Project No. 507424
ALLADIN
Natural Language Based Decision Support in Neuro-rehabilitation

SPECIFIC TARGETED RESEARCH PROJECT
PRIORITY 2.3.1.11

Deliverable 4.2:
Document containing the description of the developed software tools and the extracted knowledge, the discovered patterns and data associations

Due date of deliverable: 31/12/2006
Actual submission date: 31/01/2007

Start date of the project: 1/1/2004 Duration: 36 months

Multitel

Revision 1

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	PU
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

List of Partners: Arteveldehogeschool (B)
Language and Computing NV (B)
Budapest University of Technology and Economics (HU)
Univerza v Ljubljani, Fakulteta za Elektrotehniko (SI)
Zenon SA, Robotics and Informatics (EL)
(University of Wales Cardiff (UK))
Multitel ASBL (B)
The Provost Fellows and Scholars of the College of the Holy and Undivided Trinity of Queen Elizabeth near Dublin (IRL)
Országos Orvosi Rehabilitációs Intézet (HU)
Scuola Superiore di studi universitari e di perfezionamento Sant'Anna
Universita' Campus Bio-Medico

Document identifier: D4.2_Final.doc
Version: 1.0
Status : Final
Date: 31/01/2007
Organisation: MULTITEL, SSSA, UCBM, KUL, AHS
Workpackage: 4
Task: 4.2
Dissemination: Public
Authors: Jean-Yves Parfait, Jo De Lafonteyne, Stefano Mazzoleni, Eugenio Guglielmelli, Giuseppe Cavallo, Gert Van Dijck, Marc Van Hulle, Jo Van Vaerenbergh
Implementation Software: Jean-Yves Parfait, Xavier Ricco

Approved by: Jo De Lafonteyne, WP4 Leader – Jo Van Vaerenbergh, Coordinator

Distribution List: WP4 partners, Project Coordinator, Project manager

Table of contents

Resume

The aim of this document is to provide the reader with a complete description of the work accomplished by Multitel in the Task 4.2:

This document will address three main topics:

- The data mining. Raw force/torque measurements were regularly recorded during the process of recovery of each stroke patient involved in the clinical trial. The data mining purpose was to extract high level knowledge about the recovery state of the patients from these raw data. In a second step, the physiotherapists searched for markers and milestones in patient state evolutions and recovery patterns were identified.

- The Automatic Onset Detection. A precise estimation of the onset of force/torque recordings is of a paramount importance because the data mining process is only based on the analysis of 'movement initiation'. This analysis covers only a particular part of the signal, which duration is smaller than the total signal duration. The width of the analysis window was set to 750 ms while the recording window used by the ADD (ALLADIN Diagnostic Device) ranged from 2.4 s to 6 s. Several techniques were compared including methods using statistical models built thanks to the data mining results.

- The Developed Software. They include two categories of software: the research tools and the integration module. Software of the first category were developed to assess the algorithms and to visualize the signals and the results of various tested techniques. They were developed in Matlab. The second category gathers C components due to be used in the final integration software.

1. Introduction

Usually recovery starts with a total incapacity to move the arm and leg. Step by step, these movements become smoother and more precise. Only in a few cases recovery is perfect. More often residual stiffness or some involuntary movement components remain. Recovery is not always a continuum, sometimes it suddenly stops or slows down to resume a few weeks later. Up to now, there is no way to show why or when this happens or to gives an indication about the future evolution of the patient and his final recovery.

By means of an automatic analysis of a pool of isometric force/torque recordings, WP4 assesses recovery after stroke. It aims at detecting significant events during this process, and intends to give meaningful indications to the patient about his/her final outcome.

Following scientific hypotheses underpin the assessment paradigm in ALLADIN [1]:

- Isometric force/torque trajectory patterns at the initiation of *Activity of Daily Living* (ADL) tasks are characteristic.
- Isometric force/torque trajectory patterns can be normalised for each *Activity of Daily Living* (ADL).
- There is a relation between the isometric force/torque trajectory patterns generated at the initiation of an ADL task and the grade of functional recovery of a particular stroke patient.

In order to accomplish this task, a database of isometric force/torque recordings was built. It consists in recordings of ADL movement attempts in a constraint environment. For that purpose, a dedicated device (the *ALLADIN Diagnostic Device, ADD*) was developed.

Six ADL tasks were selected for the measurement procedure: drinking a glass, turning a key, taking a spoon, lifting a bag, reaching a bottle and lifting a bottle. First, the patient saw a video describing the target movement (Recording 1). Secondly, he was asked to mentally imagine reproducing it (Recording 2). Finally, he actually tried to perform it 3 times (Recordings 3, 4, 5). During the whole procedure, the force and the torque were recorded by 8 sensors related to 8 different parts of the body: the thumb, the index finger, the middle finger, the arm, the trunk, the seat, the foot, the big toe. This protocol was repeated for each ADL task. The whole set of data recorded for the 6 tasks by the 8 sensors during the 5 recordings will be referred to as a *Measurement Session* through this document.

The patients involved in ALLADIN were recruited by the three clinical partners, resided in three different European countries. The first 8 weeks of the clinical trial patients were measured twice a week, from then on only once a week during the remaining 4 months. The main criteria for inclusion were: 1) diagnosis of ischemic brain damage, 2) an obvious motor deficit, 3) sufficient co-operation to permit full clinical examination. Patients with a pre-stroke disability interfering with the goal of the study are not included [1].

A control group of healthy subjects, measured once for all ADL tasks was added to the database. They will be referred to as the *Normal Controls* in this document.

During WP4, data mining developed algorithms to extract high level knowledge from a pool of force/torque data with the aim to give a meaning to the recovery process of a patient. Within the framework of this approach a set of relevant features was defined and extracted

from the force/torque time series. Afterwards, research efforts focused on defining a method to assess the recovery by the mean of these features. Chapter 2, provides a complete description of the way these two key points have been addressed. In section 2.1, the adopted approach and theoretical foundations are described, in section 2.2, the experimental results and in section 2.3 a critical discussion is devoted to the data mining outcomes.

In order to be consistent with the first ALLADIN scientific hypothesis, the features mentioned in the previous paragraph have to be extracted from a limited part of the signals beginning at the movement onset. The width of that time window was set to 750 ms while the total duration of the recordings ranges from 2.4 s to 6 s according to the task and to the recording (video, mental simulation or recording 1, 2, 3). A precise estimation of the onset of movement time is thus of a paramount importance. Chapter 3 deals with the tested automatic onset detection techniques.

Chapter 4 presents the software developed in WP4.The first software package is a test tool developed in Matlab and dedicated to the assessment of the data mining algorithms. It includes many visualization functionalities. The second software package consists of a DLL developed in C. It allows an easy integration of the data mining outcomes in the final integration software.

2. Data mining

This chapter describes the complete work of the data mining task (T4.2). Section 2.1 describes the adopted approach, starting from the definition of the candidate features to the choice of the metric for assessing recovery. Section 2.2 presents the experimental results. Finally, section 2.3, makes a critical analysis of the data mining results.

2.1. General approach

This section describes the adopted approach to the WP4 data mining task. The following sections detail the individual stages of the process, starting from the definition of the candidate features to the identification of markers in the recovery curves by the physiotherapists.

2.1.1. Overview

2.1.1.1. Recovery Space Paradigm

When recovery becomes perceivable, a stroke patient is expected to have reached a certain recovery level. Data mining module has to assess the recovery state of a patient by detecting significant events in the recovery course, or at least to provide the clinicians with valuable information to support their diagnostic (*e.g.,* does the patient demonstrate continuous progress or get stalled in some state, can we predict an oncoming significant recovery within a reasonable amount of time?).

The adopted approach assumes that the state of a stroke patient at a given instant is sufficiently characterized by a set of features, a so-called feature vector, which is extracted from the current isometric torque/force measurements. The feature vectors evolve in a multivariate space, referenced as the *Recovery Space*. $X_{p,t}$ is defined as the feature vector computed for the *p*-th stroke patient and at the *t*-th day after the patient stroke, and this for several ADL tasks. Note that the measurement sessions are regularly spaced: every patient has two recording sessions during the first 8 weeks and 1 recording session per week for the consecutive 16 weeks. The time course of the feature vectors of a stroke patient is expected to depict the evolution of the patient from a 'diseased' towards a normal state (see Figure 1).

The direct observation of relevant patterns in sequences of feature vectors in the recovery space is untraceable because of the high dimension of these feature vectors (up to 6 dimensions in this study). Our proposed approach consists in defining a measure of how far feature vectors stand from normality. By normality, we mean the population of feature vectors obtained for normal controls for the same ADL task. We expect that these measures of distance to normality will allow characterizing the evolution of the stroke patient in the recovery space from a more easily interpretable perspective.

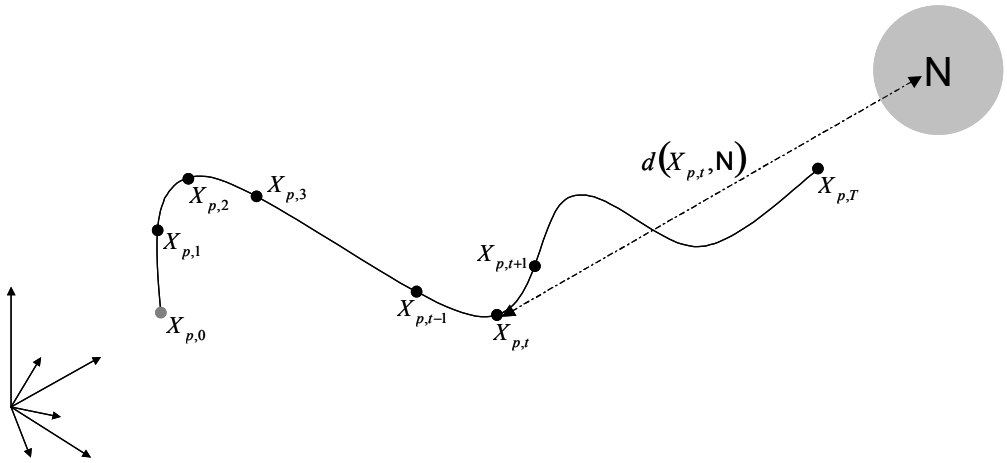


Figure 1: Recovery space paradigm: the stroke patient is represented by a feature vector that ideally evolves over time from his initial state to normality (N).

2.1.1.2. Work plan

The described approach was defined during the Leuven WP4 meeting on 28 & 29 November, and clarified in a separated meeting on the 3rd of February in the presence of the expert from the KUL, Prof. Marc Van Hulle. The main steps of the work plan are described in the Figure 2.

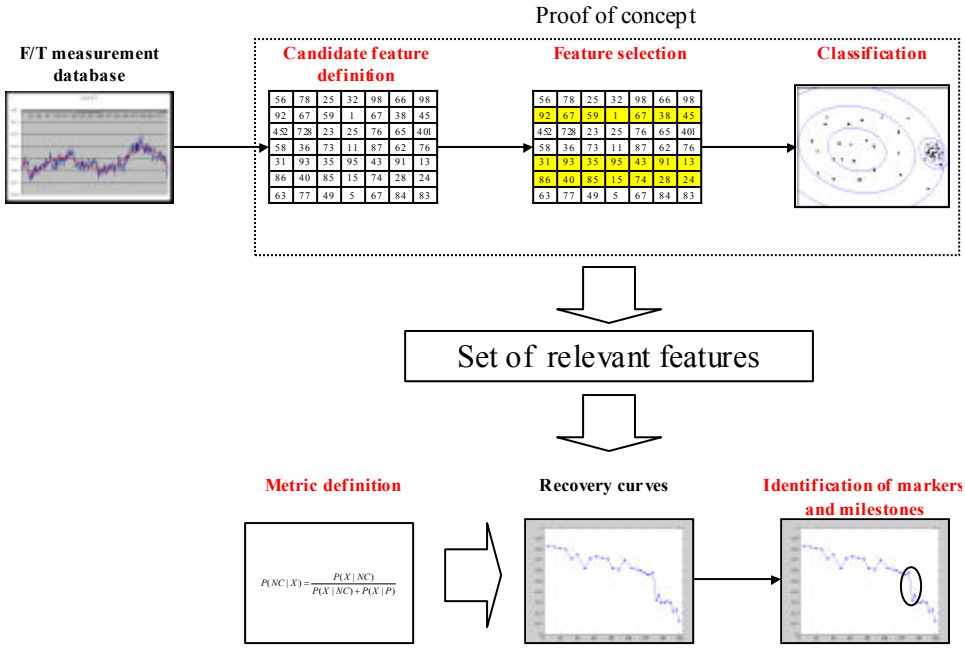


Figure 2: Main steps of the data mining approach.

The first step of task T4.2 is a **proof of concept**. The goal is to see whether the patients at the beginning of their rehabilitation can be distinguished from healthy subjects on the basis of a small number of numerical characteristics extracted from the force/torque trajectories. This first objective is a prerequisite before launching out into any deeper investigation. Not any identification of recovery patterns is to be expected without distinctive features in the data.

Prior to any processing, non informative variability (due to external factors) was carefully removed with the aim to better focus on the definition of pertinent features. External variability sources are:

- Adverse events occurred during the data acquisition such as bad calibration, disabled sensors, incomplete session,…
- Non precise detection of the onset of movement time.

A balanced force/torque measurement data subset was manually selected. The selection was performed on the basis of the first measurement session only. Data affected by an adverse event were discarded, leading to a dataset containing 57 patients and 57 normal controls.

The problem of the onset of movement time precision was initially circumvented by manually tagging all the signals of the dataset. The total number of onset of movement times is thus:

$$2\left(N_{Pat}+N_{NC}\right)N_T N_A N_S = 2.114.6.3.8 = 32832$$

Where N_{Pat} is the number of patients, N_{NC} is the number of normal controls, N_T is the number of tasks, N_A is the number of times the patient was asked to actually perform the task and N_S is the number of sensors.

This tagging work was independently carried out by physiotherapists from the three clinical sites. For this purpose a graphical user interface was developed in Matlab (see Figure 3) to assist them during this task. It allowed the physiotherapists to tag the onset by simply clicking on the visualized force and torque graphs, to save/load the tagged onset of movement times into/from a text file.

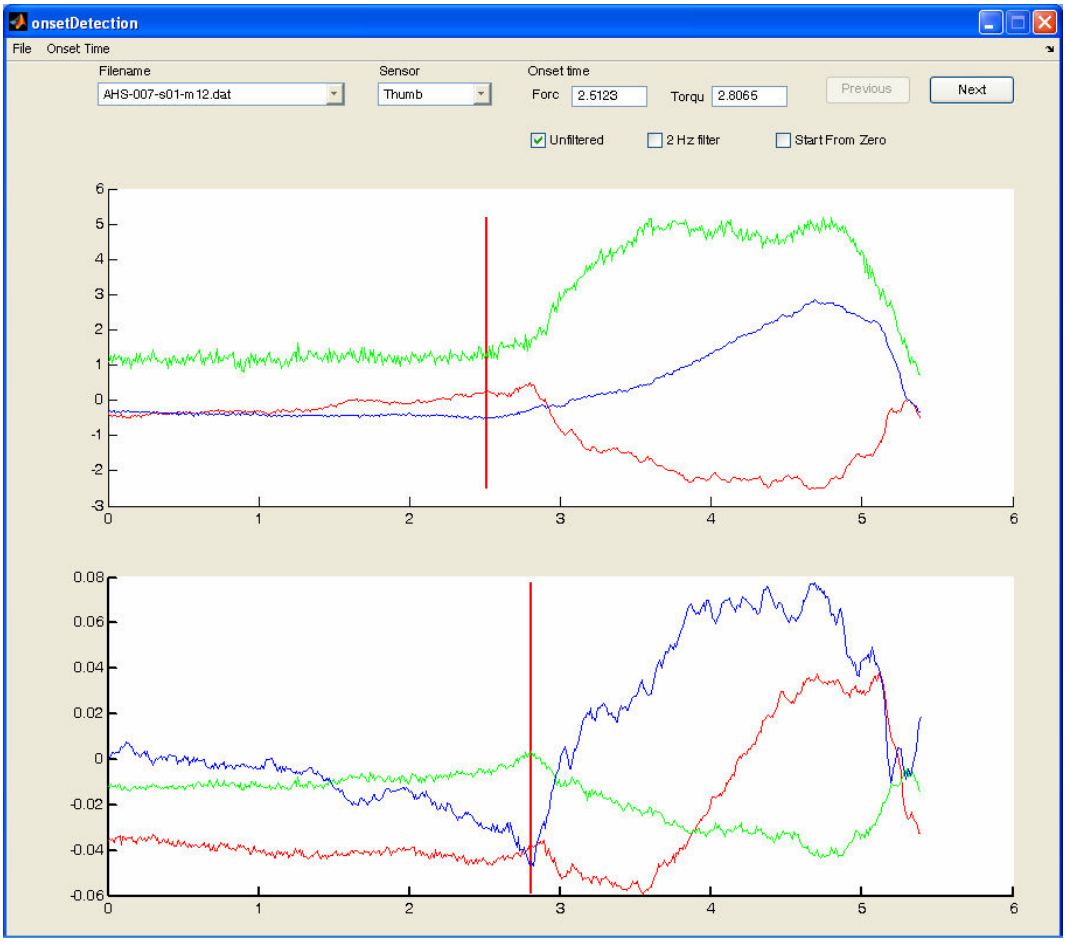


Figure 3: Onset of movement time tagging tool (screenshot).

Then, the candidate features can be extracted from this manually tagged data subset (the **candidate feature extraction** step in the Figure 2). These features are defined in section 2.1.2.3.

Afterwards, a **Feature Subset Selection** (FSS) identifies the most discriminating features in the whole list of candidate features. This task was performed by the expert's team because they have a lot of experience with this particular problem. Consequently, only the key concepts about the used algorithm will be presented in this document (see section 2.1.3). More information can be found in [5].

The last step during this proof of concept is the **classification test** (patients versus normal controls). This test is performed using the selected features. A good classification rate was decisive for any further investigation. Classification tests have also been extended to evaluate (1) the robustness of the FSS results with respect to new data (the other measurement sessions) and (2) the different statistical models for the definition of a metric.

A successful proof of concept will lead to the definition of a set of discriminating features, which on its turn will bring the trajectory of the feature vector in the recovery space in the focus. The proposed approach implies the **definition of a metric** to assess how far a patient is from a normal state. In this way, the interpretation of the evolution of the feature vector in the high-dimensionality recovery space is expected to be easier (see the *Recovery Space*

Paradigm). Once defined, the metric will enable the computation of a score, related to the degree of recovery and this for each measurement session. This will lead to the drawing of the so-called **recovery curves**.

Then, these recovery curves will be subjected to expert examination in order to identify specific and clinically sounded **recovery patterns**.

2.1.2. Candidate feature definition

This section describes all the candidate features defined during the first step of the proof of concept. Some of them are expected to contain relevant information for assessing functional recovery from stroke.

2.1.2.1. General considerations

The defined features don't explicitly depend on the energy of the signal and the reaction time given the fact that patients neither were asked to react with maximal force neither to start as fast as possible after the start signal.

The first category of features is based on scientific researches that demonstrated that stroke patients have a typically reduced ability of controlling force/torques generation, both in intensity and spatial direction. This lack of control should somehow be reflected in a set of "abnormalities" in the force/torque vector direction. Most of them are not extracted directly from the force/torque trajectories but from the following derived variables and grouped accordingly:

- Mean effort
- Angular deviation to the mean direction.
- First-order angular deviation of the effort series.
- Cumulative sum of the effort series.

A second category of features are expected to capture other characteristics of the movement dynamics (f.i.: the rise time). These are directly extracted from the effort time series. Two approaches were investigated: the *Fit of a Parameterised Curve* and a *System Identification*.

KUL experts defined a third category of features, which has to catch the cross-sensor information. The definition relies on the hypothesis that the sequence of activation of the different sensors and the relative time delays during the execution of the same task can be of clinical interest for estimating 'distance to normality'. It is expected that stroke patients will demonstrate abnormal time activation patterns due to some loss of internal models of the tasks to be performed. These internal models in the brain deal with motion planning, prediction and execution. They are tightly tied to the *Mutual Information*.

The candidate features were computed on the proof of concept database, for all the patients (first measurement session only), all the tasks, all the attempts and all the sensors. Then a feature selection algorithm was run on this pool of data to identify the most discriminative characteristics (see section 2.1.1). The calculation of a given feature for different tasks, attempts and sensors gives rise to as many different features as possible combinations of tasks, attempts and sensors.

In order to be consistent with the first hypothesis of the ALLADIN paradigm for the assessment of the post-stroke recovery, the features are to be extracted within a time window beginning at the estimated onset of movement time for a given sensor. In agreement with the physiotherapists, the width of that window is finally set to 750 ms (excepted for very few features where the analysis window width is variable) by the mean of a visual inspection of the force/torque measurements. For the proof of concept, manually tagged onset of movement times were used. A detailed description of the data pre-processing will be provided in section 2.2.

2.1.2.2. Notations

Every recording contains discrete-time trajectories of forces and torques for eight sensors with respect to three orthogonal directions. Let define $F_{s,x}[k]$, $F_{s,y}[k]$ and $F_{s,z}[k]$ as the discrete-time force signals along the three orthogonal axes *x*, *y* and z for the *s*-th sensor, *s*=1,…,8. Note the directions of these axes are specific to every sensor. Similarly, $T_{s,x}[k]$, $T_{s,y}[k]$ and $T_{s,z}[k]$ as the discrete-time torque signals around axes *x*, *y* and z for the *s*-th sensor. The sampling rate is equal to 100 Hz, that is, six force and torque values are captured every 10 ms by every sensor.

2.1.2.3. Feature definitions

2.1.2.3.1. Mean Effort Direction

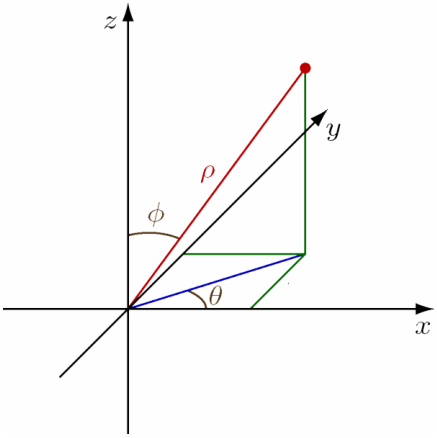
It is assumed that effort direction is more relevant for indicating recovery than pure force intensity. For example spasticity (involuntary muscle contraction) can be a source of intense force, though not functional for reaching to an object. Given a recording, for the *s*-th sensor, we compute the mean force direction features as the colatitude and azimuth angles of the mean force vector with respect to its referential. The mean force vector is defined by its components $\overline{F}_{s,x}$, $\overline{F}_{s,y}$ and $\overline{F}_{s,z}$ where

$$\overline{F}_{s,x} = \frac{1}{N}\sum_{k=k_0}^{k_0+N-1} F_{s,x}[k]$$

$$\overline{F}_{s,y} = \frac{1}{N}\sum_{k=k_0}^{k_0+N-1} F_{s,y}[k]$$

$$\overline{F}_{s,z} = \frac{1}{N}\sum_{k=k_0}^{k_0+N-1} F_{s,z}[k]$$

with k_0 being the sample index of the estimated onset time. The colatitude $\phi_{F,s}$ is the angle between the *z*-axis of the mean force vector. The azimuth $\theta_{F,s}$ is the angle between the positive *x*-axis and the line from the origin to the end of the mean force vector projected onto the *xy*-plane. These angles are obtained by converting the Cartesian coordinates of the mean force to spherical coordinates, that is,



$$\rho = \sqrt{F_{s,x}^2 + F_{s,z}^2 + F_{s,z}^2}$$

$$\phi_{F,s} = \arccos\left(\frac{F_{s,z}}{\rho}\right)$$

$$\theta_{F,s} = \arctan\left(\frac{F_{s,y}}{F_{s,x}}\right) + \pi\, u_0\left(-F_{s,x}\right)$$

where $u_0()$ stands for the Heaviside unit step function

$$u_0(x) = \begin{cases} 0 & \text{if } x \le 0 \\ 1 & \text{if } x > 0 \end{cases}$$

Angle features $\phi_{T,s}$ and $\theta_{T,s}$ can be computed similarly from the mean torque vector to characterize the mean torque "direction".

2.1.2.3.2. Angular Deviation to Mean Effort

Beside features characterizing mean direction of efforts, the angular deviation of every effort sample within the analysis frame from the mean effort is computed. It is assumed that the distribution of these angular deviations depicts some specific pattern (sudden variations, lack of smoothness, etc) in the stroke patient movements. Given a recording, for the *s*-th sensor, the angular deviation $\delta_{F,s}[k]$ between the *k*-th force sample $(F_{s,x}[k], F_{s,y}[k], F_{s,z}[k])$, within the analysis frame $k = k_0, \cdots, k_0 + N - 1$, and the mean force $(\overline{F}_{s,x}, \overline{F}_{s,y}, \overline{F}_{s,z})$ is computed as the inverse cosine of the normalized scalar product, *i.e.* the dot product of the corresponding unit-norm vectors,

$$\vec{a} = \left(\overline{F}_{s,x}, \overline{F}_{s,y}, \overline{F}_{s,z}\right)$$

$$\vec{b} = \left(F_{s,x}[k], F_{s,y}[k], F_{s,z}[k]\right)$$

$$\delta_{F,s}[k] = \arccos\left(\frac{\vec{a}}{\|\vec{a}\|} \cdot \frac{\vec{b}}{\|\vec{b}\|}\right) = \arccos\left(\frac{\overline{F}_{s,x}F_{s,x}[k] + \overline{F}_{s,y}F_{s,y}[k] + \overline{F}_{s,z}F_{s,z}[k]}{\sqrt{\overline{F}_{s,x}{}^2 + \overline{F}_{s,y}{}^2 + \overline{F}_{s,z}{}^2}\sqrt{F_{s,x}[k]^2 + F_{s,y}[k]^2 + F_{s,z}[k]^2}}\right)$$

Several features are computed in order to characterize the distribution of the angular deviations $\delta_{F,s}[k]$, $k = k_0, \cdots, k_0 + N - 1$. The angular deviations can take values between 0 to π. First, the maximum value $Max(\delta_{F,s})$ is computed in order to characterize the support

of the distribution. Next, the mean value $Mean(\delta_{F,s})$ and the standard deviation $Std(\delta_{F,s})$ are estimated in order to characterize the central tendency and the dispersion of the distribution, respectively. Then, the skewness $Skew(\delta_{F,s})$ and the kurtosis $Kurt(\delta_{F,s})$ are estimated in order to characterize the asymmetry and the peakedness of the distribution. Finally, the probability density function of the angular deviations is estimated using kernel-based method $KS(\delta_{F,s})$.

$$Max\left(\delta_{F,s}\right) = \underset{k=k_0,\cdots,k_0+N-1}{\arg\max}\left(\delta_{F,s}[k]\right)$$

$$Mean\left(\delta_{F,s}\right) = \frac{1}{N}\sum_{k=k_0}^{k_0+N-1}\delta_{F,s}[k]$$

$$Std\left(\delta_{F,s}\right) = \left(\frac{1}{N-1}\sum_{k=k_0}^{k_0+N-1}\left(\delta_{F,s}[k]-Mean\left(\delta_{F,s}\right)\right)^2\right)^{1/2}$$

$$Skew\left(\delta_{F,s}\right) = \frac{n}{(n-1)(n-2)}\frac{\sum_{k_0}^{k_0+N-1}\left(\delta_{F,s}[k]-Mean\left(\delta_{F,s}\right)\right)^3}{\left(\sum_{k=k_0}^{k_0+N-1}\left(\delta_{F,s}[k]-Mean\left(\delta_{F,s}\right)\right)^2\right)^{3/2}}$$

$$Kurt\left(\delta_{F,s}\right) = \frac{n(n+1)}{(n-1)(n-2)(n-3)}\frac{\sum_{k=k_0}^{k_0+N-1}\left(\delta_{F,s}[k]-Mean\left(\delta_{F,s}\right)\right)^4}{\left(\sum_{k=k_0}^{k_0+N-1}\left(\delta_{F,s}[k]-Mean\left(\delta_{F,s}\right)\right)^2\right)^2} - 3\frac{(n-1)^2}{(n-2)(n-3)}$$

Besides characterizing the statistical distribution of the angular deviations of the sequence of force samples to the mean force within the time region of interest, the feature extraction aims also at modelling the time correlation of the sequence of angular deviations. Such information can be provided in a compact form as the coefficients of an auto-regressive (AR) model fitting to the sequence of angular deviations. This model assumes that every angular deviation can be merely predicted by the past values, that is,

$$\delta_{F,s}[k] = \sum_{p=1}^{P} - a_p\delta_{F,s}[k-p] + \varepsilon[k]$$

where the coefficients $(a_1,\cdots,a_P)$ denote the AR coefficients. These parameters are classically estimated by minimising the mean square error between the observed angular deviations and their predicted values over the entire analysis frame,

$$(\hat{a}_1,\cdots,\hat{a}_P) = \underset{a_1,\cdots,a_P}{\arg\min}\sum_{k=k_0+P}^{k_0+N-1}\left(\delta_{F,s}[k] + \sum_{p=1}^{P} a_p\delta_{F,s}[k-p]\right)^2 .$$

The AR parameters can be obtained as the solutions to the set of Yule-Walker linear equations by estimating the correlation coefficients of the angular deviation sequence up to

the *P*-th order and applying the Levinson-Durbin recursive algorithm. Clearly, the goodness-of-fit improves as the number of parameters increases. The AR model order is chosen according to the Akaike information criterion (AIC) in order to find the best trade off between goodness-of-fit and model complexity, and to avoid overfitting the model to the data,

$$AIC = \frac{2P}{N} + \log \sum_{k=k_0+P}^{k_0+N-1} \left(\delta_{F,s}[k] + \sum_{p=1}^{P} a_p \delta_{F,s}[k-p] \right)^2 - \log N.$$

Same features are extracted for the angular deviations of the sequence of the torque samples to the mean torque vector.

