesture recognition using sEMG-signals.

# 2 Setup

This chapter introduces all the important hardware parts of the ELCS such as the MYO Armband, the Rasperry Pi and the prototype EL and explains them briefly. Finally the structure of the entire ELCS is shown.

## 2.1 Myo Gesture Control Armband

The Myo Gesture Control Armband[2] (MYO) was developed by Thalmic Labs[3] additionally to its eight EMG sensors, it has a nine-axis inertial measurement unit (IMU). Whereas the EMG sensors simply detect the electrical activity of the muscles around the forearm, the IMU delivers orientation data using accelerometer, gyroscope and magnetometer. The MYO armband is compatible with Windows, iOS and Android, the communication between armband and interface is managed via Bluetooth. Supplementary MYO has an on board gesture recognition which is able to distinguish between rest, fist, spread fingers, wave left, wave right and tip middle finger to thumb gestures. In Figure 2.1 one can see the MYO and its usage. For the ELCS it is necessary to wear the MYO on the left arm with the USB port showing in the direction of the hand.

Up to now there are many applications for the MYO like the usage for presentations where one can go forward in the slides by a simple wave right gesture, or the controlling of Adobe Acrobat Reader and lots more. There are also researches where the MYO Armband is used for controlling the prosthetic hand of an amputee. The field of application is very wide-ranging and as there is a huge developer forum for interested people, this field is growing every day.

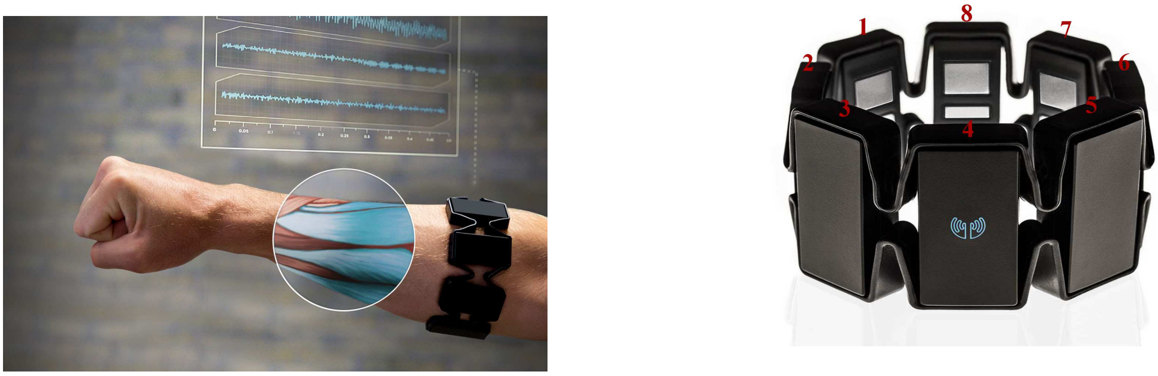
Technical specifications of the MYO Armband:

- ARM Cortex M4 Processor
- 8 medical grade stainless steel EMG (electromyography) sensors
- 1 nine-axis IMU containing three-axis gyroscope, three-axis accelerometer, three-axis magnetometer
- dual indicator LEDs

[2] https://www.myo.com/

[3] https://www.thalmic.com/

- micro-USB charging
- EMG data: 200 Hz streaming rate
- motion data: 50 Hz streaming rate
- built-in rechargeable lithium ion battery
- one full day use out of single charge



(a) Usage (worn on forearm)

(b) Overview and nomination of electrodes

*Figure 2.1: Thalmic Labs Myo Gesture Control Armband as it is used in this work with its usage on the forearm and the nomination of the electrodes.*

## 2.2 Raspberry Pi

For this work a Raspberry Pi 2 Model B (see Figure 2.2) is used for communication with the MYO Armband, implementation of excitation signal generation and classification of the sEMG signals. The Raspberry Pi is a credit card-sized single-board computer with a 900 MHz quad-core ARM Cortex-A7 CPU and 1 GB RAM. A benefit of the Raspberry Pi is that it can deal with Python which is the used programming language for this thesis. For the EL-control system the Raspberry Pi is used in connection with Griffin's iMic USB sound interface.

(a) With case

(b) Without case

*Figure 2.2: Raspberry Pi 2 Model B as it is used in this work, shown with and without case.*

## 2.3 Prototype EL

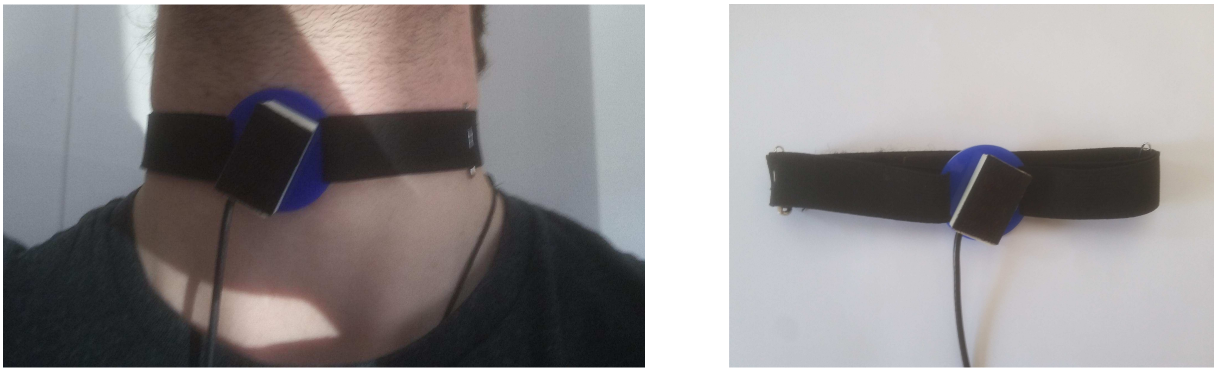
The hands-free prototype EL is based on the previous work from [Fuchs, 2015] (see Figure 2.3). A great advantage compared to commercially available EL devices is that it can be fixed on the neck through an elastic strap. For this reason no hand is needed for holding the device which leads to an enormous relief for patients dependent on such devices. The hands-free EL device has to be used in connection with a pre-amplifier.
The advantages of the used transducer are:

- Compared to state-of-the-art systems this transducer is more efficient and has therefore a lower power consumption.
- Its smaller dimensions allow the design of a wearable, hands-free device.
- Besides, it offers the possibility to playback an arbitrary excitation signal which can change the shape and frequency.

Nevertheless there are also some disadvantages:

- The efficiency decreases for higher frequencies.
- Compared to an electro-dynamic transducer it has a lower displacement.
- It has a high production of distortions.

High production of distortion is not necessarily a drawback, because non-linear distortion can cause harmonics which can support the transduction through the neck tissue.



(a) Usage (worn on the neck)

(b) Overview

*Figure 2.3: Hands-free prototype EL that is used in this work.*

## 2.4 Entire electro-larynx-control system

In the block diagram below (Figure 2.4) the entire system of the EL-control system (ELCS) is shown. When the system is started, a connection between Raspberry Pi and MYO is built. The MYO Armband detects the sEMG-signals on the forearm and sends them in addition to the IMU data via a bluetooth connection to the Raspberry Pi where a *Python* program classifies the sEMG data and calculates the actual gesture, which is either fist or garbage (everything else but fist). Out of the IMU data another program computes the vertical angle of the forearm and subsequently the fundamental frequency ($f_0$) for the excitation signal. The excitation signal, based on the LF model, is calculated for the appropriate $f_0$ and expensed over the iMic USB sound interface. From there it gets amplified via the pre-amplifier and in the end it is inducted into the vocal tract by the hands-free EL device.

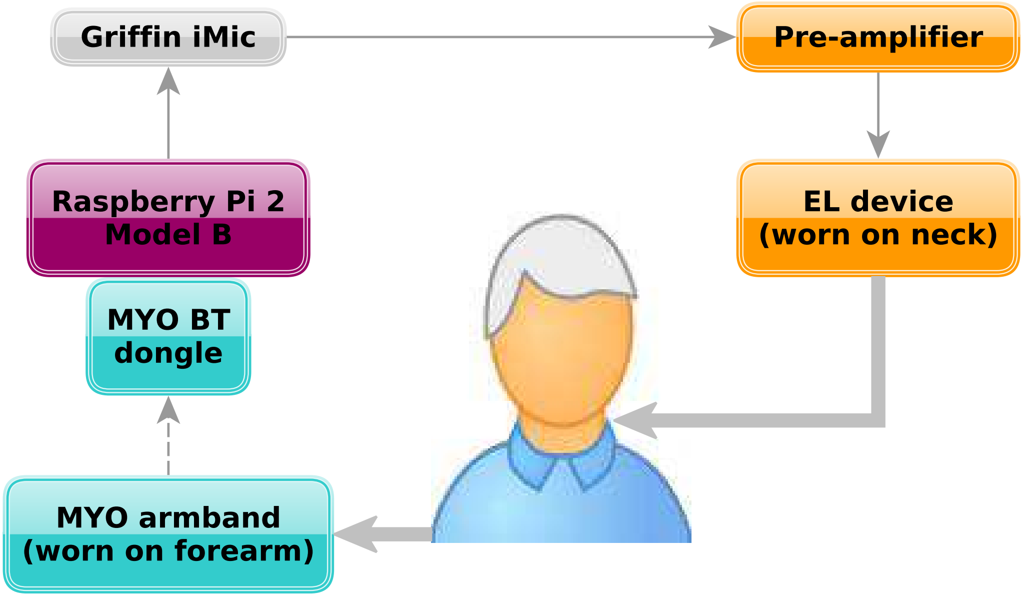


*Figure 2.4: Block diagram of the entire EL-control system (ELCS) with all its components.*

# 3 Software

This chapter of the thesis should give some insight in the software layer of the ELCS. It shortly clarifies how the MYO communicates with the Raspberry Pi and explains all the other *Python* programs involved in the ELCS like the classifying, main program, generation of excitation signal, and how they all interact.

## 3.1 General overview on the single programs

The used programming language for this thesis is *Python-2.7*.
In the following a general overview on the single software components of ELCS is given and can be seen in Figure 3.1. When ELCS is started for the first time with new TD, the used classifier is trained. The parameters for the classifier, like means and covariances in the case of GMM, are stored after training, so there is no need for new training when the system gets started again. *myo-raw-master* handles the MYO data and remits it to the main program (*el_control.py*). Furthermore the main program tells *el_play_lf_res.py* whether it should generate an output or not and which $f_0$ the excitation signal must have.

The ELCSs software consists of following parts:

- ***myo-raw-master*** project: from Danny Zhu, Provides an interface between MYO and *Linux*.
- ***myo_cls.py***: Modification of the *myo.py* file contained in *myo-raw-masters* for more classifiers.
- ***el_play_lf_res.py***: Generation of LF excitation signal.
- ***el_control.py***: Main program, contains all adjustable settings like male or female, type of TD set, functions for $f_0$ variation and tells *el_play_lf_res.py* whether to produce an axcitation signal or not.
- Training data pool: For classifier training, contains different types of TD for theTPs.

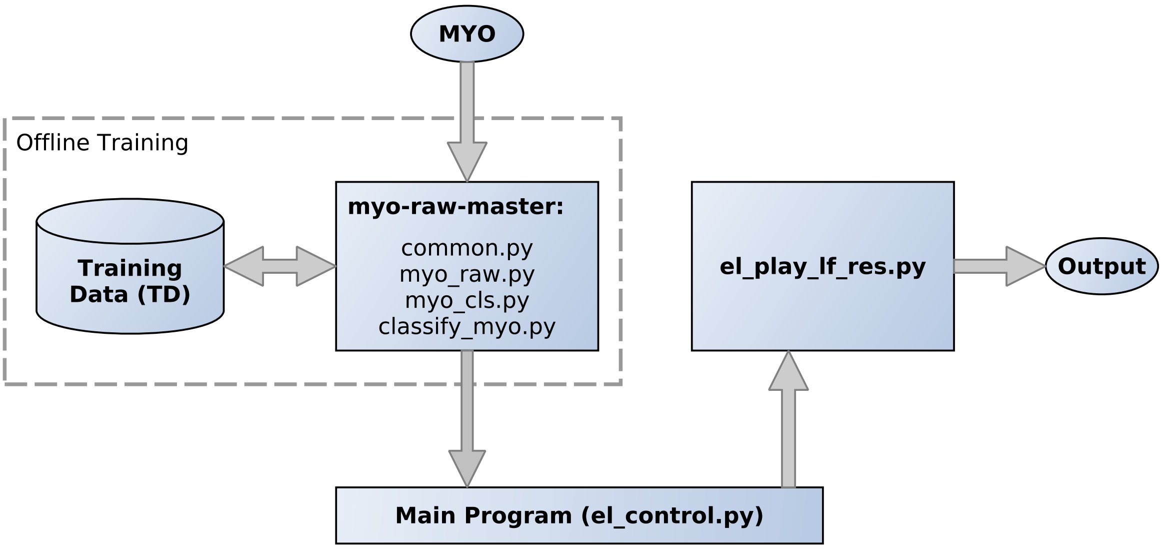


*Figure 3.1: Program flow of ELCS, myo-raw-master handles all the MYO data and remits it to main program which tells the output generation program whether to produce an excitation signal or not and which $f_0$ the signal should have.*

## 3.2 Myo interface (myo-raw-master)

First goal was to get the MYO Armband running on *Linux* because until now it supports only *Windows*, *Android* and *MacOS*. A *Python* script written by Danny Zhu[4] was found which builds a connection from the MYO to a *Linux* platform via *Bluetooth*. With this program it is possible to get access to all MYO data. Additionally a script file is included to record individual gestures (*classify_myo.py*) and classify them with a simple KNN classifier (*myo.py*).

The project from Danny Zhu found on github is called *myo-raw-master* and includes the following files:

- *common.py*: contains some common definitions
- *myo_raw.py*: contains the communication protocol
- *myo.py*: contains a KNN classifier and a class that handles self classified gestures
- *classify_myo.py*: for recording of training data (TD)

*myo_raw.py* contains a class called MyoRaw, which implements the communication protocol with the MYO Armband. Additionally it gives access to poses from MYO onboard gesture recognition and to raw sEMG- and orientation data from MYO at a rate of 50 Hz.

*classify_myo.py* is used for recording TD. When starting the program one can see a *Python pygame* window with one line for each gesture. The sEMG readings are stored as long a number key is held down. There are 10 gestures possible to record (0-9). Once there is TD recorded one can see a histogram for the actual gesture and the number of recorded samples for each gesture. Each set of TD has the same structure and for the ELCS only two of these ten possible gestures are recorded:

- vals0.dat consists of sEMG samples for the garbage gesture

[4] https://github.com/dzhu/myo-raw

- vals1.dat consists of sEMG samples for the fist gesture

- vals2.dat - vals9.dat are empty files

After recording TD with *classify_myo.py*, the Myo class in *myo.py* can be used to notify a program, which is in this case the main program (*el_control.py*), each time a self classified pose starts. The most common classification among the last 25 sEMG samples is taken as the program's best estimate of the current pose.

## 3.3 Classification of sEMG-signals

*myo-raw-masters myo.py* works with a KNN classifier. After a few considerations it was decided to try different classifiers to see which works best. In *myo.py* the most common classification among the last 25 sEMG samples is taken as the program's best estimate of the current pose. Now for the ELCS this is changed to a value of 6 to maintain a faster pose result. The new file is called *myo_cls.py.* The six different classification methods KNN, GMM, RSVM, LSVM, OCSVM and RAFO were implemented out of the box using the *Python* package *sklearn.* Figure 3.2 shows the sEMG pattern for garbage and fist gesture.

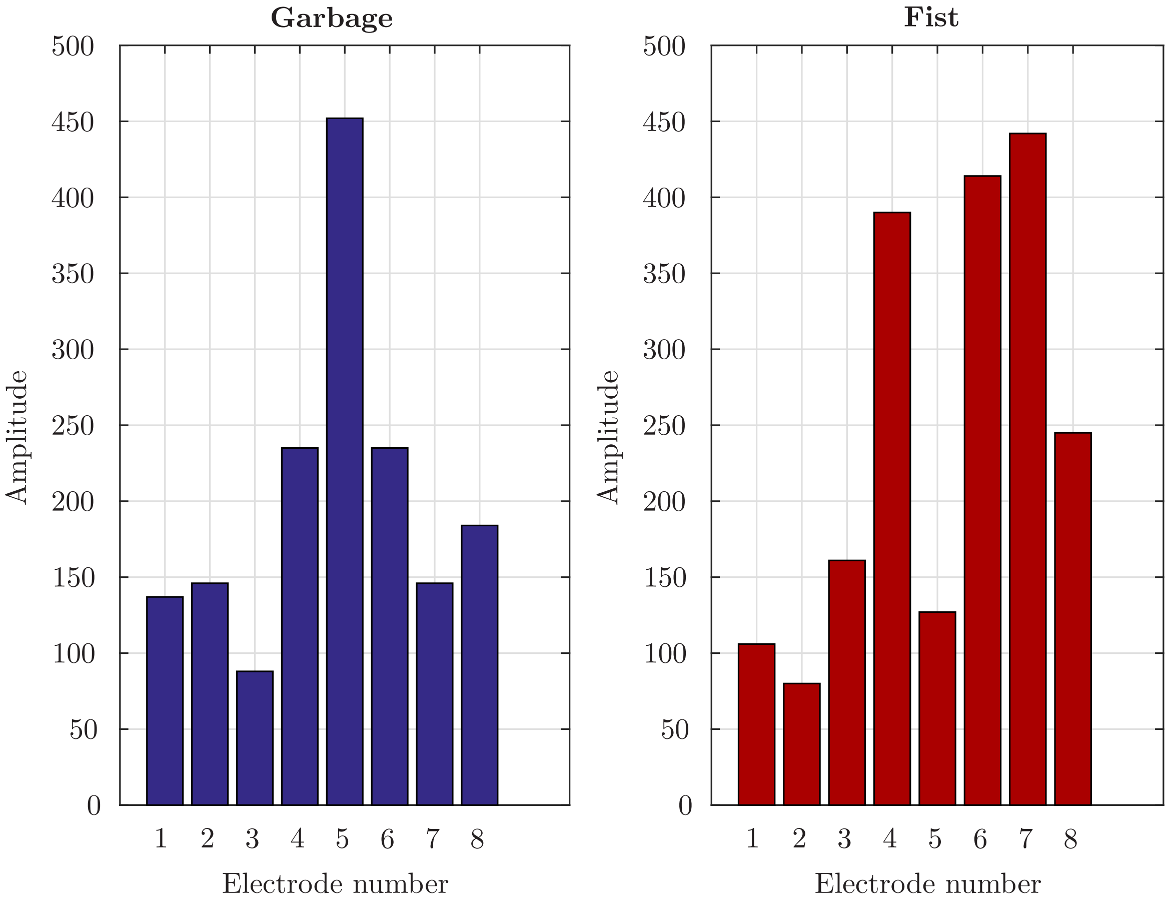


*Figure 3.2: EMG pattern for garbage and fist gesture (median over 5000 Samples for each class).*

## 3.4 EL-control main program

The main program (*el_control.py*) handles all the data it gets from the MYO Armband over *myo-raw-master*. There is a binary output (fist/no fist (garbage)) which is responsible for the ON/OFF decision of the device. Thus, it turns the excitation signal from the output function to ON when a fist gesture is recognized or to OFF when no fist gesture is recognized. Additionally it handles different functions for the $f_0$ variation:

- Method A: $f_0$ calculation due to vertical arm angle (see Listing 3.1 and 3.2)
- Method B: $f_0$ calculation due to strenght of fist gesture (see Listing 3.1 and 3.3)

Informal tests showed that $f_0$ variation by using the orientation data of MYO Armband (method A) is more practicable for this application. In Figure A.10 $f_0$ versus arm angle is shown for males and females. For this purpose the angle of the arm was mapped to a frequency range using a logarithmic function.

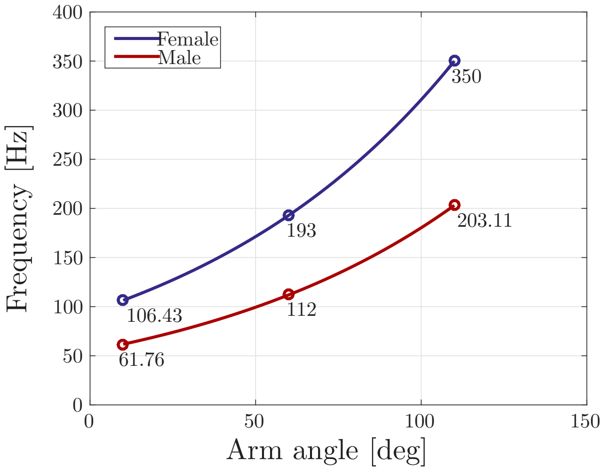


*Figure 3.3: $f_0$ versus arm angle for female and male frequency ranges. A logarithmic function was used for mapping the arm angle to a frequency.*

```
# Functions for calculating F0
class F0():
  def __init__(self, gender, sig):
    self.freq = None
    self.gen = gender
    self.s = sig
    self.fm_m = 112.0        # average f0 for male (TI projekt Julia und Klaus)
    self.fm_f = 193.0        # average f0 for female(TI projekt Julia und Klaus)
    self.fmin = 50.0         # (free chosen)
    self.fmax = 350.0        # (because of LF model!!!)
    self.max_angle = 20.0    # maximum arm angle  0 < angle <= 90
    self.min_angle = -80.0   # minimum arm angle  -90 <= angle < 0
    self.set_param()

  def set_param(self):
    self.v = self.fmax/self.fm_f   # ratio fmax/fm_f log function
    if self.gen == ’f’:
      self.fm_log = self.fm_f
      self.fl_log = self.fm_f/self.v # minimum f0 female log function
      self.fh_log = self.fm_f*self.v # maximum f0 female log function
    elif self.gen == ’m’:
      self.fm_log = self.fm_m
      self.fl_log = self.fm_m/self.v # minimum f0 male log function
      self.fh_log = self.fm_m*self.v # maximum f0 male log function
    else:
      print(’Chose either f for female or m for male!’)
      raise ValueError
```

*Listing 3.1: $f_0$ settings*

```
def f0_imu_arm_up_down_log_range(self, in_data):
  x = in_data[1]
  angle = x + 90.0    # vertical angle given from MYO (horizontal is 0 degrees)
  lb = self.min_angle + 90.0
  ub = self.max_angle + 90.0
  self.lam = np.log(self.fh_log/float(self.fl_log))/(ub-lb)
  if lb <= angle <= ub:
    f0 = self.fl_log * np.exp(self.lam*(angle-lb))
  elif angle < lb:
    f0 = self.fl_log * np.exp(self.lam*(0))
  elif angle > ub:
    f0 = self.fl_log * np.exp(self.lam*(ub-lb))
  self.s.set_f0(f0)
```

*Listing 3.2: $f_0$ variation via IMU, vertical arm position is mapped to logarithmic $f_0$ range*

```
def f0_emg_fist_strength(self, in_data):
  min_mean = 50
  max_mean = 700
  #print(in_data)
  mean_sens = np.mean(in_data)
  #print(mean_sens)
  #f0 = f0_min + ((f0_max-f0_min)/(s.ub-s.lb))*(mean_sens-s.lb)
  f0 = self.fmin + ((self.fmax - self.fmin)/(max_mean-min_mean))*(mean_sens-min_mean)
  self.s.set_f0(f0)
  #print(self.s.f0)
```

*Listing 3.3: $f_0$ variation via sEMG-signals, mean of sEMG samples is mapped to $f_0$ range*

## 3.5 Generation of the excitation signal

In order to get an excitation signal for the EL when a certain gesture is made, it was necessary to develop a program which plays a specific output when the ON-gesture is made and stops this output immediately when OFF-gestures are made.

It was decided to construct a signal according to the LF-model [Fant et al., 1985]. The *Matlab* function *glotlf.m* from the Voice Toolbox served as a guidline.

The output generating function (*el_play_lf_res.py*) calculates a prototype period of the LF-model, then this prototype period is convoluted with an impulse train. The prototype LF-impulse is always resampled according to the desired frequency and the impulse train is always generated for the appropriate $f_0$. This happens in a blockwise way. In Figure 3.4 one can see the excitation signal for a frequency of 100 Hz.

- sampling frequency = 16000 Hz
- block length of output blocks is 512 samples → 32 ms
- possible range for $f_0$ when male user and logarithmic $f_0$-function: 61.76 Hz - 203.11 Hz
- possible range for $f_0$ when female user and logarithmic $f_0$: 106.43 Hz - 350 Hz

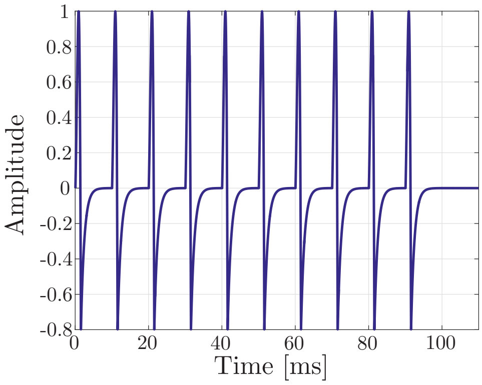


*Figure 3.4: 100 Hz excitation signal for ELCS based on LF-model.*

# 4 Training data and classifier

The following chapter describes essential steps for classification of the data delivered via MYO Armband. It covers all the steps needed to build up appropriate sets of TD and starts with general considerations. Finally a first evaluation of TD with artificial test data through cross-validation and an evaluation of the different classifiers is made.

## 4.1 General considerations

Training data (TD) is recorded with *Python* using the *Python* program *classify_myo.py* written by Danny Zhu as described in Chapter 3.2.

For examination of the general usage test there are four different sets of TD for each test person (TP) which will be explained in the following.

- TD from one TP where the position of the MYO Armband was constant during recording (MPC).
- TD from one TP where the position of the MYO Armband was varied after every 1000 samples of sEMG readings (MPV).
- MPV TD sets from all TPs appended together (MPA). This set is equal for every TP.
- MPV TD sets from all TPs except the actual TP (MPW).

## 4.2 Length of training data

In order to decide how many samples of TD are needed for appropriate classification, at first some prototype sets of MPC and MPV data were recorded for one TP and a comparison between these sets was made.

For MPC data 10 sets, where each of them contains 1000 samples for garbage gesture and 1000 samples for fist gesture, were recorded. For MPV data the same was done, but the MYO position was always varied a bit for every of the ten sets. Therefor the Armband was taken off and on again to simulate everyday life usage. The next step was to append the MPC data sets to get new TD sets with different lengths between 1000 samples and 10000 samples for each gesture.

That was done in a $k$-permutations of $n$ way $\binom{n}{k}$ where $n$ means the number of data sets (10) and $k = 1, ..., K$ means the number of how much data sets are appended. $K = 10$ denotes the maximum number of TD sets appended. After that there where $(2^K) - 1$ different data sets with length of 1000 samples to 10000 samples with a stepsize of 1000 for each gesture. Again the same steps were executed for MPV data. Table 4.1 illsutrates the different TD sets.

| **Training data sets** | | |
|---|---|---|
| Number of TD sets appended ($k$) | length per class in [Samples] | Number of new TD sets |
| 1 | 1000 | 10 |
| 2 | 2000 | 45 |
| 3 | 3000 | 120 |
| 4 | 4000 | 210 |
| 5 | 5000 | 252 |
| 6 | 6000 | 210 |
| 7 | 7000 | 120 |
| 8 | 8000 | 45 |
| 9 | 9000 | 10 |
| 10 | 10000 | 1 |
| Σ | | 1023 |

*Table 4.1: Number of prototype training data sets with different lengths after appending them in a k-permutations of n way.*

For all these 1023 data sets with different lengths from 1000 samples to 10000 samples for each class, a classification using all six classifiers was made with a test data set. The test data set contains 5017 samples for garbage and 5016 samples for fist gesture. Each gesture was recorded in one run. It is important to note that the test data set is completely artificial as it was recorded the same way as TD. On that account the test data set contains samples for fist and garbage class but it does not involve the transition between these two gestures. The statistical performance measures for every data set and every classifier were calculated and written in a text file for further visualizations in *Matlab*. The median over the performance measures for the sets with same lengths was calculated separately for each classifiers classification result. For example there were 10 different sets of performance measures for each classifier for the TD with length of 1000 samples for each gesture and over these 10 values the median was calculated. As it can be seen in Figure 4.1 for the KNN classifier the medians of the performance measures converge to a nearly constant value when TD length is about 5000 samples for each class. The same result can be recognized for all other classifiers see Appendix A.1. Out of this result it is suggested to choose a length of 5000 samples for each class to obtain an appropriate TD set. The following Equations (4.1 to 4.8) describe the statistical performance measures:

$$TPR = TP/P = TP/(TP + FN) \tag{4.1}$$
$$SPC = TN/N = TN/(TN + FP) \tag{4.2}$$
$$PPV = TP/(TP + FP) \tag{4.3}$$
$$NPV = TN/(TN + FN) \tag{4.4}$$
$$FPR = FP/N = FP/(FP + TN) = 1 - SPC \tag{4.5}$$
$$FNR = FN/(TP + FN) = 1 - TPR \tag{4.6}$$
$$FDR = FP/(TP + FP) = 1 - PPV \tag{4.7}$$
$$ACC = (TP + TN)/(TP + FP + FN + TN) \tag{4.8}$$

with

| | |
|---|---|
| P | positive instances |
| N | negative instances |
| TP | true psitive eqv. with hit, fist correctly identified as fist |
| TN | true negative eqv. with correct rejection, garbage correctly identified as garbage |
| FP | false positive eqv. with false alarm, Type I error, garbage incorrectly identified as fist |
| FN | false negative eqv. with miss, Type II error, fist incorrectly identified as garbage |
| TPR | sensitivity or true positive rate eqv. with hit rate, recall |
| SPC | specificity or true negative rate |
| PPV | precision or positive predictive value |
| NPV | negative predictive value |
| FPR | fall-out or false positive rate |
| FNR | false negative rate |
| FDR | false discovery rate |
| ACC | accuracy |

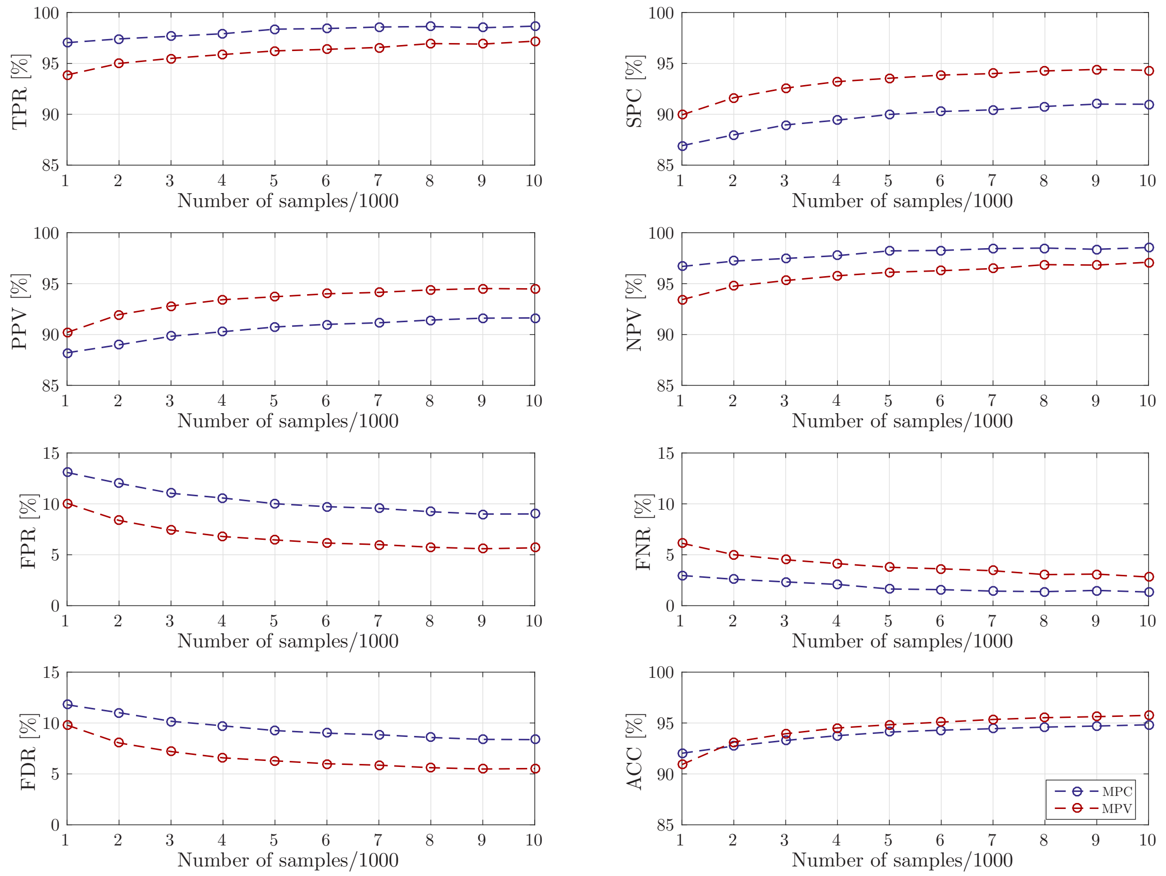


*Figure 4.1: Statistical measures of performance for prototype MPC- and MPV-data sets with different lengths using a KNN classifier.*

## 4.3 Training data of individual test persons

In the previous Section 4.2 it turned out that 5000 samples for fist gesture and 5000 samples for garbage are good values for an appropriate TD set. The next step was to record individual MPC and MPV TD sets for every TP taking part in the general usage test. This was done by recording five TD sets of length 1000 samples for fist and garbage, respectively, where the MYO position was not varied in between. These five sets where appended and led to an individual MPC set of a TP. Again the same steps were done for recording MPV sets, but after recording each of the five small sets (with length of 1000 samples for each gesture) the MYO position was varied a bit. This was done to simulate everyday life usage because every time a user puts on the MYO Armband he or she will not be able to place it exactly on the same position. The number of TPs who took part in the TD recording sessions was 13. After making these MPC and MPV sets for each TP it was possible to construct the remaining two TD sets which are MPW and MPA. The MPA set is the same for every TP and was obtained through appending all MPV sets of each TP. MPW sets for each TP were constructed through connection of MPV sets from each TP except the TP for whom this TD set was used in the general usage test. This makes it possible to evaluate if it is necessary to have TD of the person using this device, or if it is adequate to have TD from a certain number of individuals but not from the person using this device. Table 4.2 shows the lengths of the different TD sets.

| **Length of TD sets** | | | | |
|---|---|---|---|---|
| | **Garbage** | | **Fist** | |
| **TD sets:** | [Samples] | [s] | [Samples] | [s] |
| MPC(TP) | 5000 | 100 | 5000 | 100 |
| MPV(TP) | 5000 | 100 | 5000 | 100 |
| MPW(TP) | 60000 | 1200 | 60000 | 1200 |
| MPA | 65000 | 1300 | 65000 | 1300 |

*Table 4.2: Lengths of the four different types of training data.*

## 4.4 Evaluation of training data

All TD sets from each TP were evaluated for the six different classifiers. Evaluation of MPC and MPV sets was managed by cross validation where the 5000 samples of each class were split up into five parts. Four of these five parts were appended and used as TD, the remaining part served as test data. Statistical performance measures as described in Chapter 4.2 were calculated when testing TD using each classifier. As it is the case in cross validation, this was done five times so that every part was used once as test data set. Also MPA data was evaluated using cross validation. Since the MPA set is the same for every TP it was split up into five parts where four parts were used as TD and one part as test data. When splitting up MPA TD data it was taken care that every part contained material of each TP and again these steps were repeated in a way that every part was used as test data. For evaluation of MPW sets it was decided to use MPV as test data because of the fact that MPW data does not contain any material of the TP. Each TPs MPW set was tested with the corresponding MPV set. For the TD sets where cross validation was used, the median was calculated over the statistical performance measures of every classifier over the five different series of results. The results can be seen in Figure 4.2. In this Figure it can be noticed that the variances of these measures are smaller for MPC data sets. Additionally when using OCSVM classifier the true positive rate (TPR), which gives an estimate on how good fist was recognized, and accuracy (ACC) are significant worse than for the other classifiers. In exchange the true negative rate (SPC), which gives an insight on how good garbage was recognized, is higher for this classification method. In Table 4.3 one can see a ranking for the accuracy (ACC) values and the different types of TD according to the six classifiers. The results of this evaluation regarding accuracy (see Figure 4.2 and Table 4.3) confirm nearly what we expected:

- MPC data set works best because of low variances and high median values for the statistical performance measures. This data set has the highest median value for all classifiers for accuracy.

| Accuracy (ACC) ranking | | | | | | |
|---|---|---|---|---|---|---|
| **Rank** | **KNN** | **GMM** | **RSVM** | **LSVM** | **OCSVM** | **RAFO** |
| 1 | MPC<br>(93.12 %) | MPC<br>(92.13 %) | MPC<br>(94.07 %) | MPC<br>(92.88 %) | MPC<br>(72.42 %) | MPC<br>(92.80 %) |
| 2 | MPA<br>(89.18 %) | MPA<br>(85.21 %) | MPA<br>(88.13 %) | MPV<br>(77.73 %) | MPA<br>(68.05 %) | MPA<br>(83.57 %) |
| 3 | MPV<br>(83.08 %) | MPW<br>(81.72 %) | MPV<br>(82.83 %) | MPW<br>(77.60 %) | MPV<br>(67.73 %) | MPV<br>(82.39 %) |
| 4 | MPW<br>(80.47 %) | MPV<br>(78.81 %) | MPW<br>(82.51 %) | MPA<br>(76.45 %) | MPW<br>(62.71 %) | MPW<br>(81.81 %) |

*Table 4.3: Accuracy (ACC) ranking for the six classifiers using the four different types of TD (For MPC, MPV and MPW the values are the medians over the TPs. For MPA data it is not the median as this data set is the same for every TP).*

- MPA data set seizes the second place as it has the second highest median values for all classifiers except the LSVM where it is on the fourth place.
- MPV data set lays on the third place for KNN, RSVM, OCSVM and RAFO. Using GMMs this data set is on the last place and for LSVM on the second place.
- MPW data set achieved the worst results and is on the fourth place for KNN, RSVM, OCSVM and RAFO. When GMM or LSVM classification was used, this data set seizes the third place.

The expectation was that individual TD sets which only contain information of the appropriate TPs work best. On the other hand MPW data set which contains no information of the "user" was supposed to reach the worst results.

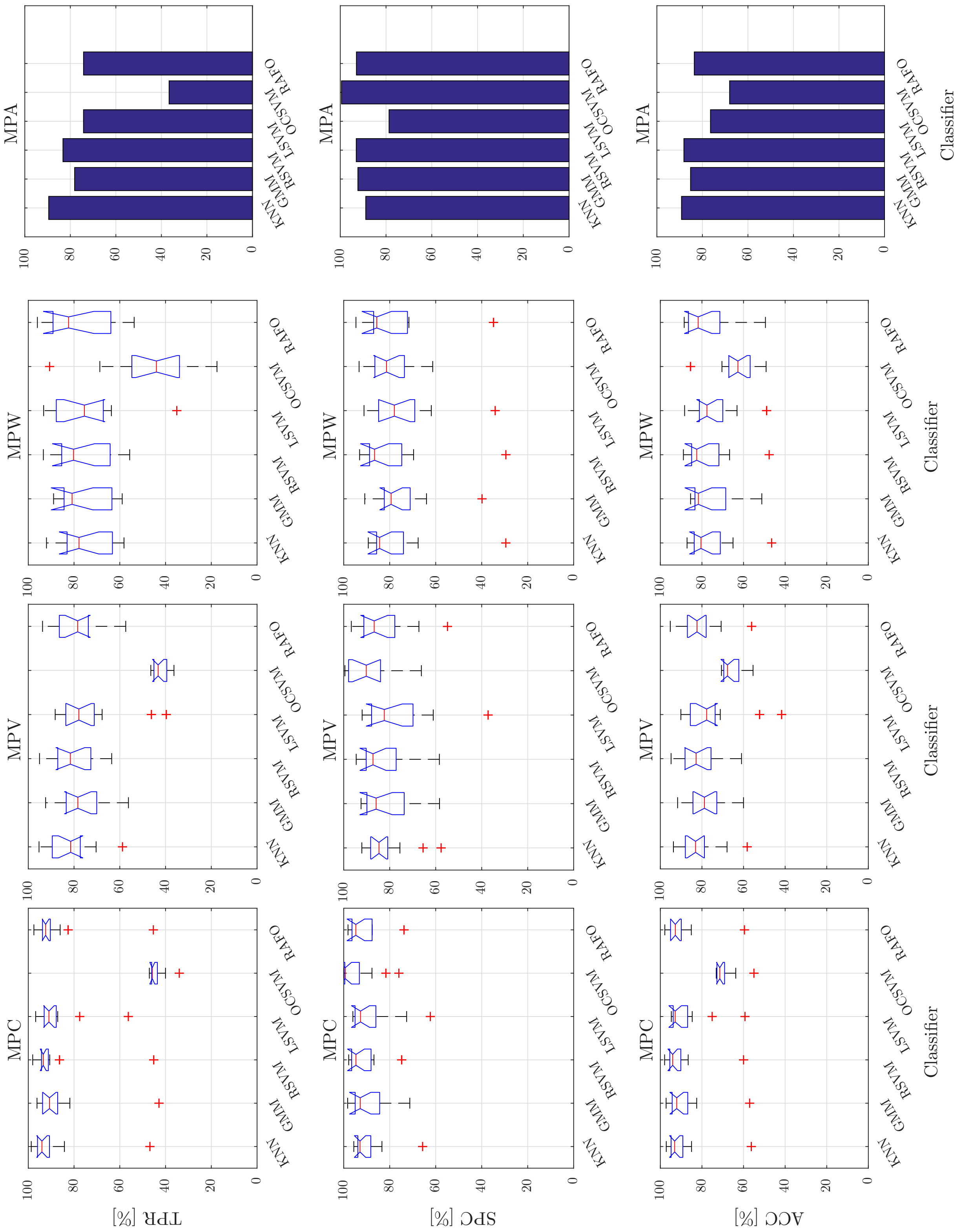


*Figure 4.2: Statistical measures of the performance (sensitivity (TPR), specificity (SPC) and accuracy(ACC)) for the different classifiers. Cross validation was used to split MPC, MPV and MPA data of TPs into TD and test data. MPW data sets were tested using MPV sets as test data.*

## 4.5 Evaluation of classifiers

For the decision, which classifier should be used in the general usage test, a simple B vs. A test with two TPs and their appropriate MPC data was conducted. The TP had to try the ELCS with all six different classifiers and had to determine if classifier B was better, equal or worse than classifier A in terms of functionality which means time-lags for turning the device to on and off, and stability including the appearing errors during ON and OFF. Every combination of classifiers came up three times in a random order. Additionally the two TPs had to give short comments which are summarized in the following. The results of this B vs. A test can be seen in Table 4.4, the symbols in brackets denote that the TP was not really sure to make a clear decision, for more details it is referred to Table A.1 in Appendix A.2 where the number of how often a classifier was better, equal or worse than the other can be seen. The ranked overall results are listed in Table 4.5 and graphically depicted in 4.3. In Appendix A.2 Figure A.8 once again shows these results but additionally with the number of chosen as equal and worse.