2.1.2.3.3. First-order Angular Deviation of Effort Series

Additional information on stroke patient's ability in controlling generated forces/torques is expected to be found in the angular deviations between successive effort samples within the analysis frame. Given a recording, for the *s*-th sensor, the angular deviation $\varphi_{F,s}[k]$ between the *k*-th force sample $(F_{s,x}[k], F_{s,y}[k], F_{s,z}[k])$ and the (*k*-1)-th force sample $(F_{s,x}[k-1], F_{s,y}[k-1], F_{s,z}[k-1])$, within the analysis frame $k = k_0 + 1, \cdots, k_0 + N - 1$, is computed as the inverse cosine of the normalized scalar product, *i.e.* the dot product of the corresponding unit-norm vectors,

$$\vec{a} = \left(F_{s,x}[k], F_{s,y}[k], F_{s,z}[k] \right)$$
$$\vec{b} = \left(F_{s,x}[k-1], F_{s,y}[k-1], F_{s,z}[k-1] \right)$$
$$\varphi_{F,s}[k] = \arccos\left(\frac{\vec{a}}{\|\vec{a}\|} \cdot \frac{\vec{b}}{\|\vec{b}\|} \right)$$
$$= \arccos\left(\frac{F_{s,x}[k]F_{s,x}[k-1] + F_{s,y}[k]F_{s,y}[k-1] + F_{s,z}[k]F_{s,z}[k-1]}{\sqrt{F_{s,x}[k-1]^2 + F_{s,y}[k-1]^2 + F_{s,z}[k-1]^2}\sqrt{F_{s,x}[k]^2 + F_{s,y}[k]^2 + F_{s,z}[k]^2}} \right)$$

Several features are computed in order to characterize the distribution of the angular deviations $\varphi_{F,s}[k]$, $k = k_0 + 1, \cdots, k_0 + N - 1$. The angular deviations can take values between 0 to π. First, the maximum value $Max(\varphi_{F,s})$ is computed in order to characterize the support of the distribution. Next, the mean value $Mean(\varphi_{F,s})$ and the standard deviation $Std(\varphi_{F,s})$ are estimated in order to characterize the central tendency and the dispersion of the distribution, respectively. Then, the skewness $Skew(\varphi_{F,s})$ and the kurtosis $Kurt(\varphi_{F,s})$ are estimated in order to characterize the asymmetry and the peakedness of the distribution. Finally, the probability density function of the angular deviations is estimated using kernel-based method $KS(\varphi_{F,s})$.

$$Max\left(\varphi_{F,s}\right) = \underset{k=k_0+1,\cdots,k_0+N-1}{\arg\max}\left(\varphi_{F,s}[k]\right)$$

$$Mean\left(\varphi_{F,s}\right) = \frac{1}{N}\sum_{k=k_0+1}^{k_0+N-1}\varphi_{F,s}[k]$$

$$Std\left(\varphi_{F,s}\right) = \left(\frac{1}{N-1}\sum_{k=k_0+1}^{k_0+N-1}\left(\varphi_{F,s}[k]-Mean\left(\varphi_{F,s}\right)\right)^2\right)^{1/2}$$

$$Skew\left(\varphi_{F,s}\right) = \frac{n}{(n-1)(n-2)}\frac{\sum_{k=k_0+1}^{k_0+N-1}\left(\varphi_{F,s}[k]-Mean\left(\varphi_{F,s}\right)\right)^3}{\left(\sum_{k=k_0+1}^{k_0+N-1}\left(\varphi_{F,s}[k]-Mean\left(\varphi_{F,s}\right)\right)^2\right)^{3/2}}$$

$$Kurt\left(\varphi_{F,s}\right) = \frac{n(n+1)}{(n-1)(n-2)(n-3)}\frac{\sum_{k=k_0+1}^{k_0+N-1}\left(\varphi_{F,s}[k]-Mean\left(\varphi_{F,s}\right)\right)^4}{\left(\sum_{k=k_0+1}^{k_0+N-1}\left(\varphi_{F,s}[k]-Mean\left(\varphi_{F,s}\right)\right)^2\right)^2} - 3\frac{(n-1)^2}{(n-2)(n-3)}$$

Besides characterizing the statistical distribution of the first-order angular deviations of the sequence of force samples within the time region of interest, the feature extraction aims also at modelling time correlation. Such information can be provided in a compact form as the order and the coefficients of an auto-regressive (AR) model fitting to the sequence of angular deviations, as described previously.

Same features are extracted for the angular deviations between successive torque samples.

2.1.2.3.4. Cumulative Sum of Effort Series

The integrals of the effort signals are expected to convey some information on the velocity of the virtual movements (since the patient's movements are constraint by the ADD), thereof on the stroke patient ability to perform some movement velocity patterns. More especially, the norm of the integral of the force/torque sample sequence is used. Given a recording, for the *s*-th sensor, the norm $\left\|\vec{\gamma}_{F,s}[k]\right\|$ of the integral vector $\vec{\gamma}_{F,s}[k]$ of the force sample vector sequence at the *k*-th time instant, within the analysis frame $k = k_0,\cdots,k_0+N-1$, is computed as the norm of the cumulative sum of the force sample vector from the k_0-th time instant up to the *k*-th time instant,

$$\gamma_{F,s,x}[k] = \sum_{l=k_0}^{k}F_{s,x}[l]$$

$$\gamma_{F,s,y}[k] = \sum_{l=k_0}^{k}F_{s,y}[l]$$

$$\gamma_{F,s,z}[k] = \sum_{l=k_0}^{k}F_{s,z}[l]$$

$$\vec{\gamma}_{F,s}[k] = \left(\gamma_{F,s,x}[k], \gamma_{F,s,y}[k], \gamma_{F,s,z}[k]\right)$$
$$\left\|\vec{\gamma}_{F,s}[k]\right\| = \sqrt{\gamma_{F,s,x}[k]^2 + \gamma_{F,s,y}[k]^2 + \gamma_{F,s,z}[k]^2}$$

Several features are computed in order to characterize the distribution of the norms of the integral force vectors $\left\|\vec{\gamma}_{F,s}[k]\right\|$, $k = k_0, \cdots, k_0 + N - 1$. First, the mean value $Mean(\left\|\vec{\gamma}_{F,s}\right\|)$ and the standard deviation $Std(\left\|\vec{\gamma}_{F,s}\right\|)$ are estimated in order to characterize the central tendency and the dispersion of the distribution, respectively. Next, the skewness $Skew(\left\|\vec{\gamma}_{F,s}\right\|)$ and the kurtosis $Kurt(\left\|\vec{\gamma}_{F,s}\right\|)$ are estimated in order to characterize the asymmetry and the peakedness of the distribution. Finally, the probability density function of the angular deviations (note that an angle is not defined in this paragraph) is estimated using kernel-based method $KS(\left\|\vec{\gamma}_{F,s}\right\|)$.

$$Mean\left(\left\|\vec{\gamma}_{F,s}\right\|\right) = \frac{1}{N}\sum_{k=k_0}^{k_0+N-1}\left\|\vec{\gamma}_{F,s}[k]\right\|$$

$$Std\left(\left\|\vec{\gamma}_{F,s}\right\|\right) = \left(\frac{1}{N-1}\sum_{k=k_0}^{k_0+N-1}\left(\left\|\vec{\gamma}_{F,s}[k]\right\| - Mean\left(\left\|\vec{\gamma}_{F,s}\right\|\right)\right)^2\right)^{1/2}$$

$$Skew\left(\left\|\vec{\gamma}_{F,s}\right\|\right) = \frac{n}{(n-1)(n-2)}\frac{\sum_{k=k_0}^{k_0+N-1}\left(\left\|\vec{\gamma}_{F,s}[k]\right\| - Mean\left(\left\|\vec{\gamma}_{F,s}\right\|\right)\right)^3}{\left(\sum_{k=k_0}^{k_0+N-1}\left(\left\|\vec{\gamma}_{F,s}[k]\right\| - Mean\left(\left\|\vec{\gamma}_{F,s}\right\|\right)\right)^2\right)^{3/2}}$$

$$Kurt\left(\left\|\vec{\gamma}_{F,s}\right\|\right) = \frac{n(n+1)}{(n-1)(n-2)(n-3)}\frac{\sum_{k=k_0}^{k_0+N-1}\left(\left\|\vec{\gamma}_{F,s}[k]\right\| - Mean\left(\left\|\vec{\gamma}_{F,s}\right\|\right)\right)^4}{\left(\sum_{k=k_0}^{k_0+N-1}\left(\left\|\vec{\gamma}_{F,s}[k]\right\| - Mean\left(\left\|\vec{\gamma}_{F,s}\right\|\right)\right)^2\right)^2} - 3\frac{(n-1)^2}{(n-2)(n-3)}$$

Besides characterizing the statistical distribution of the norms of the integral force vectors within the time region of interest, the feature extraction aims also at modelling time correlation. Such information can be provided in a compact form as the order and the coefficients of an auto-regressive (AR) model fitting to the sequence of angular deviations, as described previously.

Same features are extracted for the norm of the integral torque vector.

Note that we cannot give the real interpretation of movement while the objects are fixed. A constant force implies a linear increase in speed under the imagined situation of free moving objects. In the situation of fixed objects, the usefulness of this parameter can be less pertinent, but it will serve as a kind of low-pass filtering on the data.

2.1.2.3.5. Curve Fitting

The curve fitting approach

This approach implies the definition of a family of parameterised models $\phi(t,\theta)$ which is driven by an a priori knowledge about the relationship between y and t. y can then be expressed as follows:

$$\mathbf{y} = \phi(\mathbf{t},\theta) + \boldsymbol{\varepsilon}$$

Where:

- $\mathbf{y}$ is the vector of measured samples.
- $\phi(\mathbf{t},\theta)$ is a family of parameterised models. $\phi(\mathbf{t},\theta)$ is short for $\phi(kT_s,\theta)$ $k = 1,...,N$ (T_s is the sampling period, θ is the parameter set, N is the number of samples).
- $\boldsymbol{\varepsilon}$ is the error vector

The parameter vector values are tuned so as for minimizing the *Average Squared Error* $E(\theta)$ between the predicted data and the measured samples:

$$E(\theta) = < \left(\phi(t,\theta) - y\right)^2 >$$
$$\hat{\theta} = \arg\min_{\theta} E(\theta)$$

Where:

- $<.>$ is the average operator
- $\hat{\theta}$ is the optimal parameter set

Features can then be extracted from the value of $\hat{\theta}$.

In the case where $\phi(t,\theta)$ is linear, the minimization of $E(\theta)$ leads to a set of linear equations whose solving straightforwardly provides an optimal solution. On the contrary, when $\phi(t,\theta)$ is a non-linear function, the minimization is performed iteratively from an initial guess and is likely to lead to a locally optimal solution depending on the initial guess.

This iterative minimization problem has already been widely addressed in the literature. Among the variety of existing algorithms, the *Levenberg-Macquardt* (associated with the `lscurvefit` function from the *Matlab Optimization Toolbox*) is used in this work. The interested reader can find details about the mathematical foundations and the implementation of this method in [8] (pages 681-688) and [9] or in the *Matlab Optimization Toolbox* documentation.

In order to assess the goodness of fit for a given model we compute the *Normalized Error* $NE(\theta)$:

$$NE(\theta) = \sqrt{\frac{\sum_{i=1}^{N}\left(y_i - \phi(iT_s,\theta)\right)^2}{\sum_{i=1}^{N} y_i^2}}$$

$NE(\theta)$ is also expected to give indications about the smoothness of the voluntary movement. It is independent of the signal energy and the number of considered samples (thus of the analysis window width) for a given model and is expected to allow a consistent comparison between all the patients.

The choice of $\phi(t,\theta)$ and consequently the extracted features depends on the dynamic characteristics we want to capture. Several tests were performed with different models $\phi(t,\theta)$ and using different definitions for the analysis window (see [15]).

Fixed window length

In agreement with the first a 750ms-long window beginning at the onset of movement time is used. The Figure 4 shows the corresponding region of interest for the norm of the force measured by the thumb sensor during the *drinking a glass* task for 15 patients and 15 normal controls (randomly selected).

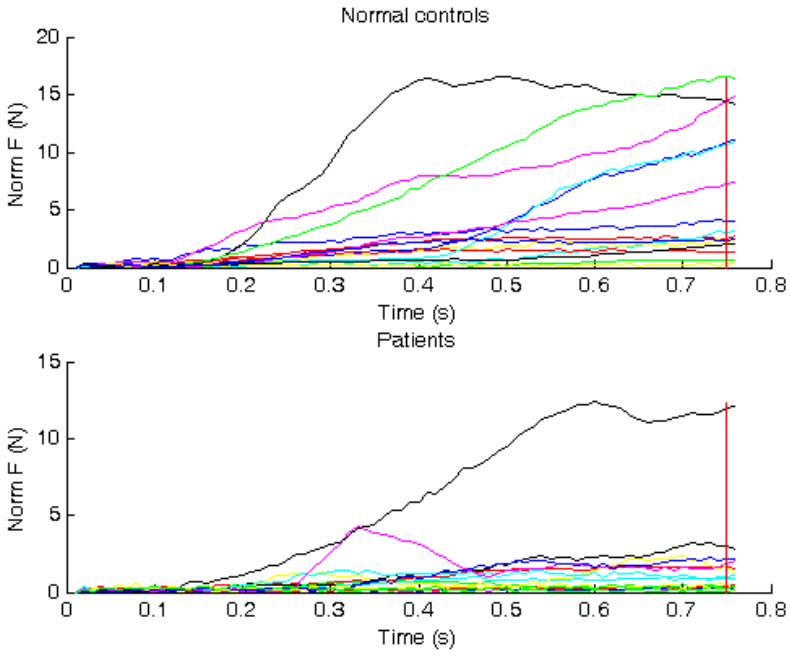


Figure 4: Region of interest (window length: 750 ms) for 15 patients (below) and 15 normal controls (above)-Signal: norm of the force, thumb sensor, drinking a glass task

Two models were tested. We focused here only on the rise time (feature T_1) and on the variability of the experimental data around the model (feature $NE\left(\hat{\theta}\right)$). Therefore, even though the proposed models are very simple, they are sophisticated enough to capture the desired characteristics in the movement dynamics.

During muscle contractions a background vibration is present of which spectral components can reach 40 Hz in frequency. Spectral components resulting from voluntary movements have generally lower frequencies. Therefore, signals are low-pass filtered at 40 Hz during the pre-processing step (10-order Butterworth filter). A zero-phase filtering is performed using the Matlab `filtfilt` function. The part of signal within the analysis window is also normalized: subtraction of the first sample value and division by the last sample value.

Model 1

$$\phi(t',\theta) = \begin{cases} a + be^{\frac{(t'-t'_e)}{T_1}} & 0 \le t' \le t'_e \end{cases}$$

Where:

- t' stands for the $t - t_{onset}$ (t represents the time and t_{onset} is the manually tagged onset time)
- t'_e is the end-of-frame time ($t'_e = 0{,}75$ s)

Parameter set: $\theta = \left(T_1, a, b\right)$

Features: $T_1, NE\left(\hat{\theta}\right)$

Initialization: $T_1 = 0{,}5$ s, $a = \dfrac{-e^{-1.5}}{1-e^{-1.5}}, b = \dfrac{1}{1-e^{-1.5}}$

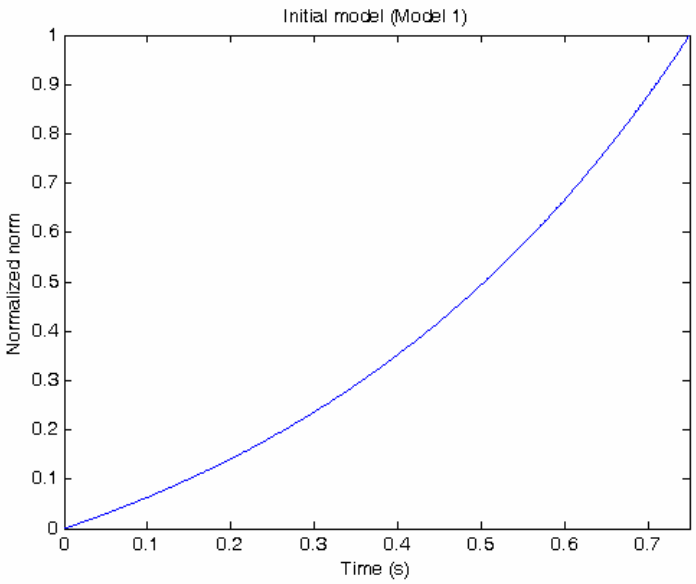


Figure 5: Initial model (Model 1)

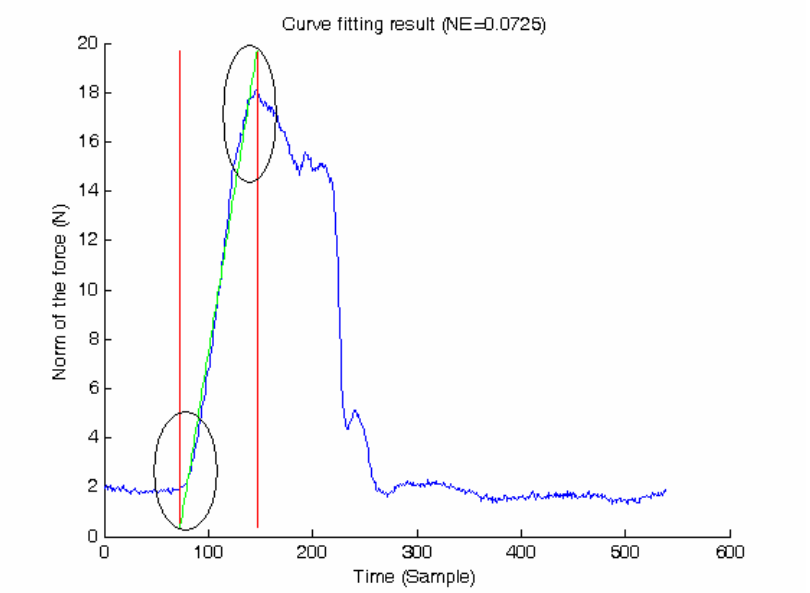


Figure 6: Curve fitting result with a 750ms-long window. Norm of the force (thumb sensor-drinking a glass task): blue, limits of the analysis window: red, model: green.

A scan of the curve fitting results showed that the parameters are sometimes ill-conditioned (T_1, a, b have simultaneously very high absolute values). Figure 6 illustrates such a case. This mainly occurs when the model has to fit a ramp-like signal. Indeed, the Taylor expansion of be^x around $x = 0$ gives:

$$be^x = b\left(1 + x + x^2 + ...\right) \approx b\left(1 + x\right)$$

With $x = \frac{(t' - t'_e)}{T_1}$. When $\lim_{T_1 \to \infty} x = 0$, the model tends to $a + b + \frac{b}{T_1}\left(t' - t'_e\right)$.

A straightforward solution should consist in putting constrains on the parameter values. Then, the algorithms for this constrained problem would be totally different from the approach described in the introduction. Another solution is to define the model in such way that the parameters never appear as a product or ratio of each other in the Taylor expansion. This is the reason why we defined the Model 2.

Model 2

$$\phi(t', \theta) = \begin{cases} e^{\frac{(t' - t'_e)}{T_1}} & 0 \le t' \le t'_e \end{cases}$$

Where:

- t' stands for the $t - t_{onset}$ (t represents the time and t_{onset} is the manually tagged onset time)
- t'_e is the end-of-frame time ($t'_e = 0{,}75$ s)

Parameter set: $\theta = (T_1)$

Features: $T_1, NE(\hat{\theta})$

Initialization: $T_1 = 0,5$ s

Model 2 is less sophisticated than Model 1 but the extracted features turned to be well-conditioned. The initial model is very similar to the initial model 1 (see Figure 5).

Variable window length

For this experiment, the whole motor activity will be taken into account. The analysis window is obtained by an empirical algorithm using hand-tuned thresholds. The onset of movement times have to be provided.

During the pre-processing step, a zero phase low-pass filtering is performed using a 10-order Butterworth filter (cut-off frequency equal to 40 Hz). The normalization of the signal performed for a fixed-length analysis window is not used anymore.

Model 3

$$\phi(t',\theta) = \begin{cases} a + (c-a)e^{\frac{t'-t_r}{T_1}} & 0 \le t' < t_r \\ c & t_r \le t' < t_d \\ b + (c-b)e^{\frac{t'-t_d}{T_1}} & t_d \le t' < t'_e \end{cases}$$

Where:

- t' stands for the $t - t_{onset}$ (t represents the time and t_{onset} is the manually tagged onset time)
- t'_e is the end-of-frame time (with t_{onset} as time origin).

Parameter set: $\theta = (T_1, T_2, a, b, c, t_r, t_d)$

Features: $T_1, T_2, NE(\hat{\theta})$

Initialization: Depends on the concavity.

If the concavity is negative:

$a = b = \min(Signal)$, $c = \max(Signal)$, $T_1 = T_2 = 0.5$, $t_r = \frac{t_e'}{4}$, $t_d = \frac{3t_e'}{4}$.

If the concavity is positive:

$a = b = \max(Signal)$, $c = \min(Signal)$, $T_1 = T_2 = 0.5$, $t_r = \frac{t_e'}{4}$, $t_d = \frac{3t_e'}{4}$

The Figure 7 shows the initial model for $a = b = 0$, $c = 2$, $T_1 = T_2 = 0.5$, $t_r = \frac{t_e'}{4}$, $t_d = \frac{3t_e'}{4}$.

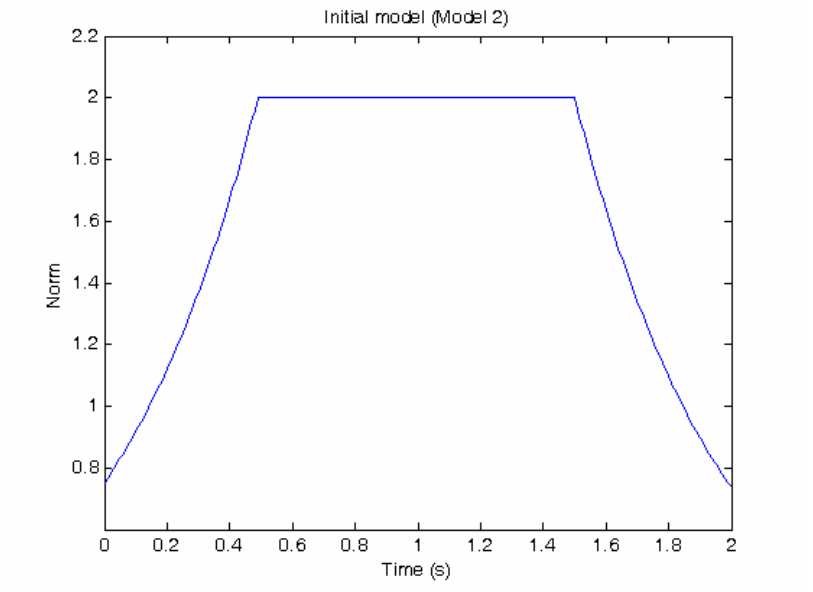


Figure 7: Initial model (T1=T2=0.5, a=b=0,c=2, te=2)

The Figure 8 shows an example of a curve fitting performed on the norm of the force (Thumb sensor, drinking a glass task).

The same remark as for the Model 1 can be formulated: features are ill-conditioned although the fit is good. However, the approach that led to Model 2 can't be applied here.

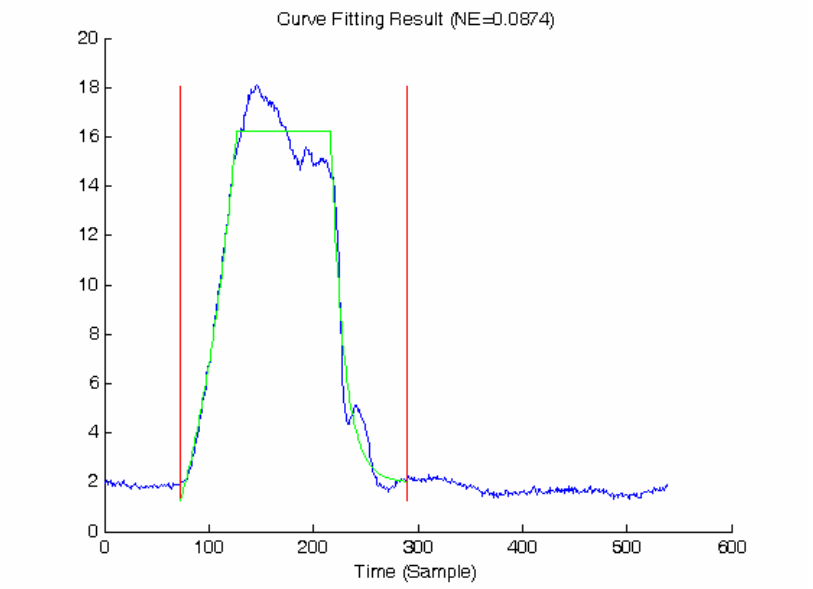


Figure 8: Curve fitting result with a variable window. Norm of the force (thumb sensor-drinking a glass task): blue, limits of the analysis window: red, model: green.

The main drawback of the curve fitting approach is the definition of the model. This requires a good a priori knowledge of the time evolution of the effort. Even if one has a precise idea of the shape of the signal, it is not easy to formulate it in a mathematical function. In addition to this, a fine modelling should require different model for every task and every sensor. Models presented here and in [15] were widely inspired by the evolution of the norm of the force during the *drinking a glass* task for the thumb sensor.

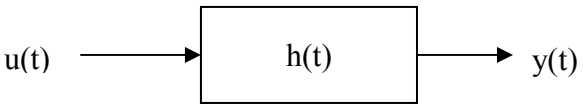
To avoid the definition of numerous complex models, we focused only on very simple characteristics such as the dominant time constant (T_1) that are well-defined for various signals.

The system identification approach, presented in the following paragraph, will allow a much finer modelling of various kinds of response. Moreover, the model complexity can easily be increased just by considering more singularities. Finally, the transfer function (the outcome of a system identification approach) can be decomposed in elementary contributions corresponding to damped cosines and exponentials, allowing an easy interpretation in terms of time constants (consequently in terms of rise times) and an easy way to include a priori knowledge about the dynamics of the system.

2.1.2.3.6. System Identification

The blind system identification approach

The model-based approach is widely used in biomedical engineering to study, simulate or control muscle dynamics [10], [11], [12], [13], [14]. Measurement data are viewed as the response of a system to an input signal $u(t)$ through a transfer function $h(t)$, describing the muscle dynamics.

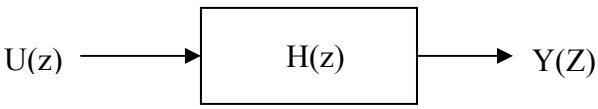


With:

$$y(t) = h(t) \otimes u(t)$$

Where $\otimes$ is the convolution operator.

When dealing with numerical data, the z-transform is a powerful tool to study discrete time systems. In the discrete time domain, the relationship above becomes:



With:

$$Y(Z) = H(Z)U(Z)$$

Various types of models can be defined within this very general scheme depending on the underlying system and on the expected level of details.

Some of them are built upon sub-system modelling. Key sub-systems are identified inside the system under consideration (for instance separate muscle or muscle groups, joints,…). Each of them is modelled with the desired level of details and the global model is obtained by assembling the sub-models in an appropriate way. Sometimes, estimates of physiological parameters are required (e.g. Hill-based model). This approach implies a detailed understanding of the studied system and how the different sub-systems interact.

On the contrary, another approach consists in considering the system as a black-box without taking care of the internal behaviour. This approach is particularly suitable when one is only interested in picking the main characteristics of the system. (e.g. dynamic features such as the transient or steady response). It does not require a detailed description of the studied system. Therefore, it is very useful when underlying processes are not well-known.

This latter approach will be adopted in this work. Indeed, the ADL tasks, used to assess the recovery level of a patient in this project, involve complex sensory and cognitive mechanisms. A complete mathematical model of the patient behaviour doesn't exist. In addition to this, the goal of this modelling study is to pick the main characteristics of the considered part of the body dynamics, and use them as features (model polynomials or derived physical properties) and not to get a detailed understanding of it.

The Figure 9 depicts the model structure adopted in this work. A unique structure will be considered for all the sensors and all the tasks: sensors are independent of each other in this modelling study. Cross-sensor features already take the interactions between sensors into account (mutual information, cross-sensor delay). $U(z)$ is the target response (for instance moving the hand from the left to the right, or grasping a glass). $T(z)$ is the actual neuro-muscular input signal to the part of the body under consideration. $N1(z)$ represents the cognitive activity transforming the willing into neuro-muscular signal (moto-neurone pool). $N2(z)$ takes the neuronal down path (spinal cord,…) into account. $Y(z)$ is the force/torque component measured by the sensor. It is assumed to be equal to the actually produced force/torque component (perfect sensor). $H(z)$ describes the muscle dynamics. $F(z)$ models the feedback. It globalizes the neuronal up path (including the sensor reaction force perception) and the cognitive activity.

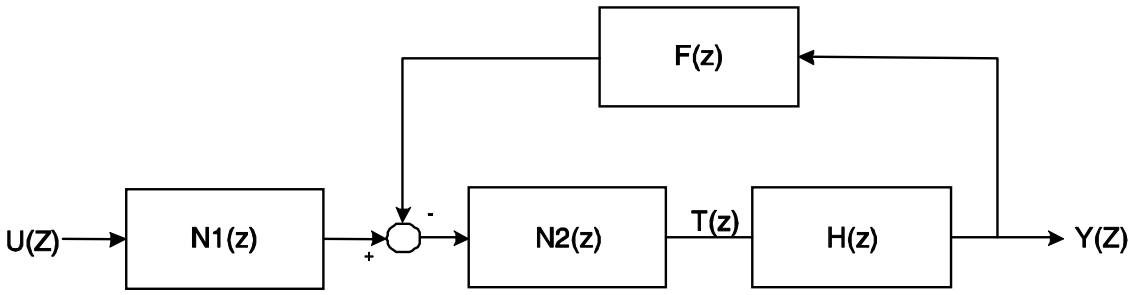


Figure 9: Model structure for all tasks and all the sensors

As already introduced above, most of these variables and transfer functions are unavailable. Moreover, the recovery level assessment is based on the ADL task accomplishment and not on the detailed internal behaviour. Therefore, we will focused only on the resulting transfer function $G(Z)$.

$$Y(z) = G(z)U(z)$$

Although most described physical sub-systems are known to be non linear, linearity will be assumed. Each transfer function is the quotient of two polynomials and:

$$G(z) = \frac{N1(z)N2(z)H(z)}{1 + F(z)N2(z)H(z)}$$

Despite it is seemingly a rough approximate, the linearity assumption is made in many works in the literature for a limited part of a system or even for a complete description. For instance, in [10], the precision grip force dynamics was modelled by a second order AR (*AutoRegressive*) transfer function, leading to results in great agreement with previously published results. In [12], the intrinsic contribution to the muscle dynamics was modelled by a second order AR transfer function. Linear systems have the advantage of being easy to predict and to interpret.

The system identification approach requires knowledge about both the output and input to fit the model parameters. In our case, input signal shapes are unknown and differ from one task to another one. To circumvent this problem, the input signal is modelled as the response of an unknown system $C(z)$ to a step function. $C(z)$ may depend on the sensor and the task but not on the patient (for a given task and a given sensor).

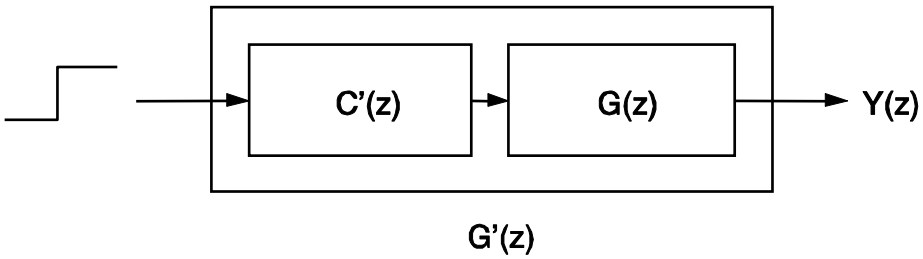


Figure 10: Input signal modeling

$$Y(z) = G(z)C(z)\frac{z}{z-1}$$

$$Y(z) = G'(z)\frac{z}{z-1}$$

$C(z)$ will be identified at the same time as $G(z)$. For instance, if the input signal is actually a ramp $C(z) = \frac{z}{z-1}$. For a rising slope followed by a steady effort, $C(z)$ is a two-pole transfer function.

As a black box, we used an *Output-Error* model for $G'(z)$.

$$Y_E(z) = G'(z)U(z) + E(z)$$

Where $E(z)$ is the z-transform of the simulation error and $Y_E(z)$ is the z-transform of the experimental data. The input signal is considered to be noise free so that any difference between experimental data and predicted samples is assumed to be measurement noise. This model uses a fraction to describe the system with two polynomials as numerator and denominator.

$$G'(z) = \frac{B(z)}{F(z)}$$
$$B(z) = b_1 + b_2 z^{-1} + b_3 z^{-2} + \ldots + b_{n_b} z^{-n_b+1}$$
$$F(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_{n_f} z^{-n_f}$$

This leads to the following relationship in the temporal domain:

$$y(t) = -\sum_{i=1}^{n_f} a_i y(t - iT_s) + \sum_{j=1}^{n_b} b_j u(t - (j-1)T_s) + e(t)$$

n_b is at most equal to n_f ($d^\circ(B(z)) < d^\circ(F(z))$). Indeed, by the *Initial Value Theorem*:

$$y(0) = \lim_{z\to\infty} Y(z) = \lim_{z\to\infty} G'(z)\frac{z}{z-1}$$

$y(0) = 0$ because no physical system can react instantaneously. Therefore
$d^\circ(B(z)) < d^\circ(F(z))$.

Parameter estimations

The parameters (a_i and b_j) estimation was handled by the `oe` function from the *Matlab System Identification Toolbox*. This method uses an iterative procedure to minimize the prediction error (see the *Matlab System Identification Toolbox* documentation) from an initial guess. The *Levenberg-Macquardt* method will be used (see [8] (pages 681-688) and [9]). A detailed description of the parameter estimation procedure can be found in A.2.

In order to assess the goodness of fit for the model that comes out from the system identification $\phi(iT_s, \theta)$ we use the *Normalized Error* $NE(\theta)$:

$$NE(\theta) = \sqrt{\frac{\sum_{i=1}^{N} (y_i - \phi(iT_s, \theta))^2}{\sum_{i=1}^{N} y_i^2}}$$

$NE(\theta)$ is also expected to give indications about the smoothness of the voluntary movement. It is independent of the signal energy and the number of considered samples (thus of the analysis window width) and is expected to allow a consistent comparison between all the patients.

Model orders

The last important issue for the system identification approach is the determination of the model order. The model order results from a trade-off: the higher the order the more the model will be close to the experimental data but the more it will be likely to overfit the data. In addition to this, we have to keep in mind that the main goal of the system identification approach is to derive discriminative features from the model. Thus, we will choose the model order that maximizes the difference between the $NE(\theta)$ averaged over the patients and the the $NE(\theta)$ averaged over the normal controls.

Ideally, the order should be determined for every task, every sensor, every X-Y-Z component if we consider components separately, for both torque and force. Considering orders ranging from $n_b = 1$ and $n_f = 1$ to $n_b = 8$ and $n_f = 8$, this would require the estimate of 3456 models for each patient and 10368 models if we consider X-Y-Z component separately.

In order to reduce the number of tests, we first experimentally determined the order for all the sensors (norm of the force) considering only the *drinking a glass* task. In a second step, model orders were determined for the thumb sensor (norm of the force) for all the other tasks. The orders of the remaining models were extrapolated using a priori knowledge about the complexity of the tasks.

Torque models were assumed to have the same orders as the force models since the dynamics of the underlying physiological system is the identical.

Similarly we estimated the model orders for the X-Y-Z components of the efforts partly by an experimental way, partly by extrapolation. Although driven by the need for reducing of the number of test, this sub-optimal approach was justified by the fact that the model orders don't vary a lot. The resulting orders for the norm of an effort, the X-component, the Y-component, the Z-component are presented in Table 1, Table 2, Table 3, Table 4 respectively.

TID/SID	1	2	3	4	5	6	7	8
1	55	55	55	44	44	33	44	33
2	55	55	55	44	44	33	44	33
3	66	66	66	44	44	33	44	33
4	66	66	66	44	44	33	44	33
5	55	55	55	44	44	33	44	33
6	66	66	66	44	44	33	44	33

Table 1: Model order (1rst digit=nb, second digit=nf) for the norm of an effort (force or torque) for every task and every sensor.

TID/SID	1	2	3	4	5	6	7	8
1	44	44	44	44	44	33	44	33
2	44	44	44	44	44	33	44	33
3	55	55	55	44	44	33	44	33
4	55	55	55	44	44	33	44	33
5	44	44	44	44	44	33	44	33
6	55	55	55	44	44	33	44	33

Table 2: Model order (1rst digit=nb, second digit=nf) for the X-component of an effort (force or torque) for every task and every sensor.

TID/SID	1	2	3	4	5	6	7	8
1	44	44	44	44	44	33	44	33
2	44	44	44	44	44	33	44	33
3	55	55	55	44	44	33	44	33
4	55	55	55	44	44	33	44	33
5	44	44	44	44	44	33	44	33
6	55	55	55	44	44	33	44	33

Table 3: Model order (1rst digit=nb, second digit=nf) for the Y-component of an effort (force or torque) for every task and every sensor.

TID/SID	1	2	3	4	5	6	7	8
1	44	44	44	44	44	33	44	33
2	44	44	44	44	44	33	44	33
3	55	55	55	44	44	33	44	33
4	55	55	55	44	44	33	44	33
5	44	44	44	44	44	33	44	33
6	55	55	55	44	44	33	44	33

Table 4: Model order (1rst digit=nb, second digit=nf) for the Z-component of an effort (force or torque) for every task and every sensor.

2.1.2.3.7. Cross-Sensor delay and mutual information

Different approaches will be considered here. A first lead could consist in measuring the difference in time between the initiation points from the 2 different sensors. This is very simple to calculate, while the onset of movement times is already estimated from a particular method.

Another more statistically founded method is to calculate the delay between different sensors under which the *Mutual Information* between different sensors is maximized. Inspiration should be found here from image registration. In order not to calculate the mutual information between every possible component (X, Y or Z) with every other component from another sensor, one could e.g. correlate the energy or magnitude between different sensors. With means of the mutual information the optimal delay $a_{optimal}$ can be found as:

$$a_{optimal} = \max_{a}\left(I\left(\left\|\vec{F}_{s1}(k)\right\|,\left\|\vec{F}_{s2}(k-a)\right\|\right)\right)$$

With, for two discrete random variables X and Y:

$$I(X,Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log\left(\frac{p(x,y)}{p(x)\,p(y)} \right)$$

The mutual information under this optimal delay $a_{optimal}$ could be a useful feature:

$$I_{a\_optimal} = I\left(\left\| \vec{F}_{s1}(k) \right\|, \left\| \vec{F}_{s2}(k - a_{optimal}) \right\| \right)$$

Note that only a part of the force vectors in time can be taken to allow for a time shift.
This mutual information can be a measure of how good synchronization is between different movements from different sensors. Mutual information can be computed from histograms (however not optimal), to keep things simple.

2.1.3. Feature Subset Selection

2.1.3.1. Overview

Once all the features were extracted from the proof of concept database, the aim of the *Feature Subset Selection* (FSS) was to identify the most discriminative feature among all the candidate features. This task was carried out by the KUL data mining expert's team. Therefore, we will only mention the key steps of the algorithm. More details can be found in [5] and [6] from which this section was widely inspired.

An exhaustive search for a subset of features of size N_f leading to the best classification rate in a feature set of size N should require:

$$\frac{!N}{!(N - N_f)!N_f}$$

estimates of classification rate. The dataset submitted to the FSS algorithm counted 59472 features (see section 2.2.2). According to the expression above, an exhaustive search for the 6 most pertinent features should lead to the estimate of about $6.5\ 10^{25}$ classification rates! Therefore, such approach is not feasible.

The proposed strategy is a hybrid filter/wrapper approach. A schematic overview is presented in Figure 11. Although, this hybrid approach does not guarantee an optimal solution, it has been shown to yield results very closed to the optimal. Up to now, no technique (except an exhaustive search) can guarantee an optimal solution.

First, irrelevant and redundant features are removed in the filtering step. Second, an optimal subset is searched among the remaining features. The filter step is independent of the induction model used in the feature subset search. The hybrid approach presents two major advantages: the increase in interpretability of the feature subsets and the speed-up of the wrapper approach in the search for the optimal feature subsets.

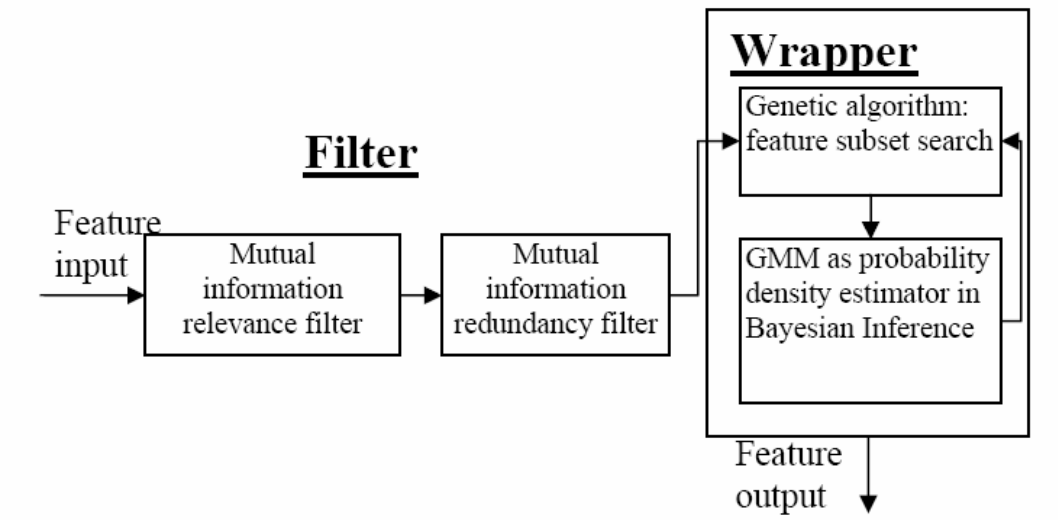


Figure 11: Schematic overview of the overall feature subset selection strategy for classification of normal control and patients. First, irrelevant and redundant features are removed in the filter. Second, the wrapper approach focuses on the smaller set of interesting features.

2.1.3.2. Relevance filter

As mentioned in section 2.1.3.1, the aim of the relevance analysis is to remove features that don't carry information about a label[1] (here, either *Patient* or *Normal Controls*). The relevance of a feature F_i with regard to a label C is assessed by the *Mutual Information* $I(F_i, C)$:

$$I(F_i, C) = \int_{F_i} \int_{C} P(f_i, c) \log_2 \left(\frac{P(f_i, c)}{P(f_i) P(c)} \right) df_i dc$$

Where $P(f_i)$ is the marginal probability distribution of the feature F_i, $P(c)$ is the marginal probability distribution of the label C, $P(f_i, c)$ is the joint probability distribution of F_i and C. A high mutual information value means that F_i is relevant while a low value means it is not. In practice, the mutual information is computed on finite sample size so that the probability distributions are not known. The estimated value will thus depend on the estimator, the sample size and the sample distribution (see [7]). This makes risky to judge whether a feature is relevant only on the basis of this value. A smart solution consists in computing the distribution of the mutual information under random permutations π_k of the labels:

$$I_{\pi_k}(F_i, C_{\pi_k}) = \int_{F_i} \int_{C} P(f_i, c_{\pi_k}) \log_2 \left(\frac{P(f_i, c_{\pi_k})}{P(f_i) P(c)} \right) df_i dc$$

[1] In the following, *label* refers to the set of classes. A *class* thus corresponds to a particular value of a label.

since $P(c) = P(c_{\pi_k})$. Considering different permutations, we can derive the distribution of the mutual information under the null hypothesis (H_0) of feature F_i being irrelevant. Then, it becomes trivial to deduce the following statistical test:

$$\begin{cases} H_0 \text{ accepted } (F_i \text{ irrelevant}) \text{ if } \hat{I}(F_i, C) \leq \tau_\alpha \\ H_0 \text{ rejected } (F_i \text{ relevant}) \text{ if } \hat{I}(F_i, C) > \tau_\alpha \end{cases}$$

Where α is the confidence level and $\hat{I}(F_i, C)$ is the mutual information estimator and τ_α is defined as $P(\hat{I}(F_i, C) < \tau_\alpha) = \alpha$.

2.1.3.3. Redundancy filter

The goal of the redundancy filter is to merge features carrying highly overlapping information. The metric used to measure the distance between two features in terms of information is based on the *Normalized Mutual Information*:

$$I_n(F_i, F_j) = \frac{2I(F_i, F_j)}{H(F_i) + H(F_j)} = \frac{2\left(H(F_i) + H(F_j) - H(F_i, F_j)\right)}{H(F_i) + H(F_j)}$$

Where $H(F) = \int_F P(f) \log_2 (P(f)) df$ is the entropy. The distance is then defined as follows:

$$d(F_i, F_j) = 1 - I_n(F_i, F_j)$$

The distance between two sets is the maximum distance between any two elements from the two sets.

2.1.3.4. Wrapper search

In this final step, the optimal feature subset is selected among features that were not filtered in the previous stage. The search strategy was a genetic algorithm specially tuned for the FSS problem (see [5]). The induction model was a *Gaussian Mixture Model* (GMM), suitable for the available amount of data in the proof of concept.

2.1.4. Classification

2.1.4.1. General approach

A small set of relevant features have been identified among the list of candidates defined in section 2.1.2.3. The recovery space paradigm (see section 2.1.1.1) relies on the assumption that the state of a stroke patient at a given instant is sufficiently characterized by the set of features, the so-called feature vector. The primary goal of the classification is thus to see whether patients and normal controls can be discriminated in the recovery space (see section 2.1.1.1) on the basis of these features. However, although this condition is a prerequisite to any further investigation, we have to keep in mind that the final aim of the data mining task is to identify markers and milestone in the patient recovery trajectories.

As the rehabilitation progresses, the feature vector of a patient should move toward the region characterizing the normal control in the recovery space. Therefore, discriminating patient and normal controls is expected to become more and more difficult and the classification rate to progressively decrease.

A last issue classification has to deal with is the choice of a statistical inference model. Indeed, the statistical approach adopted for the definition of a recovery metric relies on a statistical modelling (see section 2.1.5) of the two classes (namely patients and normal controls). Two criteria will be taken into account in that choice: the classification rate, assumed to be a good indicator of the ability to model the classes and its evolution during the rehabilitation, expected to point out a potential overfitting problem.

Two models will be discussed: a *Gaussian Mixture Model* (GMM) and a *Kernel Density Estimation* (KDE).

2.1.4.2. Gaussian Mixture Model

These models expressed the probability density function $P(Y)$ as a weighted sum of Gaussian functions,

$$p(Y) = \sum_{i=1}^{K} w_{\mathrm{N},i}\phi(Y;\mu_{\mathrm{N},i},\Sigma_{\mathrm{N},i})$$

with the i-th component $\phi(X;\mu_{\mathrm{N},i},\Sigma_{\mathrm{N},i})$ given as a Gaussian density function with mean vector $\mu_{\mathrm{N},i}$ and covariance matrix $\Sigma_{\mathrm{N},i}$, that is,

$$\phi(Y;\mu_{\mathrm{N},i},\Sigma_{\mathrm{N},i}) = \frac{1}{(2\pi)^{L/2}\left|\Sigma_i\right|^{1/2}}\exp\left(-\frac{1}{2}(Y-\mu_{\mathrm{N},i})^T\Sigma_i^{-1}(Y-\mu_{\mathrm{N},i})\right)$$

where L denotes the dimension of the feature vector. Besides, the mixing weights $w_{\mathrm{N},i}$ satisfy the constraint $\sum_{i=1}^{K} w_{\mathrm{N},i} = 1$.

The estimates $(\hat{w}_{\mathsf{N},i}, \hat{\mu}_{\mathsf{N},i}, \hat{\Sigma}_{\mathsf{N},i})$ with $i = 1, \cdots, K$ of the GMM parameters can be obtained by maximizing the log-likelihood of a given set of feature vectors drawn from the population under consideration. Assuming that the feature vectors are statistically independent, it is expressed as follows,

$$(\hat{w}_{\mathsf{N},i}, \hat{\mu}_{\mathsf{N},i}, \hat{\Sigma}_{\mathsf{N},i}) = \arg\max_{w_{\mathsf{N},i}, \mu_{\mathsf{N},i}, \Sigma_{\mathsf{N},i}} \sum_{Y_n \in \mathsf{N}} \log\left(\sum_{i=1}^{K} w_{\mathsf{N},i} \phi(Y_n; \mu_{\mathsf{N},i}, \Sigma_{\mathsf{N},i}) \right).$$

This Maximum Likelihood (ML) problem is actually a missing data problem that is classically addressed via the Expectation-Maximization (EM) algorithm. Given an initial estimate of the parameters, the EM algorithm iteratively refines the estimates in a two-step procedure in order to increase monotically the log-likelihood of a learning set of feature vectors. Some learning algorithms are described hereafter.

Basic EM algorithm for GMM estimation

1. Collect learning feature vectors $\{Y_1, \cdots, Y_N\} \in \mathsf{N}$.

2. Get initial estimates of the parameters $(\hat{w}_{\mathsf{N},i}^{(0)}, \hat{\mu}_{\mathsf{N},i}^{(0)}, \hat{\Sigma}_{\mathsf{N},i}^{(0)})$ and set $l = 0$.

3. Compute *a posteriori* data membership probability for every feature vector and for each component (E-step),

$$\gamma_i^{(l)}(Y_n) = \frac{\hat{w}_{\mathsf{N},i}^{(l)} \phi(Y_n; \hat{\mu}_{\mathsf{N},i}^{(l)}, \hat{\Sigma}_{\mathsf{N},i}^{(l)})}{\sum_{i=1}^{K} \hat{w}_{\mathsf{N},i}^{(l)} \phi(Y_n; \hat{\mu}_{\mathsf{N},i}^{(l)}, \hat{\Sigma}_{\mathsf{N},i}^{(l)})}, \quad i = 1, \cdots, K \text{ and } n = 1, \cdots, N.$$

4. Update model parameters one component at a time (M-step),

$$\hat{w}_{\mathsf{N},i}^{(l+1)} = \frac{1}{N} \sum_{n=1}^{N} \gamma_i^{(l)}(Y_n), \; i = 1, \cdots, K$$

$$\hat{\mu}_{\mathsf{N},i}^{(l+1)} = \frac{\sum_{n=1}^{N} \gamma_i^{(l)}(Y_n) Y_n}{\sum_{n=1}^{N} \gamma_i^{(l)}(Y_n)}, \; i = 1, \cdots, K$$

$$\hat{\Sigma}_{\mathsf{N},i}^{(l+1)} = \frac{\sum_{n=1}^{N} \gamma_i^{(l)}(Y_n)(Y_n - \hat{\mu}_{\mathsf{N},i}^{(l+1)})(Y_n - \hat{\mu}_{\mathsf{N},i}^{(l+1)})^T}{\sum_{n=1}^{N} \gamma_i^{(l)}(Y_n)}, \; i = 1, \cdots, K.$$

5. Stop if convergence is reached, otherwise set $l = l + 1$ and go to 3.

Reliable estimation of full covariance matrices is a sensitive issue in GMM estimation. The estimation algorithms often lead to badly conditioned matrices that poorly inverse. It is often assumed that the coefficients of the feature vectors are mutually independent; thereby the covariance matrices are diagonal. The re-estimation equations presented above for full covariance matrices can be straightforwardly adapted to diagonal matrices.

The choice of the initial estimates is another sensitive issue in the EM algorithm. The algorithm is only guaranteed to converge to a local maximum of the log-likelihood function and it is highly dependent on the initial conditions. Various schemes can be proposed to address this issue. For instance, several initial conditions can be randomly generated within the parameter space. The estimation procedure is applied for every starting point and the final estimates are picked up such that the final log-likelihood is maximized across all candidates. Alternatively, unsupervised quantization can be used to cluster the learning feature vectors (see section 5) and the cluster relative sizes. Mean vectors and covariance matrices are then used as initial parameters. Combinations of both approaches are possible. Still another approach consists in inserting one component after the other in the mixture. It first estimates a one-component mixture whose initial parameters are trivially estimated over the whole set of learning vectors. Then, a new component is inserted and the EM algorithm is applied to the partial mixture. These steps are repeated until proper mixture size is reached. The complete algorithm is described hereafter.

Greedy EM algorithm for GMM estimation

More details about this algorithm can be found in [16].

1. Collect learning feature vectors $\{Y_1, \cdots, Y_N\} \in \mathsf{N}$.

2. Set $k = 1$ and compute initial parameters,

$$\hat{w}_{\mathsf{N},1} = 1$$

$$\hat{\mu}_{\mathsf{N},1} = \frac{1}{N}\sum_{n=1}^{N} Y_n$$

$$\hat{\Sigma}_{\mathsf{N},1} = \frac{1}{N}\sum_{n=1}^{N} (Y_n - \hat{\mu}_{\mathsf{N},1})(Y_n - \hat{\mu}_{\mathsf{N},1})^T .$$

3. Find next mixture component $\phi(X; \mu_{\mathsf{N},k+1}, \Sigma_{\mathsf{N},k+1})$ and mixing coefficient $w_{\mathsf{N},k+1}$

   3.1. Compute *a posteriori* data membership probability for every feature vector and for each available component,

$$\gamma_i(Y_n) = \frac{\hat{w}_{\mathsf{N},i}\phi(Y_n; \hat{\mu}_{\mathsf{N},i}, \hat{\Sigma}_{\mathsf{N},i})}{\sum_{i=1}^{k} \hat{w}_{\mathsf{N},i}\phi(Y_n; \hat{\mu}_{\mathsf{N},i}, \hat{\Sigma}_{\mathsf{N},i})}, \quad i = 1, \cdots, k \text{ and } n = 1, \cdots, N .$$

3.2. Perform maximum *a posteriori* classification of every feature vector,

$$\mathsf{S}_i = \left\{ Y_n : i = \arg\max_{j=1,\cdots,k} \gamma_j(Y_n) \right\},\ i = 1,\cdots,k\,.$$

3.3. Apply unsupervised quantization of every subset S_i into J clusters (*e.g.*, $J = 2$),

$$\mathsf{S}_i = \bigcup_{j=1}^{J} \mathsf{S}_{ij}\ ,\ \mathsf{0} = \bigcap_{j=1}^{J} \mathsf{S}_{ij}\ ,\ i = 1,\cdots,k\,.$$

3.4. For $i = 1,\cdots,k$ and for $j = 1,\cdots,J$

3.4.1. Compute statistics over feature vectors in cluster under consideration

$$w_{S_{ij}} = \hat{w}_{\mathsf{N},i} \frac{\sum_{Y_n \in S_{ij}} 1}{\sum_{Y_n \in S_i} 1}$$

$$\mu_{S_{ij}} = \frac{\sum_{Y_n \in S_{ij}} Y_n}{\sum_{Y_n \in S_{ji}} 1}$$

$$\Sigma_{S_{ij}} = \frac{\sum_{Y_n \in S_{ij}} (Y_n - \mu_{S_{ij}})(Y_n - \mu_{S_{ij}})^T}{\sum_{Y_n \in S_{ji}} 1}\,.$$

3.4.2. Set $l = 0$ and use $(w_{S_{ij}}, \mu_{S_{ij}}, \Sigma_{S_{ij}})$ as initial conditions $(\hat{w}^{(0)}_{\mathsf{N},k+1}, \hat{\mu}^{(0)}_{\mathsf{N},k+1}, \hat{\Sigma}^{(0)}_{\mathsf{N},k+1})$ to EM procedure for estimating $(w_{\mathsf{N},k+1}, \mu_{\mathsf{N},k+1}, \Sigma_{\mathsf{N},k+1})$ in two-component mixture problem

$$(\hat{w}_{\mathsf{N},k+1}, \hat{\mu}_{\mathsf{N},k+1}, \hat{\Sigma}_{\mathsf{N},k+1}) = \arg\max_{w_{\mathsf{N},k+1}, \mu_{\mathsf{N},k+1}, \Sigma_{\mathsf{N},k+1}} \sum_{n=1}^{N} \log \left(\begin{array}{c} (1 - w_{k+1}) \sum_{i=1}^{k} \hat{w}_{\mathsf{N},i} \phi(Y_n; \hat{\mu}_{\mathsf{N},i}, \hat{\Sigma}_{\mathsf{N},i}) \\ + w_{k+1} \phi(Y_n; \mu_{\mathsf{N},k+1}, \Sigma_{\mathsf{N},k+1}) \end{array} \right).$$

3.4.3. Compute *a posteriori* data membership probability for every feature vector and for candidate component (E-step),

$$\gamma^{(l)}_{k+1}(Y_n) = \frac{\hat{w}^{(l)}_{\mathsf{N},k+1} \phi(Y_n; \hat{\mu}^{(l)}_{\mathsf{N},k+1}, \hat{\Sigma}^{(l)}_{\mathsf{N},k+1})}{(1 - \hat{w}^{(l)}_{\mathsf{N},k+1}) \sum_{i=1}^{k} \hat{w}_{\mathsf{N},i} \phi(Y_n; \hat{\mu}_{\mathsf{N},i}, \hat{\Sigma}_{\mathsf{N},i}) + \hat{w}^{(l)}_{\mathsf{N},k+1} \phi(Y_n; \hat{\mu}^{(l)}_{\mathsf{N},k+1}, \hat{\Sigma}^{(l)}_{\mathsf{N},k+1})}\,.$$

3.4.4. Update **only** parameters for candidate component (M-step),

$$\hat{w}_{\mathsf{N},k+1}^{(l+1)} = \frac{1}{N}\sum_{n=1}^{N}\gamma_{k+1}^{(l)}(Y_n)$$

$$\hat{\mu}_{\mathsf{N},k+1}^{(l+1)} = \frac{\sum_{n=1}^{N}\gamma_{k+1}^{(l)}(Y_n)Y_n}{\sum_{n=1}^{N}\gamma_{k+1}^{(l)}(Y_n)}$$

$$\hat{\Sigma}_{\mathsf{N},k+1}^{(l+1)} = \frac{\sum_{n=1}^{N}\gamma_{k+1}^{(l)}(Y_n)(Y_n-\hat{\mu}_{\mathsf{N},k+1}^{(l+1)})(Y_n-\hat{\mu}_{\mathsf{N},k+1}^{(l+1)})^T}{\sum_{n=1}^{N}\gamma_{k+1}^{(l)}(Y_n)}.$$

3.4.5. Stop if convergence is reached, otherwise set $l = l+1$ and go to 3.4.3.

3.4.6. Store final parameters $(\hat{w}_{\mathsf{N},k+1}^{(l)}, \hat{\mu}_{\mathsf{N},k+1}^{(l)}, \hat{\Sigma}_{\mathsf{N},k+1}^{(l)})$ as parameters of candidate component $(w_{\mathsf{S}_{ij}}, \mu_{\mathsf{S}_{ij}}, \Sigma_{\mathsf{S}_{ij}})$ for later selection.

3.5. Select best candidate component, *i.e.* leading to the largest log-likelihood increase

$$(\hat{w}_{\mathsf{N},k+1}, \hat{\mu}_{\mathsf{N},k+1}, \hat{\Sigma}_{\mathsf{N},k+1}) = \arg\max_{i,j}\sum_{n=1}^{N}\log\begin{pmatrix}(1-w_{\mathsf{S}_{ij}})\sum_{i=1}^{k}\hat{w}_{\mathsf{N},i}\phi(Y_n;\hat{\mu}_{\mathsf{N},i},\hat{\Sigma}_{\mathsf{N},i}) \\ +\, w_{\mathsf{S}_{ij}}\phi(Y_n;\mu_{\mathsf{S}_{ij}},\Sigma_{\mathsf{S}_{ij}})\end{pmatrix}.$$

3.6. Update weight coefficients $\hat{w}_{\mathsf{N},i} = (1-\hat{w}_{\mathsf{N},k+1})\hat{w}_{\mathsf{N},i}$, $i = 1,\cdots,k$.

4. Refine mixture parameters via basic EM algorithm using $(\hat{w}_{\mathsf{N},i}, \hat{\mu}_{\mathsf{N},i}, \hat{\Sigma}_{\mathsf{N},i})$, $i = 1,\cdots,k+1$ as initial parameters.

5. Stop if maximum number of component is reached, otherwise set $k = k+1$ and go to 3.

Like for the basic EM algorithm for GMM estimation, the greedy algorithm can be modified in order to use diagonal covariance matrices instead of full covariance matrices. Besides, more elaborate stopping criterions can be used instead of simply check that the maximum number of component has been reached. For instance, model order selection techniques (*e.g.*, Minimum Description length (MDL), Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), etc) can be applied to stop the mixture expansion.

Remark – Other methods for estimating GMM can be tested if EM-based methods fail: Figueiredo-Jain (FJ) algorithm, non-linear least square estimation by Gauss-Newton (GN) algorithm, Markov Chain Monte Carlo (MCMC) method, Cross-Entropy (CE) algorithm, etc.

An important particular case in this study will be the 1-order GMM. Due to the dataset size the number of kernels (order of the GMM) is limited. This first-order model will be widely used with the normalized likelihood metric. Then, the GMM reduces to a multivariate Gaussian distribution. In this case the parameter estimation becomes trivial since it consists only in the calculation of the mean vector and the covariance matrix from the feature dataset:

$$\hat{\mu}_{\mathrm{N}} = \frac{1}{N}\sum_{n=1}^{N} Y_n$$

$$\hat{\Sigma}_{\mathrm{N}} = \frac{1}{N}\sum_{n=1}^{N} (Y_n - \hat{\mu}_{\mathrm{N}})(Y_n - \hat{\mu}_{\mathrm{N}})^T$$

The initialization is not a problem anymore.

2.1.4.3. Kernel Density Estimation

The *Kernel Density Estimation* (KDE) is a non parametric density estimator commonly used in the statistical data analysis field. In this approach, the unknown true *Probability Density Function* (pdf) $f(x)$ is approximated by as a sum of sample-centred elementary contributions (the so-called *Kernel Functions*). In the multivariate case, product kernels can be used. The estimated probability density function $\hat{f}(x)$ inferred from n observations of the random variable $x = (x_1, ..., x_d)$ is then:

$$\hat{f}(x) = \frac{1}{n h_1 ... h_d}\sum_{i=1}^{n}\left\{\prod_{j=1}^{d} K\left(\frac{x_j - x_{ij}}{h_j}\right)\right\} = \frac{1}{n}\sum_{i=1}^{n}\left\{\prod_{j=1}^{d} K_{h_j}\left(x_j - x_{ij}\right)\right\}$$

With $$\int_{\Re_d} K(x)\,dx = 1$$

Where n is the sample size, d is the number of dimensions, h_k (k=1...d) are the kernel bandwidths, $x_i = (x_{i1}, ..., x_{id})$ is the i^{th} observation of the random variable x, x_{ij} is the j^{th} component of x_i and K stands for the kernel density function under consideration. Some commonly used kernel functions are reminded in Table 5. The kernel bandwidths can be constrained to be uniform ($h_i = h_j, \forall i, j$).

Kernel	K_{h_j}
Uniform	$K_{h_j} = \frac{1}{2} \quad (\left\| \frac{x_j - x_{ij}}{h_j} \right\| \leq 1)$
Epanechnikov	$K_{h_j} = \frac{3}{4}\left(1 - \left(\frac{x_j - x_{ij}}{h_j}\right)^2\right) \quad (\left\| \frac{x_j - x_{ij}}{h_j} \right\| \leq 1)$
Gaussian	$K_{h_j} = \frac{1}{\sqrt{2\pi} h_j} \exp\left(-\frac{1}{2}\left(\frac{x_j - x_{ij}}{h_j}\right)^2\right)$

Table 5: Commonly used kernel functions.

Once the kernel function has been chosen, the only parameters to be set are the kernel bandwidths. The estimate of the h_k (k=1...d) is the most sensitive issue of the KDE approach. Too small values of h_k will result in spurious features in the density estimate whereas too big values will exceedingly smooth the density estimate and hide informative details of the distribution. The influence of the kernel bandwidth value on the probability density function estimate is illustrated in the Figure 12, Figure 13 and Figure 14 for the 1-dimension case (Gaussian kernels). Then, $\hat{f}(x)$ reduces to:

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} \left\{ K\left(\frac{x_j - x_{ij}}{h}\right) \right\}$$

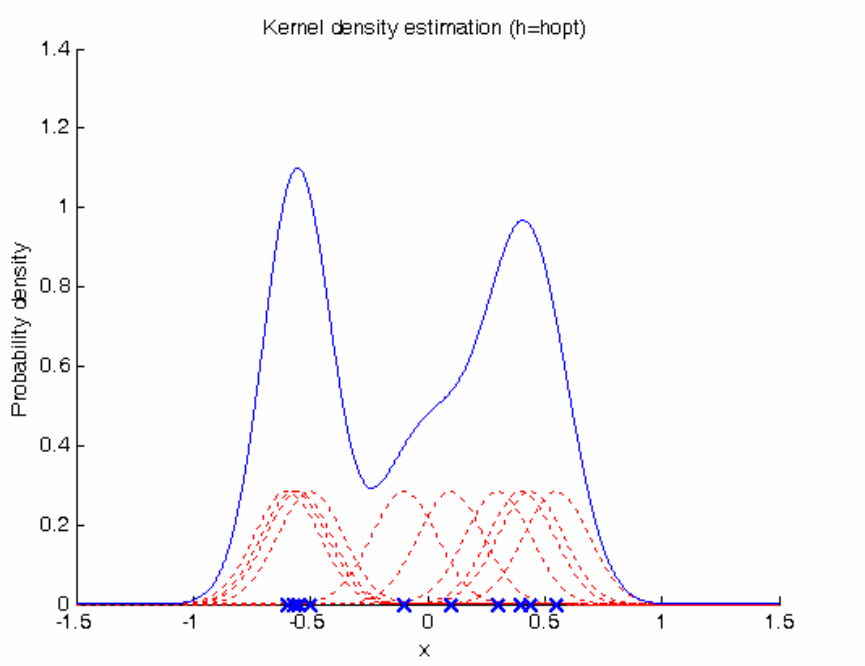


Figure 12: 1-dimension Kernel Density Estimate (Gaussian kernels). The kernel bandwidth was estimated using the maximum leave-one-out likelihood cross-validation criterion (hopt=0.14). Red dotted curves are the elementary probability masses (kernels), the blue thick stroke curve is the resulting pdf, the blue crosses represent the samples.