| **Functionality (time, stability)** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **B vs. A** | | **TP 1:** | | | **TP 2:** | | |
| **A** | **B** | **trial: 1** | **trial: 2** | **trial: 3** | **trial: 1** | **trial: 2** | **trial: 3** |
| KNN | GMM | - | - (=) | - (=) | - | - | - |
| KNN | RSVM | + | + (=) | + | - (=) | - | - |
| KNN | OCSVM | - | - | - | - | - | - |
| KNN | LSVM | + | = (-) | - (=) | - | - | - |
| KNN | RAFO | - | - | - | - | - | - |
| GMM | RSVM | = | + | + (=) | + | = | = |
| GMM | OCSVM | - | - | - | - | - | + |
| GMM | LSVM | + | = | = | = | = | + |
| GMM | RAFO | - | - | - | - | - | - |
| RSVM | OCSVM | - | - | - | = | - | = |
| RSVM | LSVM | = | - | - | - | = | = |
| RSVM | RAFO | - | - | - | - | - | - |
| OCSVM | LSVM | + | + | + | - | = | - |
| OCSVM | RAFO | + | + | + | - | - | - |
| LSVM | RAFO | - | - | - | - | - | - |

*Table 4.4: Classifier evaluation regarding functionality (time, stability) via a user test involving two subjects (B vs. A test).*

### Comments of TPs regarding the B vs. A test:

The KNN classifier seems to work quite good in nearly all cases. It seems to be more slowly in recognizing the garbage gesture and thus slowly in turning the device off. It rarely recognizes fist when garbage gesture is made, it mostly performs better than the other classifiers. When using GMM classifier TPs said that there are sometimes dropouts meaning that the classifier recognized garbage instead of fist. Furthermore it was somewhat more slow than KNN when turning off. Using RSVM classifier there where dropouts during ON and little disturbing noises during OFF, which means that the classifier recognized fist although the TPs made garbage gestures. With OCSVM classification it quickly turned out that this classifier is unusable for this application because it abiding turns to ON when a garbage gesture is made and there are many dropout errors when a fist gesture is made. The LSVM classifier was ascribed to turn faster off than the other classifiers but there appeared dropouts and disturbing noises. Although the TPs did not really agree in their opinion regarding this classifier. RAFO classifier delays highly when TPs want to turn the device to ON or OFF. Furthermore it often turns the device undesired to ON. One TP said that this classifier operates a bit arbitrary.

| Classifier ranking: | | | | | |
|---|---|---|---|---|---|
| TP | Place: | Classifier: | + | - | = |
| TP 1 | 1 | RSVM | 13 | 0 | 3 |
| | 2 | KNN | 10 | 4 | 1 |
| | 3 | LSVM | 8 | 3 | 4 |
| | 4 | GMM | 6 | 6 | 3 |
| | 5 | RAFO | 3 | 12 | 0 |
| | 6 | OCSVM | 0 | 15 | 0 |
| TP 2 | 1 | KNN | 15 | 0 | 0 |
| | 2 | RSVM | 6 | 3 | 6 |
| | 2 | GMM | 6 | 5 | 4 |
| | 2 | OCSVM | 6 | 6 | 3 |
| | 3 | LSVM | 4 | 6 | 5 |
| | 4 | RAFO | 0 | 15 | 0 |
| Overall | 1 | KNN | 25 | 4 | 1 |
| | 2 | RSVM | 19 | 3 | 8 |
| | 3 | GMM | 12 | 11 | 7 |
| | 3 | LSVM | 12 | 9 | 9 |
| | 4 | OCSVM | 6 | 21 | 3 |
| | 5 | RAFO | 3 | 27 | 0 |

*Table 4.5: Overall and individual ranking of the classifiers.*

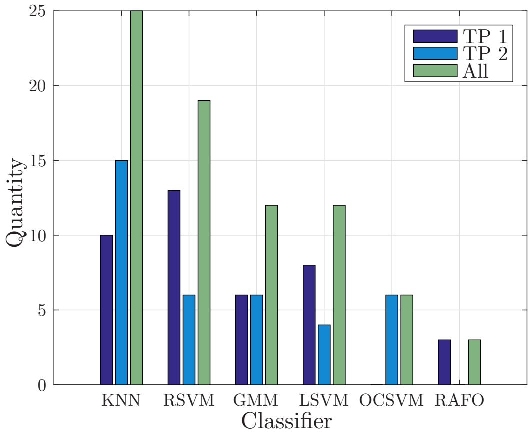


*Figure 4.3: Classifier evaluation, shows how many times a classifier was ranked as better as another one.*

After evaluating the results of this B vs. A test depicted in Table 4.5 and pictured in Figure 4.3, it was decided to use the KNN classifier for the ELCS and its further evaluation through the general usage test, as it was ranked as best most frequently.

# 5 Evaluation of electro-larynx-control system and results

This chapter explains the process of evaluation from the first considerations through realisation of the general usage test with the software SpeechRecorder [Draxler and Jänsch, 2004] to the point of evaluating the results in *Matlab* and *Praat*.

## 5.1 General considerations

First thoughts went in the direction of how can the ELCS be evaluated in terms of usability, functionality and accuracy. Therefore, the decision was made to conduct a general usage test and a user test. A user test needs a lot of time and resources like patients who are suffering from the consequences of a laryngectomy. Therefore it was decided only to explain it in theory (see Chapter 5.3). In the general usage test the two different features of the ELCS:

1.) ON/OFF-control, and

2.) $f_0$ variation

were evaluated.

ON/OFF-control was evaluated with a traffic light test that gives some insight on the time-lags and the occurring errors. For evaluation of $f_0$ variation each TP had to imitate a graphically given fundamental frequency contour.

This all was done for the four different types of TD as described in Chapter 4.1 while using the KNN classifier which turned out to be the best choice (see Chapter 4.5).

## 5.2 General usage test

With the examination of a general usage test the general functions of the ELCS should be evaluated. Therefor it is not necessary that TPs who take part in this test have undergone a laryngectomy. This general usage test should give an insight on how well the two main features (ON/OFF-control and $f_0$ variation) of the device work in general, and on the performance of the four different types of TD sets.

The general usage test was conducted using SpeechRecorder [Draxler and Jänsch, 2004] which is a recording tool with some special features like a traffic light countdown with variable times for green, yellow and red light phases (compare Figure 5.1). For more detailed information about SpeechRecorder [Draxler and Jänsch, 2004] it is referred to the manual [Draxler, 2004].

This test is split up in three different tasks:

- time-lag and error evaluation,
- false ON evaluation, and
- $f_0$ variation evaluation

At the beginning of each task there was a short training sequence to make the TP familiar with the whole system and test setup. For time-lag and error evaluation the TPs had to produce the vowel /a/, using the ELCS, for as long as SpeechRecorder [Draxler and Jänsch, 2004] showed a green light. During the False ON evaluation task the TPs had to make critical gestures while green light was on. For this task the green light phase lasted 20 seconds. For the $f_0$ variation evaluation where the TPs had to imitate a given $f_0$ contour, the recording was started and stopped manually by the instructor. Hence, for this task there was no need to use the traffic light function provided by SpeechRecorder [Draxler and Jänsch, 2004]. In the case of time-lag and error evaluation, red light meant that the TP had to do nothing, red-yellow light means prerecording. At the beginning of this phase the recording started and the TP had to prepare for the task. Green light phase is called recording and when this phase started the TP had to execute the task until the next red-yellow phase (postrecording) started. During this red-yellow phase it was still recorded until the end of this phase. The last phase was again red light phase, during this phase the TP had to do nothing. For the time-lag and error task in the general usage test the prerecording- and recording-times were always different to avoid a training effect. Postrecording-time was always two seconds. The different times can be seen in Table 5.1.

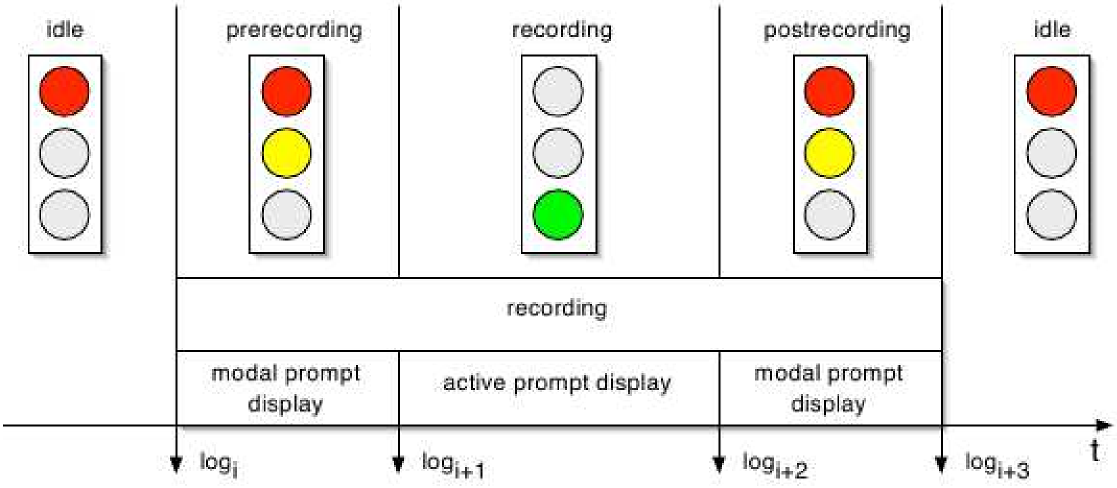


*Figure 5.1:* SpeechRecorder *recording phases in combination with the traffic lights [Draxler, 2004].*

The instructions for the TPs can be seen in Appendix A.3. 12 out of 13 people examined the general usage test. One of those 12 participants was the author of this thesis and his results were not included in the evaluation.

| **Recording times** | | | |
|---|---|---|---|
| **Number of trial** | **Prerecording** | **Recording** | **Postrecording** |
| **i** | $t_{prerec,i}$ [s] | $t_{rec,i}$ [s] | $t_{postrec,i}$ [s] |
| 1 | 3 | 5 | 2 |
| 2 | 3 | 4 | 2 |
| 3 | 2 | 3 | 2 |
| 4 | 1 | 4 | 2 |
| 5 | 2 | 3 | 2 |
| 6 | 3 | 5 | 2 |
| 7 | 1 | 3 | 2 |
| 8 | 1 | 4 | 2 |
| 9 | 3 | 4 | 2 |
| 10 | 2 | 5 | 2 |
| 11 | 2 | 3 | 2 |
| 12 | 1 | 5 | 2 |
| 13 | 2 | 3 | 2 |
| 14 | 3 | 4 | 2 |
| 15 | 1 | 5 | 2 |
| 16 | 3 | 5 | 2 |
| 17 | 2 | 3 | 2 |
| 18 | 1 | 4 | 2 |
| 19 | 2 | 3 | 2 |
| 20 | 1 | 4 | 2 |

*Table 5.1: Recording times of the different trials for the general usage test (settings of SpeechRecorder [Draxler and Jänsch, 2004]).*

### 5.2.1 Time-lag and error evaluation

For each TPs' TD sets (MPC, MPV, MPW and MPA) a time-lag evaluation was made in the following way. The TP had to produce the vowel /a/ as long as the green light was on. Producing the vowel /a/ means that the TP has to make a fist in order to produce an excitation signal and form the vocal tract in an appropriate way. This task was repeated 20 times for each TD set. The resulting recordings contain the clean excitation signal. Additionally signals were recorded via a headset. For the evaluation of ON/OFF time-lags and occurring errors only the clean excitation signal was used. Analysis of these recordings was done in *Matlab*. First the recordings were normalized and the start and end indizes of each signal block were searched. Afterwards the start and end point of the main block was searched. By subtracting the start point of the recording phase (green light) from the start point of the main block, the ON time-lag ($T_{on,i}(TP)$) was calculated for every recording. OFF time-lag ($T_{off,i}(TP)$) was calculated by subtracting the record end point from the main block end point, respectively. Additionally, for these recordings the ON errors (false OFFs) and the OFF errors (false ONs) were detected. ON errors occur in the main block and OFF errors occur at the beginning and at the end of the record, that is before main block and after main block. Figure 5.2 shows the parts of a recording for an ideal and a defective one including errors and time-lags. Time-lags are composed by the reaction time of the TP and the delay of the system. In [Jain et al., 2015] they made a study on the visual reaction time of 120 healthy medical students between 18 and 20 years. Test persons were told to concentrate on the black cross on the screen and press the "space bar" key, as soon as possible once the red circle (target stimulus) appeared on the screen. Target stimulus was always appearing after a variable time and the outcome of this study was a mean reaction time of 247.6 ms.

For some recordings the time-lags are negative which means that the TP either made a fist before the green light started or released the fist before green light phase ended. These negative time-lags were not included in the evaluation. It is also important to mention that false ONs and false OFFs respectively do not necessarily have to be errors caused by the system, they could also have been caused by the TP when he or she accidentally made or released the fist gesture. This case could not be evaluated.

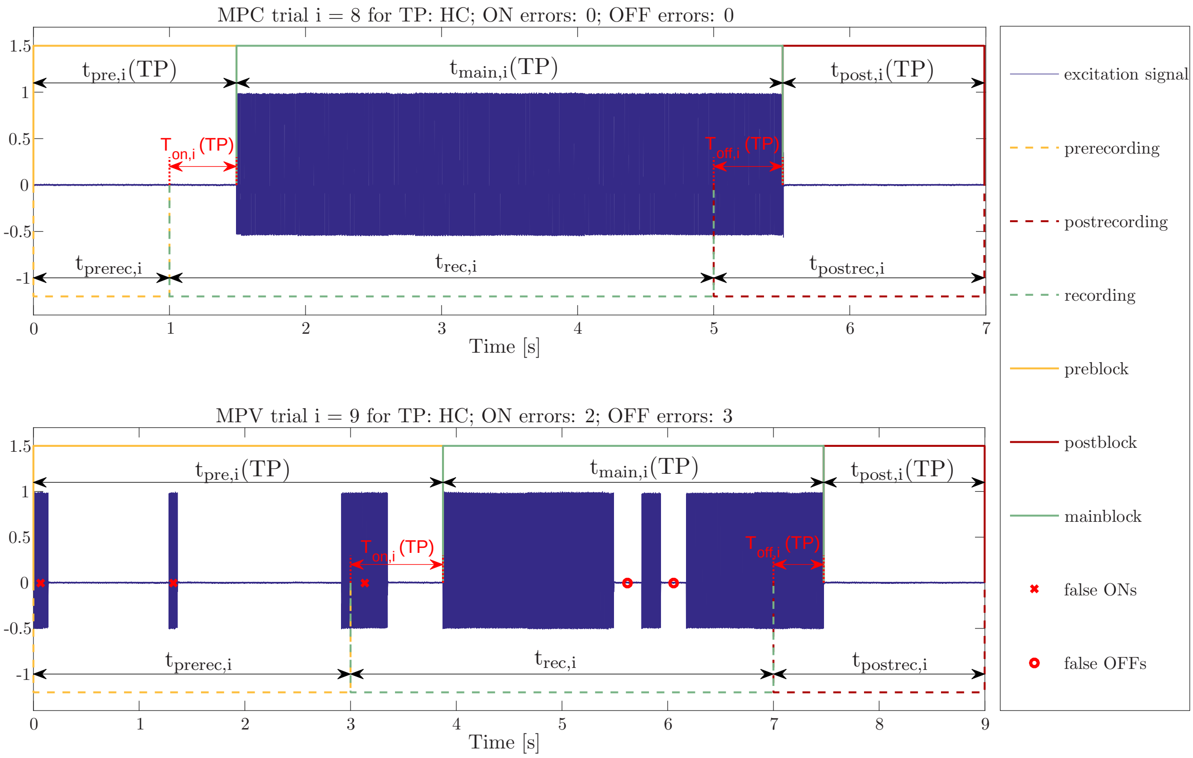


*Figure 5.2: Different sections of the recordings including time-lags and errors, on top for an ideal trial and at the bottom for a defective trial.*

### Calculations

$$T_{on,i}(TP) = t_{pre,i}(TP) - t_{prerec,i} \tag{5.1}$$

$$T_{off,i}(TP) = t_{postrec,i} - t_{post,i}(TP) \tag{5.2}$$

$$N_{Eon}(TP) = \sum_{i=1}^{N_t} n_{Eon,i}(TP) \tag{5.3}$$

$$N_{Eoff}(TP) = \sum_{i=1}^{N_t} n_{Eoff,i}(TP) \tag{5.4}$$

$$f_{OFF}(TP) = \frac{N_{Eon}(TP)}{\sum_{i=1}^{N_t} t_{main,i}(TP)} \tag{5.5}$$

$$f_{ON}(TP) = \frac{N_{Eoff}(TP)}{\sum_{i=1}^{N_t} t_{pre,i}(TP) + t_{post,i}(TP)} \tag{5.6}$$

$$f_{OFFtotal} = \frac{\sum_{TP=1}^{N_{TP}} N_{Eon}(TP)}{\sum_{TP=1}^{N_{TP}} \sum_{i=1}^{N_t} t_{main,i}(TP)} \tag{5.7}$$

$$f_{ONtotal} = \frac{\sum_{TP=1}^{N_{TP}} N_{Eoff}(TP)}{\sum_{TP=1}^{N_{TP}} \sum_{i=1}^{N_t} t_{pre,i}(TP) + t_{post,i}(TP)} \tag{5.8}$$

with

| | |
|---|---|
| $T_{on,i}(TP)$ | ON-time-lag of a TP and a single trial i |
| $T_{off,i}(TP)$ | OFF-time-lag of a TP and a single trial i |
| $\tilde{T}_{on}(TP)$ | Median ON-time-lag over all trials for a TP |
| $\tilde{T}_{off}(TP)$ | Median OFF-time-lag over all trials for a TP |
| $\tilde{T}_{on}$ | Median ON-time-lag over all trials and all TPs |
| $\tilde{T}_{off}$ | Median OFF-time-lag over all trials and all TPs |
| $\tilde{T}_{on+off}$ | Median of the sum of ON- and OFF-time-lags over all trials and all TPs |
| $t_{prerec,i}$ | prerecording time for trial i |
| $t_{postrec,i}$ | postrecording time for trial i |
| $t_{rec,i}$ | recording time for trial i |
| $t_{pre,i}(TP)$ | preblock time for trial i and testperson TP |
| $t_{post,i}(TP)$ | postblock time for trial i and testperson TP |
| $t_{main,i}(TP)$ | mainblock time for trial i and testperson TP |
| $n_{Eon,i}(TP)$ | number of ON errors in trial i for testperson TP |
| $n_{Eoff,i}(TP)$ | number of OFF errors in trial i for testperson TP |
| $N_{TP}$ | number of test persons (11) |
| $N_t$ | number of trials for each TP and TD set(20) |
| $N_{Eon}(TP)$ | number of ON-errors in all trials for TP |
| $N_{Eoff}(TP)$ | number of OFF-errors in all trials for TP |
| $f_{OFF}(TP)$ | number of false OFFs per second over all trials for a particuar TP |
| $f_{ON}(TP)$ | number of false ONs per second over all trials for a particuar TP |
| $\tilde{f}_{OFF}$ | median of false OFFs per second over all TPs |
| $\tilde{f}_{ON}$ | median of false ONs per second over all TPs |
| $f_{OFFtotal}$ | number of false OFFs per second over all trials and TPs |
| $f_{ONtotal}$ | number of false ONs per second over all trials and TPs |

### Results

The calculations are depicted in Equations 5.1 to 5.8. Figure 5.3 shows three boxplots of the time-lags each for the different TD sets. In the left boxplot $T_{on}$, in the middle $T_{off}$ and on the right side the sum of both ($T_{on+off}$) is shown. Each box contains 220 values (11 TPs times 20 trials). The median of all $T_{on,i}(TP)$ over all trials and TPs ($\tilde{T}_{on}$) is between 577 $ms$ and 592 $ms$ for the different types of TD sets and there is no significant difference. Considering $\tilde{T}_{off}$

it catches ones eye that MPW has a significant lower value than MPC and MPV data. This could arise from the case that MPW data has no information about the actual TP hence the classifier is not that robust in recognizing the fist gesture, which means garbage is recognized once the TP looses the fist somewhat. The median over the sums of $T_{on,i}(TP)$ and $T_{off,i}(TP)$ which is denoted as $\tilde{T}_{on+off}$ shows this significance only compared to MPC data set. Table 5.2 again shows the median values $\tilde{T}_{on}$, $\tilde{T}_{off}$ and $\tilde{T}_{on+off}$ for the different TD sets. In Table 5.3 the number of negative time-lags, which, as said before, were not taken into account, can be seen. The median ON- and OFF-time-lags for each TP and TD set can be seen in Figure 5.4. Considering this plot it catches ones eye that there is no TD set which has always a minimum, as it was supposed for MPC and MPV data sets.

The occurring ON and OFF errors per second are shown in boxplot 5.5. Each box includes eleven values and each of these values is the sum of errors over all 20 trials devided by the time. That is in case of ON errors the sum of all false OFFs during the 20 trials devided by the time of all mainblocks in these trials. Vice versa for OFF errors. It can be seen for false OFFs, that the variance for MPC and MPV data is much smaller than for MPW and MPA TD. This is not the case for false ONs where the variance is nearly the same as for ON errors using MPC or MPV data and for MPV it is even lower. The high variances for false OFFs when using MPW and MPA data could be a result of the fact that these TD sets are not so specific for a TP anymore and on that account the classifier is not so sensitive in recognizing fist gesture because the fist training samples for one TP could be garbage for another TP. On the other side, regarding false ONs, this is not the case as being not sensitive in recognizing fist means being more sensitive in recognizing garbage. The medians of ON and OFF errors are nearly the same for all types of TD except for false OFFs for MPW which is significantly higher than for MPC or MPV which is a consequence of the certainty that MPW data contains no information from the actual TP. Additionally the median values can be seen in Table 5.4.