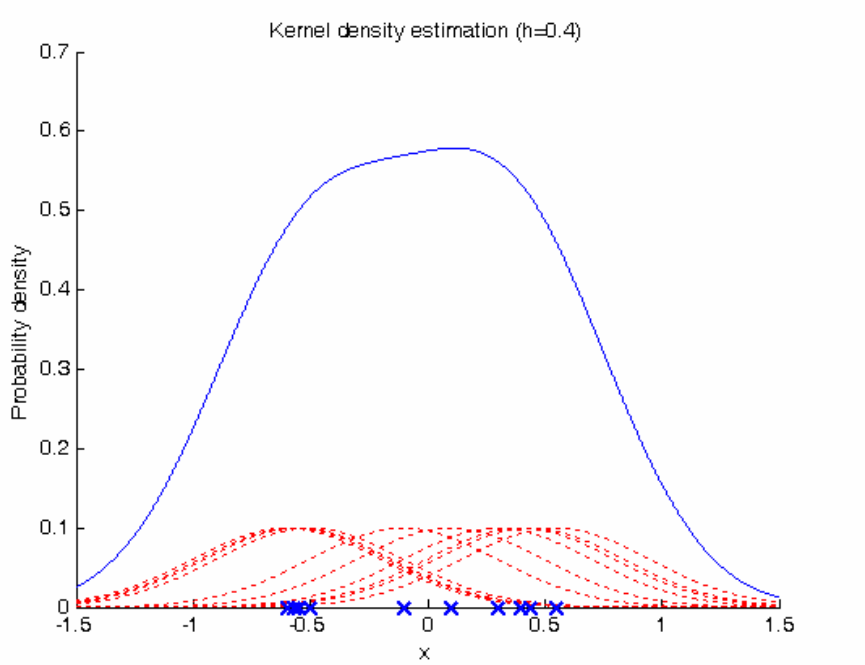


Figure 13: 1-dimension Kernel Density Estimate (Gaussian kernels) with a non-optimal kernel bandwidth (h=0.4). Red dotted curves are the elementary probability masses (kernels), the blue thick stroke curve is the resulting pdf, the blue crosses represent the samples. A too big value of the kernel bandwidth results in an exceedingly smooth probability density function.

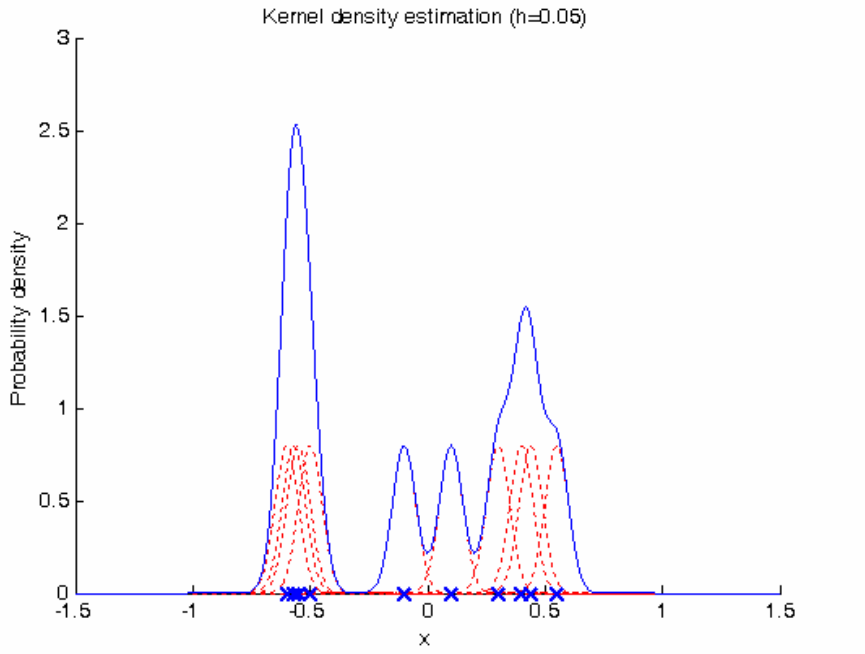


Figure 14: 1-dimension Kernel Density Estimate (Gaussian kernels) with a non-optimal kernel bandwidth (h=0.05). Red dotted curves are the elementary probability masses (kernels), the blue thick stroke curve is the resulting pdf, the blue crosses represent the samples. A too small value of the kernel bandwidth results in spurious features in the density estimate.

The bandwidth estimate has been widely addressed in the literature (see [17] for an overview). In this study, we will use a technique based on the maximum likelihood cross-validation criterion (see [18]). In the multivariate case, it relies on a 1-dimension search. Gaussian kernels will be used, with uniform bandwidths.

2.1.5. Distance To Normality (DTN)

The direct observation of relevant patterns in sequences of feature vectors in the recovery space is difficult because of the high dimension of these feature vectors (up to 6 dimensions in this study). The approach that we propose consists in the definition of a metric to measure how far feature vectors stand from normality.

2.1.5.1. The statistical approach

The distance to normality of a feature vector can be expressed in a statistical sense. To do so, we need first to estimate the probability distribution of observing some feature vector in the normal population given some samples $\mathsf{N} = \{Y_1, \cdots, Y_N\}$. Then, we can infer a probabilistic metric.

We propose two different metrics: the *A Posteriori Probability* of normality and the *Normalized Likelihood*. In both case the metric is not actually a distance since the higher score, the closer to the normal state.

2.1.5.1.1. A Posteriori Probability of normality

By the Bayes' rule, the *A Posteriori Probability* of normality $P(N \mid X)$ can be expressed as the product of the likelihood of observing the feature vector in case of normality $P(X \mid N)$ with the *a priori* probability of normality $P(N)$, divided by the *a priori* probability of observing the feature vector $P(X)$, that is:

$$P(N \mid X) = \frac{P(X \mid N) P(N)}{P(X)}$$

This expression can be developed further:

$$P(N \mid X) = \frac{P(X \mid N) P(N)}{P(X \mid N) P(N) + P(X \mid P) P(P)}$$

Where $P(X \mid P)$ is the likelihood of observing the feature vector when being a stroke patient, $P(P)$ is the a priori probability of being a patient.

The a posteriori probability is a very strongly founded theoretical concept in classification problems. Indeed, let Y be a feature vector corresponding to an item to be classified, it can be shown that assigning the category maximizing the likelihood:

$$C(Y) = \arg\left(\max_i \left(P(C_i \mid Y)\right)\right) = \arg\left(\max_i \left(\frac{P(Y \mid C_i)P(C_i)}{P(Y)}\right)\right)$$

asymptotically leads to the lowest classification error rate.

For classification, the a priori probabilities $P(C_i)$ have a precise definition: the relative part of C_i in the population. Here for calculating a posteriori probability based distance, the a priori probabilities, $P(P)$ and $P(N)$ are not well defined: in the recovery space, the normal controls are a reference population from which we try to measure how far we are.

In a classification problem, we intuitively understand that, all the other parameters let unchanged, most populated categories C_i are favored in the decision process ($P(C_i \mid Y)$ increases as $P(C_i)$ increases as well). On the contrary, when the a posteriori probability is used as a metric, assuming that inferred estimates of $P(X \mid N)$ and $P(X \mid P)$ are stable (samples from the two classes are representative), it is not acceptable that the distance to the normality changes if the database composition varies. Consequently, $P(P)$ and $P(N)$ will be set to the arbitrary value 0.5.

Two drawbacks can be formulated about this metric definition. First, the estimate of $P(X \mid P)$ may be problematic if there is a big variability in the patient population (limited amount of training data). Secondly, the a posteriori probability $P(N \mid X)$ can take high value although $P(X \mid N)$ is low when $P(X \mid P)$ is low simultaneously. The patient is thus said to be close to normality while his behaviour differs widely from the normal control population.

2.1.5.1.2. Normalized likelihood

This metric is simply defined as the normalized likelihood of observing the feature vector in case of normality:

$$P_{Norm}(X \mid N) = \frac{P(X \mid N)}{\max\left(P(X \mid N)\right)}$$

The major drawback of this metric is that $\max\left(P(X \mid N)\right)$ is difficult to calculate for a complex probability distribution. Therefore, it will be used only with a very simple distribution: a 1-order Gaussian Mixture Model (see section 2.1.5.2). The calculation of $\max\left(P(X \mid N)\right)$ becomes trivial:

$$\max\left(P\left(X \mid N\right)\right) = P\left(\overline{X}_{NC} \mid N\right)$$

Where $\overline{X}_{NC}$ is the mean feature vector corresponding to normal controls.

2.1.5.2. Probability distribution estimation

The metrics proposed in the section 2.1.5.1 imply the estimate of the density probability function $P\left(X \mid N\right)$ and $P\left(X \mid P\right)$ (both referred to as $P\left(Y\right)$ in this section). Two models were considered: the *Gaussian Mixture Model* (GMM) and the *Kernel Density Estimation* (KDE) model (see sections 2.1.4.2 and 2.1.4.3).

2.1.6. Markers and milestones

Clinical partners will interpret the distance to normality curves in order to find characteristic patterns. A preliminary study can be found in [3].

2.2. Experimental results

2.2.1. Databases

2.2.1.1. Methodology

ALLADIN uses ADL tasks as prime variables for its measurements because these tasks are indispensable for the well-being of a stroke patient. The assessment of a patient ability to perform the task is based on the analysis of repeated measurements of isometric force/torque trajectories related to different parts of the body. A complete description of the methodology followed during the clinical trial is described in details in [2]. Therefore, only concepts indispensable for framing the data mining task is reminded here.

Various factors (age, gender, lesion location and its consequences, co-morbidities, social factors,…) are likely to influence the stroke recovery (see [3]). Therefore, it is important to make sure that patients in the ALLADIN database are representative of the European stroke population.

The adopted approach was to choose the subjects on the basis of availability using the consecutive sampling method. This involved recruiting all patients who meet the inclusion and exclusion criteria as they become available (see Table 6 and Table 7) at the three clinical centres, resided in three different European countries.

Hemiparesis due to stroke
The brain impairment must be proved by CT or MRI
Minimum age is 18 years
The subject must be suitable to endure physical load during the measurements
The subject must be cooperative
Signed informed consent

Table 6: Inclusion criteria in ALLADIN

Restricted disposing capacity or legal incapacity
Prisoner
Movement or other disorder that makes it impossible for the patient to sit calmly during the treatment
Skin problem where use of an orthosis is contra- indicated
Patients shorter than 1530 mm Patients taller than 1870 mm
Patients with a weight in excess of 110 kg
Patients with a pre-stroke disability interfering with the goal of the study

Table 7: Exclusion criteria in ALLADIN. Patients who matched at least one of these criteria were automatically excluded from the clinical trials.

In practice, it is difficult, if not impossible, to obtain a 100 % true random sample. However, the more the clinical trial lasts, the more the group of patients in the database is representative of the overall stroke population. In ALLADIN, the measurement period began in February

2005 and ended September 2006. A total of 208 patients volunteered to participate. A survey of the literature shows that this number is similar to the sample sizes of other data mining experiments where measurements from clinical devices were used (see [2]).

Patients were measured and assessed twice a week during the first 2 months period and once a week during four consecutive months.

Six different tasks with a varying complexity were considered in the ALLADIN project (see Table 8). Motivations that led to the choice of these tasks are explained in [2]. The task descriptions are summarized in section B.1.

Task index	Task name	Short Task name
1	'*Grasping a glass*'	'*Glass*'
2	'*Turning a key*'	'*Key*'
3	'*Grasping a spoon*'	'*Spoon*'
4	'*Lifting a bag*'	'*Bag*'
5	'*Grasping a bottle*'	'*Grasping bottle*'
6	'*Bringing the bottle to the other side*'	'*Bringing bottle*'

Table 8: Names of the six ADL tasks considered in the ALLADIN project.

The data acquisition followed a precise protocol detailed in [2]. For each task, the patient watched a video showing the movement (recording 1). Secondly, he was asked to mentally imagine and reproducing it (recording 2). Finally, he actually tried to perform it 3 times (recordings 3, 4, 5). The detailed schedule of a measurement session is reminded in section B.2.

During all the sessions, the patient sat in a dedicated device developed at the start the ALLADIN project (see [3]): the *ALLADIN Diagnostic Device* (ADD). It is equipped with 8 force/torque sensors related to 8 different body parts. (see Table 9). The 8 ADD sensors allowed the recording of the isometric efforts (force and torque) produced along the X-Y-Z direction during the execution of an ADL task under the 3 conditions.

Sensor Index	Part of the body
1	Thumb
2	Index finger
3	Middle finger
4	Arm
5	Trunk
6	Seat
7	Foot
8	Big toe

Table 9: The 8 parts of the body monitored by the ADD

Besides the experimental group of stroke patient, a normal control group of healthy subjects underwent the same tests. The data acquisition for that group followed exactly the same protocol.

The 6 components of the effort samples (force and torque in the X-Y-Z directions) coming from the 8 sensors of the ADD were stored in a binary format, one separate file per recording for each task. Therefore, 30 files were registered per measurement session. A header containing identification information (e.g. the file format version, sampling rate,…) was added at the beginning of the file. The recording durations and the thereof number of samples depended on the type task (see Table 10).

Task Index	Baseline (Rec0)	Video (Rec1)	1st rep. (Rec2)	2nd rep. (Rec3)	3rd rep. (Rec4)
Glass	3	5.4	5.4	5.4	5.4
Key	3	3.7	3.7	3.7	3.7
Spoon	3	3.4	3.4	3.4	3.4
Bag	3	2.4	2.4	2.4	2.4
Grasping bottle	3	4	4	4	4
Bringing bottle	3	6	6	6	6

Table 10: Recording durations (in seconds) for the different tasks and attempts of an ADD measurement session.

The acquired data coming at the three clinical partners were weekly uploaded to and synchronized with the ALLADIN global database located at the Budapest University of Technology and Economics (BUTE). For this purpose a special tool enabling an easy transfer of the data to the WP4 partners was developed. Besides this an Access database contained general information about all the patients involved in the clinical trial. It consisted of:

Patient identification information: patient identifier, dominant hand, gender, admission date, stroke date, marital status…

And for each measurement session:

- Session identification information: session identifier, date, free comments,…
- Session description information: translated text of the diagnosis, corresponding speech filenames,…
- Session measurement information: measurement file names, free comments, flag indicating whether a problem occurred during the session,…
- Alternative recovery assessments based on scoring techniques.

2.2.1.2. Proof of concept database

Particular care was taken to eliminate non informative variability (due to external factors) to improve the focus on the definition of pertinent features. Targeted external variability sources are:

- Adverse events occurred during the data acquisition such as bad calibration, disabled sensors, incomplete session, the bouncing balls or treadmill in the neighbourhood,…
- Non precise detection of the onset of movement time.

A patient/normal control balanced force/torque measurement data subset was manually selected. Only the first measurement session was considered. Data clearly affected by adverse events were discarded, leading to a dataset containing 57 patients and 57 normal controls. The selection was performed by a visual inspection of the signals.

It is important to highlight that no information about the patients' profile group was used. Therefore, the proof of concept database is guaranteed to be representative of the global ALLADIN database.

Another critical point is the sample size. The required amount of data depends on the targeted analysis and on the statistical models (more precisely the number of parameters to tune) used to represent the classes: separating a high number of classes or picking rare characteristics increases the need for data, properly modelling a given class by a neural network requires much more training data than a Gaussian mixture model. In this proof of concept, there will be only two classes to separate and the statistical models will be either low-order GMMS or KDE, of which training requires few data. Under such conditions, the selected sample (57 patients and 57 normal controls) has an appropriate size.

The problem of the non precise onset detection was circumvented by a manually tagging of the signals for all the sessions (equal to the number of subjects since only the first session was considered for the proof of concept), all the tasks, for the recordings 3-4-5 (see section 1) and all the sensors. The total number of onset times is thus:

$$2\left(N_{Pat}+N_{NC}\right)N_T N_A N_S = 2.114.6.3.8 = 32832$$

Where N_{Pat} is the number of patients, N_{NC} is the number of normal controls, N_T is the number of tasks, N_A is the number of attempts and N_S is the number of sensors.

This tagging was independently carried out by physiotherapists from the three clinical sites. A graphical user interface was developed in Matlab (see Figure 3) to assist them with the tagging.

In deliverable 2.2 [3], the authors have highlighted an aliasing effect that polluted the effort recordings. After more thorough investigation, they identified the sensor power supply as a potential source of noise. Although the spectral analysis showed that the importance of the disturbance was significant, it was not possible to isolate the proof of concept dataset from this additional external variability source since it affects all the collected data.

2.2.1.3. DTN Database

The selected patients/Normal controls are the same as in the proof of concept (57 patients and 57 normal controls). No additional selection was performed, though, unlike the proof of concept database, now all the measurement sessions were considered in order to focus on the recovery course.

It was planned to assess all patients twice a week during the first two months post stroke and once a week for the consecutive four months. Thus, the number of available sessions per patient should be 32. Nevertheless, the assessment periods were often shortened for various

reasons so that the actual number of available sessions per patient is much lower. Indeed, Figure 16 shows that about 30% of the patients have less than 10 valid sessions and only 58% of the patients have more than 23 valid sessions. Solely 7 patients have 32 valid measurement sessions.

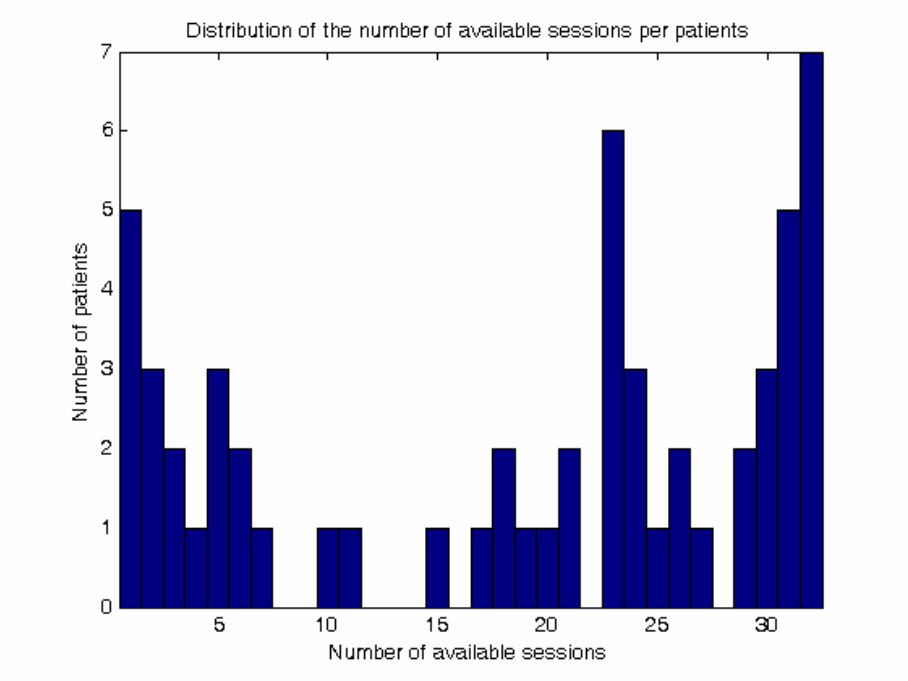


Figure 15: Distribution of the number of available sessions per patient.

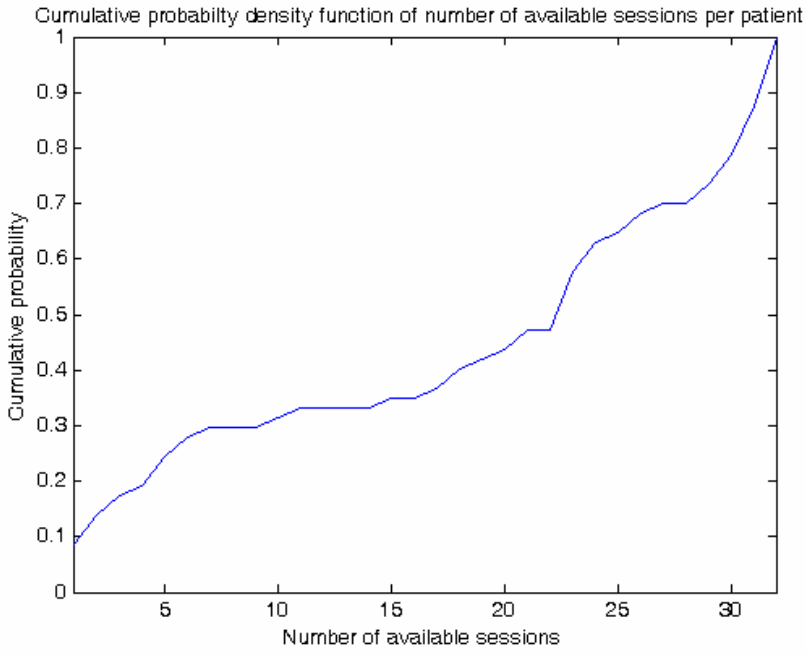


Figure 16: Cumulative probability density function of the number of available sessions per patient.

Up to now, no automatic onset detection method achieved a high performance. Thus, this database was also manually tagged. Given that the most relevant features had been identified

and that two features are extracted from the same signal, the number of signals to be tagged was equal to:

$$N_{Session} N_{Signals} = 1069.5 = 5345$$

2.2.2. Feature extraction

2.2.2.1. Data pre-processing

2.2.2.1.1. Windowing

One of ALLADIN hypotheses is that isometric force/torque trajectory patterns at the initiation of Activity of Daily Living tasks contain characteristics of stroke recovery (see section 1). Consequently, the analysis was performed on a limited region of interest beginning at the estimated onset of movement time. In agreement with the physiotherapists the analysis window width was set to 750 ms. Figure 17 shows the corresponding region of interest for some randomly selected patients and normal controls (norm of the force, thumb sensor, 'glass' task).

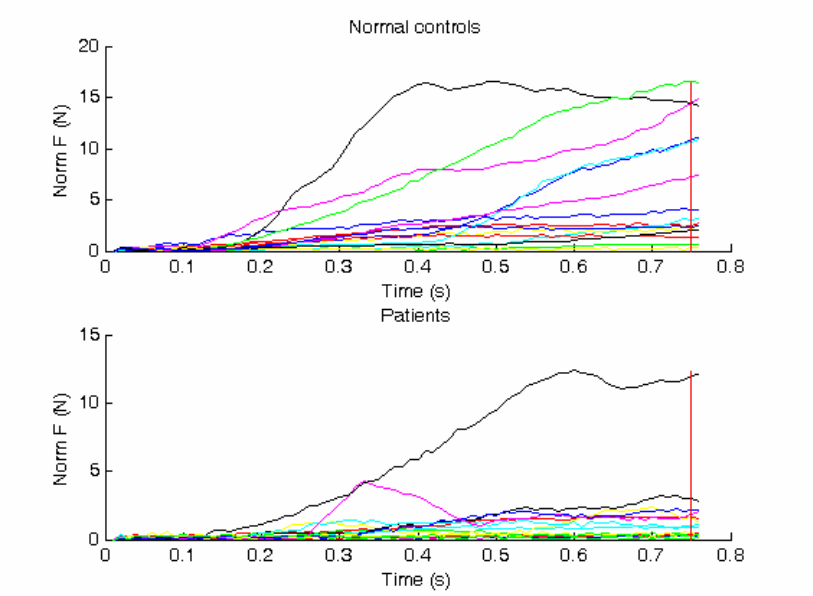


Figure 17: Region of interest (window length: 750 ms) for 15 patients (below) and 15 normal controls (above)-Signal: norm of the force, thumb sensor, drinking a glass task

2.2.2.1.2. Onset of movement time

No automatic method achieved good results. So, we used, for all the experiments, the manually tagged onset of movement time.

2.2.2.1.3. Filtering frequency

During muscle contractions a background vibration is present of which spectral components can reach 40 Hz in frequency. Spectral components resulting from voluntary movements

have generally lower frequencies. Therefore, signals are low-pass filtered at 40 Hz during the pre-processing step (10-order Butterworth filter). A zero-phase filtering is performed using the Matlab `filtfilt` function. The part of signal within the analysis window is also normalized: subtraction of the first sample value and division by the last sample value. The frequency response of the filter is represented in Figure 18.

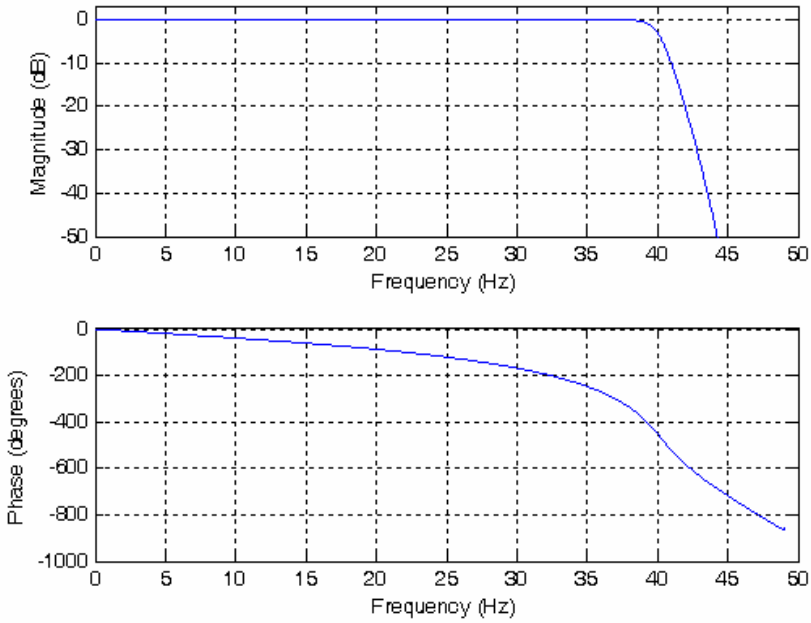


Figure 18: Frequency response of the Butterworth 10-order filter used in the data pre-processing.

2.2.2.1.4. Coordinate system changes

The coordinate system mapping aimed at making the functional interpretation of the effort directions consistent with regards to the measured side of a patient (left handed/right handed). For instance, the 'pushing forward' action results in opposite-signed Y-component for the thumb sensor in the body coordinate system defined in [3]. The mapping functions (actually sign changes) were established on the basis of the functional interpretation of the ADD coordinate system provided in [3].

The coordinate systems related to the trunk and the seat sensors, were first rotated to compensate the angular position of the sensors on the ADD (see [3]) before any further processing.

2.2.2.1.5. Data normalization

For some sensors, efforts are characterized by a dominant component. For instance, the force at the seat sensor is characterized by a very high Z-component (several hundreds Newton) corresponding to the patient weight. Values of some candidate features (f.i.: features based upon Cumulative Sum of the Effort Series or the Angle To the Mean Direction) are likely to be driven by such a dominant component. Consequently, informative variability of the features may be hidden.

In order to circumvent this problem, the value of the signal under consideration at the onset of movement time was subtracted from the effort time series within the analysis window:

$$F_{Norm}(k) = F(k_{Onset} + i) - F(k_{Onset}) \quad i = 0, ..., N$$

Where $F_{Norm}(k)$ stands for the normalized sample, $F(.)$ is the original sample, k_{Onset} is the onset time in samples, N is the analysis window width in samples (here $N = 75$).

A final normalization step prevents the three effort components from simultaneously being zero and thereof avoids to have undefined angles and divisions by zero.

2.2.2.2. Proof of concept candidate features

All the candidate features were computed for all the patients in the proof of concept database for all tasks, for all sensors and the recordings 2-3-4 (the actual attempts to perform the described ADL task) and this both for force and torque signals. The data were pre-processed as described above.

The candidate features extracted from the three different attempts (same task, same sensor) were considered as three different features since the patient may have learned.

Features were computed not only for the 3-dimension signals but also from the X-Y-Z components considered independently and from the projection of the effort vector in the three reference plans (XY-plan, YZ-plan and XZ-plan). The underlying idea is that, for a given task and a given sensor, some components can convey most of the movement characteristics, making the other components appearing more as a consequence of noise sources. This made the total number of candidate features increasing to 59472.

The feature extraction was performed with a modified version of the tool already developed in task 4.1 (see [4]).

Features were stored in a $N_{Pat} \times N_{Feat}$ matrix, where N_{Pat} is the number of patients/normal controls (114 for the proof of concept database), N_{Feat} is the number of candidate features. Information about features was generated together with the matrix building. A Matlab script allowed retrieving the feature description given its index. The matrix was finally submitted to the KUL data mining expert's team for the feature selection step.

2.2.3. Feature selection

The feature set submitted to the KUL data mining expert's team contained a total of 59472 features as a result of the computation of all the candidate features defined in section 2.1.2 for all the patients present in the proof of concept database, for all tasks, all sensors and recordings 2-3-4 (actual attempts to perform the ADL task), both for the force and the torque signals.

The feature selection process identified the six following relevant feature subsets:

Subset 1:

Feature 1:

Description:	Standard deviation value of the integral of the torque Y-component within the time region of interest.
Sensor: Middle finger	
Task: Glass	
Record ID: 2	

Subset 2:

Subset 1 + *Feature 2*:

Description:	Maximum value of the angular deviation between the torque vector projected onto the XZ-plan and the mean torque vector within the time region of interest.
Sensor: Thumb	
Task: Lifting a bottle	
Record ID: 4	

Subset 3:

Subset 2 + *Feature 3*:

Description:	Mean value of the angular deviation between the torque vector projected onto the XZ-plan and the mean torque vector within the time region of interest.
Sensor: Seat	
Task: Glass	
Record ID: 3	

Subset 4:

Subset 3 + *Feature 4*:

Description: Standard deviation value of the integral of the torque X-component within the time region of interest.

Sensor: Thumb

Task: Lifting a bottle

Record ID: 4

Subset 5:

Subset 4 + *Feature 5*:

Description: Normalized sum of the residuals resulting from fitting the 'Exponential2' model to the norm of the force sample series within the time region of interest.

Sensor: Thumb

Task: Glass

Record ID: 4

Subset 6:

Subset 5 + *Feature 6*:

Description: Mean value of the angular deviation between the force sample vector and the previous force vector in the YZ-Plan within the time region of interest.

Sensor: Index finger

Task: Bag

Record ID: 4

2.2.4. Classification results

2.2.4.1. Introduction

A small set of relevant features was identified among the list of feature candidates defined in section 2.1.2.3. The discrimination based on these features of patients and normal controls in the recovery space was necessary to appropriately assess recovery after stroke and to subsequently identify markers and milestone in the recovery trajectory (see section 2.1.1.1).

A first set of classification tests was performed on the proof of concept data in order to assess the ability of the selected features to distinguish the patients at the beginning of their rehabilitation and the normal controls.

As the rehabilitation progresses, the feature vector of a patient should move toward the region that characterizes the normal control in the recovery space (see section 2.1.1.1). As a consequence, the discrimination of patient and normal controls is expected to become more and more difficult with a classification rate that progressively decreases.

It is important to remember that the proof of concept database on which the feature selection was performed only contained data from the first measurement sessions. Consequently, there is a risk that the analysis overfits the state of the patients at the very beginning of their rehabilitation. Therefore, a second set of classification tests was performed at increasing time intervals distant from stroke onset in order to assess the capability of the selected features to characterize the progressing state of the patients during the process of rehabilitation. If the overfitting scenario occurs, a brutal decrease in the classification rate to 0.5 (random decision) will be observed instead of the expected smooth evolution.

Classification results presented in this section have been obtained by assigning a sample to the class (Patient or Normal Control) maximizing the a posteriori probability:

$$C(Y) = \arg\left(\max_i \left(P(C_i \mid Y)\right)\right) = \arg\left(\max_i \left(\frac{P(Y \mid C_i)P(C_i)}{P(Y)}\right)\right)$$

Using the Bayes rule, this expression can be further developed:

$$C(Y) = \arg\left(\max_i \left(\frac{P(Y \mid C_i)P(C_i)}{\sum_{i=1}^{|C|} P(Y \mid C_i)P(C_i)}\right)\right)$$

Where Y is the feature vector, $P(C_i \mid Y)$ is the a posteriori probability of belonging to the class C_i (here patient or normal control), $P(Y \mid C_i)$ is the conditional probability of belonging to the class C_i given the feature vector Y (also called the likelihood), $P(C_i)$ is the a priori probability of belonging to the class C_i.

The last critical issue discussed in this section will concern the statistical distribution inference ($P(Y \mid C_i)$). Indeed, the statistical approach adopted for the definition of a recovery metric relies on a statistical modelling (see section 2.1.5) of the patients and normal controls. Two approaches will be compared: a *Gaussian Mixture Model* (GMM) and a *Kernel Density Estimation* (KDE). Results of a feature pre-processing procedure will also be presented.

The order of the GMM model was set to one. Indeed, higher orders resulted in ill-conditioned covariance matrices. In this particular case the training becomes trivial (see section 2.1.4.2): the estimation of the distribution parameters is straightforward and leads to an optimal solution. Both diagonal (denoted as *diag*) and full (denoted as *full*) covariance matrices were used.

The KDE was handled using a Matlab toolbox downloaded from the MathWorks website. The *Likelihood Cross-Validation* (LCV) technique was used for the kernel bandwidth selection.

The classification performances are assessed by the *Classification Rate* (CR) using the leave-one-out method. The CR is defined as follows:

$$CR = \frac{N_{Correct}}{N}$$

Where $N_{Correct}$ is the number of correctly classified patients/normals, N is the database size.

2.2.4.2. Experiment 1: Classification rate at the beginning of the rehabilitation

The classification tests presented here were performed on the proof of concept database (57 patients/57 normal controls). Only the 6 features identified by the feature selection were computed using the data pre-processing described in section 2.2.2.1. The manually tagged onset of movement times was used. The feature subsets are described in section 2.2.3.

The classification rate was assessed using the leave-one-out method. The results are presented in Table 11 for a 1-Order GMM (Diagonal and full covariance matrix) and for a KDE.

Globally, the classification rate progressively rises as the feature vector size increases for the two inference models. This means that the patients and normal controls can be better discriminated in a high-dimension recovery space.

For the subset 1 to 5, the KDE approach outperforms the GMM. However, for the highest-size feature subset, the best performance is achieved by the GMM with a diagonal covariance matrix.

	1-Order GMM (diag)	1-Order GMM (full)	KDE
Subset 1	0.7368	0.7368	0,8333
Subset 2	0.7807	0.8158	0,8333
Subset 3	0.8421	0.8158	0,8684
Subset 4	0.7982	0.8246	0,8772
Subset 5	0.8421	0.8070	0,8509
Subset 6	0.9211	0.8860	0,8947

Table 11: Classification rate assessed on the proof of concept database (without feature pre-processing). 1-Order GMM (diag): 1-order GMM with a diagonal covariance matrix. 1-Order GMM (full): 1-Order GMM with a full covariance matrix, KDE: Kernel Density Estimation using the Likelihood Cross-Validation for the selection of the kernel bandwidth.

In order to improve the classification performances, we tested a two-pass feature pre-processing before training the model. The first step consists in removing the outliers (when an outlier was identified, its value was set to the mean of the corresponding component of the feature vector). The second step was a *Principal Component Analysis* (PCA), expected to better separate patients and normal controls in the recovery space. Classification results are presented in Table 12.

	1-Order GMM (diag)	1-Order GMM (full)	KDE
Subset 1	0,807	0,807	0,8509
Subset 2	0,8421	0,8596	0,8596
Subset 3	0,8333	0,9123	0,8947
Subset 4	0,8947	0,9035	0,8772
Subset 5	0,8684	0,9123	0,9035
Subset 6	0,8947	0,9386	0,9211

Table 12: Classification rate assessed on the proof of concept database (with feature pre-processing). 1-Order GMM (diag): 1-order GMM with a diagonal covariance matrix. 1-Order GMM (full): 1-Order GMM with a full covariance matrix, KDE: Kernel Density Estimation using the Likelihood Cross-Validation for the selection of the kernel bandwidth.

Table 12 shows that the feature pre-processing leads to a significant improvement of the classification performance for the GMM with a full covariance matrix and the KDE with a classification rate rise ranging from 4.4% to 10.5% for the former approach and from 0% to 5,3% for the latter. The performances for the two approaches are very close. However, when the covariance matrix of the GMM is constrained to be diagonal, the trend is less clear. For the subset 3 and 6, the classification is even smaller than in the case were no feature pre-processing is performed. This can be explained by the fact that the PCA defines a linear transformation of the feature:

$$x_i = \sum_{j=1}^{|X|} a_{ij} x_j$$

Where x_i is the i^{th} component of the feature vector, X the feature vector, $|X|$ the dimensionality of the feature vector and a_{ij} the coefficient of the linear transformation. Therefore, if $a_{ij} \neq 0$ if $i \neq j$ assuming that out-of-diagonal elements of the covariance matrix are null causes a mismatch between the model and the experimental data.

Conclusions

A first classification test performed on the proof of concept database showed that the KDE approach outperforms the GMM approach (both with a full and a diagonal covariance matrix) for all the feature subsets excepted for the biggest-size feature subset for which the best performance is achieved with a diagonal covariance matrix GMM model.

A two-pass feature pre-processing applied before training significantly improved the classification performance of the KDE approach and of the GMM when a full covariance matrix is assumed. The highest classification performance is then (93,86%). It is important to note that it is very close to the best classification rate reached for a diagonal covariance matrix GMM without feature pre-processing (92,11%) so that the usefulness of the feature pre-processing is not so evident when the subset 6 is used.

We can conclude that the selected features can successfully discriminate the patients at the beginning of their rehabilitation from the normal controls since the best classification rate rises to 93,86%. This classification rate is reached using the subset 6. As a consequence, these 6 features will always be used in the following developments. However, no conclusion can be drawn about the inference model.

2.2.4.3. Experiment 2: Classification rate evolution as the rehabilitation progresses

The classification tests presented here were performed on the DTN database (57 patients/57 normal controls). Only the 6 features identified by the feature selection were computed using the data pre-processing described in section 2.2.2.1. The manually tagged onset of movement times were used. The feature subsets were described in section 2.2.3.

The aim of this second experiment is to assess the capability of the selected features to characterize the changing state of the patients during the rehabilitation process. Therefore, the patient measurement sessions of the DTN database were grouped according to the time elapsed since the date of the stroke:

$$PDS_k = \left\{ s_{ij} \mid \left(MD_{ij} - SD_i \right) \in \left[B_k - \frac{\Delta}{2}, B_k + \frac{\Delta}{2} \right] \right\}$$

Where PDS_k is the patient database subset k, s_{ij} represents the measurement data of the j^{th} session of the i^{th} patient, MD_{ij} is the date of the measurement session, SD_i is the date of the stroke of the i^{th} patient, B_k is the centre of the time bin related to PDS_k, Δ is the width of the time bins. Classification will then be performed on the data subsets DS_k resulting from the merge of the PDS_k with the NDS data subset containing the data related to all the normal control.

NDS is constant while PDS_k contains a varying number of measurement sessions. In order to prevent the most populated class (biggest $P(C_i)$) from being favoured during the classification decision (see the classification criterion in section 2.2.4.1), a balanced dataset will be simulated by arbitrarily setting the a priori probabilities to 0.5. In this way, the classification will rely only on the position of the feature vectors in the recovery space.

Two inference models will be tested: a 1-Order GMM and a KDE. The two-pass feature pre-processing described in the previous section will also be discussed. In the previous section, we showed that the use of this technique with a diagonal covariance matrix GMM degrades the classification performances. Consequently, these experimental conditions will be discarded. The classification rate is assessed using the leave-one-out method.

Since the number of patient measurement sessions varies, it may occur that not enough training data is available to train the patient GMM. The required amount of training data depends on the number of parameters to be tuned (see Figure 19). The GMM order is equal to one and we use the subset 6 (6 features). Figure 19 shows that properly tuning the patient GMM parameters requires a minimum of 14 sessions. Consequently, classification will not be performed if the number of sessions is smaller.

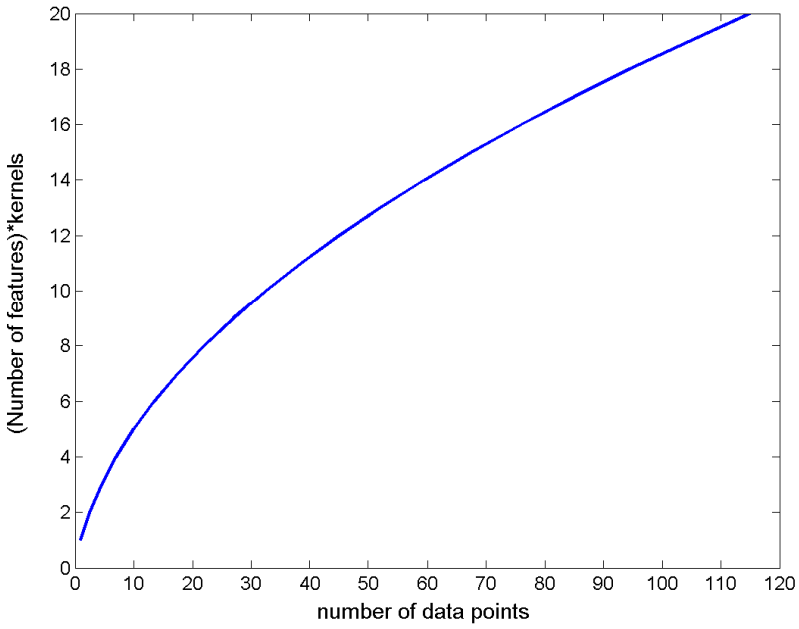


Figure 19: Number of model parameters correctly tuned as a function of the amount of training data.

The results are presented in the Figure 20. The evolution can globally be decomposed in two phases. From week 1 to about the week 7, the classification rates steadily decreases but doesn't collapse. Although the classification performance decrease is high (more than 15% on average), the relatively smooth evolution tends to invalidate a strong overfitting scenario that would have undermined any further investigation and is in favour of the hypothesis that feature vectors are characteristic of patients and progressively overlap with the normal control population in the recovery space. In that region, diagonal covariance matrix GMM and the full covariance matrix GMM without feature pre-processing globally perform worse but there is a similar trend for all classifiers.

From the week 7 on, the classification rate does no longer follow a clear trend and varies around 62%. The variability around this medium performance is quite high. No approach towers above all others.

Weeks 22 to 27 are characterized by an evolution of the classification rate more erratic. The classification rate rises sometimes to 70%, similar to performance achieved at the beginning of the rehabilitation. A smaller number of available measurement sessions leading to an unstable estimate of the classification rate can explain this. In addition to this, some patients who recovered fast left the ALLADIN assessment program before its completion. So, the patient population from weeks 22 to 27 was likely to contain relatively more patients with serious impairments.

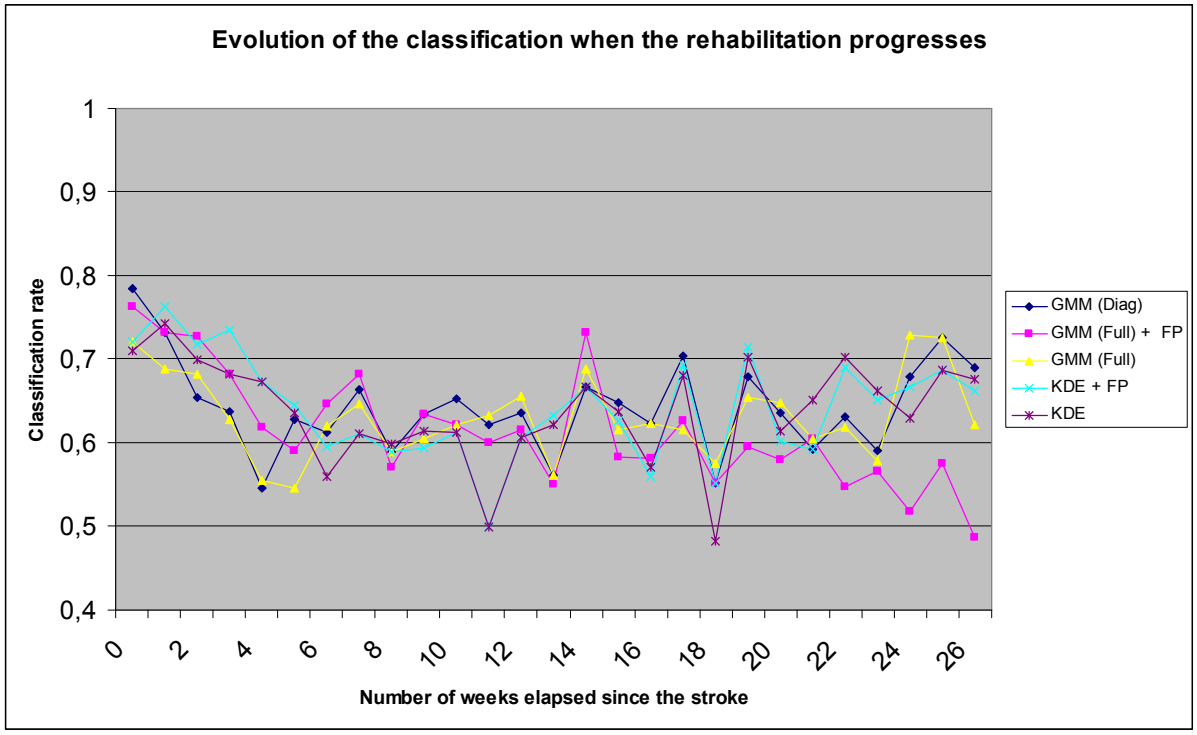


Figure 20: Evolution of the classification rate when the rehabilitation progresses. GMM (Diag): diagonal covariance matrix GMM, GMM (Full): full covariance matrix GMM, GMM (Full) + FP: full covariance matrix GMM + feature pre-processing, KDE: Kernel Density Estimation, KDE + FP: Kernel Density Estimation + feature pre-processing

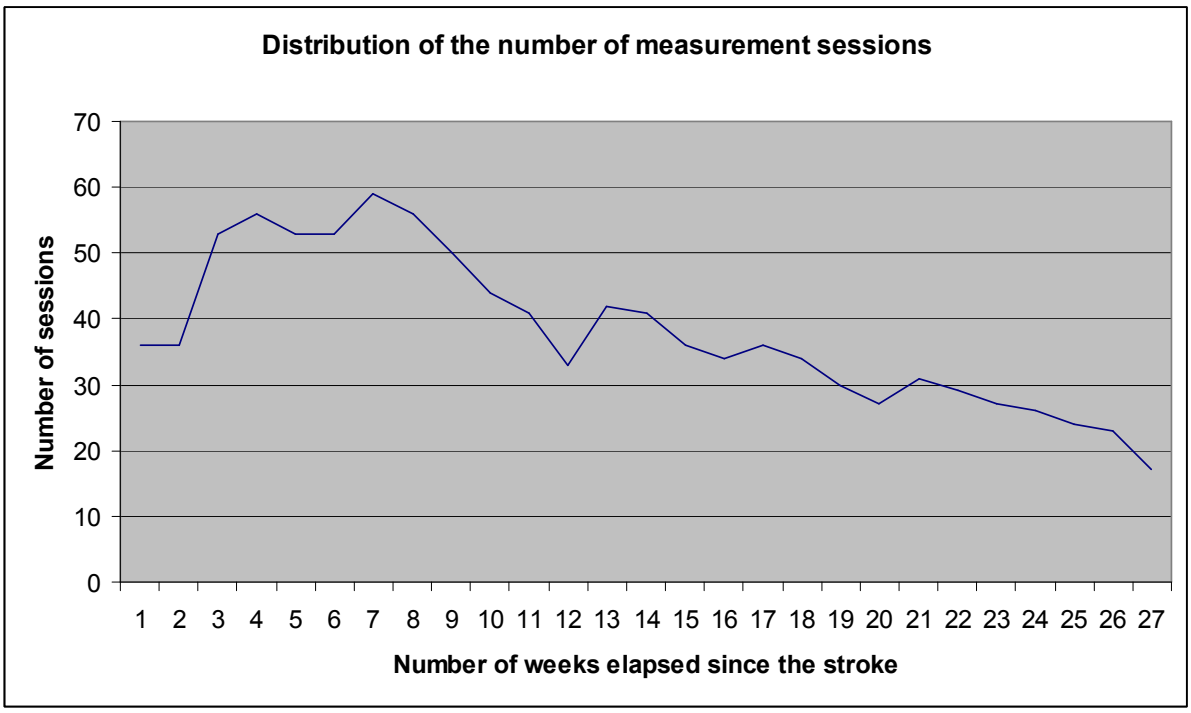


Figure 21: Distribution of the number of sessions.

Conclusions

In this section, the capability of the selected features to characterize the recovery state of a patient was assessed. The classification performances were significantly lower than the results obtained in the previous section and their evolution during the process of rehabilitation is noisy. However, the presence of a region where the classification rate decreases relatively smoothly followed by a region characterized by a stagnation of the performances tends to invalidate a strong overfitting scenario that would have undermined any further investigation and is in favour of the hypothesis of feature vectors characteristic of the patients and progressively overlapping with the normal control population in the recovery space.

Figure 20 also shows that the full covariance matrix GMM combined with the feature pre-processing has the smoothest evolution for the first six weeks. Nevertheless, no conclusion about the appropriateness of the different statistical inference approaches can be drawn because there is not a straightforward relation between the recovery curves and the classification rates.

2.2.5. Distance to normality curves

The results presented in this section are obtained on the DTN database (see section 2.2.1.3) with manually tagged onset of movement times. The feature subset 6 is used. Two metrics will be discussed: the *A Posteriori Probability* of normality and the Normalized *Likelihood* (see section 2.1.5). It is important to remind that these metrics are actually recovery scores: the higher, the better recovery.

A visual inspection of the recovery curves resulting from the possible combinations of the two different metrics and the different inference models described above, unveiled some

recurrent characteristics. The following graphs will illustrate these characteristics by the mean of the analysis of a particular patient (NIMR-093).

Figure 22 illustrates the evolution of the *Normalized Likelihood* for different 1-Order GMM. The thick lines are smoothing splines (same color as the raw data). Their monotonously rising shapes show that the features and the defined metric are capable of picking the patient recovery evolution. However, for the three models, the DTN presents an important variance around the smoothed trends. The full covariance matrix GMM (red) and the diagonal covariance matrix (blue) GMM lead to very similar DTN. Their average evolution (estimated by the splines) is almost identical. On the contrary, the DTN values obtained with a full covariance matrix GMM (green) and using a feature pre-processing are quite different. The average evolution diverges from the two others models and the variance is bigger. An examination of the whole database shows that these observations can be generalized.

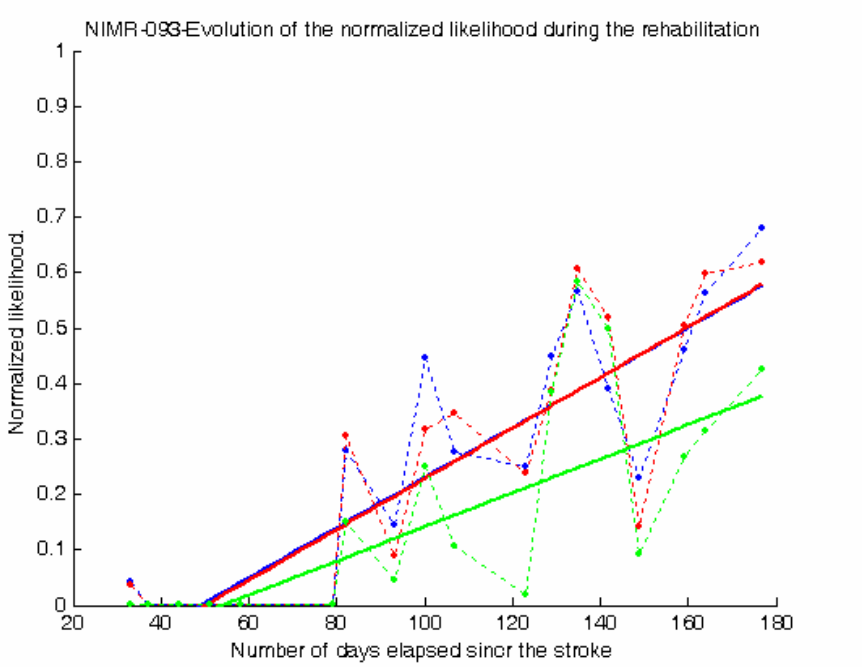


Figure 22: Evolution of the normalized likelihood during the rehabilitation for different variants of the GMM. Dashed blue: Diagonal covariance matrix GMM, dashed red: Full covariance matrix GMM, dashed green: Full covariance matrix GMM + Feature preprocessing. The thick lines are smoothing splines (same color as the raw data).

Figure 23 compares the evolution of the Normalized Likelihood metric (blue) and the evolution of the A Posteriori Probability metric. The inference model is the same for the two metrics (Full covariance matrix GMM without a feature pre-processing). The DTN values based on the a Posteriori Probability sharply rise to rapidly reach 1. The increase is faster than for the Normalized Likelihood based metric but the variance around the average trajectory is much higher. A complete examination of whole the database shows that this first intuition can be generalized to all the patients.

Figure 24 shows the evolution of the a Posteriori probability of normality for different inference models: Full covariance matrix GMM, KDE, KDE + feature pre-processing. The evolution of the DTN is very similar for the three models and presents the same characteristics: fast increase and high variance around the average trajectory. These two

characteristics are thus related to the a Posteriori probability metric. In addition to this, we can conclude that the inference model has not a lot of influence on the DTN value.

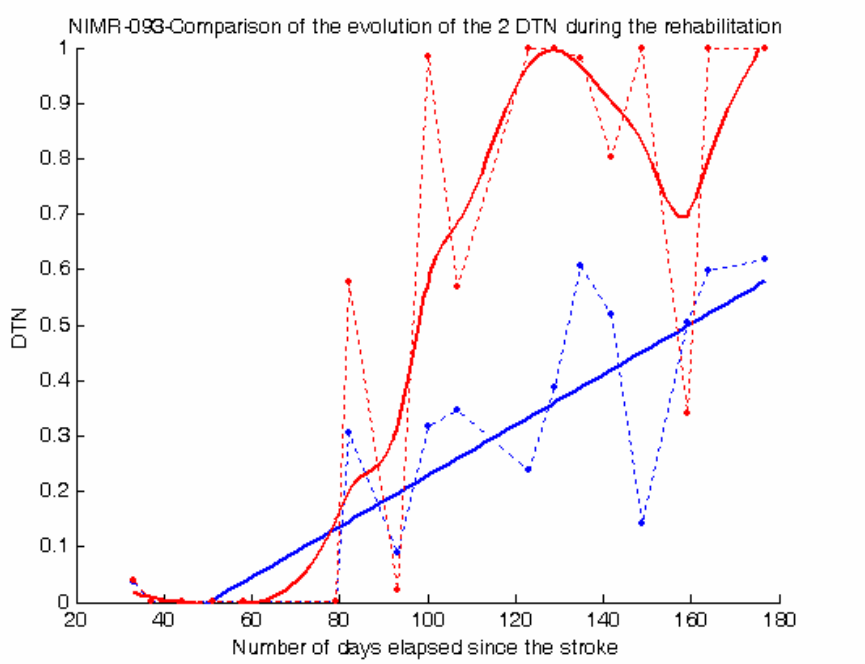


Figure 23: Comparison of the evolution of the DTN based on the two different metrics during the rehabilitation (Full covariance matrix GMM). Dashed red: A posteriori probability of normality, Dashed blue: normalized likelihood. The thick lines are smoothing splines (same color as the raw data).

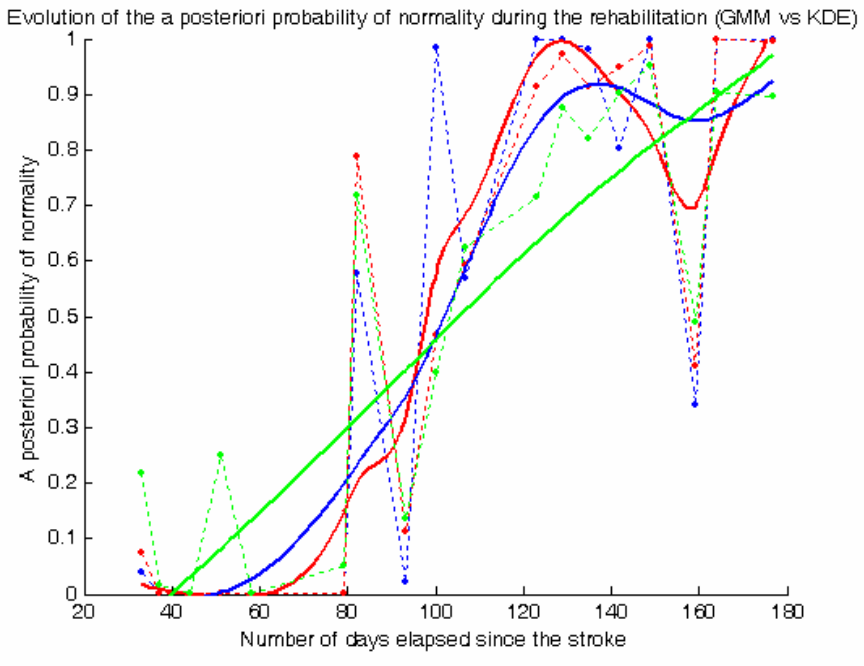


Figure 24: Evolution of the a posteriori probability of normality during the rehabilitation for different inference model. Dashed blue: Full covariance matrix GMM, dashed red: KDE, dashed green: KDE + Feature preprocessing. The thick lines are smoothing splines (same color as the raw data).

In order to appreciate the recovery progress on the whole database, the DTN distribution was built within non overlapping time bins (one week) centred upon increasing time intervals distant from stroke onset. To do so, the patient measurement sessions of the DTN database were grouped according to the time elapsed since the date of the stroke:

$$PDS_k = \left\{ s_{ij} \mid \left(MD_{ij} - SD_i \right) \in \left[B_k - \frac{\Delta}{2}, B_k + \frac{\Delta}{2} \right] \right\}$$

Where PDS_k is the patient database subset k, s_{ij} represents the measurement data of the j^{th} session of the i^{th} patient, MD_{ij} is the date of the measurement session, SD_i is the date of the stroke of the i^{th} patient, B_k is the centre of the time bin related to PDS_k, Δ is the width of the time bins.
The DTN of all the sessions each time bin contained were computed. In this way, we have the distribution of the DTN related to the patient population of the whole database at different times B_k (increasingly distant from the date of the stroke). The DTN were computed with the normalized likelihood metric and a full covariance matrix GMM as an inference model (no feature pre-processing).

The following graphs compare the recovery curve of some patients with the global performance for the whole database (Δ=1 week). The blue curve represents the time trajectory of the DTN distribution mean. It is composed of a region characterized by a rising slope between the week 1 and the week 8 followed by a plateau where the DTN mean does not vary anymore. The upper dashed red curve is the time trajectory of 75-percentile, the lower red curve is the time trajectory of the 25-percentile. Consequently, half of the patients evolve between the two red curves. The 25-percentile and the 75-percentile are quite distant, which means that the recovery curves widely vary from one patient to another one. The thick lines are smoothing splines (same colour as the raw data).

The Figure 25 shows the DTN evolution of a patient presenting a recovery faster than the patient population.

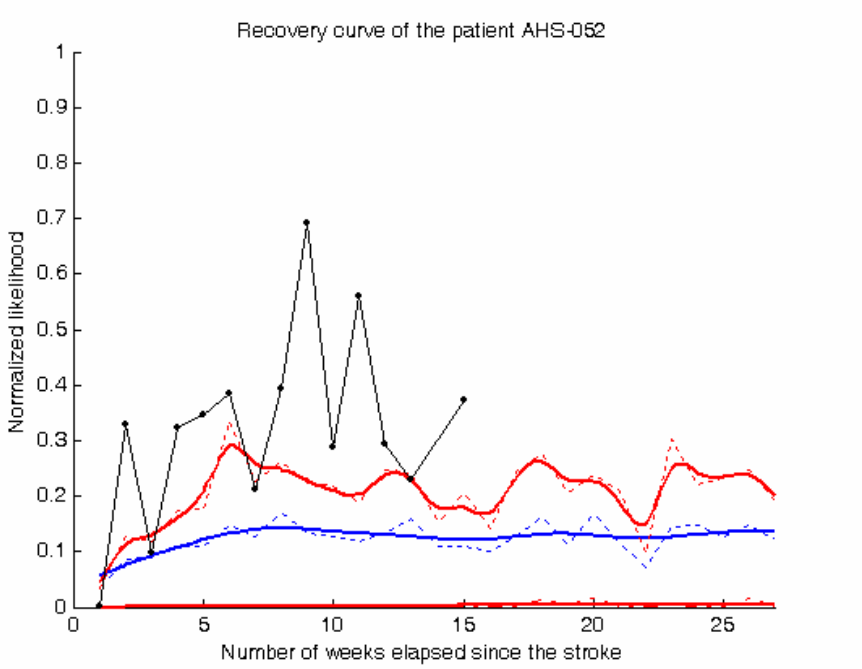


Figure 25: DTN evolution for a patient that presents a faster recovery than the average recovery of the patient population. Dashed blue curve: time trajectory of the DTN distribution mean, upper dashed red curve: 75-percentile, lower red curve: 25-percentile, black curve: recovery curve of the patient AHS-052. The thick lines are smoothing splines (same color as the raw data).

The Figure 26 illustrates the case of a patient that doesn't evolve.

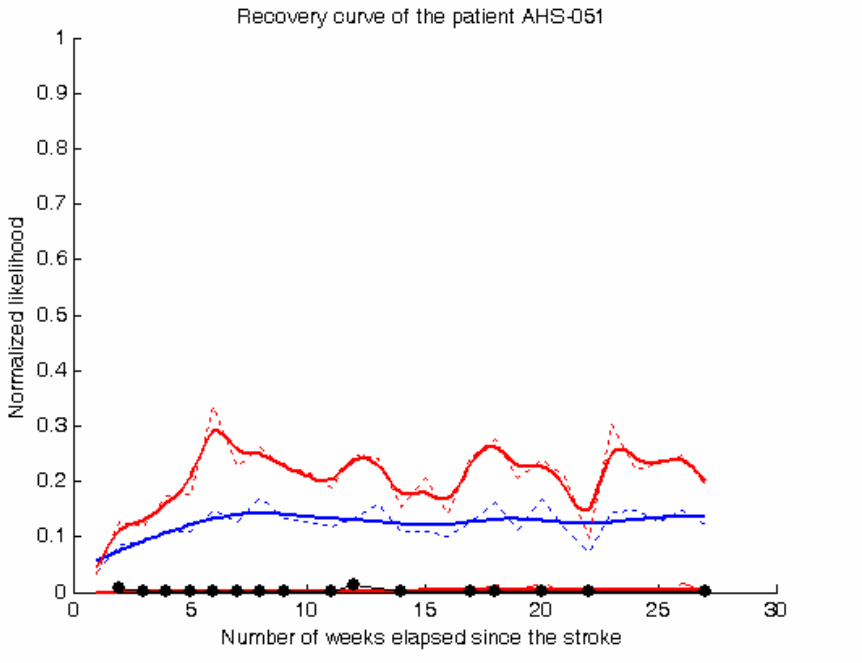


Figure 26: DTN evolution for a patient that doesn't recover. Dashed blue curve: time trajectory of the DTN distribution mean, upper dashed red curve: 75-percentile, lower red curve: 25-percentile, black curve: recovery curve of the patient AHS-051. The thick lines are smoothing splines (same color as the raw data).