Figure 5.6 represents ON and OFF errors for each TP and set of TD per second (same values as in Figure 5.5) and the sum of errors over all TPs per second. It can be observed that these values vary highly between the TPs. Regarding the sum of errors it is obvious that for each TD set the value of false OFFs per second is higher than for false ONs per second. Especially for MPW and MPA data this value reaches a much higher level. These observations can be traced back to the same reasons as explained above.

Additionally it was checked if time-lags and errors are normal distributed. Therefor the Anderson-Darling test was performed in *Matlab* and the outcome is listed in Table 5.5. The result that data is not normal distributed is the same for all types of TD, hence a median calculation as can be seen in Figure 5.4 was used for evaluation.

| | **Median time-lags** | | |
|---|---|---|---|
| **TD** | $\tilde{T}_{on}$ [ms] | $\tilde{T}_{off}$ [ms] | $\tilde{T}_{on+off}$ [ms] |
| MPC | 592.26 | 564.44 | 1194.6 |
| MPV | 584.63 | 579.47 | 1185.2 |
| MPW | 580.46 | 520.25 | 1130.0 |
| MPA | 577.05 | 556.34 | 1174.6 |

*Table 5.2: Median time-lags*

| | **Number of negative time-lags** | |
|---|---|---|
| **TD** | $T_{on,i}(TP)$ | $T_{off,i}(TP)$ |
| MPC | 2 | 1 |
| MPV | 1 | 0 |
| MPW | 0 | 3 |
| MPA | 2 | 2 |

*Table 5.3: Number of negative time-lags over all test persons and trials (with $i = 1, ..., N_t$ and $TP = 1, ..., N_{TP}$).*

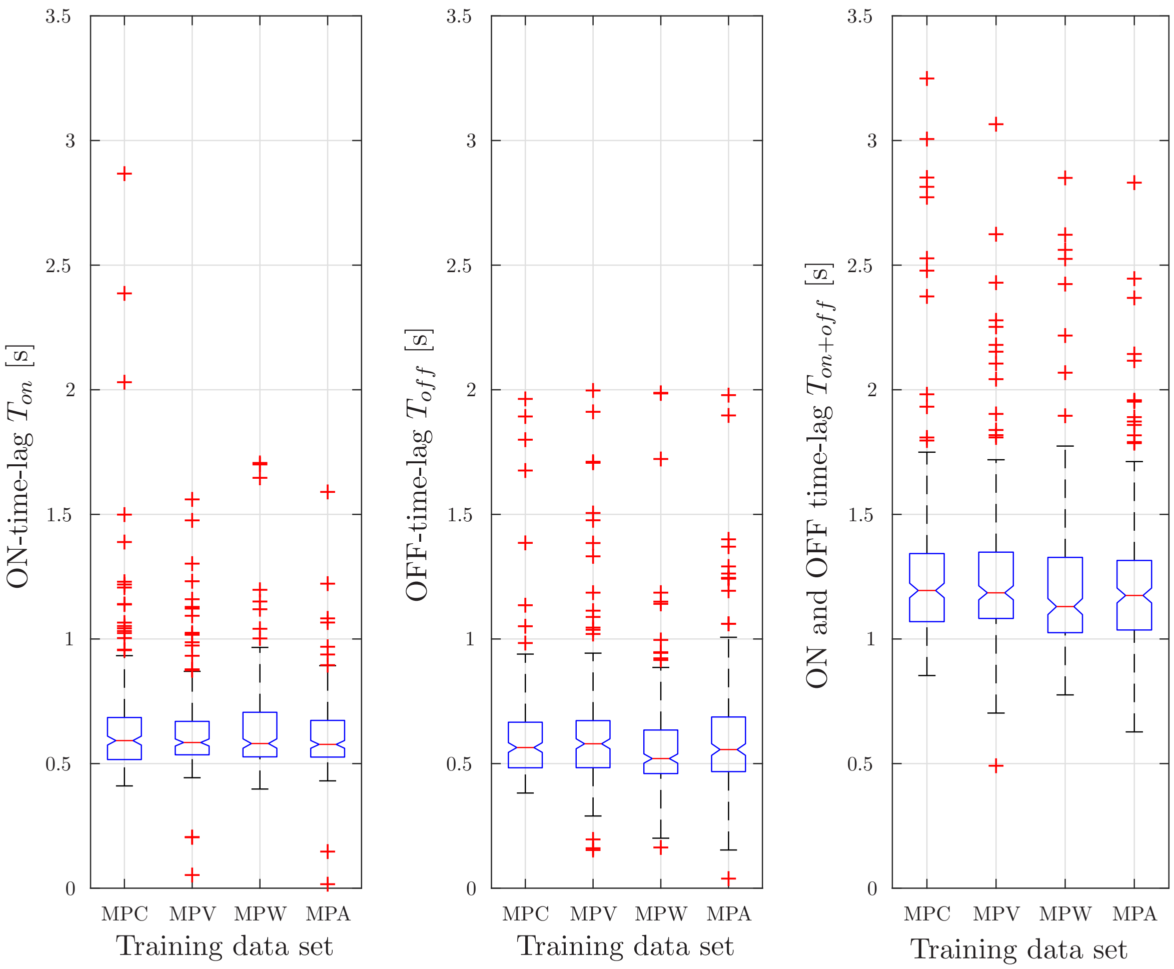


*Figure 5.3: ON/OFF-time-lags for the different types of TD over all TPs and trials (220 values for each box).*

| Median errors per sec | | |
|---|---|---|
| **TD** | $\tilde{f}_{OFF}$ [1/s] | $\tilde{f}_{ON}$ [1/s] |
| MPC | 0.012626 | 0.013236 |
| MPV | 0.012813 | 0.012825 |
| MPW | 0.16269 | 0.026503 |
| MPA | 0.025201 | 0.013435 |

*Table 5.4: Median errors per second*

| Check if normal distributed | | | | |
|---|---|---|---|---|
| **Anderson-Darling test** | **Time-lags** | | **Errors** | |
| | $\{T_{on,i}(TP)\}$<br>$i = 1, ..., N_t$<br>$TP = 1, ..., N_{TP}$ | $\{T_{off,i}(TP)\}$<br>$i = 1, ..., N_t$<br>$TP = 1, ..., N_{TP}$ | $\{n_{Eon,i}(TP)\}$<br>$i = 1, ..., N_t$<br>$TP = 1, ..., N_{TP}$ | $\{n_{Eoff,i}(TP)\}$<br>$i = 1, ..., N_t$<br>$TP = 1, ..., N_{TP}$ |
| Normal distributed<br>probability p | ×<br>$< 0.0005$ | ×<br>$< 0.0005$ | ×<br>$< 0.0005$ | ×<br>$< 0.0005$ |

*Table 5.5: Check if time-lags and errors are normal distributed using the Anderson-Darling test. The result is equal for all four types of TD.*

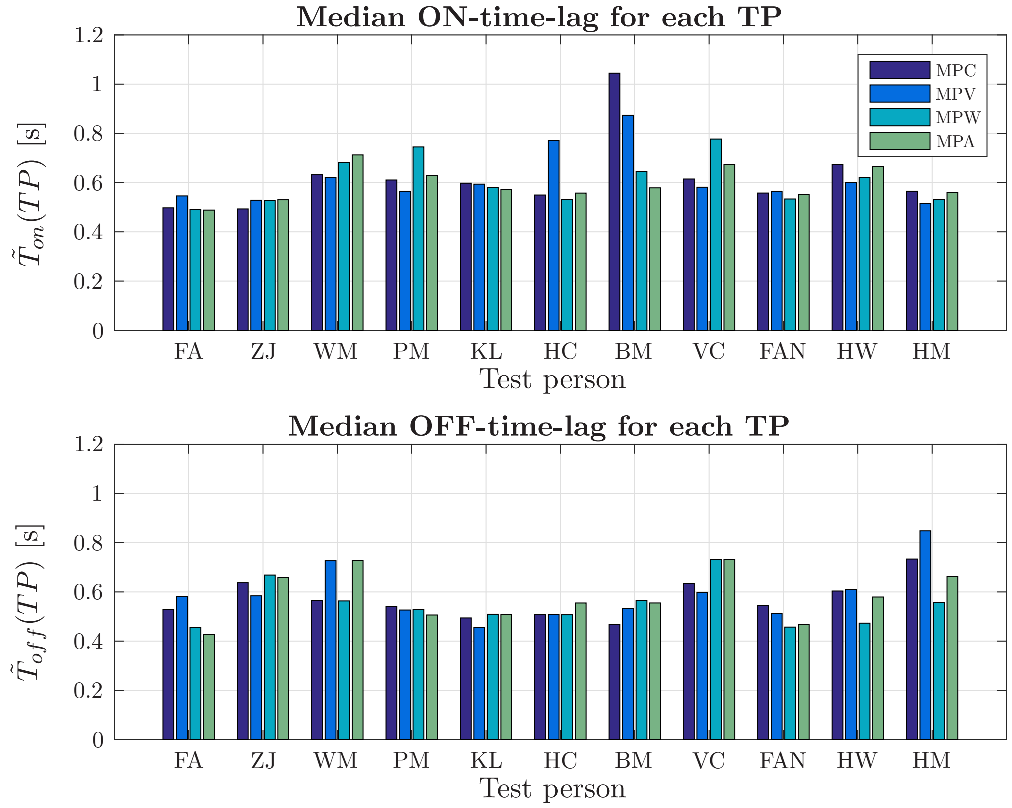


*Figure 5.4: ON/OFF-time-lags for each TP, values are the medians over the trials (20 trials for each set of TD and TP).*

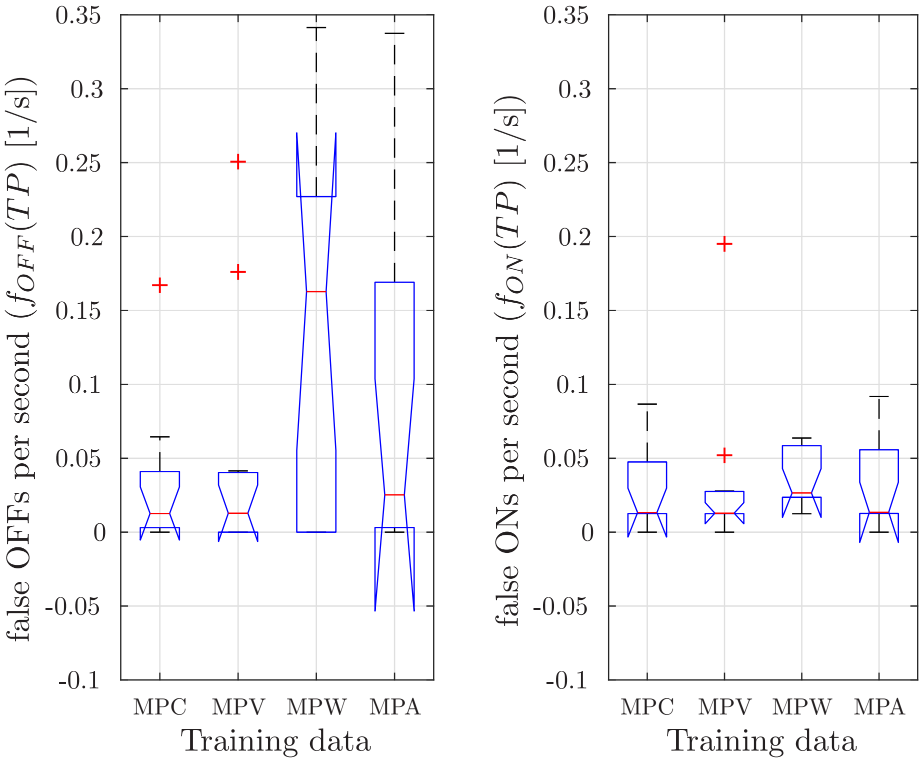


*Figure 5.5: ON/OFF-errors per second over all TPs for the different types of TD. Every box contains 11 values where each of them was calculated through division by the sum of errors (all trials of one TP) by the appropriate total time (either sum of all trials mainblocks or sum of all trials pre- and postblocks).*

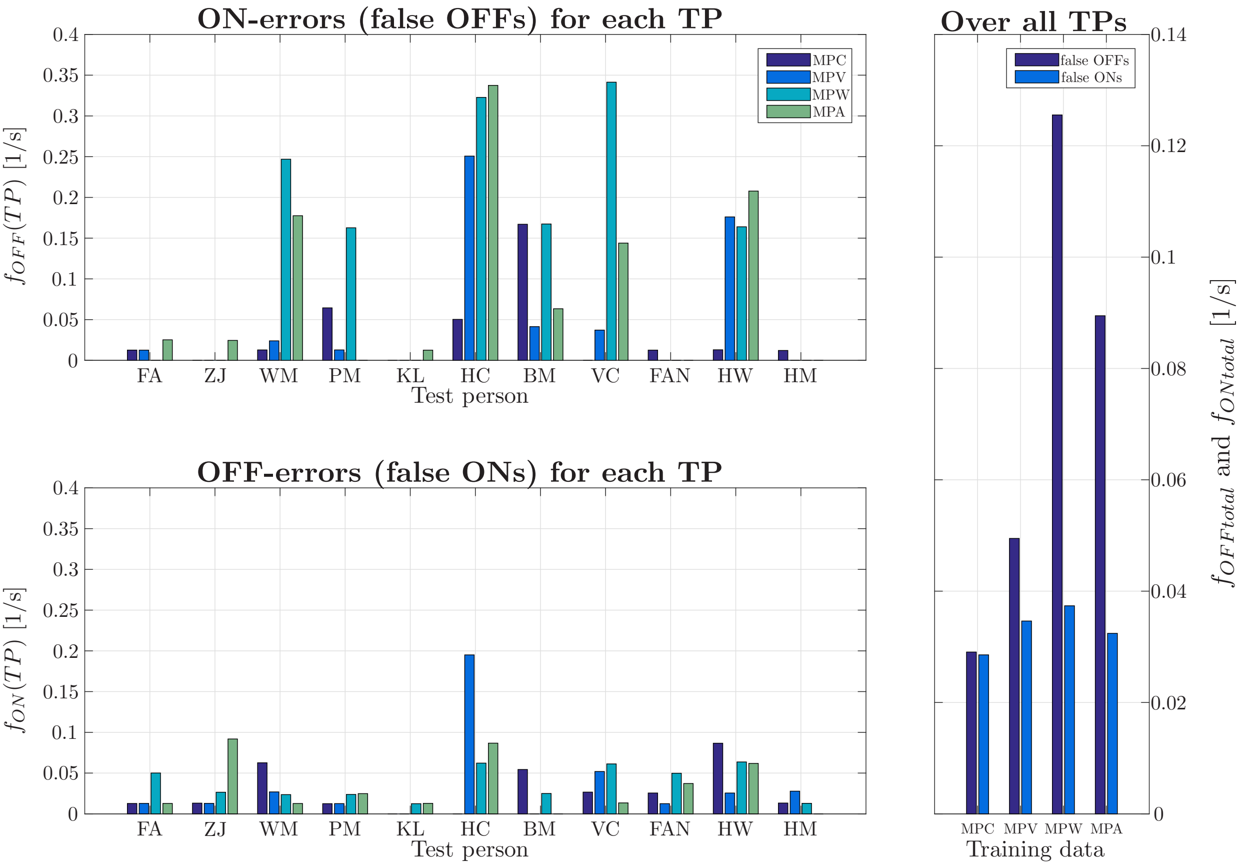


*Figure 5.6: Left: ON/OFF-errors per second for each TP (sum of errors during all trials divided by the appropriate total time which is either the sum of mainblocks for all trials or the sum of pre- and postblocks for all trials); Right: ON/OFF-errors per second over all TPs (sum of all occurring errors devided by the appropriate total time which is either the sum of all mainblocks over all trials and TPs or the sum of pre- and postblocks over all trials and TPs).*

### 5.2.2 False ON test evaluation

Equations 5.9 to 5.11 show the calculations for this task. When trying different types of TD and classifiers in informal tests, it turned out that there are some critical gestures which are classified as fist by mistake. This testing should give a little insight on this matter.
Evaluation of false ONs was likewise done with SpeechRecorder [Draxler and Jänsch, 2004]. For this task every TP had to make critical gestures like moving fingers, scratching the back of the head, grab the pencil case and turn over a few pages of a book. When the light in SpeechRecorder [Draxler and Jänsch, 2004] turned to red-yellow the TP had to prepare for this task. During green light phase the TP had to make the described gestures until the recording stopped. Prerecording and postrecording time were one second each and recording time was 20 seconds. This led to a total recording length of 22 seconds. This task was repeated for all four types of TD.

#### Calculations

$$t_r = t_{rec} + t_{postrec} = 21s \tag{5.9}$$

$$t_{fONrel}(TP) = \frac{t_{fON}(TP)}{t_r} \tag{5.10}$$

$$t_{fONrelALL} = \frac{\sum_{TP=1}^{N_{TP}} t_{fON}(TP)}{N_{TP} \cdot t_r} \tag{5.11}$$

with

| | |
|---|---|
| $t_{rec}$ | recording time (20 seconds) |
| $t_{postrec}$ | postrecording time (1 second) |
| $t_r$ | relevant recording time |
| $t_{fON}(TP)$ | time where ELCS device was ON for testperson TP |
| $t_{fONrel}(TP)$ | relative time where ELCS device was ON for testperson TP |
| $t_{fONrelALL}$ | relative time where ELCS device was ON over all testpersons |

#### Results

In Figure 5.7 the results of this evaluation are depicted. It shows the time where ELCS generated an output relative to the recording time which was 21 seconds (recording time plus postrecording time). On the left side of this plot the relative ON time is illustrated for the particular TP and on the right side this value is shown as sum over all TPs. It stands out that the lowest value over all TPs, which is about 15 percent, is reached when using MPW data. This again could be the case because of the fact that MPW data does not contain any information of the actual TP, that is why the classifier is not that sensitive in recognizing fist gesture meaning that even critical gestures are classified more likely as garbage. Considering the relative ON time for each TP separately one can not say that every TP has the minimum value when using MPW data. Table 5.6 shows the TD set where the minimum relative ON time lays for each TP. Four TPs reached a minimum when using MPC data, three TPs reached it when MPW data was used. Using MPV or MPA data in each case two TPs reached a minimum relative ON time.

| Minimum of relative ON times | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Min** | **FA** | **ZJ** | **WM** | **PM** | **KL** | **HC** | **BM** | **VC** | **FAN** | **HW** | **HM** | $\sum$ |
| MPC | - | - | - | - | × | - | - | × | × | × | - | 4 |
| MPV | × | - | - | - | - | - | - | - | - | - | × | 2 |
| MPW | - | × | - | - | - | × | × | - | - | - | - | 3 |
| MPA | - | - | × | × | - | - | - | - | - | - | - | 2 |

*Table 5.6: Shows the TD set (marked with crosses) for which the appropriate TP obtained a minimum relative ON time.*

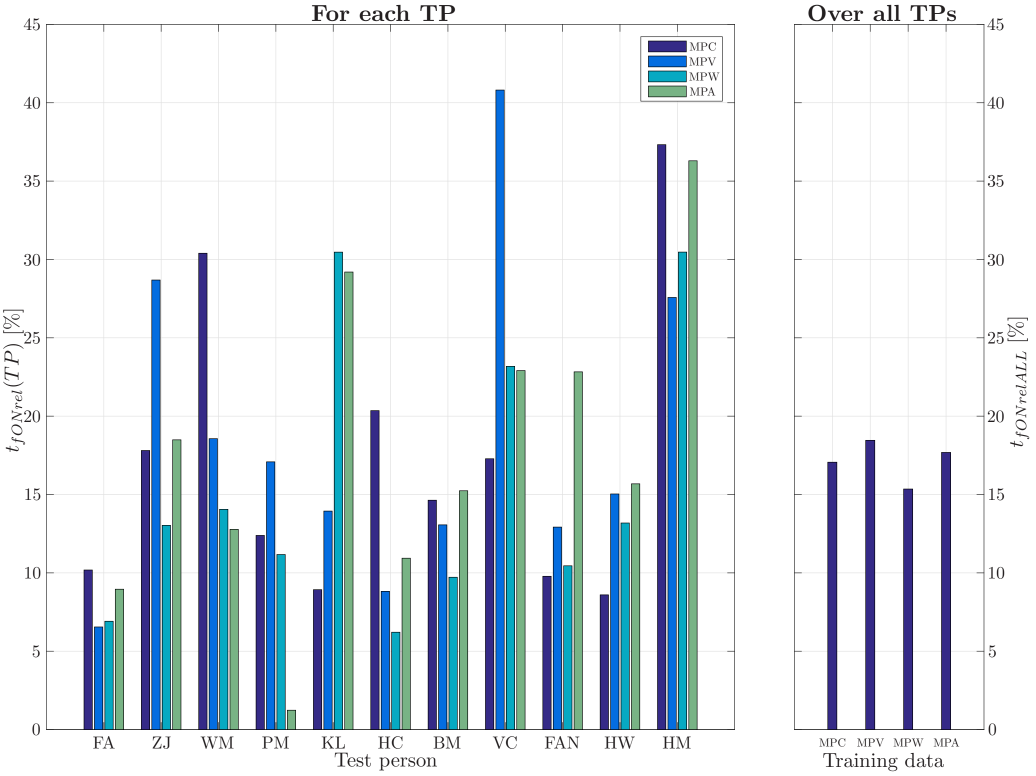


*Figure 5.7: Left: ON time of the ELCS device relative to recording time for each TP; Right: ON time of the ELCS device relative to recording time over all TPs.*

### 5.2.3 Comparison with ON/OFF-control via electrodes placed on the neck

In the dissertation of Anna Katharina Fuchs "The Bionic Electro-Larynx Speech System" [Fuchs, 2015] ON/OFF-control was managed via electrodes placed on the neck. Therefore, voice initiation time VIT which is equivalent to $\tilde{T}_{on}$ and voice termination time VTT which is equivalent to $\tilde{T}_{off}$ were evaluated. For evaluation of voice initiation time and voice termination time 40 samples were recorded before training and 40 samples after training for three female and one male speaker. Training was used to make the participants familiar with using the device. It consisted of 9 sessions within two weeks. The outcome of this evaluation was that the medians for every speaker are in a range between 400 $ms$ and 480 $ms$ regardless if pre- or post-training, but the variance of the values was lower for post-training samples. Compared to the median time-lags of this thesis, which are listed in Table 5.2 the median values for VIT and VTT are all around 120 $ms$ to 200 $ms$ lower. One reason for this result perhaps lies in the processing of sEMG data. Classification of the sEMG data received by the MYO Armband probably needs more time than the algorithm used in [Fuchs, 2015]. Another explanation is that in this work making a fist gesture is used for turning on the device. Making a fist gesture immediately when a visual stimulus is noticed could maybe engage more time.