The patient recovery curve in that doesn't evolve.

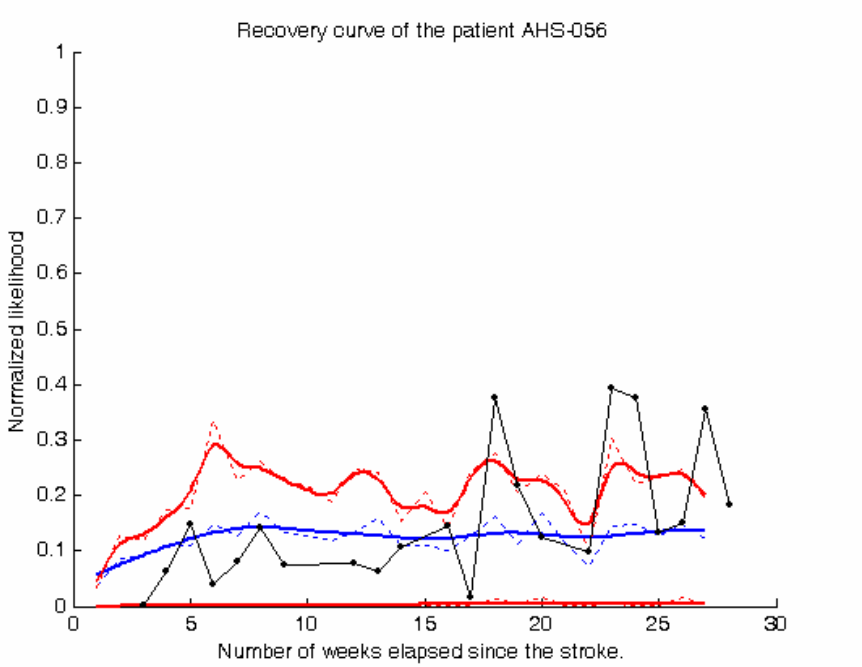


Figure 27: DTN evolution for a patient that recovers roughly like the overall patient population. Dashed blue curve: time trajectory of the DTN distribution mean, upper dashed red curve: 75-percentile, lower red curve: 25-percentile, black curve: recovery curve of the patient AHS-056. The thick lines are smoothing splines (same color as the raw data).

Conclusions

A visual inspection of the recovery curves resulting from the possible combinations between the two different metrics defined in the section 2.1.5 and the different inference models discussed in the sections 2.2.4.2 and 2.2.4.3 for the whole database unveiled some general characteristics. In this section, the analysis of the recovery curves for a particular patient illustrated these characteristics.

Firstly, the evolution of the DTN as the rehabilitation progresses, presents a high variance around its average trajectory. We showed the a posteriori probability of normality metric resulted in a more varying smoothed DTN trajectory but also in a higher DTN variance when compared with the normalized likelihood metric.

Secondly, the recovery curves built on the same metric but with different inference model were similar. Therefore, the DTN are not widely influenced by the statistical modelling though significant differences in classification performance were highlighted in the sections 2.2.4.2 and 2.2.4.3.

In order to appreciate the global evolution of the recovery course of the overall patient population of the database, we also built the DTN distribution at increasing time intervals from stroke onset. The time trajectory of the mean of these distributions presents a rising slope from the week 1 to the week 8, followed by a plateau, which is not contradicting clinical experience. The time trajectory of the 25-percentile and the 75-percentile are far away. This means that the recovery curves widely vary from one patient to another one.

2.3. Conclusions

Six relevant features were identified from a set of 59472 candidate features extracted from the proof of concept database using manually tagged onset of movement times. Several classification tests were performed with different statistical models and showed that these features could discriminate the patients at the beginning of their rehabilitation and the normal controls in the recovery space. These conclusions successfully concluded the proof of concept.

Focus was then set on the definition of the DTN. The two metrics were discussed together with different statistical inference models. The recovery curves present a mean trajectory that seems to correspond to the patient evolution. Indeed, the mean trajectory for the global patient population is composed by a rising slope from the week 1 to the week 8 followed by a plateau, which is in agreement with the physiotherapist experience. However, the evolution of the DTN is affected by a high variance.

Now, the next step is the validation of the DTN by the clinical experts.

3. Automatic onset detection

This chapter describes the different approaches to the onset of movement detection problem. A brief description of the tested techniques will be provided in section 3.2. Afterwards a comparative study of the performances will be presented.

3.1. Introduction

The features mentioned in the previous paragraph were extracted within a time window that started at the movement onset time for the sensor under consideration. The width of that window was set to 750 ms while the total duration of the recordings ranges from 2.4 s to 6 s according to the task and to the recording index (video, mental simulation or attempts 1, 2, 3). A precise estimation of the onset of movement time is thus of a paramount importance

In order to properly identify the onset of the motion pattern responsible for the executed ADL task with other words, the starting point of the time window of interest, a dedicated approach was implemented.

The proposed approach made a comparative analysis of the performance of different proposed techniques for onset time estimation and this with respect to the manual tagging of the ALLADIN clinical experts, operating by direct visual inspection. To this aim the experts were provided with a dedicated onset time estimation tool which has been derived by the AVM (*ALLADIN Visualization Module*), and that is enclosed in the D4.1 software package (see Figure 3).

Starting from in-depth review of state-of-the-art techniques and after internal debate between engineers and clinical experts, a shortlist of candidate methodologies for automatic onset of movement time estimation were identified, namely:

- To detect the time when the force/torque signal reaches 2% of its peak value;
- To identify the onset time by using the spectroflatness measure (SFM) of the force/torque signal.
- To identify the onset time by using PDF/Ks-density measure of the force/torque signal.
- To identify the onset time by using the 2-nd order derivitative of the force/torque signal (previously low-pass filtered at 3Hz or at 5Hz).
- To identify the onset time by analysis the statistical properties of a small number of relevant features. This approach is based on the data mining results.

Each approach is briefly described in section 3.2. The first four methods were investigated by SSSA. Therefore, only the final results of the comparative study will be provided in this document. The interested reader can find more details in [4].

3.2. Description of the candidate techniques

3.2.1. The 2% rule

A neuro-rehabilitation research paper (see [20]) inspired the proposal of the here presented technique. The inputs to the present threshold-based algorithm are the three components of the force (or torque) signals. It computes the 2% of the peak value on the signal and finds the time corresponding to that value for each component and then the minimum value is taken as onset time.

Results from different versions of the present technique, consisting of different thresholds (4%, 6%, 8%, 10%) and the 2% computed on the integral of the signal are reported.

3.2.2. The second derivative method

The input to this algorithm is the absolute value of the force (or torque) signals. In figure 1, the absolute value of the force is represented by the blue line. Also the filtered lines are shown. The figure shows 4 red markers, which show different onset locations. The steps of the algorithm are the following ones:

1) Find the threshold point on the 1st derivative at the 15% of its maximum (the red marker on the red line at about 50th point) this point is the basis of the next three points. The threshold scan starts from the 15th point to exclude the usual starting peak, and the 3Hz filter is applied before the derivation.

2) Search the nearest maximum peak of the second derivative of the 3 Hz filtered line (marker on the cyan line). This step represents a second version of the technique: henceforth the results from this version will be labelled as *"2nd derivative (filtered 3Hz)"*.

3) Search the nearest maximum peak of the second derivative of the 5 Hz filtered line (marker on the dark yellow). This step represents a second version of the technique: henceforth the results from this version will be labeled as *"2nd derivative (filtered 5Hz)"*.

4) search backward the zero crossing of the first derivative line (marker on the red line at about the 30th point). This is similar to the 2% MIT rule, except that it scans backward from a higher speed, so initial small velocity peaks are neglected. This step represents the main version of the technique: henceforth the results from this version will be labeled as *"2nd derivative (zero crossing)"*.

The application of the second derivative technique is illustrated in the Figure 28.

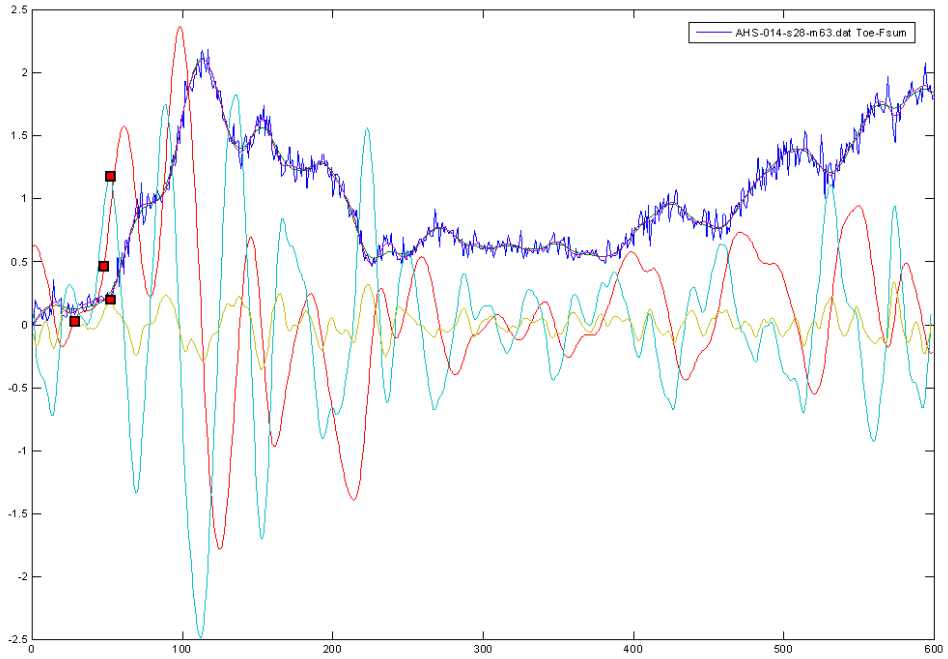


Figure 28: Application of the second derivative technique on a sample force measurement. Blue: raw data, green: : raw data filtered with 3 Hz low-pass Butterworth filter, red: first derivative of the 3 Hz filtered, cyan second derivative of the 3Hz filtered, pink: raw data filtered with 5 Hz low-pass Butterworth filter, dark yellow: second derivative of the 5Hz filtered.

3.2.3. The Spectral Flatness Method (SFM)

The Spectral Flatness Measure (SFM) is a well-known method for quantifying the amount of randomness (or "stochasticity") that is present in a signal. This measure has been widely used in signal compression, audio characterization and retrieval.

In many signal-processing applications, such as compression, modelling, detection or retrieval, one deals with the problem of determining the amount of randomness that is present in a signal. A standard method to measure randomness is based on estimation of the amount of correlation structure by means of a Spectral Flatness Measure (SFM).

SFM is defined as the ratio of the geometric mean to the arithmetic mean of the power spectral components in every spectral band. Sometimes called also "tonality coefficient", it is used to quantify how much tone-like a sound is, as opposed to being noise-like.

It can be shown that 0<SFM<1. SFM=0 corresponds to a structured or non-random process, while SFM=1 corresponds to a random signal in the sense that no extra information can be obtained by looking at longer blocks of signal samples, i.e. having no additional structure when considering these measurements as a "process". The Spectral Flatness for a signal x(t), can be defined as:

$$\theta = \frac{\exp\left(\int_{-0.5}^{0.5} \ln S_X(f)\,\mathrm{d}f\right)}{\int_{-0.5}^{0.5} S_X(f)\,\mathrm{d}f}$$ where $S_x(f)$ is the power spectral density of x(t).

When the θ becomes smaller than a predefined threshold, we indicate this point as the movement initiation point. The use of the measure in this application is motivated by the fact that once the subject intends to perform the movement, the recorded signals become less random.

The application of spectral flatness method is illustrated in Figure 29.

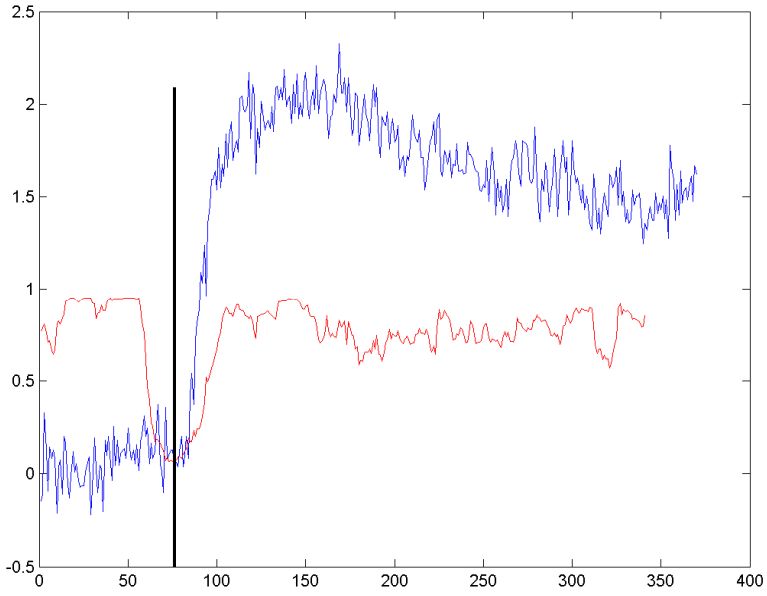


Figure 29: Application of the spectral flatness method. The blue curve is the raw signal, the red curve is the spectral flatness.

3.2.4. The ks-density based method (PDF)

The ks-density function computes a Probability Density Function (PDF) estimate of the input vector. It is a continuous function and for that reason it has been preferred to the histogram function which is a discrete function.

Typically, in signals of interest, stationary values (e.g. flat regions) of force/torque signal correspond to maxima of the PDF while values where the slope of the signal is high generally correspond to minima of the PDF.

The algorithm locates the minimum of the local minima (Minimum Density Point, MDP) in the ks-density function (Local minimum 1 in Figure 30).

A first version of the PDF estimation algorithm outputs the MDP as onset. In the second version, a line passing through the MDP (green line in Figure 30) is drawn with a slope equal to the mean value of the first derivatives of an interval around the MDP.

The ks-density also allows calculating the mean value of the signal before the task starts that generally corresponds to the first maximum in the PDF (Maximum 1 in Figure 30). The onset point is then determined when the tangent crosses this threshold value.

The application of the ks-density based technique is illustrated in Figure 30.

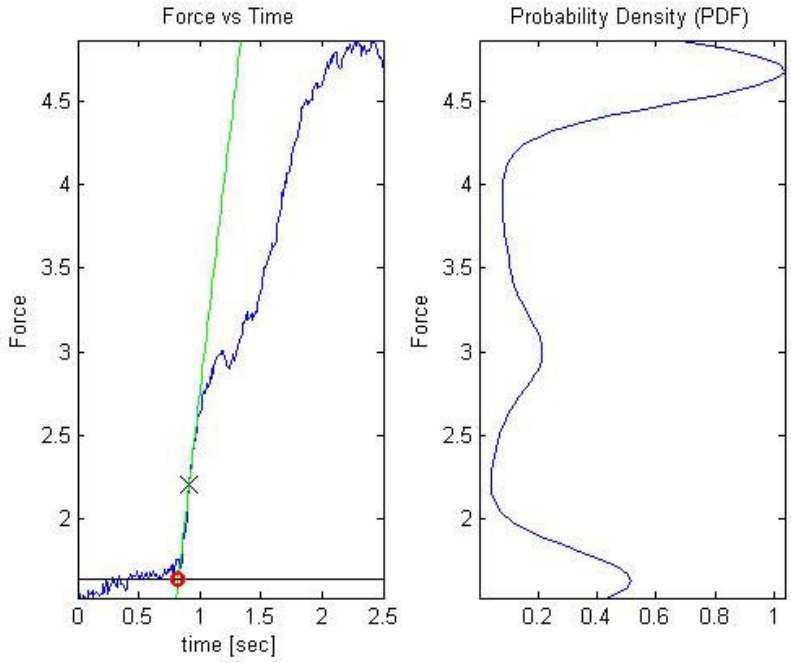


Figure 30: Application of the ks-density based technique on a sample force measurement.

3.2.5. The statistical modelling based method

The research carried out during the data mining task led to the definition of a set of feature characteristics of the movement initiation. The underlying principle consists in using these results to build a statistical model of the movement initiation for the signals the features are extracted from (two features are extracted from the same signal):

$$P_i(X_i \mid P) \quad \text{i=1,...,5}$$

Where X_i is the feature extracted from the signal i (i=1,…,5), $P_i(X_i \mid P)$ is the statistical model of the movement initiation for a patient. In this study, the $P_i(X_i \mid P)$ will be approximated by a mono-variable Gaussian distribution. $P_i(X_i \mid P)$ is trained on the proof of concept database using the manually tagged onset of movement times.

First, the beginning of the analysis window is set on the first sample and is shifted sample after sample toward the end of the signal. The corresponding feature is computed at each window shift. At the end of this process, a value of the feature for each sample for each signal is yielded.

In a second step, we searched for patterns typical for the initiation of movement by the mean of $P_i(X_i \mid P)$ and an appropriate criterion. Several criteria were used. The first approach consists in detecting the onset time when the local distribution of the feature is the closest to $P_i(X_i \mid P)$:

$$\text{C1:} \quad k_{OT} = \max_k(\overline{P}_i(X_i(k) \mid P))$$

Where k is the sample index and $\overline{P}_i(X_i(k) \mid P) = \frac{1}{a}\sum_{i=0}^{a-1} P(X_i(k+i) \mid P)$ the local average of $P_i(X_i \mid P)$. It is important to note that the value of a is a hyper parameter to be tuned to minimize the detection error. One can argue that since the patient progressively moves toward the normal population, the distribution of X_i should tend to $P_i(X_i \mid N)$ where $P_i(X_i \mid N)$ is the statistical modelling of the movement initiation for the signal i for the normal control population. This variant leads to the following criteria:

$$\text{C2:} \quad k_{OT} = \max_k(\overline{P}_i(X_i(k) \mid N))$$

Where $\overline{P}_i(X_i(k) \mid N) = \frac{1}{a}\sum_{i=0}^{a-1} P(X_i(k+i) \mid N)$. $P_i(X_i \mid N)$ is trained on the normal control population from the proof of concept database.

The last investigated approach consisted in detecting the onset of movement time when the local distribution of X_i differs from the distribution of X_i on the noise[2] part of the signal. The criterion is based on classification approach:

$$\text{C3:} \quad k_{OT} = k \text{ if } \overline{P}_i(P \mid X_i(k)) > \overline{P}_i(Noise \mid X_i(k))$$

With the Bayes rule, $P_i(P \mid X_i)$ and $P_i(Noise \mid X_i)$ can be developed as follows:

$$P_i(P \mid X_i) = \frac{P_i(X_i \mid P)P(P)}{P_i(X_i \mid P)P(P) + P_i(X_i \mid Noise)P(Noise)}$$

$$P_i(Noise \mid X_i) = \frac{P_i(X_i \mid Noise)P(Noise)}{P_i(X_i \mid P)P(P) + P_i(X_i \mid Noise)P(Noise)}$$

The a priori probability $P(Noise)$ and $P(P)$ are arbitrarily set to 0.5. Consequently, C3 can be re-written:

$$\text{C3:} \quad k_{OT} = k \text{ if } \overline{P}_i(X_i(k) \mid P) > \overline{P}_i(X_i(k) \mid Noise)$$

[2] By 'noise part of the signal', we mean here the part of the signal that does not correspond to a voluntary contraction.

Where $\overline{P}_i(X_i(k)\,|\,P) = \frac{1}{a}\sum_{i=0}^{a-1} P(X_i(k+i)\,|\,P)$ and $\overline{P}_i(X_i(k)\,|\,Noise) = \frac{1}{a}\sum_{i=0}^{a-1} P_i(X_i(k+i)\,|\,Noise)$ and a is a hyper parameter to be tuned to minimize the detection error.

$P_i(X_i\,|\,Noise)$ is trained on the inactive parts of the signals (situated before the manually tagged onset of movement time) contained in all the measurement sessions of the DTN database. For the same motivation that leads to the definition of C2, we also define C4:

$$\text{C4:} \quad k_{OT} = k \text{ if } \overline{P}_i(X_i(k)\,|\,N) > \overline{P}_i(X_i(k)\,|\,Noise)$$

3.3. Assessment methodology

3.3.1. Introduction

The proof of concept database (see section 2.2.1.2) was distributed to the clinical experts in order to understand more in details how they decide which time point in the signal corresponded to the onset of the signal related to the execution of the ADL task.

We discovered that they used a more sophisticated analysis, when compared to the results obtained by the calculated onset. They gave a physiological interpretation to the signals, which became reflected in the way the onset time was tagged.

Their work importantly contributed to the data mining activities.

3.3.2. Methodology

The adopted methodology for the comparative study carried out by SSSA is detailed in [4]. The four candidate methods are:

- The 2% rule
- The second derivative method
- The spectral flatness method
- The ks-density based method

The dataset used to assess the performance of the onset of movement time detection algorithm, contained data from 96 patients (48 patients and 48 normal controls), all selected from the proof of concept database.

The statistical property based method was compared with the best technique identified by the SSSA comparative study. In order to prevent the comparisons from being biased by a mismatch between the experimental conditions adopted by two partners, the two methods were assessed on the same database under the same experimental settings.

The test dataset was the DTN database (see section 2.2.1.3). The onset of movement times are estimated for all the patients (not for the normal controls) and for all the sessions. Only the signals from which the features were extracted from[3] were considered, namely:

- Torque, Middle finger sensor, 'Glass' task, Recording ID: 2
- Torque, Thumb sensor, 'Lifting a bottle' task, Recording ID: 4
- Torque, Seat sensor, 'Glass' task, Recording ID: 3
- Force, Thumb sensor, 'Glass' task, Recording ID: 4
- Force, Index finger sensor, 'Bag' task, Recording ID: 4

3.4. Experimental results

3.4.1. Assessment of the Automatic Onset Detection Algorithms

The results of the comparative study carried out by SSSA are presented in the Table 13.

Onset technique	Mean value	Standard deviation	Variance	Median	Mean absolute
2% rule	-0.3300	1.3568	1.8410	-0.0554	0.8520
Spectral flatness	N/A	N/A	N/A	N/A	N/A
Ks-density (interpolation method)	-0.3131	0.8592	0.7382	-0.1396	0.5629
Ks-density (max speed method)	-0.5022	0.9035	0.8164	-0.3081	0.6958
2nd derivative (zero crossing)	-0.0676	0.6900	0.4760	-0.0100	0.3974
2nd derivative (filtered 3Hz)	-0.1934	0.6936	0.4810	-0.1358	0.4441
2nd derivative (filtered 5Hz)	-0.2012	0.6944	0.4822	-0.1458	0.4490

Table 13: Results of the SSA comparative study of different onset of movement detection methods. Mean value, standard deviation, variance and median are related to the error distribution. The mean absolute value is the mean of the error absolute value distribution.

The best method in Table 13 is the 2nd derivative (zero crossing) method.

[3] When the features are extracted from an angular distribution, the second derivative method was applied on the norm of the effort.

As explained in the section, this winning method was assessed on the DTN database together with the statistical approach in order to avoid a mismatch between the experimental conditions.

The mean (bias), the standard deviation (std), the median (median) of the distribution of the error between the automatically detected onset of movement time and the manually tagged value were computed. The mean of the absolute value of the error was also calculated.

The results are summed up in the Table 14.

Technique	Bias (s)	Std (s)	Median (s)	Mean abs (s)
2nd derivative (zero crossing)	-0,099493	0,74741	-0,065806	0,43962
C1	1.292850	1.449872	0.962903	1.443073
C2	1.231265	1.426477	0.935323	1.406306
C3	-0.174144	0.895032	-0.136936	0.546866
C4	-0.140038	1.036150	-0.182097	0.614586

Table 14: Assessment of the statistical approach.

Among the statistical approaches, the maximum-likelihood-based methods lead to the weakest results with a mean of the absolute error superior to 1.4 second. The techniques based on a classification criterion perform much better but their performances are lower than the 2nd derivative (zero crossing) algorithm for which the mean of the absolute error is about 0.4 s.

Conclusions

The best performances are achieved by the 2nd derivative (zero crossing) method. The mean of the resulting absolute error distribution is equal to 0.4 s. We have to keep in mind that the automatic onset detection performances are to be appreciated with regards to the analysis window width. The width of that window was set to 0.75 s. Thus, the mean absolute error appears to be more than half this value!

3.4.2. DTN sensitivity analysis

Section 3.4.2.1 is devoted to the sensitivity of the DTN to a small variation of the identified onset of movement times. Section 3.4.2.2 presents a new feature set defined on the basis of the results of the section 3.4.2.1. Section 3.4.2.3 briefly describes the influence of the automatic onset detection algorithm on the DTN.

3.4.2.1. DTN and feature sensitivity

The DTN related to the j^{th} measurement session of the i^{th} patient is computed on the basis of 6 relevant features identified as such by the feature subset selection algorithm (see section 2.1.3), extracted from 5 different signals:

$$DTN_{ij} = f\left(F_1,...,F_6\right)$$

Where F_k k=1,...6 are the features. Each feature is extracted within an analysis window beginning at the identified onset of movement time for the corresponding signal so that:

$$F_k = F\left(t_{ij}\left(k\right)\right) \text{ k=1,...6}$$

Where $t_{ij}\left(i\right)$ stand for the onset times.

The sensitivity of the DTN with regards to the onset detection error[4] is assessed by comparing DTN values computed using the tagged onset of movement times (DTN_{ij}^0) with DTN computed when these onset times vary around these values (DTN_{ij}). In order to prevent the estimator from being influenced by scaling effects and to make the interpretation of the results easier, we defined the relative DTN error:

$$\Delta_R DTN_{ij} = \frac{DTN_{ij} - DTN_{ij}^0}{\max(DTN_{ij}, DTN_{ij}^0)}$$

The distribution of $\Delta_R DTN_{ij}$ is obtained by computing $\Delta_R DTN_{ij}$ for all the measurement sessions of the DTN database (see section 2.2.1.3). Let denote $t_{ij}\left(k\right)$ (k=1,…,6), the current onset time and $t_{ij_{Manual}}\left(k\right)$ the manually tagged values, $t_{ij}\left(k\right)$ can be written as follows:

$$t_{ij}\left(k\right) = t_{ij_{Manual}}\left(k\right) + e_{ij}\left(k\right)$$

Where $e_{ij}\left(k\right)$ is the onset detection error.

[4] In this section, the difference between the manually tagged onset time and the time used to extract the features will be sometimes called 'Onset detection error', although it does not result from an automatic onset detection algorithm. Here, the purpose is not to simulate the influence of such algorithm.

The average of the absolute value of $\Delta_R DTN_{ij}$ was used as a global estimator of the DTN variations for the whole database:

$$\overline{|\Delta_R DTN|} = \frac{1}{N}\sum_{i=1}^{N}|\Delta_R DTN_{ij}|$$

Where N is the number of sessions in the DTN database and $|.|$ is the absolute value operator.

Figure 31 illustrates the evolution of $\overline{|\Delta_R DTN|}$ for onset detection errors ranging from -1 to 1 second (constant distribution: $e_{ij}(k) = cst \quad \forall i, j, k$). The DTN were computed according to the normalized likelihood metric with a full covariance matrix 1-Order GMM (no feature pre-processing) for the six feature subsets (see section 2.2.3).

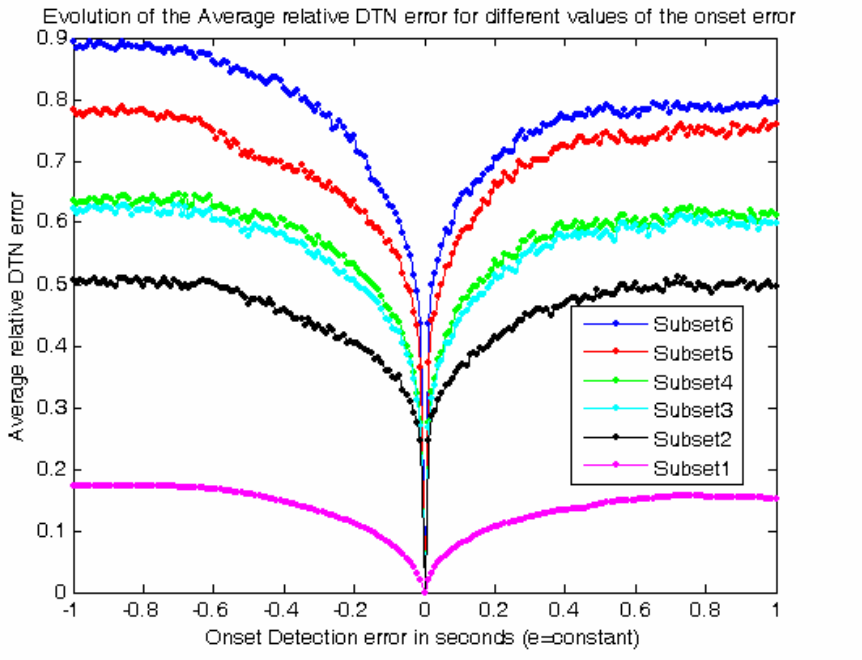


Figure 31: Evolution of $\overline{|\Delta_R DTN|}$ (normalized likelihood metric) for different values of the onset detection error (constant distribution).

For all the subsets, Figure 31 shows that the average of the relative absolute value of the DTN error starts rising when the onset detection error increases. The rise is first very sharp and progressively smoothens afterwards. Thereafter, once the onset error exceeds 0.75 s, $\overline{|\Delta_R DTN|}$ does not evolve anymore. The evolution of $\overline{|\Delta_R DTN|}$ is not exactly the same for positive and negative errors. This suggests that some features are differently affected according to the sign of the error.

For a given value of the onset detection error, $\overline{|\Delta_R DTN|}$ progressively increases when more features are considered. Nevertheless, the rise in $\overline{|\Delta_R DTN|}$ is not similar for all the features.

Indeed, considering the features 2 and 5 results in a dramatic increase of $\overline{|\Delta_R DTN|}$ while for the features 1 and 4 the $\overline{|\Delta_R DTN|}$ rises only slightly.
It is worth noting that for an onset detection error of 0.43 s in absolute value (the performance achieved by the best onset detection method), $\overline{|\Delta_R DTN|}$ reaches more than 15 % for the feature subset 1 and rises to about 85 % (80 % for positive errors) for the feature subset 6. The DTN variability is thus significant for onset errors commonly encountered in the project.

A fine understanding of the DTN variability sources requires a detailed analysis of how the features behave when the onset times are affected by detection errors. To do so, we compute the average feature error when the onset of movement time varies around the tagged value:

$$\overline{|\Delta F_i|} = \sum_{j=1}^{N} \left| F_{ij} - F_i^0{}_j \right|$$

Where F_i^0 is the feature value computed with tagged onset times and N is the number of measurement sessions. In order to avoid scaling effects we use the normalized features:

$$F_{iN} = \frac{F_i - \overline{F_i}}{\sigma_i}$$

Where $\overline{F_i}$ and σ_i are the mean and the standard deviation of the feature F_i (in the dataset used to train the inference model). Figure 32 plots the evolution of the normalized feature errors for different onset detection errors (e=constant).