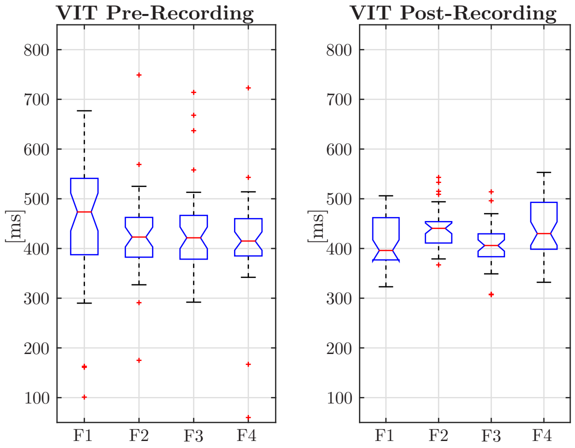


*Figure 5.8: Voice initiation time (VIT) for pre- and post-training for three female speakers (F1, F2, F3) and one male speaker (M1) [Fuchs, 2015].*

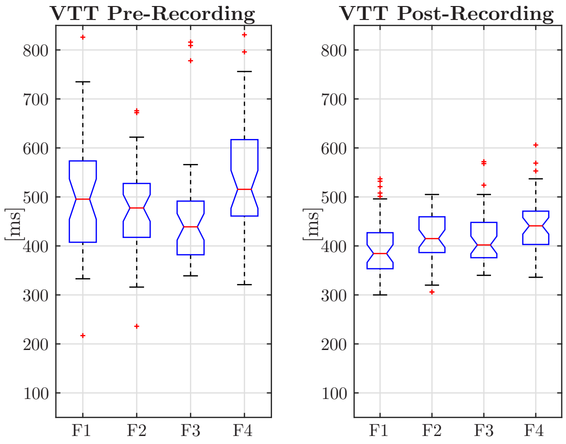


*Figure 5.9: Voice termination time (VTT) for pre- and post-training for three female speakers (F1, F2, F3) and one male speaker (M1) [Fuchs, 2015].*

### 5.2.4 $f_0$ test evaluation

The third experiment in the general usage test was $f_0$ variation. For this task the instructor started the recording and the TP had to imitate the graphically given $f_0$-contour. When the TP was finished, the instructor stopped the recording. This evaluation was done for five different $f_0$-contours. Since the general usage test should not take too long it was decided to evaluate $f_0$ variation only for MPC data set as it was supposed to work best. Besides the processed information for changing the frequency comes from MYOs orientation data and has nothing to do with TD or classifiers which are responsible for ON/OFF- control. This evaluation should give information if it is generally possible to imitate a $f_0$-contour by varying the vertical position, or in other words the angle, of the arm. Imitated $f_0$-contours were extracted from the recordings using *Praat*.

#### Results

Figure 5.10 shows the given $f_0$ contours and the imitated ones for a TP where there were no errors and the $f_0$ variation feature did well. Figure 5.11 shows the results for a TP where the MPC data set did not work that well and on account of this there occurred false OFF errors. Anyway, the imitated $f_0$ contour looks rather like the given one. For the results of the other TPs it is referred to Figures A.11 to A.20 in Appendix A.4. Contemplating these Figures, and comparing the given and imitated contours, it is obvious that the variation of $f_0$ works still good. However this result has to be enjoyed with some caution because as said before it is very theoretical. For making a statement about how well this feature works in practice it would be necessary to examine a much more complex and time intensive test where TPs have to practice with the device for getting used to control $f_0$ in real conversations.

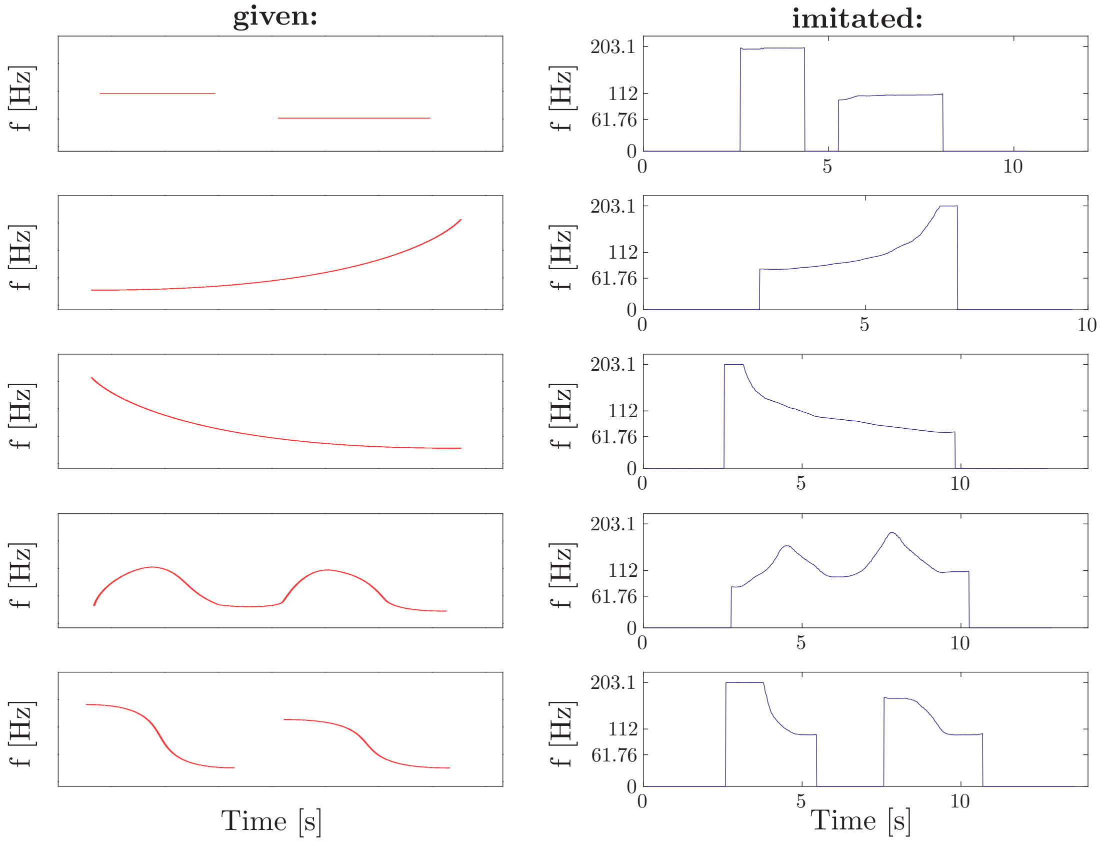


*Figure 5.10: $f_0$ for test person HK*

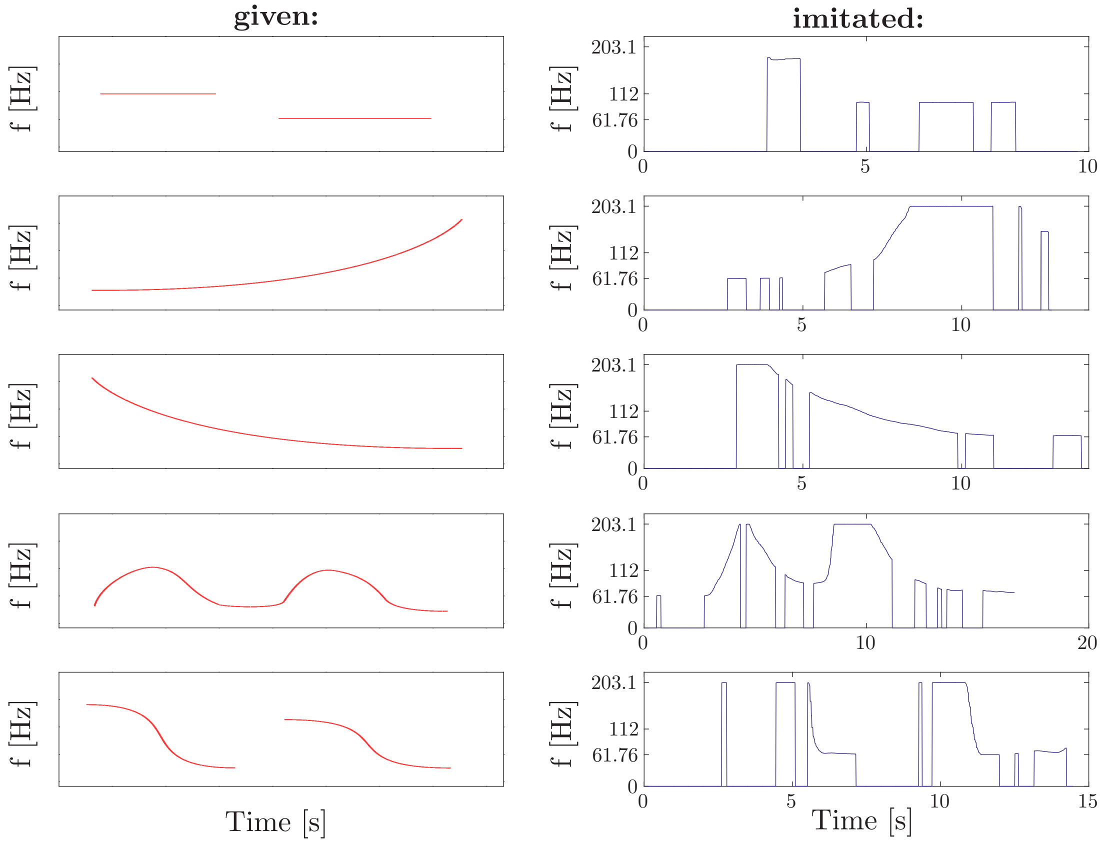


*Figure 5.11: $f_0$ contours for test person HC*

## 5.3 User test

This testing was not examined and is only explained theoretically for future work.
The user test should give some insight in the general usability of the ELCS. Therefor it is important to execute the test with help from persons who have undergone a laryngectomy. The test persons will be detailed instructed on how the device is working. They will have a short trainings phase (about 30 minutes). Afterwards they will have to perform some scenarios of daily life like the following:

- Telephone conversation: the TP has to simulate a telephone conversation while making handwritten notes.

- A chalk talk lecture: the TP has to simulate a chalk talk lecture using a microphone.

In both scenarios it is essential to have a hands-free device as the ELCS. One hand is used for holding a pencil while the other hand holds the telephone receiver or the microphone, respectively.

In the end every test person should fill out a questionnaire as follows. Because the mother tongue of the participants taking part in this user test probably is German, the questionnaire is also written in German:

# Fragebogen zum Benutzertest des Freihand-Elektro-Larynx

## A) Angaben zur Person:

Name:

Wie alt sind Sie: ______

Geschlecht:

□ männlich<br>
□ weiblich

**1.)** Vor wievielen Jahren hatten Sie eine Laryngektomie: ______

**2.)** Welches ist Ihre primäre Methode der alternativen Spracherzeugung:

**3.)** Nennen Sie den Grund für diese Wahl:

**4.)** Verwenden Sie eine sekundere Methode der alternativen Spracherzeugung?
Wenn ja, welche:

**5.)** Falls Sie einen Elektro-Larynx verwenden,
seit wievielen Jahren haben Sie Erfahrung im Umgang damit: ______

## B) Fragen zum handelsüblichen Elektro-Larynx:

**1.)** Was sind im alltäglichen Leben auftretende Probleme
bei der Verwendung eines handelsüblichen Elektro-Larynx:

**2.)** Wo sehen Sie, bei mittels Elektro-Larynx erzeugter Sprache,
einen Verbesserungsbedarf:

## C) Fragen zum Freihand-Elektro-Larynx:

**1.)** Würden Sie dieses Gerät im alltäglichen Leben verwenden:

□ Ja<br>
□ Nein

**2.)** Wo sehen Sie Probleme hinsichtlich dieses Gerätes:

**3.)** Wo sehen Sie Verbesserungsbedarf hinsichtlich dieses Gerätes:

**4.)** Was sind für Sie die positiven Aspekte dieses Gerätes:

**5.)** Wie schätzen Sie die Schwierigkeit ein,
mit diesem Gerät sprechen zu ”lernen”:

□ sehr einfach □ einfach □ mittel □ schwierig □ sehr schwierig

**6.)** Wie schätzen Sie die Benutzerfreundlichkeit ein:

□ sehr gut □ gut □ mittel □ schlecht □ sehr schlecht

**7.)** Wie würden Sie die Sprachverständlichkeit bewerten:

□ sehr gut □ gut □ mittel □ schlecht □ sehr schlecht

**8.)** Wie würden Sie die ”Natürlichkeit” der erzeugten Sprache bewerten:

□ sehr gut □ gut □ mittel □ schlecht □ sehr schlecht

**9.)** Wie würden Sie den Sprachfluss der erzeugten Sprache bewerten:

□ sehr gut □ gut □ mittel □ schlecht □ sehr schlecht

## D) Vergleich zwischen Freihand-Elektro-Larynx und handelsüblichen Elektro-Larynx:

**1.)** Wie schätzen Sie die Benutzerfreundlichkeit im Vergleich zu
einem handelsüblichen Elektro-Larynx ein:

□ viel besser □ besser □ gleich □ schlechter □ viel schlechter

**2.)** Wie würden Sie die Sprachverständlichkeit im Vergleich zu
einem handelsüblichen Elektro-Larynx bewerten:

□ viel besser □ besser □ gleich □ schlechter □ viel schlechter

**3.)** Wie würden Sie die ”Natürlichkeit” der erzeugten Sprache im Vergleich zu
einem handelsüblichen Elektro-Larynx bewerten:

□ viel besser □ besser □ gleich □ schlechter □ viel schlechter

**4.)** Wie würden Sie den Sprachfluss der erzeugten Sprache im Vergleich zu
einem handelsüblichen Elektro-Larynx bewerten:

□ viel besser □ besser □ gleich □ schlechter □ viel schlechter

**5.)** Wo sehen Sie die Vorteile im Vergleich zu einem handelsüblichen Elektro-Larynx:

# 6 Conclusion, discussion and outlook

The final Chapter of this thesis examines the sufficiencies and deficits of the ELCS in general and in terms of TD and classification. Additionally, a recommendation for using this device is given. The usability and functionality of the ELCS are dependent on many things and some of them could of course be improved.

In general one can say that ON/OFF-control of the ELCS all in all works well and the ON and OFF time-lags are all, except some outliers, in an acceptable range. Even though, the median time-lags are higher than for the work of Fuchs [Fuchs, 2015], where the electrodes were placed on the neck for ON/OFF-control, they are acceptable for this kind of application.
There is a tradeoff between time-lags and error occurrences which both are dependent on TD and probably the used classifier. For the smaller TD sets as MPC and MPV which contain only information of the individual TPs there are less errors, but the time-lags are a small amount higher as for the bigger TD sets like MPW and MPA which contain either information of all TPs except the actual one or information of all TPs including the actual TP. However the difference of time-lags can not necessarily be traced back to the different types of TD as there is no significance visible except for the median OFF-time-lag when using MPW TD, then this value is significantly better as for MPC and MPV TD.
Since there was not enough time to find the optimal parameter settings for all the different classifiers, time-lag and error values probably could be improved by an extensive testing and classifier evaluation.
Regarding TD there are some important things to mention. The best case for the end-user would be when there is no need to record his or her specific training data, but in exchange it would be necessary to provide a large TD set including data from many different people, because everyones muscle structure is different. The implementation of the classifiers and the length of the TD set are occasionally responsible for the processing time of the device when new sEMG data comes in. The arising problem could be, on the one hand that classification would not work as fast for such a large TD set, and on the other hand classification would maybe not be that robust for the individual user.
Investing some time in recording user specific TD before using the device perhaps would improve robustness of classification and lead to smaller time-lags and less error occurrences by the expense of needed time before the device can be utilised by the end-user at all.
Technical problems which appeared during working on this thesis were that *Python* sometimes crashed when running the ELCS on the Raspberry Pi. The reasons for this crashes are not

known, but the error message was an IOError in connection with *pyaudio*. This problem was solved by unplugging the USB-interface, replug it again and starting the program new. Also it was observed that the program sometimes got stuck, which means that either there was always an excitation signal, when it got stuck during fist gesture, or it was not able to produce an output anymore in the other case. This problem appeared with different frequency for the various classifiers.

### Recommendation of use for the ELCS

The ELCS developed in this work can be utilized in different ways regarding TD. The easiest way for the end-user, meaning a person which took not part in the recording of TD, would be to use it with MPA TD set. This set contains information from 13 different TPs and therefore there is no need for the end-user to record individual TD. However, as this TD set contains no information about the end-user, one can expect almost the same results concerning time-lags and errors, as can be seen for MPW data in the general usage test. Hence, it is recommended for the end-user to record his or her own individual MPV TD set by the expense of some time before the device can be used. The recording of such a MPV TD set as it was used in the evaluation of the system takes about 20 minutes, if there is an instructor who guides the end-user through this process.

### Outlook

During finalization of this thesis, it came out that there are a some things regarding the ELCS which can be done in future works. Such things are:

- Improve training data
  - Composition of training data
  - Recording of training data
  - Length of training data
  - Type of training data
- Improve classifiers
  - Parameters of classifiers
  - Try other classification methods
- Add sleep mode
- Extensive $f_0$ variation testing

The tasks of future works could be to improve the TD sets regarding the length of TD sets and their type. One could record a user specific TD set with more samples for each class or different lengths for each class. Perhaps TD of an individual user could be improved by recording smaller sets on different days and times of day and appending them together. Another approach would be to try bigger sets of TD meaning to append the individual TD sets of much more persons than it was the case in this work.

The second, not less important, task for further works is to improve the classifiers in terms of their parameters and to try different classification methods and invest more time for the evaluation of these.

Additionally, it would be beneficial to implement a feature that offers the possibility to put the device in sleep mode, where it produces no excitation signal even if a fist gesture is made.

This could be achieved either by another specific gesture, which sets the system in sleep mode or wakes it up, respectively. Another possibility would be for example making a fist gesture three times within a period of two seconds, to get the divice in sleep mode, or to wake it up.

As the evaluation of the $f_0$ variation feature in this work was very theoretical, it would make sense to evaluate the functionality of this feature more extensive. This could be managed by creating a more realistic user scenario like real conversations where the TP has to vary the fundamental frequency. Therefor it would be necessary to perform a training program for the TPs to get them used to this feature.

# A Appendix

In this appendix chapter additional figures and tables of the results of several experiments and evaluations, examined during working on this thesis can be seen.

## A.1 Training data evaluation results

Chapter 4.2 deals with the length of TD and in Figure 4.1 one can see the different statistical performance measures for the KNN classifier. Figure A.1 shows this statistical performance measures as a function of TD length for all classifiers and the MPC TD set. Figure A.3 shows the results for MPV data and in Figure A.2 and A.4 the same results are depicted, but without showing the OCSVM classifier.

The evaluation of the individual TD from each TP can be seen in Figures A.5 to A.7. Compared to boxplot 4.2 in Chapter 4.4 now the statistical performance measures are illustrated separately for each TP.

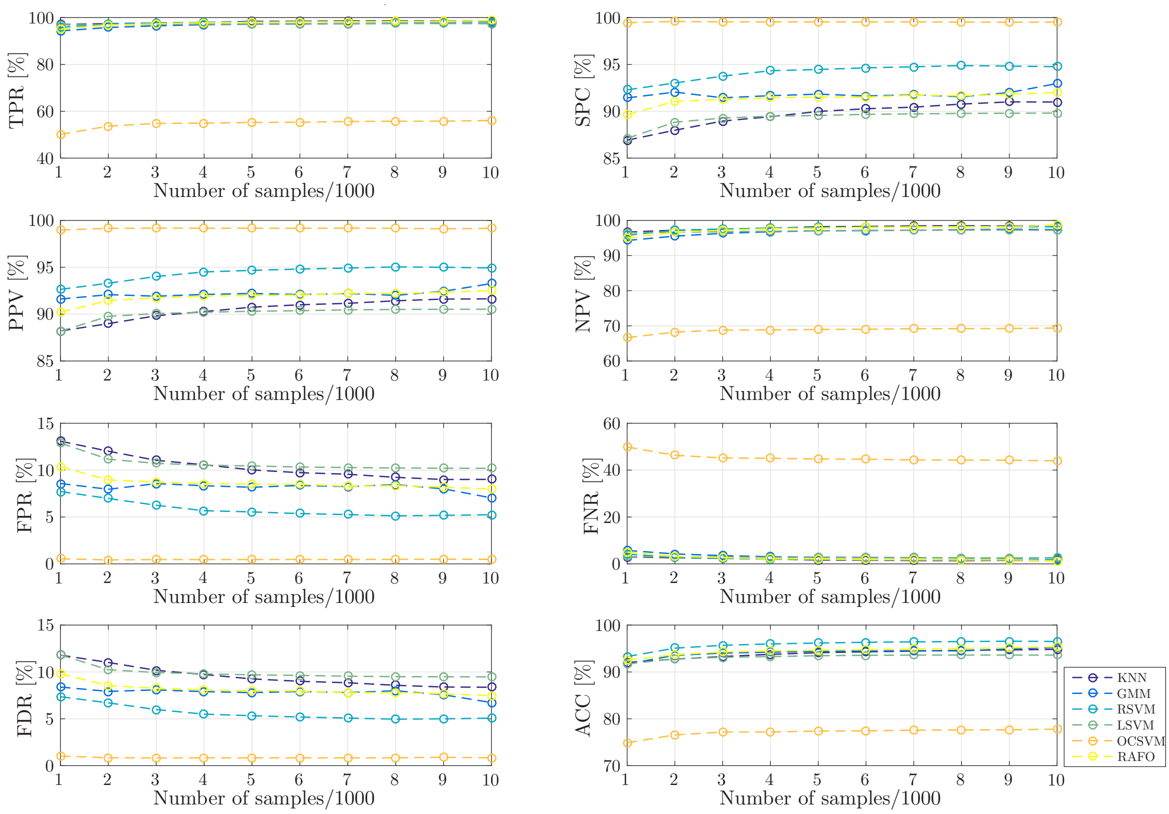


*Figure A.1: Statistical performance measures as a function of TD length for the MPC TD set.*

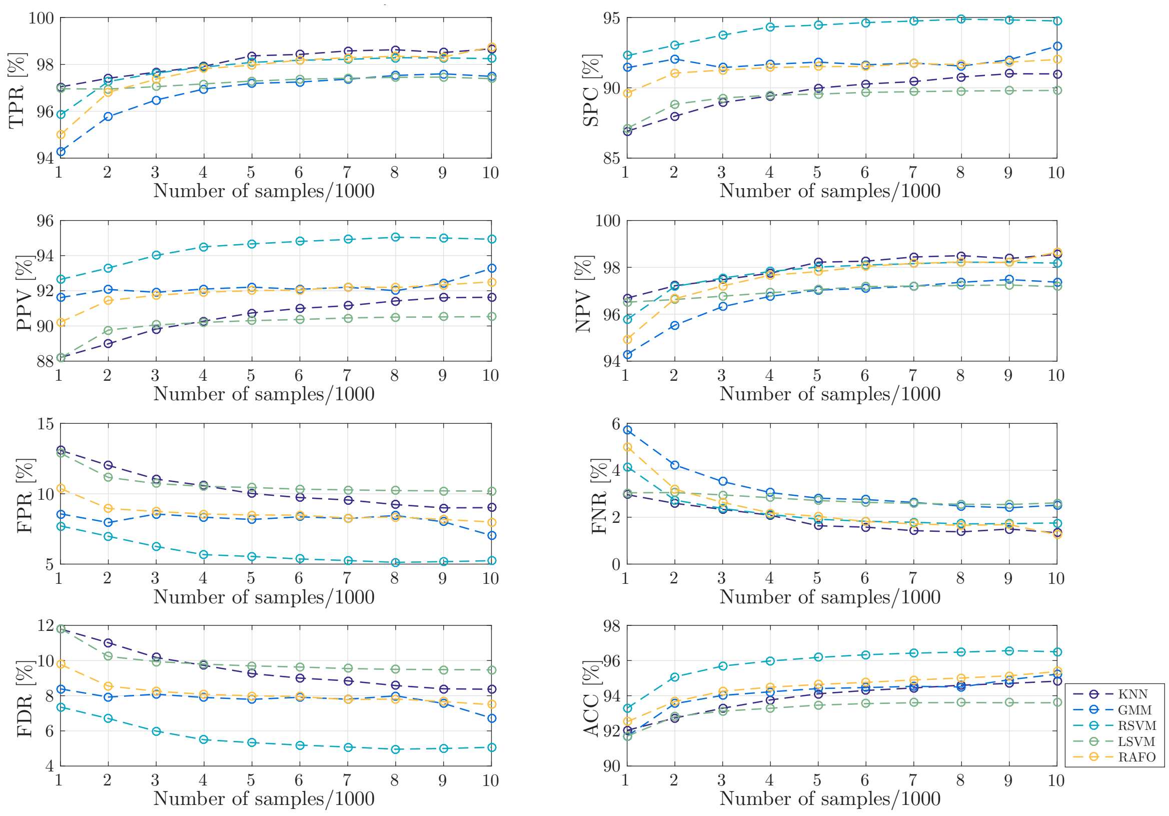


*Figure A.2: Statistical performance measures as a function of TD length for the MPC TD set and without OCSVM classifier.*

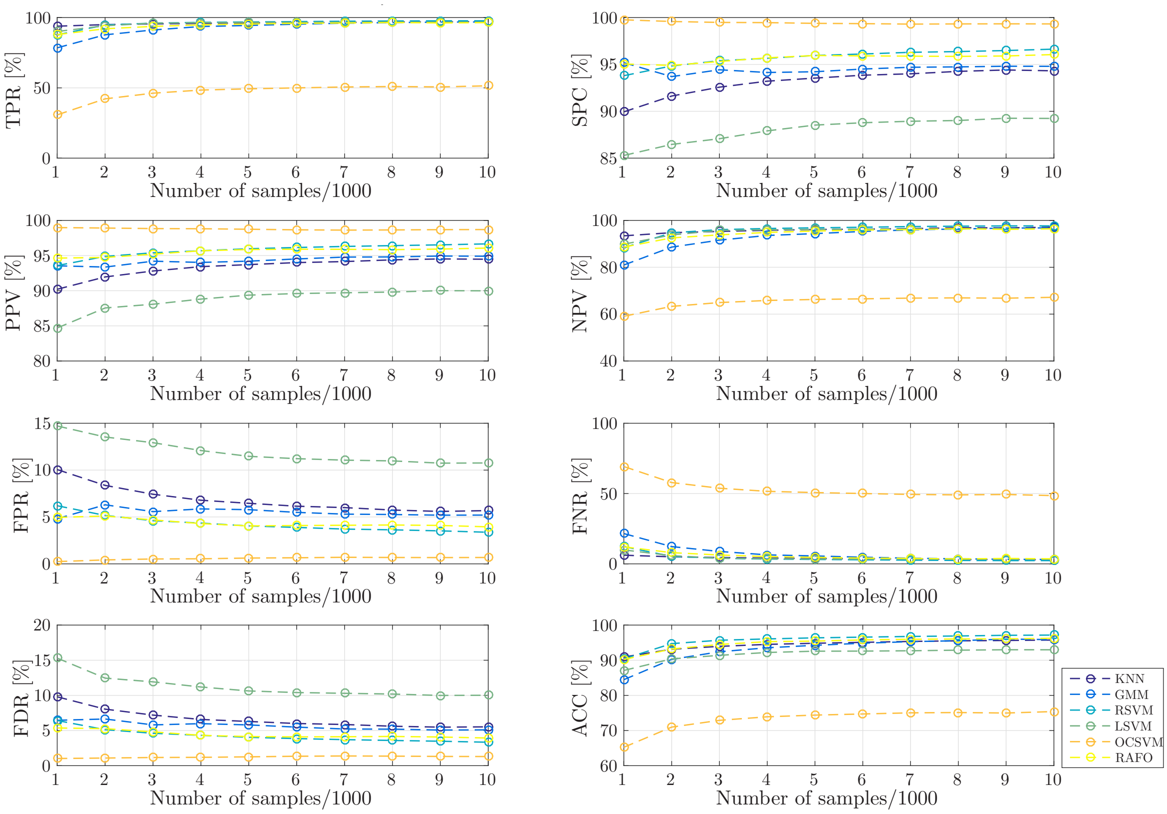


*Figure A.3: Statistical performance measures as a function of TD length for the MPV TD set.*

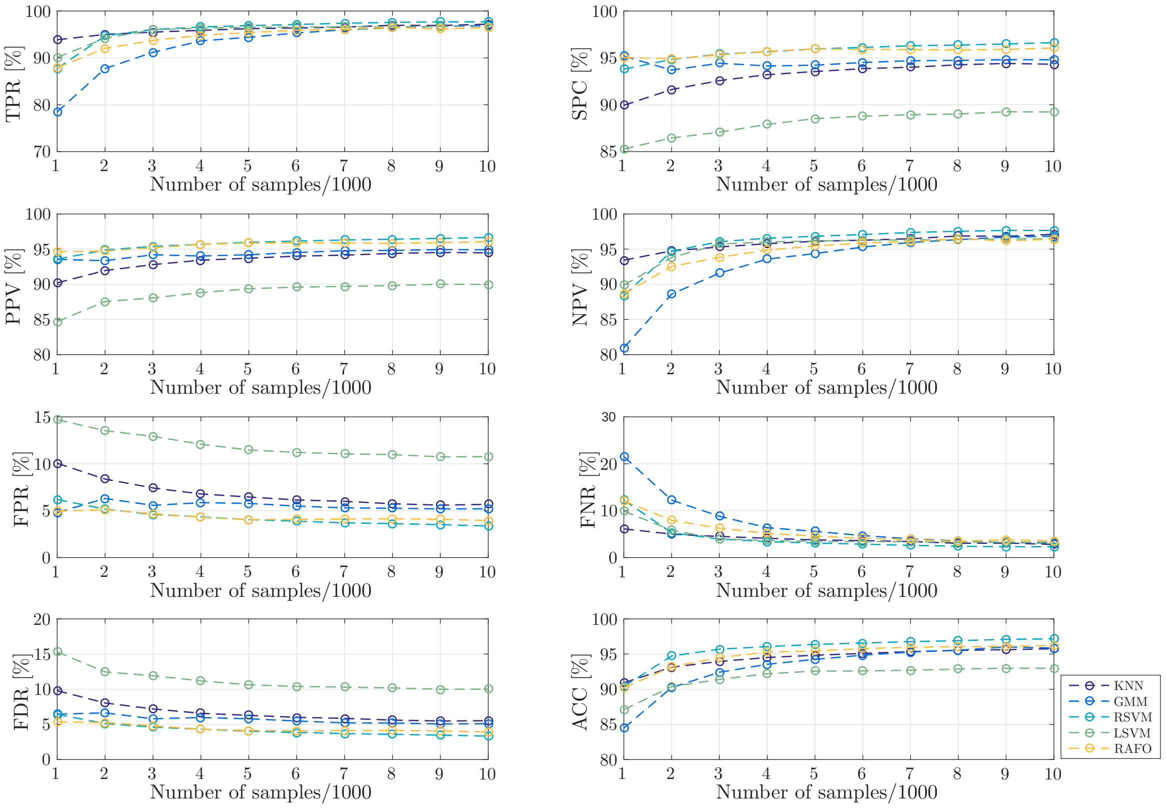


*Figure A.4: Statistical performance measures as a function of TD length for the MPV TD set and without OCSVM classifier.*

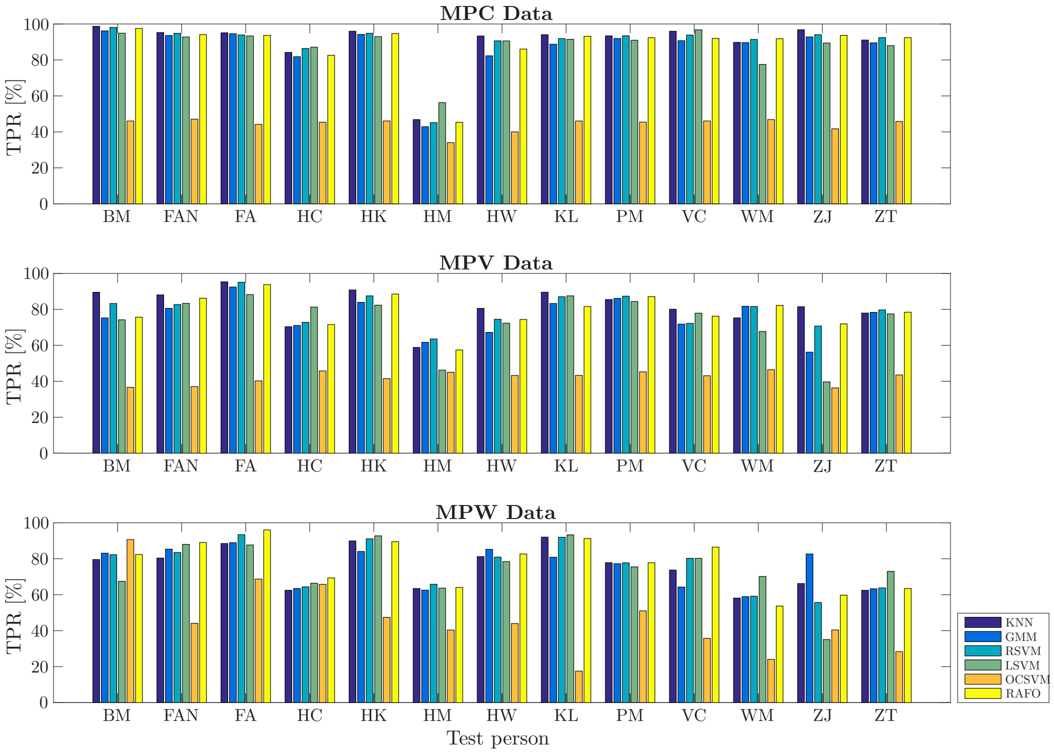


*Figure A.5: Sensitivity (TPR) for each TP, all classifiers and for MPC, MPV and MPW TD sets.*

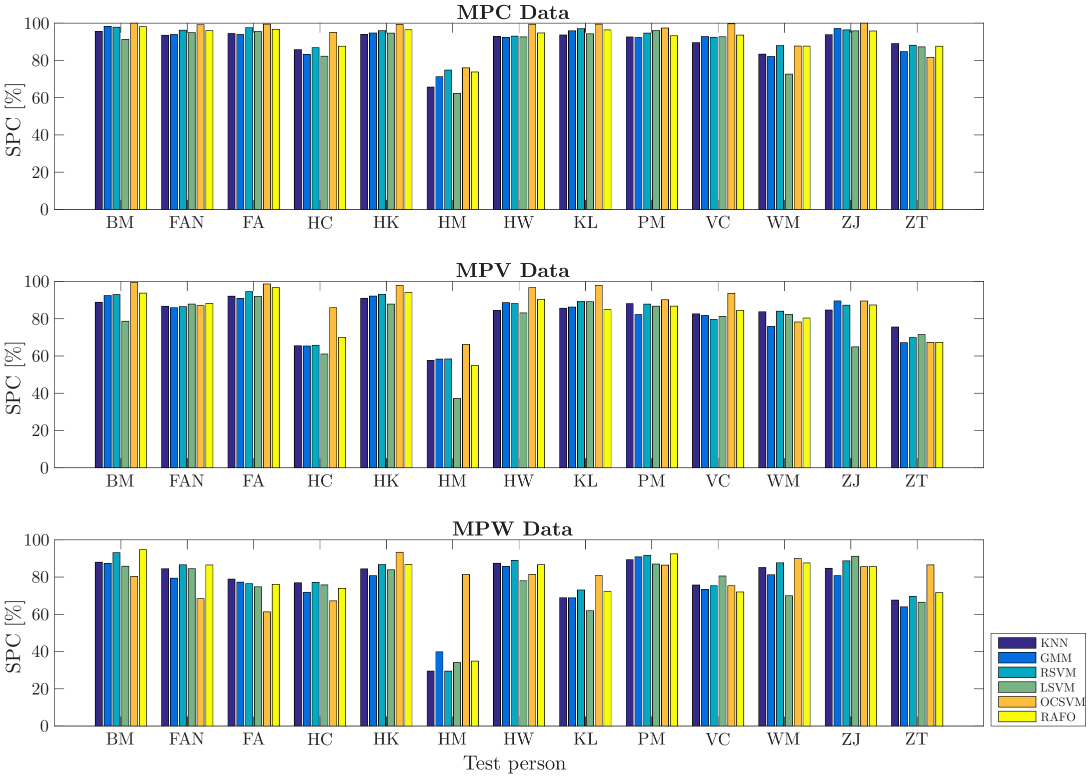


*Figure A.6: Specificity (SPC) for each TP, all classifiers and for MPC, MPV and MPW TD sets.*

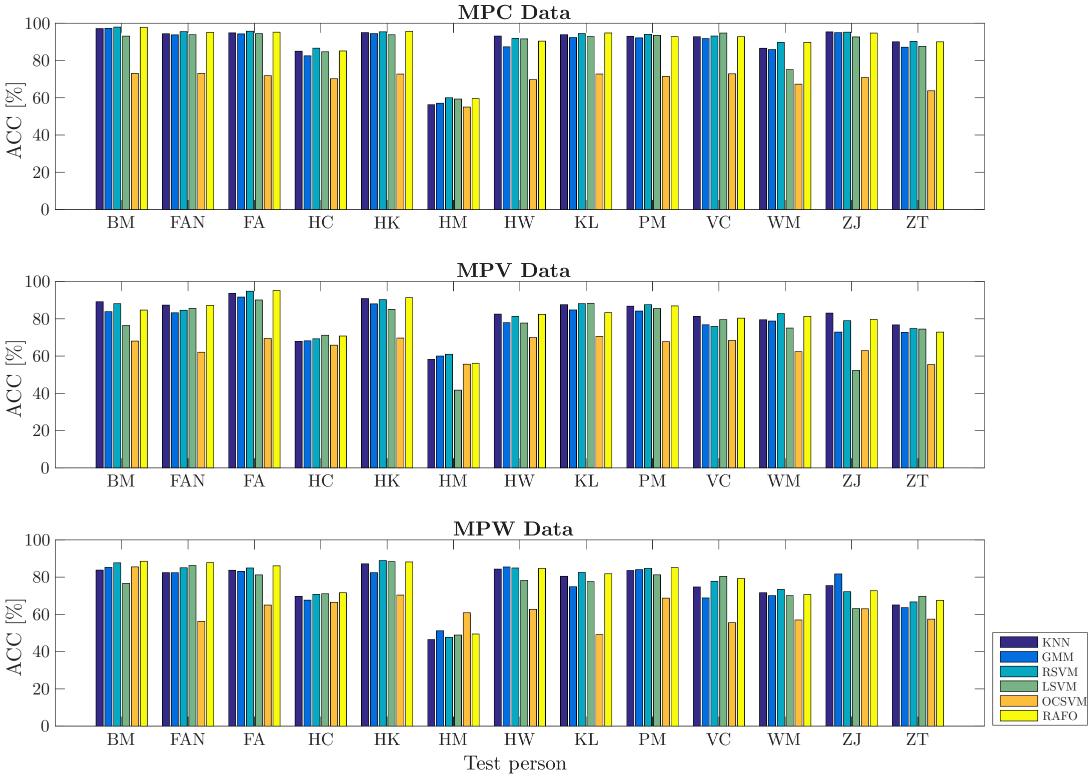


*Figure A.7: Accuracy (ACC) for each TP, all classifiers and for MPC, MPV and MPW TD sets.*

## A.2 Classifier evaluation results

Table A.1 shows the results of the classifier testing through simple A vs. B test examined by two TPs in more detail. In contrast to Table 4.4, this Table shows the results for each classifier compared to all other classifiers. It shows how many times classifier A was ranked better, equal or worse than classifier B.
In Figure A.8 this results are graphically illustrated.