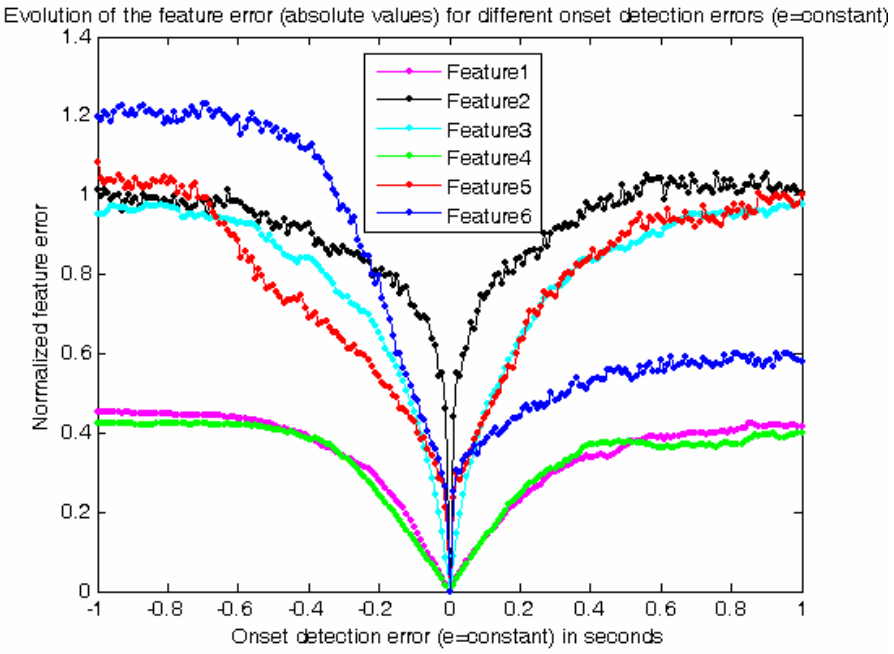


Figure 32: Evolution of the normalized feature error for different onset detection errors (e=constant).

As the Figure 31 suggested, Figure 32 shows that values of the features 1 and 4 evolve the most smoothly when the onset time error increases while the other features are characterized by a bigger variability. We also notice that some features are differently affected according to the sign of the error. This explains the asymmetric shape of the $\overline{|\Delta_R DTN|}$ evolution observed in the Figure 31. The feature 6 is particularly concerned. The reason is that this feature is extracted from the distribution of the angle between a sample and the previous one. When the signal does not correspond to an active part of the signal (negative error), the norm of the effort vectors are almost equal to zero and the angles between these vectors are not well defined.

The most robust features (features 1 and 4) are extracted from the cumulative sum distribution. The feature based on the normalized residuals of the curve fitting approach (feature 5) has a variability similar to the features extracted from an angular distribution (features 2, 3, 6).

One can argue that a constant error $e_{ij}(k) = cst \ \ \forall i, j, k$ without any dispersion is not a realistic experimental condition to assess the DTN sensitivity. We also computed $\overline{|\Delta_R DTN|}$ when the onset of movement error is a Gaussian process. Figure 33 plots the evolution of $\overline{|\Delta_R DTN|}$ for different values of μ and σ ranging from -1 to 1second and from 0.1 to 1.5 second respectively ($e_{ij}(k) = N(\mu, \sigma)$).

We see that considering the dispersion of the onset detection errors results in significant changes of the average relative DTN error only for small values of the mean. In that region, the bigger is the standard deviation, the higher is the DTN error. For big μ values, increasing the standard deviation only causes a small decrease in $\overline{|\Delta_R DTN|}$.

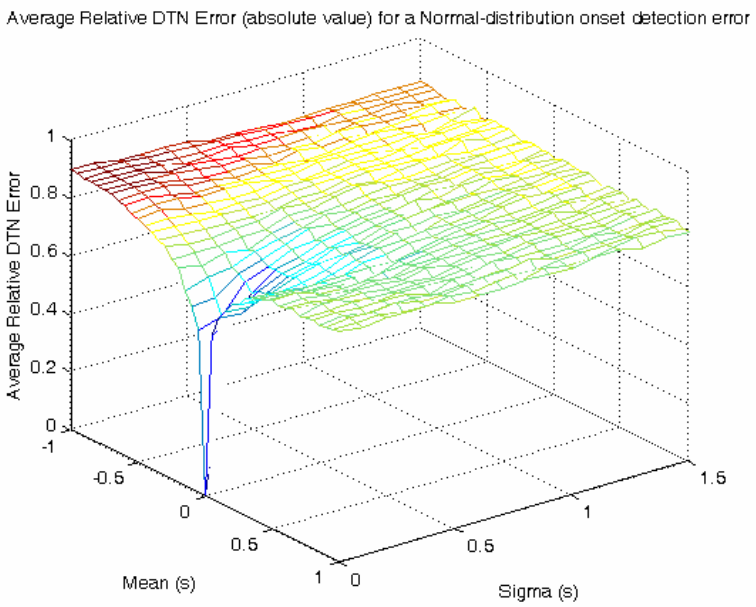


Figure 33: Evolution of $\overline{|\Delta_R DTN|}$ for a normal distribution of the onset detection error for different values of μ and σ ranging from -1 to 1 second and from 0.1 to 1.5 second respectively (feature subset 6).

Conclusions

As an overall estimator of the DTN variability, we defined the average relative DTN error (in absolute value) $\overline{|\Delta_R DTN|}$. The DTN sensitivity to the onset time was assessed by computing the value of $\overline{|\Delta_R DTN|}$ when the time used to extract the features is progressively taken away from the tagged value. The evolution of $\overline{|\Delta_R DTN|}$ for all the feature subsets unveiled that the sensitivity continuously increases as the subset dimensionality rises. However, all the features do not influence the DTN value in the same way. A detailed analysis of the feature variability showed that the two most robust features (and thereby causing the smallest DTN variability) were extracted from a cumulative sum distribution.

3.4.2.2. Robust feature subset definition

The previous section suggests that removing features causing a lot of $\overline{|\Delta_R DTN|}$ variability may lead to DTN values less sensitive to the onset time value. Such robustness is a necessary condition to appropriately assess the recovery stroke because it is a sign that variations of the feature values are due to physiological characteristics (at least partly). Indeed, we suspect that the variability of some features is partly due to purely numerical effects (particularly for feature extracted from an angular distribution).

The features 1 and 4 were shown to be the most robust with regards to the onset time. The Figure 34 plots the evolution of $\overline{|\Delta_R DTN|}$ when only these features are used to compute the DTN.

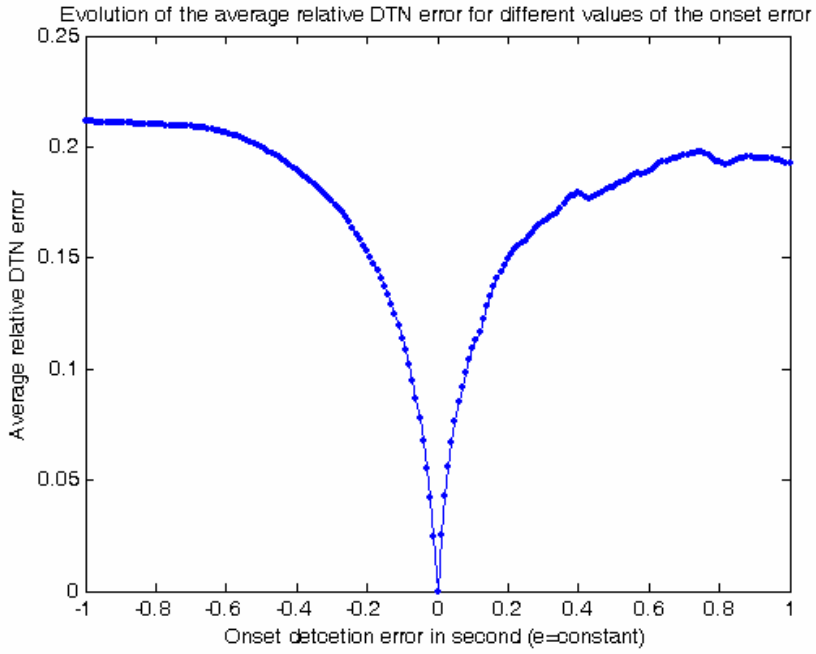


Figure 34: Evolution of $\overline{|\Delta_R DTN|}$ for different values of the onset detection error (e=constant). Only the two most robust features (feature 1-4) are used to compute the DTN.

As expected, Figure 34 shows that $\overline{|\Delta_R DTN|}$ is much lower than all the feature subsets presented in the Figure 31 at the exception of the subset 1. When compared with the subset 1, $\overline{|\Delta_R DTN|}$ increases only slightly.

When the error dispersion is taken into account by modelling the onset detection error by a Gaussian distribution, the conclusions are the same as for the feature subset 6. For small values of the mean, a bigger standard deviation results in a higher average relative DTN error. When the mean of the distribution is important, $\overline{|\Delta_R DTN|}$ only decreases slightly when the standard deviation rises. The gain in robustness when using only the features 1 and 4 for small errors is verified with a normal distribution of the error.

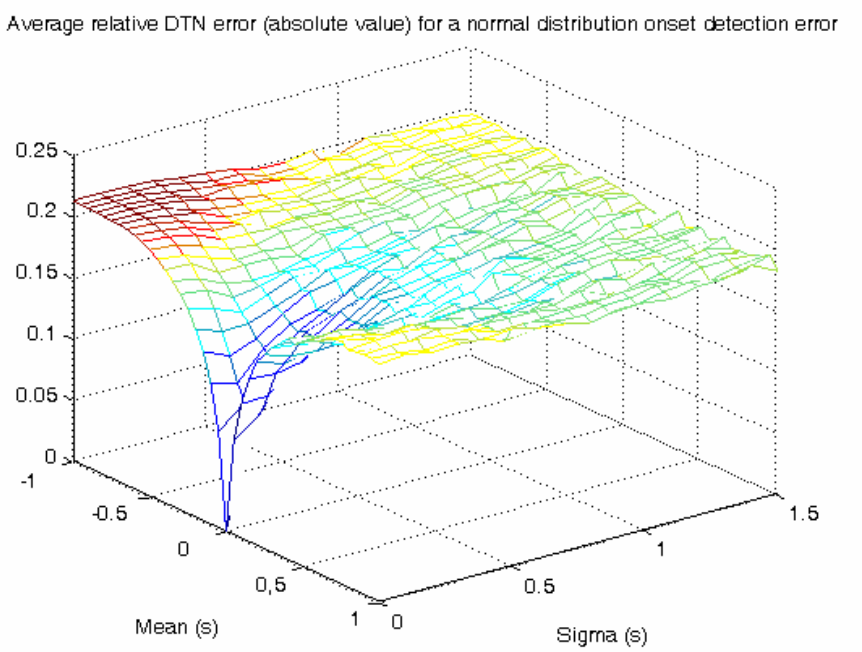


Figure 35: Evolution of $\overline{|\Delta_R DTN|}$ for a normal distribution of the onset detection error for different values of μ and σ ranging from -1 to 1 second and from 0.1 to 1.5 second respectively (features 1-4).

It is important to note that although the robustness of the features to a small onset time error is a necessary condition to appropriately assess the recovery stroke, it is not a sufficient condition. Classification tests showed that patients and normal controls were more successfully separated in a high-dimension recovery space. Therefore, using less features, like we did, may also result in important information losses and the analysis may fail in revealing the recovery state of a patient. The fact that the feature 1 was shown to be strongly discriminative during the classification test (classification rates reached about 80 % with this single feature) motivated this approach.

The following graphs compare the DTN curves (based on the normalized likelihood metric) obtained when the subset 6 is used with those that are obtained when only the features 1 and 4 are considered. The patients are the same as those presented in section 2.2.5. The DTN are

computed using the normalized likelihood metric. The inference model is a full covariance matrix 1-Order GMM. The graphs are to be examined pair-wise: the first one is related to the feature subset 6, the second one corresponds to the use of the features 1 and 4 only.

The blue curve represents the time trajectory of the DTN distribution mean. The upper dashed red curve is the time trajectory of 75-percentile, the lower red curve is the time trajectory of the 25-percentile. Consequently, the half of the patients evolves between the two red curves. The thick lines are smoothing splines (same colour as the raw data). The procedure adopted to draw these curves is the same as in the section 2.2.5.

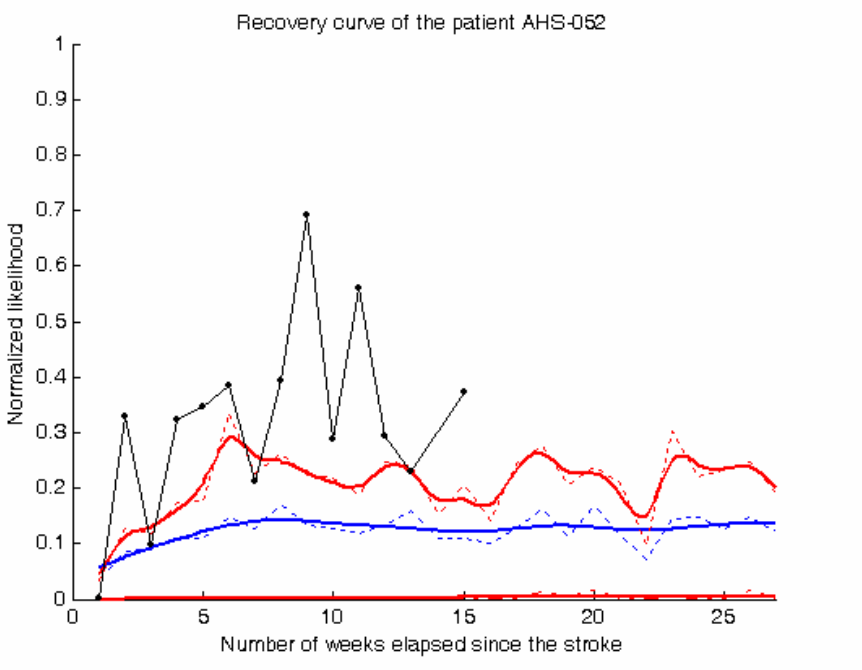


Figure 36: DTN evolution using the feature subset 6. Patient that recovers faster than the overall patient population. Dashed blue curve: time trajectory of the DTN distribution mean, upper dashed red curve: 75-percentile, lower red curve: 25-percentile, black curve: recovery curve of the patient AHS-052. The thick lines are smoothing splines (same color as the raw data).

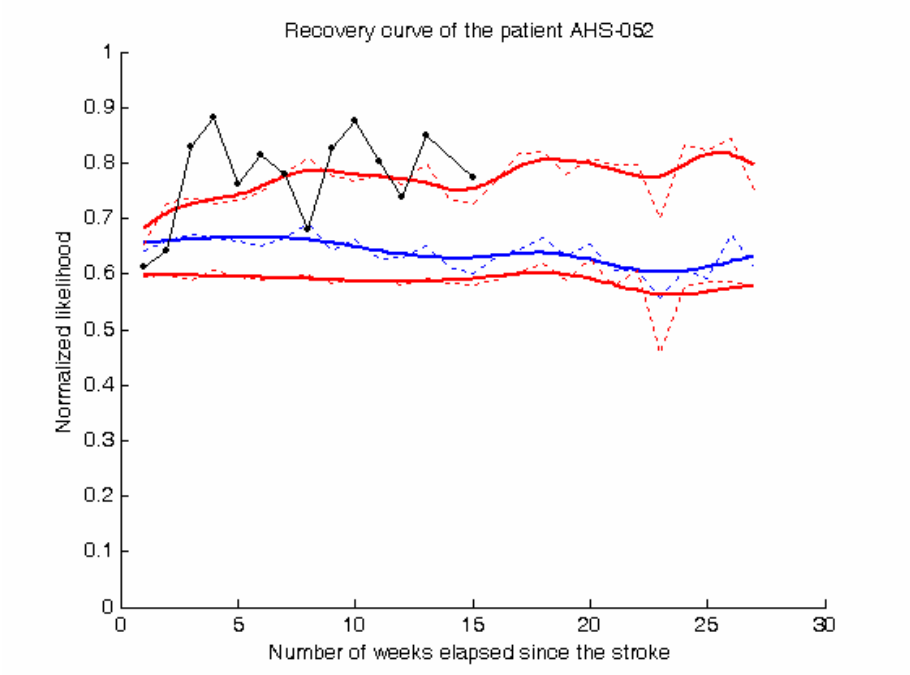


Figure 37: DTN evolution using the features 1 and 4. Patient that recovers faster than the overall patient population. Dashed blue curve: time trajectory of the DTN distribution mean, upper dashed red curve: 75-percentile, lower red curve: 25-percentile, black curve: recovery curve of the patient AHS-052. The thick lines are smoothing splines (same color as the raw data).

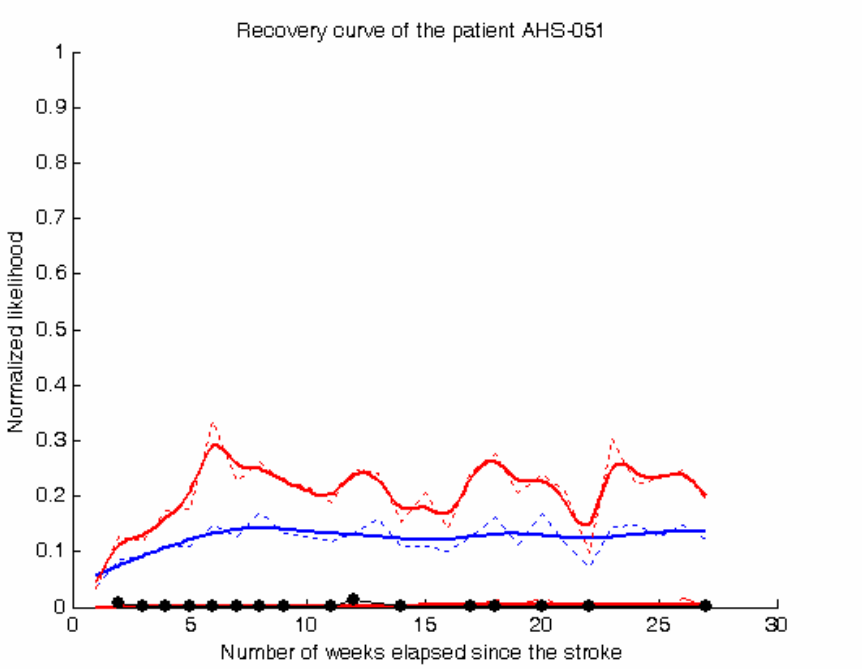


Figure 38: DTN evolution using the feature subset 6. Patient that doesn't recover. Dashed blue curve: time trajectory of the DTN distribution mean, upper dashed red curve: 75-percentile, lower red curve: 25-percentile, black curve: recovery curve of the patient AHS-052. The thick lines are smoothing splines (same color as the raw data).

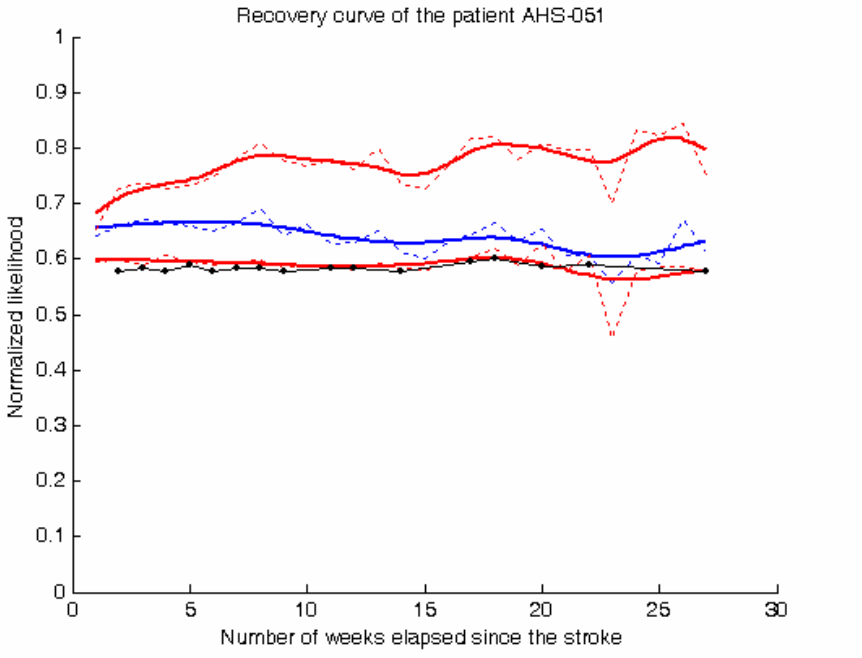


Figure 39: DTN evolution using the features 1 and 4. Patient that doesn't recover. Dashed blue curve: time trajectory of the DTN distribution mean, upper dashed red curve: 75-percentile, lower red curve: 25-percentile, black curve: recovery curve of the patient AHS-052. The thick lines are smoothing splines (same color as the raw data).

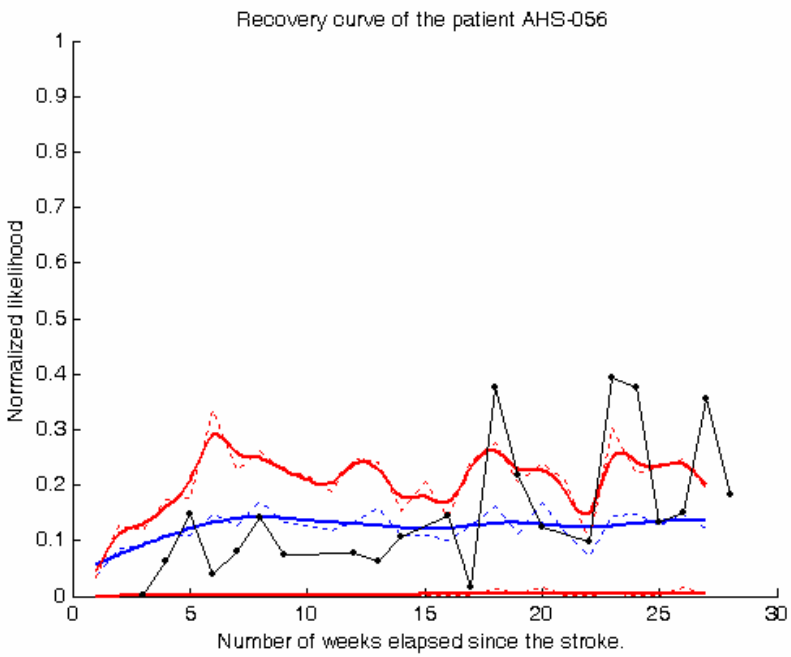


Figure 40: DTN evolution using the feature subset 6. Patient that recovers roughly like the overall patient population. Dashed blue curve: time trajectory of the DTN distribution mean, upper dashed red curve: 75-percentile, lower red curve: 25-percentile, black curve: recovery curve of the patient AHS-056. The thick lines are smoothing splines (same color as the raw data).

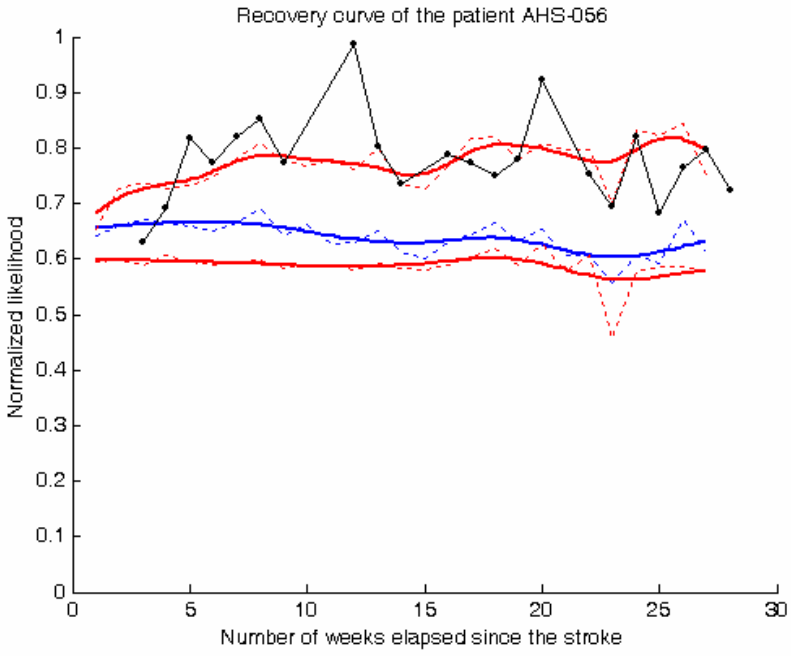


Figure 41: DTN evolution using the features 1 and 4. Patient that recovers roughly like the overall patient population. Dashed blue curve: time trajectory of the DTN distribution mean, upper dashed red curve: 75-percentile, lower red curve: 25-percentile, black curve: recovery curve of the patient AHS-056. The thick lines are smoothing splines (same color as the raw data).

The three patients have now a recovery score much higher than zero at the beginning of the rehabilitation process (see Figure 37, Figure 39 and Figure 41). The trajectories of the DTN distribution characteristics (the mean, the 25-percentile and the 75-percentile) also shifted, meaning that this observation can be generalized to the whole database. So far, we cannot give a physiological interpretation.

The 25-percentile and the 75-percentile trajectories are equally distant from each other for the two feature subsets. Thus, the use of the features 1 and 4 only does not result in more scattered recovery curves. Their evolution is also a little bit smoother. The average DTN trajectory is different for the two feature subsets. When the DTN values are computed with the feature subset 6 is used, there is a region characterized by a rising slope from the week 1 to the week 8 followed by a plateau where the DTN mean does not vary anymore. The decreasing part of the trajectory can be explained by the fact that patients who quickly recovered left the ALLADIN assessment program before its end so that patients with serious impairments were relatively more represented in the population than at the beginning of the project.

When focusing only on the evolution (without paying attention to the absolute value of the recovery score), the examination of the recovery curve of all the patients in the DTN database reveals some recurrent characteristics. They are summarized in the following points:

- When the recovery curve presents a clear increasing trend, it is conserved when the features 1 and 4 are used instead of feature subset 6. The variability is then generally lower (see Figure 37 and Figure 41).

- When the recovery curve is composed of a succession of peaks without a clear trend with the feature subset 6, the use of the features 1 and 4 sometimes reveals a more clear evolution (either a increasing slope or no evolution).

- When the DTN values computed with the feature subset 6 decrease and stay stuck to zero, those obtained with the features 1 and 4 rapidly go down below the 25-percentile trajectory.

Conclusions

In this section, we defined a new feature set containing only the most robust features identified in the section 3.4.2.1 (feature 1-4). The resulting recovery curves presented some interesting characteristics when compared with the curves obtained with the feature subset 6. The new feature set conserves a strong trend for the DTN evolution and the variability is smaller. Sometimes, a clear and relatively smooth evolution is revealed for recovery curves that presented only a succession of peaks when they were computed with the feature subset 6.

3.4.2.3. Sensitivity to the automatic onset detection error

Figure 42 plots the distribution of onset time estimation error (DTN database) for the 2nd derivative (zero crossing) method.

Table 15 presents the value of the average relative DTN error when this method is used to determine the onset of movement time. We immediately see that, $\overline{|\Delta_R DTN|}$ drops to a bit less than 15 % when the features 1 and 4 are used instead of the feature subset 6 (about 61 %).

It is important to note that these values are quite different from those obtained when the error detection was modelled by a Gaussian process. This can be explained by the fact that the error distribution in the Figure 42 is not actually a normal distribution (a statistical test rejected the normal hypothesis at a confidence level of 0.01). The important probability mass for small errors and the sensitivity of the DTN in that region justifies the smaller than expected values of $\overline{|\Delta_R DTN|}$.

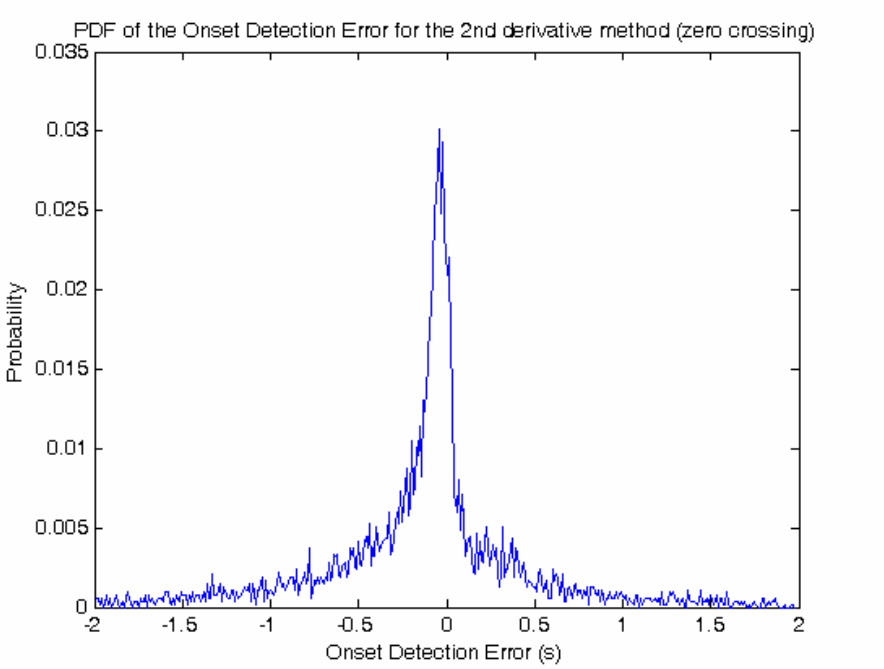


Figure 42: Distribution of onset time estimation error (DTN database) for the 2nd derivative (zero crossing) method.

	$\overline{\lvert\Delta_R DTN\rvert}$ onset error	$\overline{\lvert\Delta_R DTN\rvert}$ predicted
Feature subset 6	0.61	0.86
Features 1 and 4	0.15	0.18

Table 15: Average relative DTN error when the best automatic onset detection identifies the onset of movement time used to extract the features.

3.5. Conclusion

A basic problem has to be faced by the automatic onset detection techniques:

- Automatic onset techniques are, in one way or another, trying to identify the time when the signal reaches a threshold that can be related to information deriving from voluntary isometric contraction of the patient, rather than from noise sources.

- Clinical experts are detecting the onset time by interpreting the physiological meaning of the signal and, in some cases, neglecting initial portion of the signals, clearly deriving from voluntary isometric contraction, but not directly (physiologically) related to the ADL task being executed.

The sensitivity analysis revealed that the DTN changes rapidly and significantly when the time used to extract the features progressively takes away from the manually tagged value. The performance achieved by the best onset detection method does not allow a fully automated process. Thus, a manual feedback at the pre-processing stage should be included in the final integration module.

A detailed study of the behaviour of the features showed that some features were more robust than the other with regards to the onset detection error. A new feature set composed only of the two most robust features was defined. Then, the average relative DTN error (assessed with the automatically detected onset times) dramatically decreased from about 62 % when the feature subset 6 is used to a bit less than 15 %. The resulting recovery curves also presented interesting characteristics: no significant information loss, less variability. However, recovery scores were very high even at the beginning of the rehabilitation process, which is difficult to justify. The set of new recovery curves will be submitted the clinical partners for evaluation. Other feature combinations can also be tested in the future.

4. Developed software

The data mining work led to the development of two software modules. The first one is a Matlab graphical interface developed for research purpose only. The second one consists of a DLL programmed in C allowing integration of the data mining results in the final integration module.

4.1. The Diagnostic tool

4.1.1. General window

The diagnostic tool is a graphical interface and was developed in Matlab for research purpose only. A screenshot of the application is presented in the Figure 43.