During the examination of this B vs. A test the program crashed a few times, the number of this crashes and which classifier was used when it crashed, are listed in Table A.2. Sometimes *Python* crashed because something with *pyaudio*-package and the USB sound interface went wrong, this is denoted in "other crashes".

| **Functionality overview** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **A vs. B** | | **TP 1:** | | | **TP 2:** | | |
| **A** | **B** | + | - | = | + | - | = |
| KNN | GMM | 3 | 0 | 0 | 3 | 0 | 0 |
| KNN | RSVM | 0 | 3 | 0 | 3 | 0 | 0 |
| KNN | OCSVM | 3 | 0 | 0 | 3 | 0 | 0 |
| KNN | LSVM | 1 | 1 | 1 | 3 | 0 | 0 |
| KNN | RAFO | 3 | 0 | 0 | 3 | 0 | 0 |
| GMM | KNN | 0 | 3 | 0 | 0 | 3 | 0 |
| GMM | RSVM | 0 | 2 | 1 | 0 | 1 | 2 |
| GMM | OCSVM | 3 | 0 | 0 | 2 | 1 | 0 |
| GMM | LSVM | 0 | 1 | 2 | 1 | 0 | 2 |
| GMM | RAFO | 3 | 0 | 0 | 3 | 0 | 0 |
| RSVM | KNN | 3 | 0 | 0 | 0 | 3 | 0 |
| RSVM | GMM | 2 | 0 | 1 | 1 | 0 | 2 |
| RSVM | OCSVM | 3 | 0 | 0 | 1 | 0 | 2 |
| RSVM | LSVM | 2 | 0 | 1 | 1 | 0 | 2 |
| RSVM | RAFO | 3 | 0 | 0 | 3 | 0 | 0 |
| OCSVM | KNN | 0 | 3 | 0 | 0 | 3 | 0 |
| OCSVM | GMM | 0 | 3 | 0 | 1 | 2 | 0 |
| OCSVM | RSVM | 0 | 3 | 0 | 0 | 1 | 2 |
| OCSVM | LSVM | 0 | 3 | 0 | 2 | 0 | 1 |
| OCSVM | RAFO | 0 | 3 | 0 | 3 | 0 | 0 |
| LSVM | KNN | 1 | 1 | 1 | 0 | 3 | 0 |
| LSVM | GMM | 1 | 0 | 2 | 1 | 0 | 2 |
| LSVM | RSVM | 0 | 2 | 1 | 0 | 1 | 2 |
| LSVM | OCSVM | 3 | 0 | 0 | 0 | 2 | 1 |
| LSVM | RAFO | 2 | 0 | 0 | 3 | 0 | 0 |
| RAFO | KNN | 0 | 3 | 0 | 0 | 3 | 0 |
| RAFO | GMM | 0 | 3 | 0 | 0 | 3 | 0 |
| RAFO | RSVM | 0 | 3 | 0 | 0 | 3 | 0 |
| RAFO | OCSVM | 3 | 0 | 0 | 0 | 3 | 0 |
| RAFO | LSVM | 0 | 3 | 0 | 0 | 3 | 0 |

*Table A.1: Results of classifiers tested by B vs. A test for each TP (number of how many times classifier A was better, equal or worse then classifier B for every classifier compared to all others).*

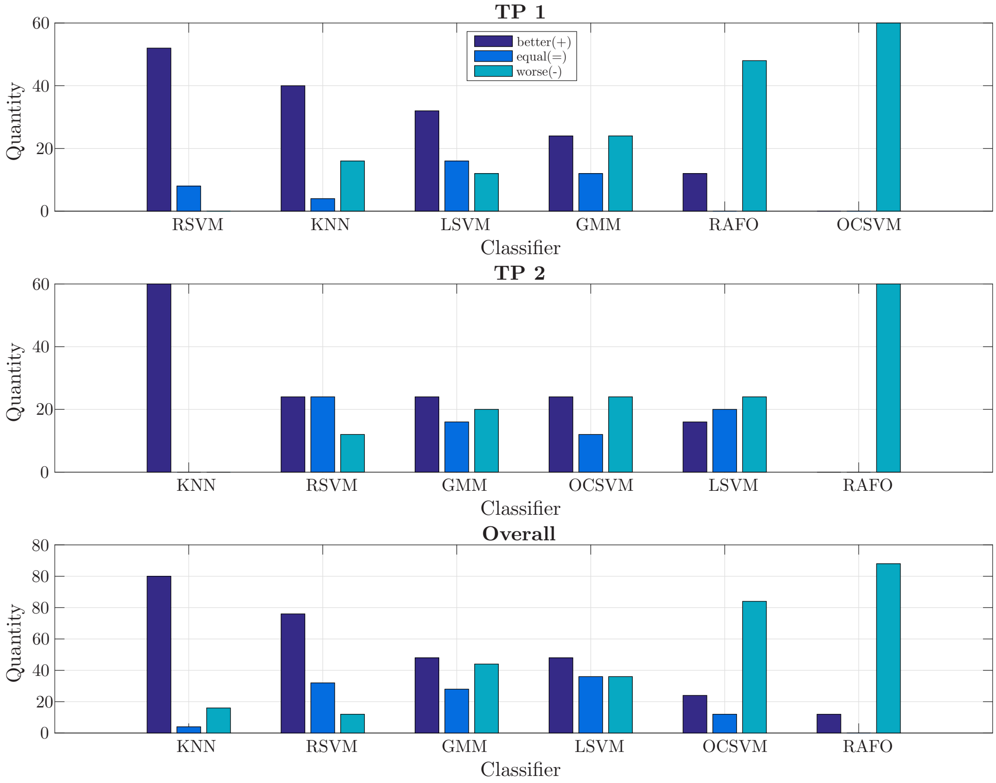


*Figure A.8: Classifier evaluation (shows number of chosen as better, equal or worse).*

| Number of program crashes: | | | |
|---|---|---|---|
| **Classifier:** | **TP 1:** | **TP 2:** | $\sum$ |
| KNN | 1 | 0 | 1 |
| GMM | 2 | 0 | 2 |
| RSVM | 0 | 0 | 0 |
| OCSVM | 0 | 0 | 0 |
| LSVM | 0 | 0 | 0 |
| RAFO | 3 | 0 | 3 |
| other crashes | 4 | 5 | 9 |

*Table A.2: Number of program crashes during classifier testing for each subject and overall.*

## A.3 Instructions for the general usage test of the EL-control-system

### For the instructor:

- Create a new test person (Speakers → Speaker... → Add)
- Set 'Force post recording phase' check mark (Project → Preferences... → Recording → Options → Force post recording phase)
- Make sure that everything in Python is set up correctly.
- Let the test person read the instructions.
- Let the test person try out the EL-control system for getting familiar with it.

# For the test person:

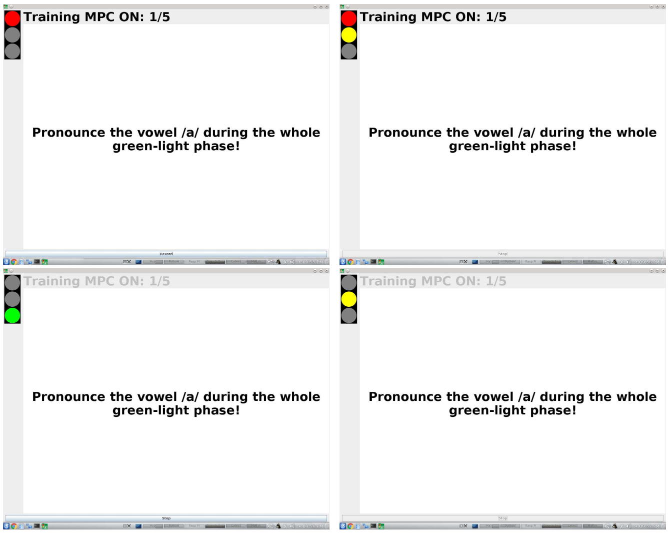
## The Tasks:

- **1.) ON/OFF task** (Time-lag and error evaluation)
- **2.) Critical gestures task** (False ON evaluation)
- **3.) Frequency variation task** ($f_0$ variation evaluation)

## In general:

Put on the MYO armband on your left arm in a way that the USB port shows to the wrist and the logo points upwards when you reach out your arm. For each task you will see traffic lights as shown in Figure A.9. Before every task there will be a short training phase.

- RED: You are allowed to do whatever you want with your arm.
- RED YELLOW: Prepare for the task, but **don't** do the task until you see the green light.
- GREEN: Do the task as long as you see the green light.
- YELLOW: Stop doing the task and do nothing with your arm.



Training MPC ON: 1/5

Pronounce the vowel /a/ during the whole green-light phase!

Training MPC ON: 1/5

Pronounce the vowel /a/ during the whole green-light phase!

Training MPC ON: 1/5

Pronounce the vowel /a/ during the whole green-light phase!

Training MPC ON: 1/5

Pronounce the vowel /a/ during the whole green-light phase!

*Figure A.9:* SpeechRecorder *[Draxler and Jänsch, 2004] traffic lights.*

The main goal of this test is to evaluate the functionality of the electro-larynx-control system in terms of ON/OFF-control and $f_0$ variation. ON/OFF control is accomplished by measuring and classifying the muscle activity of the forearm. When you make a fist the system should give an output and when you do something else with your hand there should be no output. $f_0$ variation is controlled via the vertical angle of the arm, a downward hanging arm leads to a low $f_0$, when the arm is moved upwards $f_0$ gets higher. If you want to say something than you have to make a fist and form your vocal tract in a particular way. The EL-control system will generate an excitation signal for the electrolarynx mounted on your neck. This excitation signal in combination with the formed vocal tract enables you to speak.

### 1.) ON/OFF task:

When light turns to green imitate to say the vowel /a/ during the whole green light phase. (Make a fist when you see the green light and form your vocal tract in a way as you would say the vowel /a/. Sustain the vowel /a/ until green light phase ends.)

### 2.) Critical gestures task:

For this task, you should perform the following activities. Please **don't** make a fist.

- move fingers

- scratch the back of your head
- grab the pencil case
- turn over a few pages of the book

You have 20 seconds for doing the above gestures.

**3.) Frequency variation task:**
For this task imitate a given $f_0$ contour (see Figure A.10) with your arm. Horizontal arm means 0°, −80° → low $f_0$, +20° → high $f_0$. The recording will be stopped manually by the instructor when you finished imitating the contour.

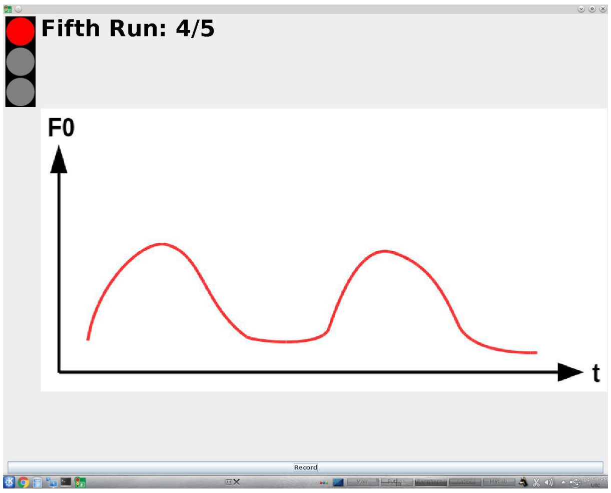


*Figure A.10: Given $f_0$ contour for imitation.*

## A.4 General usage test results

### $f_0$ evaluation

The results for every single TP of the $f_0$ variation evaluation task in the general usage test can be seen in Figures A.11 to A.20.

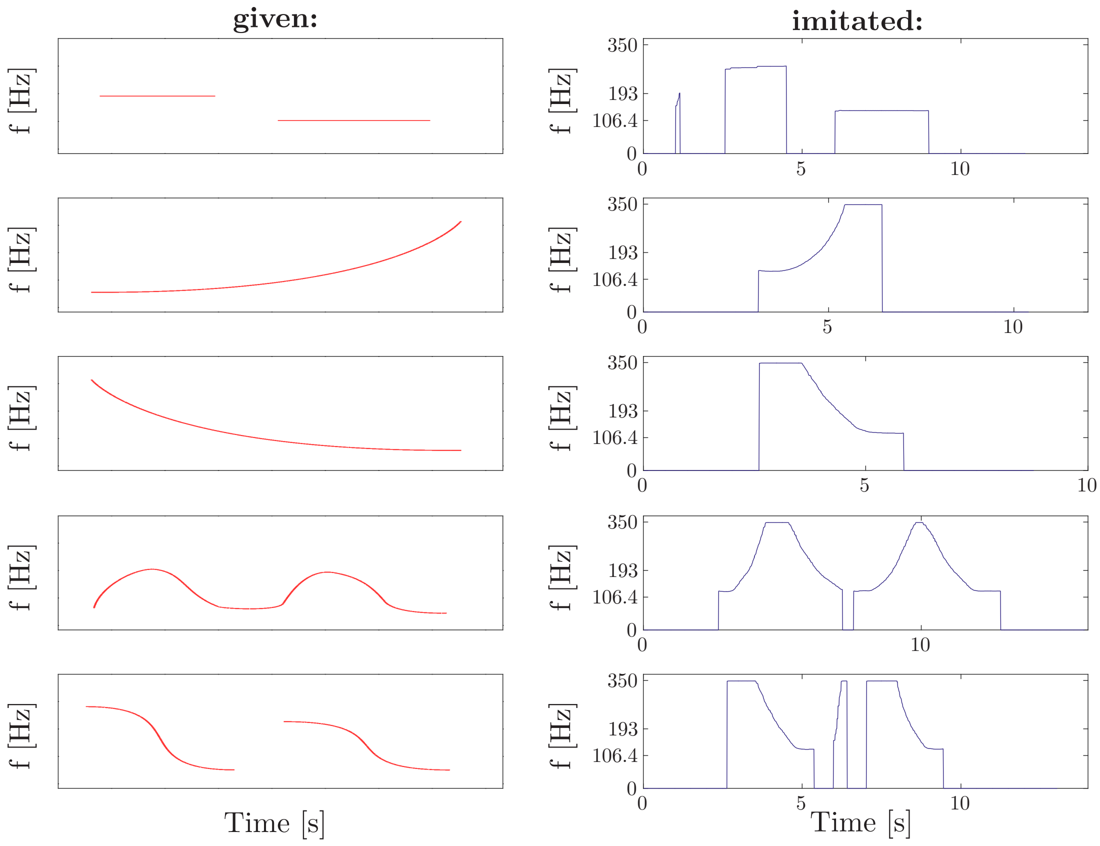


*Figure A.11: $f_0$ contours for test person FA*

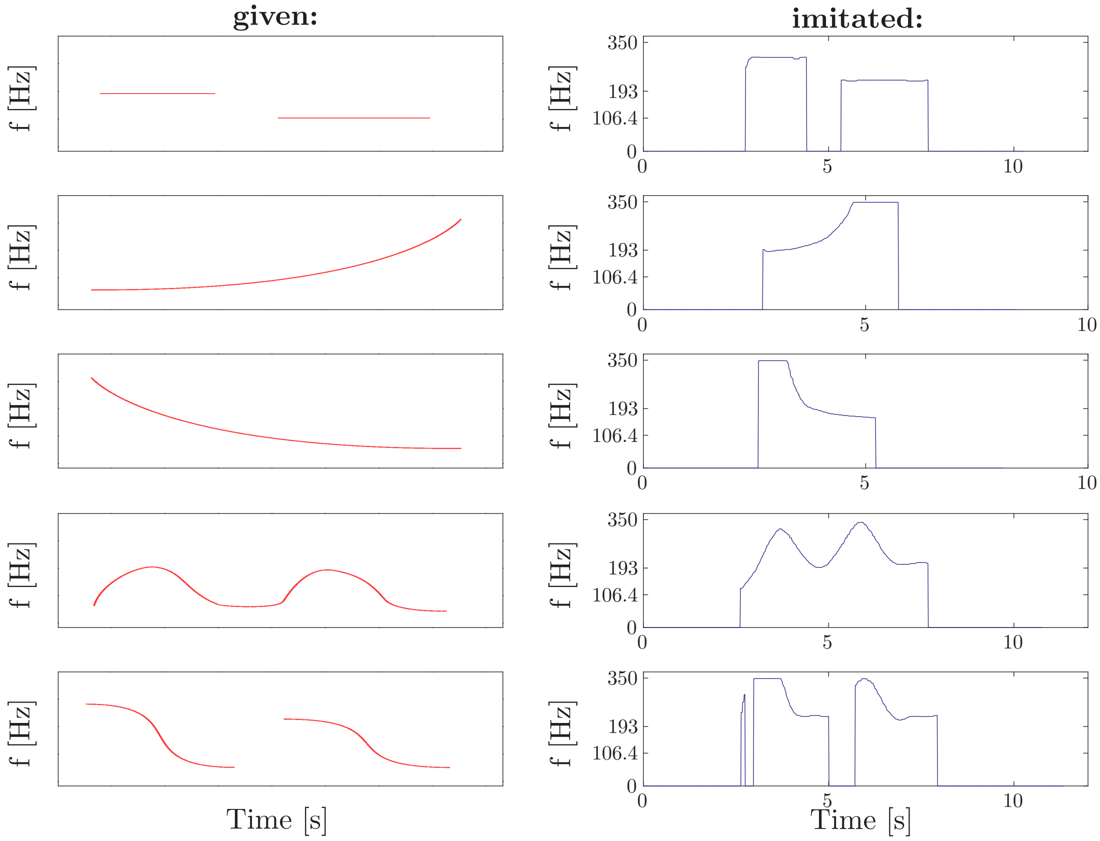


*Figure A.12: $f_0$ contours for test person ZJ*

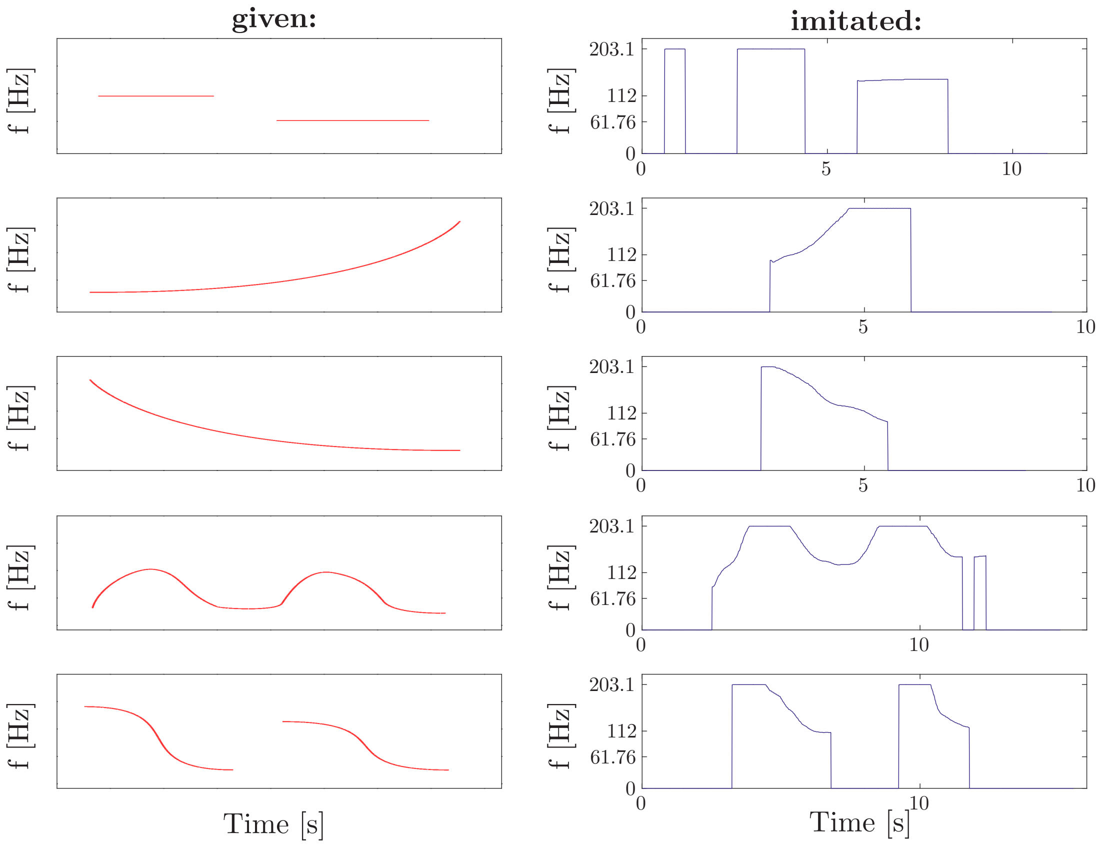


*Figure A.13: $f_0$ contours for test person WM*

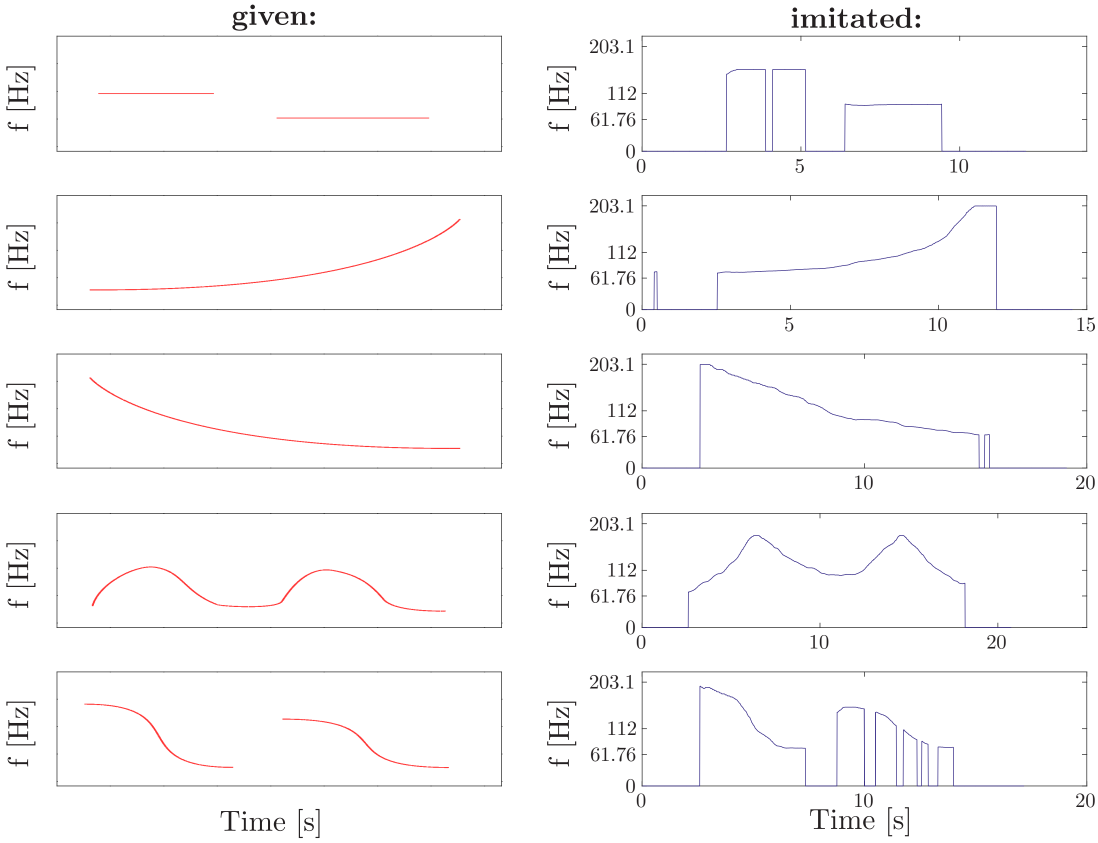


*Figure A.14: $f_0$ contours for test person PM*

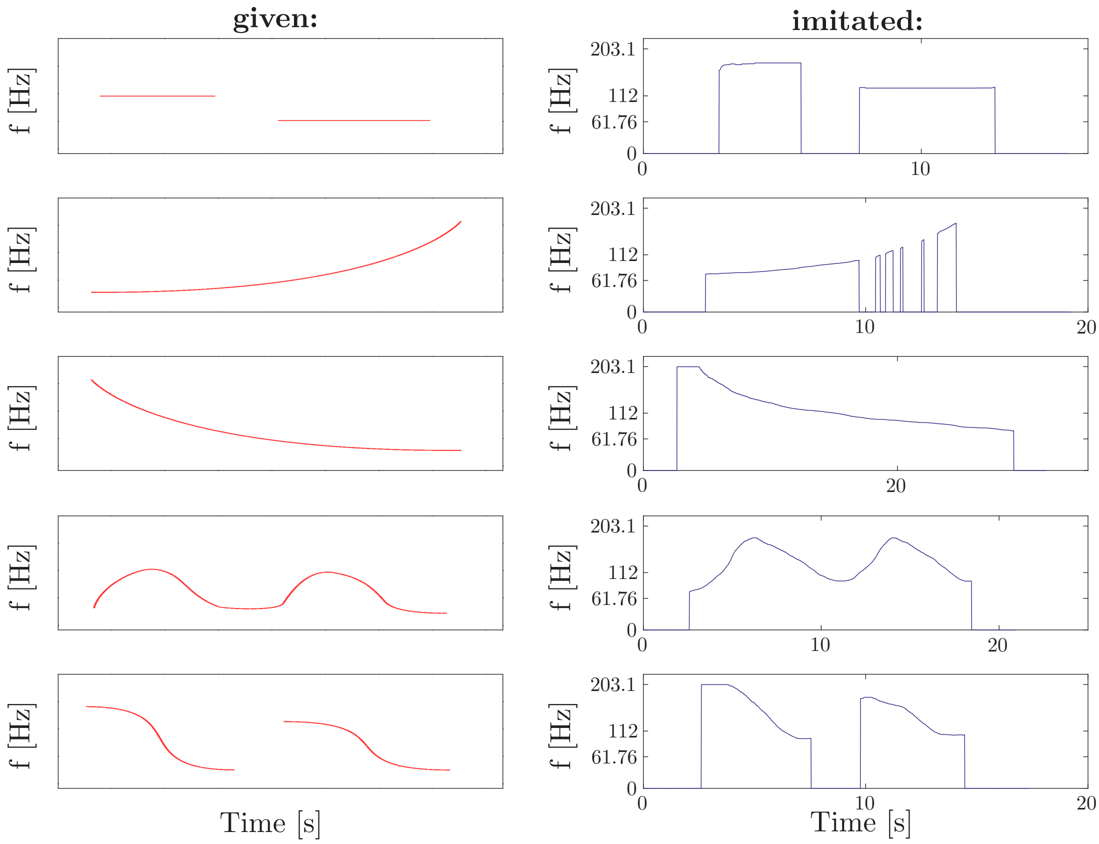


*Figure A.15: $f_0$ contours for test person KL*

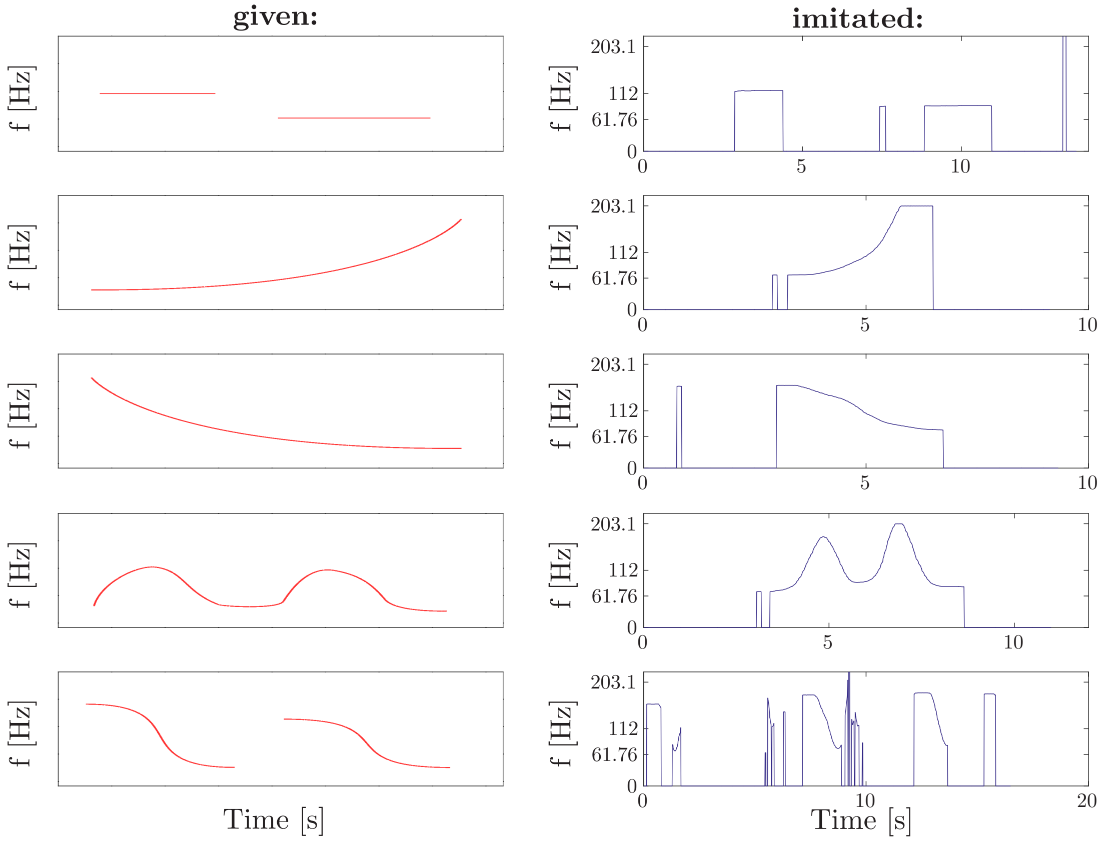


*Figure A.16: $f_0$ contours for test person BM*

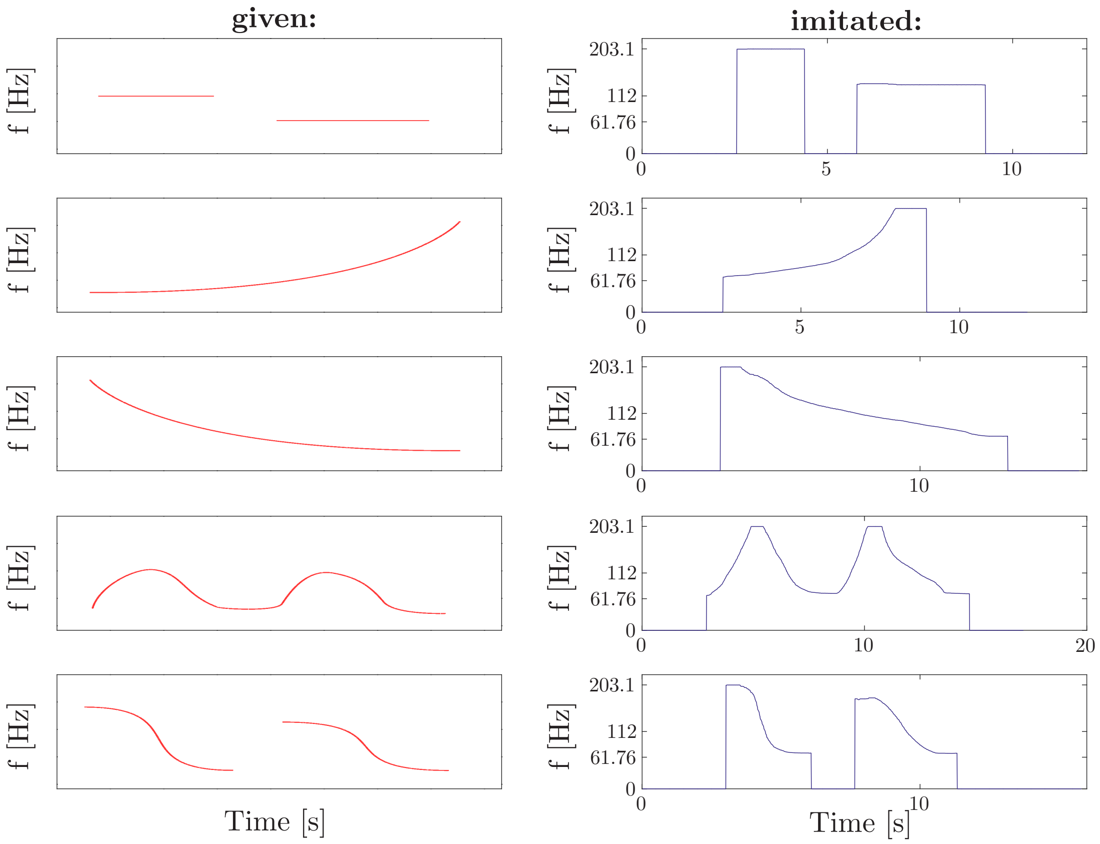


*Figure A.17: $f_0$ contours for test person VC*

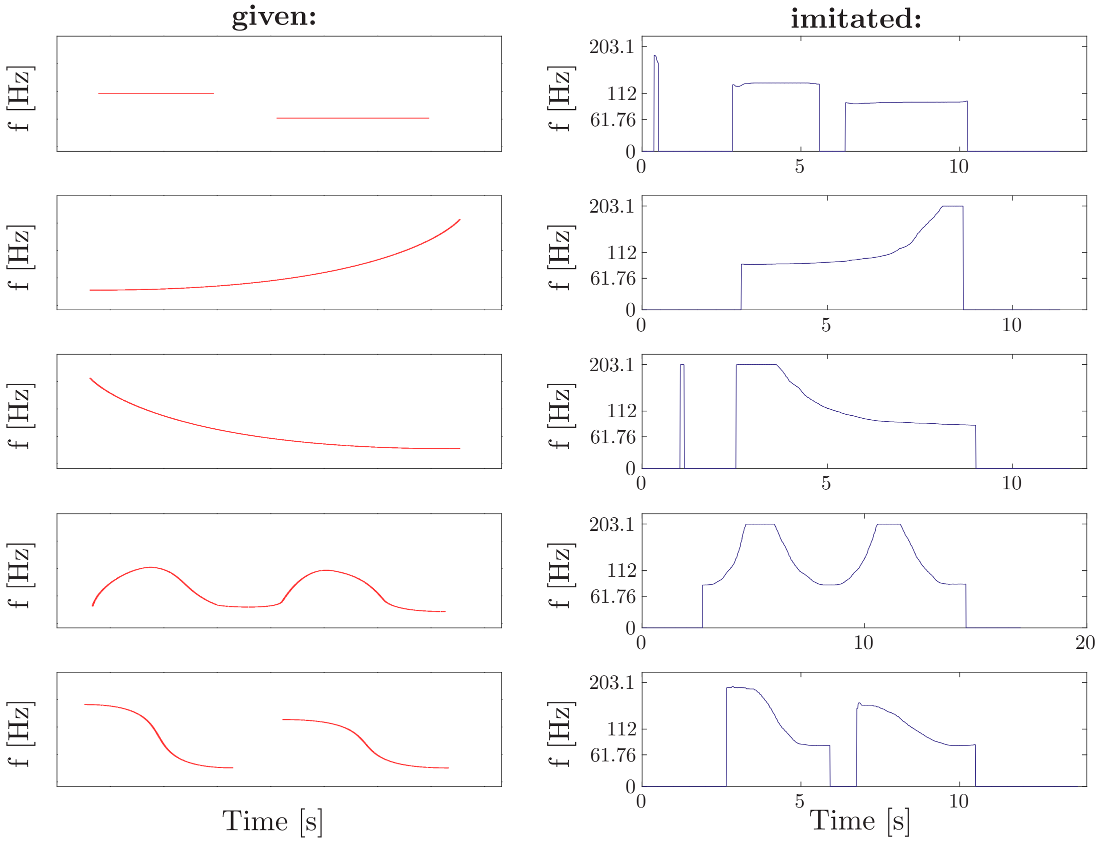


*Figure A.18: $f_0$ contours for test person FAN*

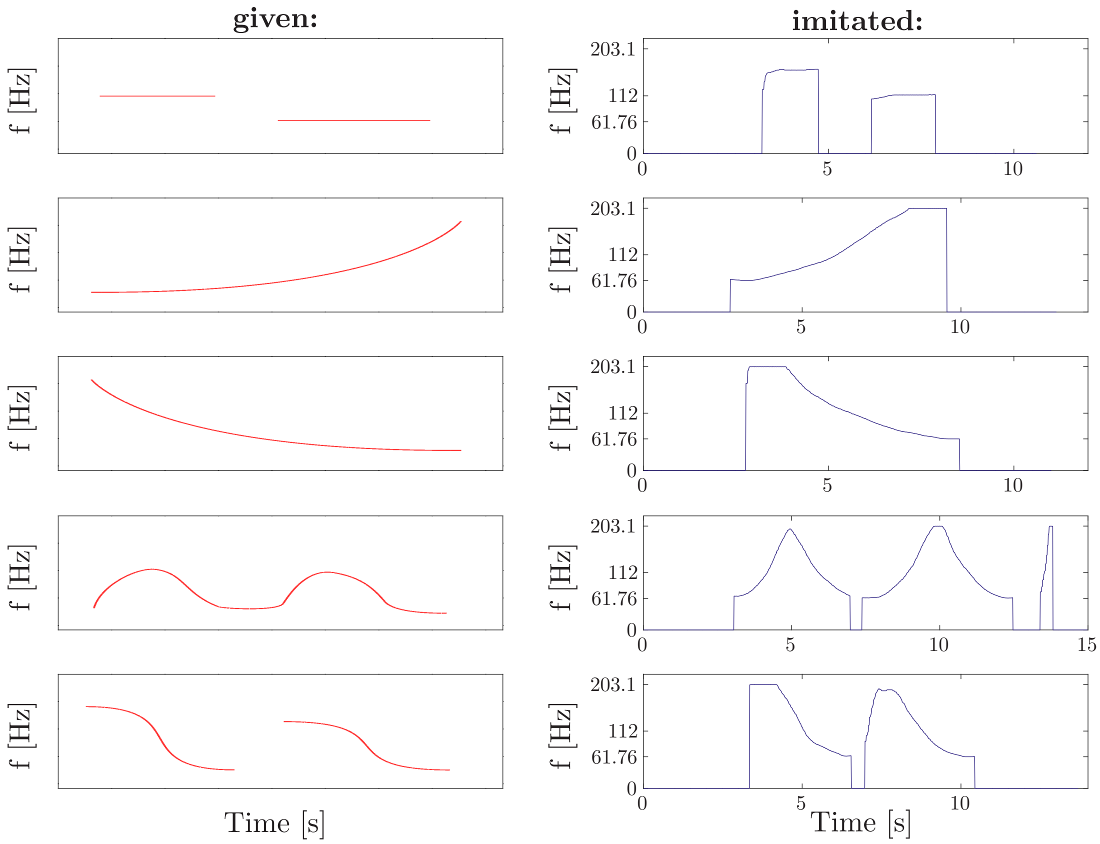


*Figure A.19: $f_0$ contours for test person HW*

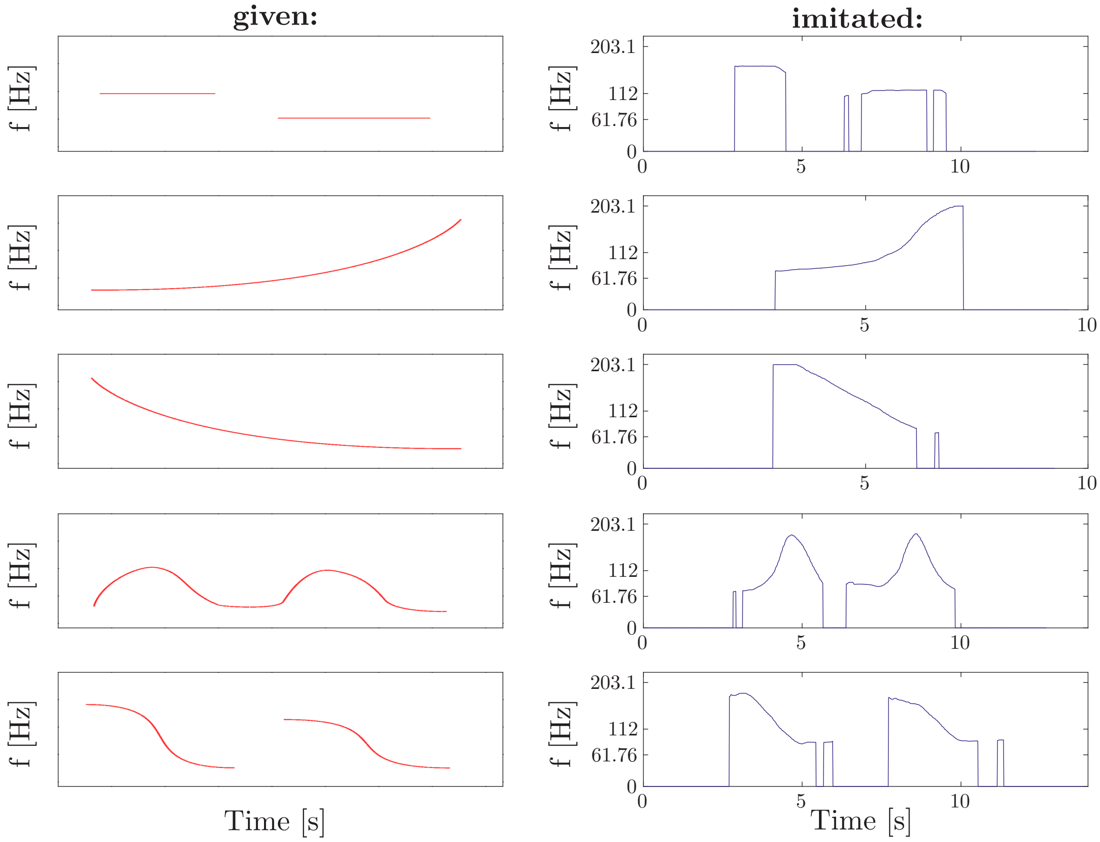


*Figure A.20: $f_0$ contours for test person HM*

# Bibliography

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[Cunningham and Delany, 2007] Cunningham, P. and Delany, S. J. (2007). K-nearest neighbour classifiers. Technical report, University College Dublin.

[Draxler, 2004] Draxler, C. (2004). *SpeechRecorder Quick Start and User Manual*. Institut für Phonetik und Sprachverarbeitung Universität München.

[Draxler and Jänsch, 2004] Draxler, C. and Jänsch, K. (2004). Speechrecorder – a universal platform independent multi-channel audio recording software. In *Proceedings of the IV. International Conference on Language Resources and Evaluation (LREC)*, Lisbon, Portugal.

[Fant et al., 1985] Fant, G., Liljencrants, J., and Lin, Q.-G. (1985). A four parameter model of glottal flow. Technical Report STL-QPSR Nos. 2-3, Royal Institute of Technology, Stockholm, Sweden.

[Fuchs, 2015] Fuchs, A. K. (2015). *The Bionic Electro-Larynx Speech System – Challenges, Investigations, and Solutions*. PhD thesis, Graz University of Technology.

[Goldstein et al., 2004] Goldstein, E., Heaton, J., Kobler, J., Stanley, G., and Hillman, R. (2004). Design and implementation of a hands-free electrolarynx device controlled by neck strap muscle electromyographic activity. *IEEE Transactions on Biomedical Engineering*, 51(2):325–332.

[Hashiba et al., 2007] Hashiba, M., Sugai, Y., Izumi, T., Ino, S., and Ifukube, T. (2007). Development of a wearable electro-larynx for laryngectomees and its evaluation. In *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pages 5267–5270.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference and prediction*. Springer New York Inc.

[Haykin, 2009] Haykin, S. (2009). *Neural Networks and Learning Machines*. Number Bd. 10 in Neural networks and learning machines. Prentice Hall.

[Heaton et al., 2011] Heaton, J., Robertson, M., and Griffin, C. (2011). Development of a wireless electromyographically controlled electrolarynx voice prosthesis. In *Proceedings of Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 5352–5355, Boston, MA, USA.

[Jain et al., 2015] Jain, A., Bansal, R., Kumar, A., and Singh, K. (2015). A comparative study of visual and auditory reaction times on the basis of gender and physical activity levels of medical first year students. *International Journal of Applied and Basic Medical Research*, 5(2):124–127.

[Kikuchi and Kasuya, 2004] Kikuchi, Y. and Kasuya, H. (2004). Development and evaluation of pitch adjustable electrolarynx. *Speech Prosody*, pages 1–4.

[Kumar et al., 2004] Kumar, S., Kumar, D. K., Alemu, M., and Burry, M. (2004). EMG Based Voice Recognition. *Intelligent Sensors, Sensor Networks and Information Processing Conference*, pages 593–597.

[Matsui et al., 2014] Matsui, K., Kimura, K., Pérez, A., Rodríguez, S., and Corchado, J. M. (2014). Development of electrolarynx by multi-agent technology and mobile devices for prosody control. In *Highlights of Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection Communications in Computer and Information Science*, volume 430, pages 54–65. Springer Science + Business Media.

[sklearn, 2012] sklearn (2012). scikit-learn, machine learning in python user guide. `http://scikit-learn.org/stable/user_guide.html`.

[Statistics and Breiman, 2001] Statistics, L. B. and Breiman, L. (2001). Random forests. In *Machine Learning*, pages 5–32.

[Stepp, 2008] Stepp, C. E. (2008). Electromyographic Control of Prosthetic Voice after Total Laryngectomy. Master's thesis, Massachusetts Institute of Technology.

[Uemi et al., 1994] Uemi, N., Ifukube, T., Takahashi, M., and Matsushima, J. (1994). Design of a new electrolarynx having a pitch control function. In *3rd IEEE Proceeding of International Workshop on Robot and Human Communication (RO-MAN)*, pages 198–203, Nagoya.

[Vary and Martin, 2006] Vary, P. and Martin, R. (2006). *Digital Speech Transmission: Enhancement, Coding and Error Concealment*. John Wiley & Sons.