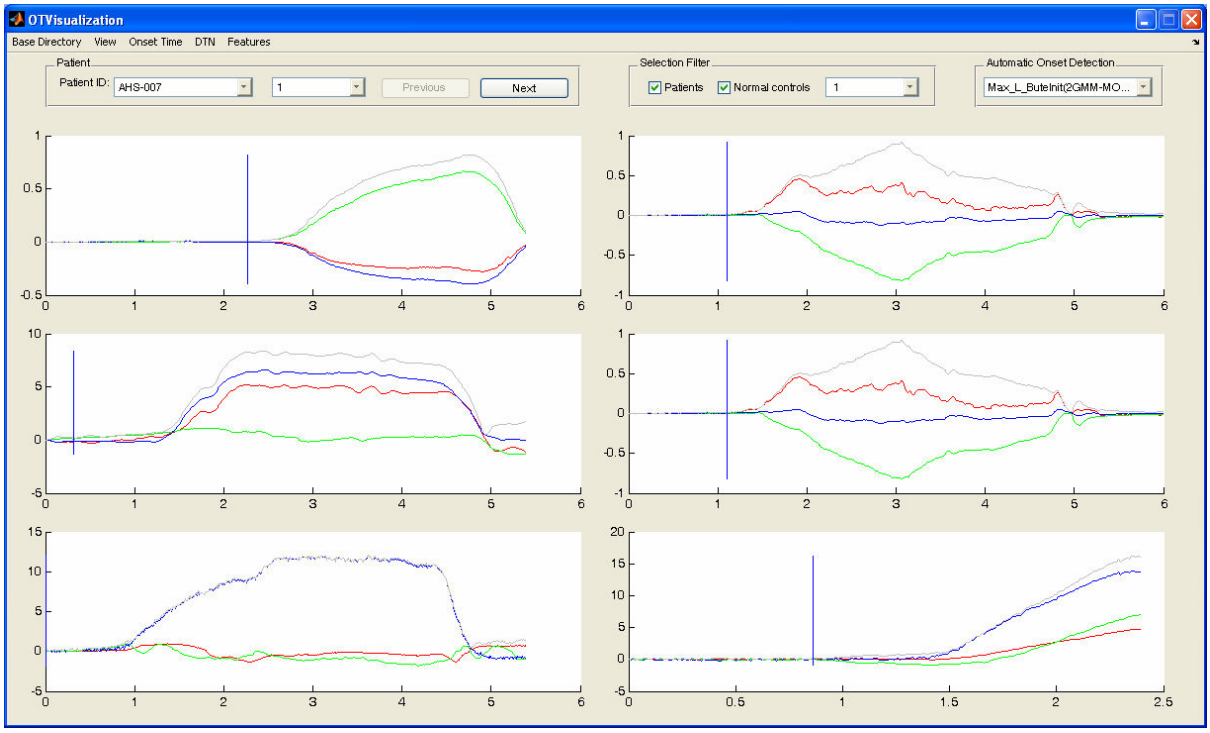


Figure 43: Diagnostic tool. Screenshot of the application.

A first group of controls *(Patient*) at the top of the window allows the user to select the current patient and the current session.

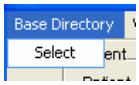
A second group of controls (*Selection filter*) allows the user to set a filter on the selection of the current patient. For instance, he may want that only patients are present in the patient selection combo box.

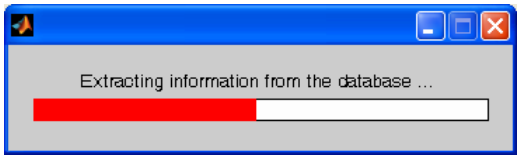
The last group of controls allows the user to select the current automatic onset of movement detection method.

The signals are displayed in six separate subplots.

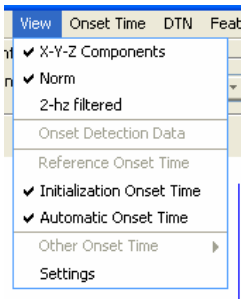
4.1.2. The view menu and the main supported functionalities

- The *base directory* menu item. It contains only one sub item launching a browse for folder menu to select the base directory of the database. After selecting the base directory, the progress bar below is displayed while the program is retrieving the database content.



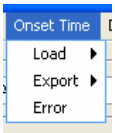


- The *View* menu item. Allows to tune the display settings (hidden /plotted signals, onset of movement time visualisation,…).



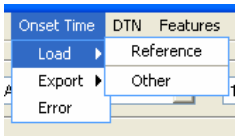
- The *onset time* menu item handles all the operations supported for the onset time. In this application we distinguish:

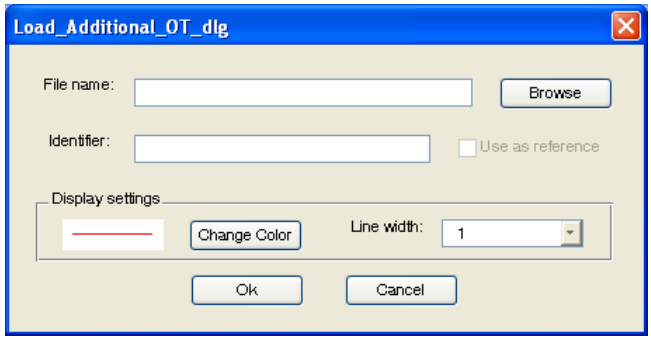
  - The *reference* onset time (generally manually tagged).
  - The *initialization* onset time. Initial guess for the onset of movement if the currently selected automatic detection method uses a two-pass procedure.
  - The automatic onset time. Onset of movement time estimated by the automatic currently selected detection algorithm.



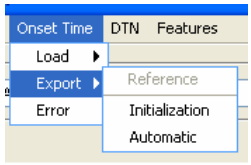
The user can:

  - ✓ Load a list of onset of movement time from a text file. The loading dialog box below is then displayed. The dialog allows the user to specify the file containing the onset of movement times to load (*browse* button), an application identifier for this data, some display settings that will be used when these data will be plotted.

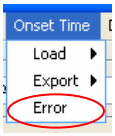




✓ Export the onset of movement time computed by the currently selected automatic detection method.



✓ Compute the distribution of the error between the onset of movement estimated by the currently selected automatic detection algorithm and the reference onset.



- The *DTN* menu item allows the user to plot the recovery curve of the current patient in a separate figure. The following modes are available: KDE (the DTN is computed with a KDE as an inference model), GMM (DTN is computed with a full covariance GMM as an inference model).

- The *feature* menu item allows the user to plot the feature trajectories in a separate figure.

4.2. The DTN-Integration package

The DTN-Integration package consists of DLL developed in C, an example program showing the usage of the DLL, the code of a function implementing dynamic loading.

This package computes the DTN using the results of the data mining research.

Function:

```
int DTN(double* in, double* outNL, double* outPost, unsigned int nbSession)
```

Arguments:

Input

in:	double array containing the features of nb sessions, grouped per session.
nbSession:	unsigned int containing the number of sessions for which the DTN must be computed.

Output

outNL:	Double array containing the DTN based on the normalized likelihood metric with a full covariance matrix GMM (one value per session).
outpost:	Double array containing the DTN based on the a posteriori probability of normality likelihood metric with a full covariance matrix GMM (one value per session).

A Appendix-A: Curve fitting and system identification modeling.

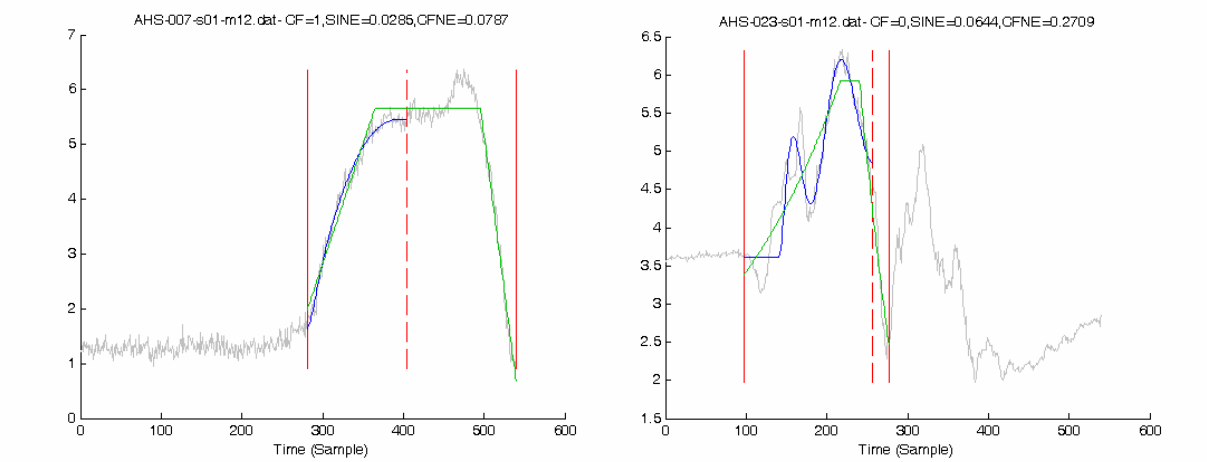
A.1 Curve fitting and system identification modeling: illustrating plots

The following figures show the curves resulting from the curve fitting and the system identification modelling for some patients and normal controls (variable length window). The considered signal is the norm of the force (Thumb sensor, drinking a glass). The model order for the system identification is $n_b = 3$, $n_f = 4$.

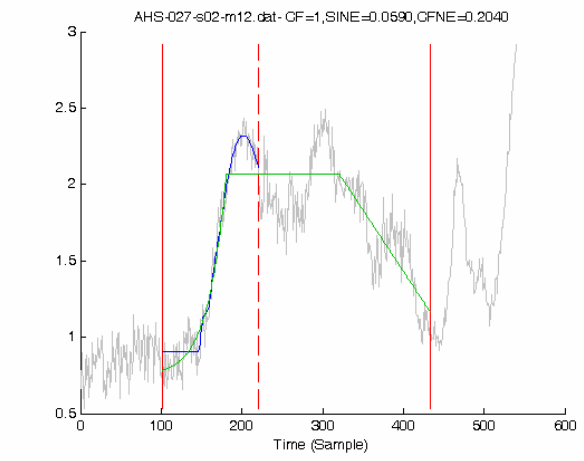
- The vertical red plain lines delimit the window used for the curve fitting.
- The vertical red dashed lines delimit the window used for the system identification.
- The signal is plotted in grey.
- The curve resulting from the curve fitting approach is plotted in blue.
- The curve resulting from the system identification approach is plotted in green.

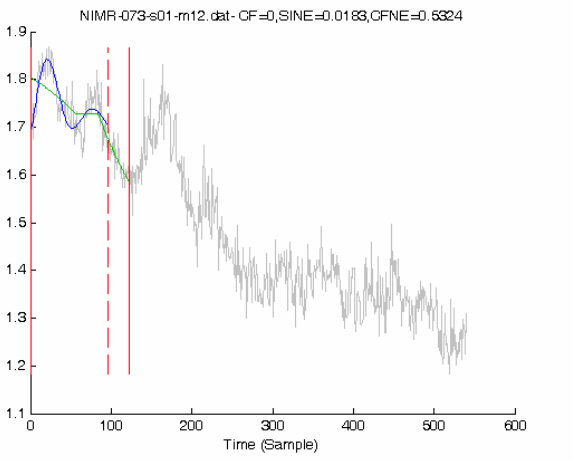
In the title:

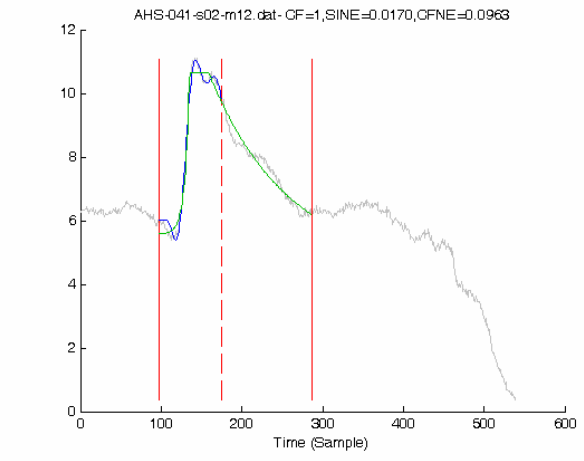
- CF indicates whether the patient is a normal control (0:patient, 1:normal control)
- SINE is the normalized error for the system identification approach.
- CFNE is the normalized error for the curve fitting approach[5].

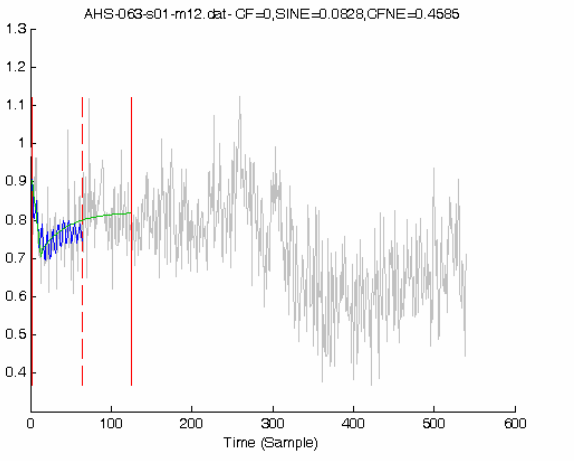


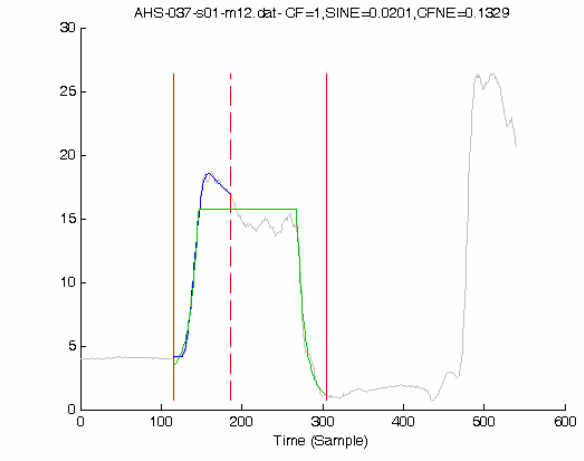
[5] The modeled part of the signal is not the same for the two approaches since the analysis window lengths differ. Therefore, the comparison between the normalized error is not straightforward.

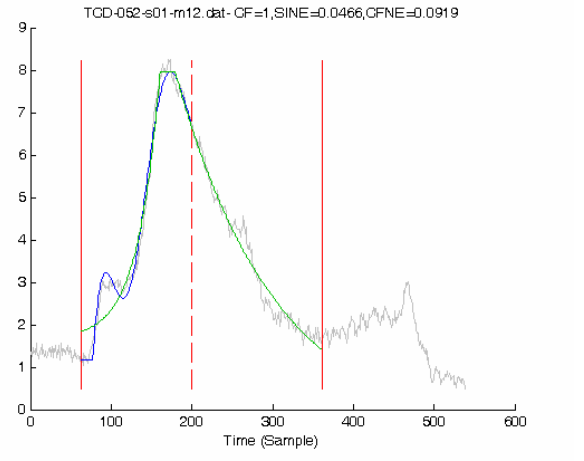

AHS-027-s02-m12.dat - CF=1,SINE=0.0590,CFNE=0.2040
Time (Sample)

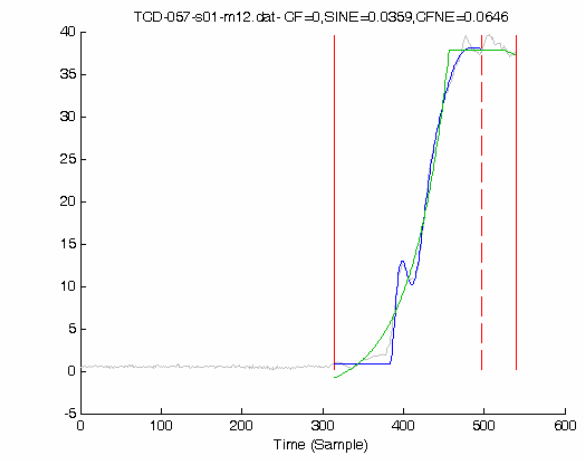

NIMR-073-s01-m12.dat - CF=0,SINE=0.0183,CFNE=0.5324
Time (Sample)

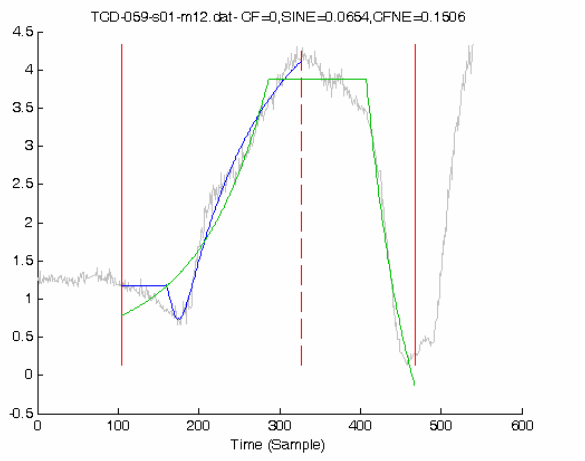

AHS-041-s02-m12.dat - CF=1,SINE=0.0170,CFNE=0.0963
Time (Sample)

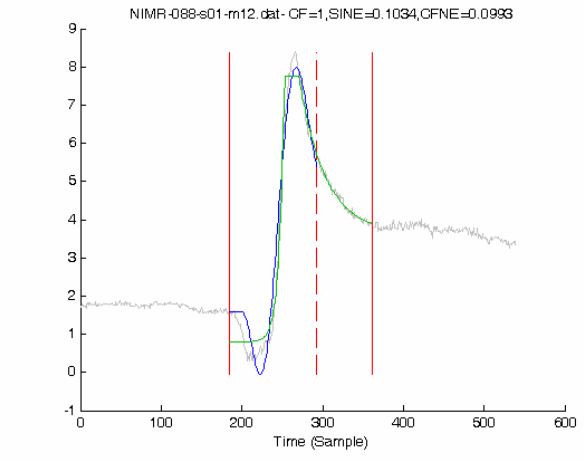

AHS-063-s01-m12.dat - CF=0,SINE=0.0828,CFNE=0.4586
Time (Sample)

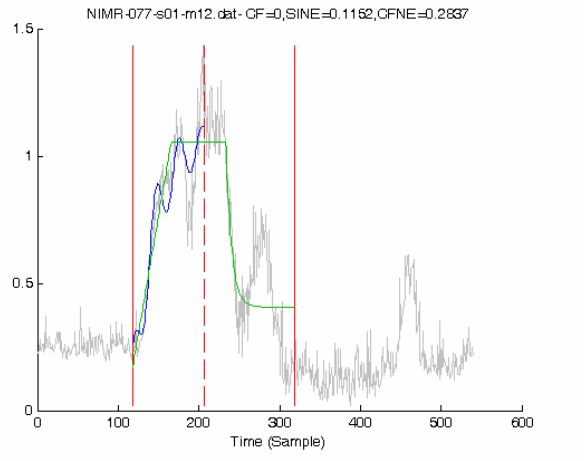

AHS-037-s01-m12.dat - CF=1,SINE=0.0201,CFNE=0.1329
Time (Sample)

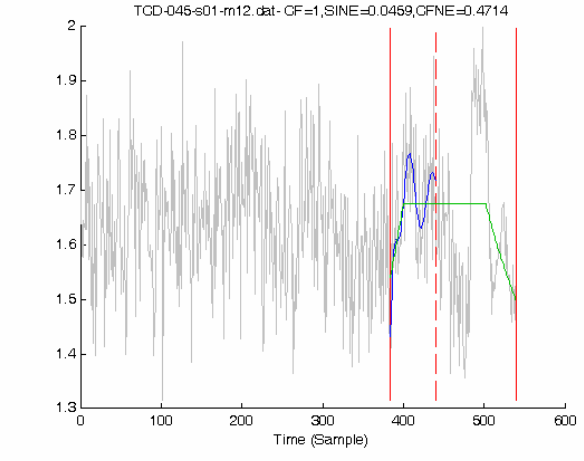

TCD-052-s01-m12.dat - CF=1,SINE=0.0466,CFNE=0.0919
Time (Sample)

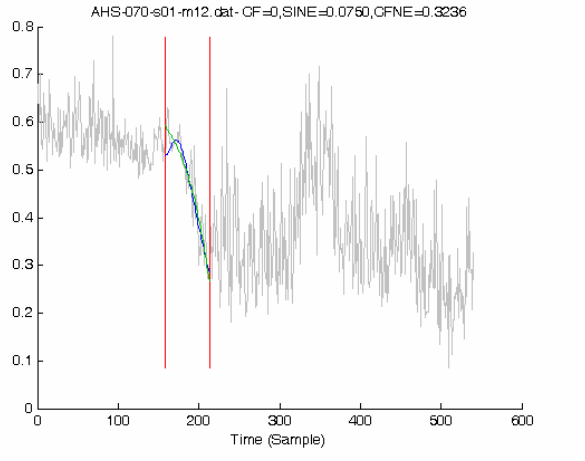

TCD-057-s01-m12.dat - CF=0,SINE=0.0359,CFNE=0.0646
Time (Sample)


TCD-059-s01-m12.dat - CF=0,SINE=0.0654,CFNE=0.1506
Time (Sample)


NIMR-088-s01-m12.dat - CF=1,SINE=0.1034,CFNE=0.0993
Time (Sample)


NIMR-077-s01-m12.dat - CF=0,SINE=0.1152,CFNE=0.2837
Time (Sample)


TCD-045-s01-m12.dat - CF=1,SINE=0.0459,CFNE=0.4714
Time (Sample)


AHS-070-s01-m12.dat - CF=0,SINE=0.0750,CFNE=0.3236
Time (Sample)

A.2 System identification: parameter estimation

The global parameter estimation procedure has to deal with two important issues: the local minima of the loss function (particularly problematic when the number of parameters is high) and uncertainty about the real onset time value. These two issues were tackled respectively by considering several start-up models and several delayed inputs. The adopted parameter estimation procedure is detailed below.

1) Determination of the start-up models.
2) A minimization is performed for each start-up model for 5 different delayed inputs with a maximum number of steps (`Maxiter`) equal to 30 and a tolerance (`Tol`) equal to 0.0001[6]. The delay values are linearly spaced between the estimated onset time and a maximum value depending on the signal maximum position. The model with the smallest loss function (error prediction for the `oe` function) is used as the start-up model for 3).
3) A new minimization is performed with the best model from 2) as an initial guess for several delayed inputs linearly distributed around the best delay that comes out of 2) (`Maxiter=100, Tol=0.0001`).

Several methods were tested to determine the start-up models. The first one was based on the pole-zero representation of $G'(z)$, that is to say the roots of $F(z)$ and $B(z)$ respectively:

$$B(z) = b_1 \prod_{i=1}^{N_{Cb}} (z - z_{cb_i})(z - z^*_{cb_j}) \prod_{j=1}^{N_{Rb}} \left(z - z_{rb_j}\right)$$

$$F(z) = \prod_{i=1}^{N_{Cf}} (z - p_{cf_i})(z - p^*_{cf_j}) \prod_{j=1}^{N_{Rb}} \left(z - p_{rf_j}\right)$$

Where:

- N_{Cb} and N_{Cf} are the number of complex roots of $B(z)$ and $F(z)$ respectively
- N_{Rb} and N_{Rf} are the number of real roots of $B(z)$ and $F(z)$ respectively

Singularities (particularly the poles) give a good indication about various characteristics of the system dynamics (time constant, resonance frequencies,…). Indeed, assuming that all the roots are simple, since $Y(z) = G'(z)U(z)$, it comes:

$$\frac{Y(z)}{z} = \frac{R_0}{1-z} + \sum_{i=1}^{N_{Cf}} \left(\frac{R_{ci}}{z - p_{cf_i}} - \frac{R^*_{ci}}{z - p^*_{cf_i}} \right) + \sum_{j=1}^{N_{Rf}} \left(\frac{R_{ri}}{z - p_{rf_i}} \right)$$

Where the $R.$ are the residues. By taking the inverse of the z-transform $Z^{-1}[\ ]$, we get the following contributions:

[6] The minimization algorithm stops when the norm of the correction vector is less than `Tol` or when the maximum number of iterations (`Maxiter`) is reached.

- $Z^{-1}\left[\frac{R_0 z}{1-z}\right] = R_0 u(k)$ where $u(k)$ is the step function.
- $Z^{-1}\left[\frac{zR_{ci}}{z-p_{cf_i}} - \frac{zR_{ci}^*}{z-p_{cf_i}^*}\right] = 2|R_{ci}|\rho_i^k \cos(k\varphi_i + \alpha_{ci})$ by posing $p_{cf_i} = \rho_i e^{j\varphi_i}$ and $R_{ci} = |R_{ci}| e^{j\alpha_i}$.
- $Z^{-1}\left[\frac{zR_{ri}}{z-p_{rf_i}}\right] = p_{rf_i}^k$

The response to a step function is thus composed of a constant term, damped cosines ($\rho_i < 1$ for stable systems) and decreasing exponentials. Time constants and resonance frequency solely depend on the pole values. Therefore, zero-pole representation easily allows using a priori knowledge about the system dynamics to choose a start-up model. In this work, it will be assumed that rise times are longer than 0.15 second and that effort variation frequency is smaller than 5Hz.

The determination of the start-up models depends on the orders n_b, n_f and previously fitted lower order models will be used. The order of $B(z)$ and $F(z)$ are progressively increased by adding single real poles or pairs of complex poles:

- A real zero/pole is added to $\hat{B}(z)^{n_b-1}/\hat{F}(z)^{n_f-1}$ if the increased order n_b/n_f is odd.
- A pair of conjugate complex poles/zeros to $\hat{B}(z)^{n_b-2}/\hat{F}(z)^{n_f-2}$ if the increased order n_b/n_f is even.

Where $\hat{B}(z)^{n_b-i}$ and $\hat{F}(z)^{n_f-j}$ are the numerator and denominator of a previously fitted lower order $\hat{G}'^{n_b-i}_{n_f-j}(z)$. The start-up values for $G'^{n_b}_{n_f}(z)$ result from the merge of the singularities of $\hat{G}'^{n_b-i}_{n_f-j}(z)$ and the initial positions of the added singularities. The parameters of $\hat{G}'^{n_b-i}_{n_f-j}(z)$ are not frozen and can change during the optimization procedure together with the position of the new singularities. The underlying idea is to progressively refine an initial rough approximation of $G'(z)$ by adding new contributions in order to pick more and more details in the experimental data. Thus, the identification system approach allows dealing with various complex responses simply by increasing the model order. This is an advantage over the curve fitting approach.

The initial singularity positions are chosen according to the a priori knowledge we have about the system dynamics. A particular attention will be paid to the poles (finest initial approximation) because there are expected to have a bigger influence on the response shape than the zeros have.

This method applied with 4 initial values for a couple zeros added to $\hat{B}(z)^{n_b-2}$ and 15 initial values for a couple of poles added to $\hat{F}(z)^{n_f-2}$ (leading to a maximum of 60 start-up models,

less if real singularities are added) turned to be very time consuming (several weeks to extract features from the proof of concept database). Several variants were derived from the approach described above in order to speed up the process without important performance losses.

Although, these attempts were successful, the calculation time required to extract features from the whole proof of concept database remained too important and another method was investigated. It consists in a least square approach using a QR factorization. This approach requires reasonable computation time and losses in modelling performances. It is used in the Matlab built-in initialization of the `oe` function.

In order to assess the goodness of fit for the model that comes out from the system identification $\phi(iT_s,\theta)$ we used the *Normalized Error* $NE(\theta)$:

$$NE(\theta)=\sqrt{\frac{\sum_{i=1}^{N}\left(y_i-\phi(iT_s,\theta)\right)^2}{\sum_{i=1}^{N}y_i^{\,2}}}$$

It is independent of the signal energy and the number of considered samples (thus of the analysis window width) and is expected to allow a consistent comparison between all the patients.

Modelling performances losses between the initialization method based on previously fitted lower order models (described above) and the Matlab built-in initialization function were assessed by averaging the $NE(\theta)$ over a balanced subset of 96 Patients/Normal Control extracted from the proof of concept database for different model orders (with $n_b \le 3$ and $n_f \le 4$ to reduce the computation time). The signal modelled was the norm of the force registered by the thumb sensor during the *drinking a glass* task. The model and optimization parameters were the following: `Maxiter=100, Tol=0.0001, ZStability=1.01`. For the initialization method based on previously fitted lower order models, the parameters values are those mentioned above. The least-square based initialization technique hasn't any parameter. The results for a variable-width analysis window are presented in the Table 16, Table 17.

n_f / n_b	1	2	3
1	0,1302		
2	0,0961	0,0943	
3	0,0942	0,0912	0,0839
4	0,0906	0,0876	0,0744

Table 16: Average $NE(\theta)$ over 96 Patients/Normal Controls with the initialization method based on previously lower order fitted model-Norm of the force-thumb, *drinking a glass* task)-Variable-width analysis window.

n_f / n_b	1	2	3
1	0,110705		
2	0,090106	0,087002	
3	0,083425	0,080085	0,074953
4	0,085173	0,079439	0,073801

Table 17: Average $NE(\theta)$ over 96 Patients/Normal Controls with the least-square based initialization method. (Norm of the force-thumb, *drinking a glass* task)-Variable-width analysis window.

The least-square based initialization significantly outperforms the initialization method based on previously fitted lower order models for all the values of n_b and n_f. Moreover, it is less time consuming. Consequently, this method will always be used.

B Appendix-B: Isometric force/torque measurements

B.1 Description of the Activity of Daily Living tasks

ADL Task Name	Description
Grasping a glass (no reaching)	Arm is placed close to the body, close to the mid line, foot position is standard, finger positions prepared for a cylindrical grasp. ***Object***: glass placed close to the hand
Turning a key	Start position is the same as for grasping the glass. ***Object***: key in a lock located in front of the hand. The key should be oriented horizontally in the lock
Grasping a spoon	Start position is the same as for grasping the glass. Reaching towards the spoon and opening of the hand is measured. The middle finger should also be measured, foot position is standard. ***Object***: spoon is placed a bit higher than the glass and on the side of the back of the hand
Lifting a bag	Start position of the arm: at side of the body, elbow in natural position (slightly flexed), hand, and foot positions are standard, finger positions as normal (cylindrical grasp). ***Object***: a bag placed on the ground
Grasping a bottle	Start position is an almost extended arm over the midline. Start position of the hand is the same as for grasping the glass. Foot position is slided backward, and the back should be leaned forward. ***Object***: bottle placed in front of the hand
Bringing the bottle to the other side	Start positions of the arm, hand, and the foot are the same as for grasping a bottle. ***Object***: bottle placed in front of the affected shoulder at arm reach distance

B.2 Time summary of one isometric force measurement session

Operation	Time [sec]
Patient entry, setting of the ADD to ADL1, start of the Measuring SW	300
Video presentation of ADL1	5
Memorizing	5
Exercising	2
Resting	20
Exercising	2
Resting	20
Exercising	2
Resetting of the ADD to ADL2	30
Video presentation of ADL2	5
Memorizing	5
Exercising	2
Resting	20
Exercising	2
Resting	20
Exercising	2
Resetting of the ADD to ADL3	5
Video presentation of ADL3	5
Memorizing	5
Exercising	2
Resting	20
Exercising	2
Resting	20
Exercising	2
Resetting of the ADD to ADL4	5
Video presentation of ADL4	5
Memorizing	5
Exercising	2
Resting	20
Exercising	2
Resting	20
Exercising	2
Resetting of the ADD to ADL5	180
Video presentation of ADL5	5
Memorizing	5
Exercising	2
Resting	20
Exercising	2
Resting	20
Exercising	2
Resetting of the ADD to ADL6	25
Video presentation of ADL6	5
Memorizing	5
Exercising	2
Resting	20
Exercising	2
Resting	20
Exercising	2
Patient exit	300
Total	**1181**

Reference

[1] *Annex 1, Description of work*, 2005

[2] *D1.1: Methodology for multi centre trial*, Jo Van Vaerenbergh, all partners, April 2004.

[3] *Deliverable D2.2: Preliminary markers and milestones in stroke recovery*, Arteveldehogeschool

[4] *Deliverable D4.1: Data definition for patients, tools for elimination of noisy and software for data pre-processing and description of data mining algorithms*, Scuola Superiore Sant'Anna.

[5] Van Dijck, G. and Van Hulle M. M., "*Hierarchical Feature Subset Selection for Features Computed from the Continuous Wavelet Transform*", In proc. of the IEEE Workshop Machine Learning for Signal Processing (2005), September 2005, Mystic, Connecticut, pp. 81-86.

[6] *ALLADIN FSS report*, Van Dijck, G. and Van Hulle, 2006

[7] Van Dijck, G. and Van Hulle M. M., "*Speeding up the Wrapper Feature Subset Selection in Regression by Mutual Information Relevance and Redundancy Analysis*", submitted for publication to the 16th International Conference on Artificial Neural Networks (ICANN 2006).

[8] *Numerical Recipies in C, The Art of Scientific Computing 2nd edition*, W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Cambridge University Press, 1992

[9] *Levenberg-Marquardt Optimization*, Informal Note, Sam Roweis, 1996

[10] *Precision Grip Force Dynamics: A System Identification Approach*, A. Fagergren, O. Ekeberg and H. Forssberg, IEEE Transactions on Biomedical Engineering, vol. 47, N°. 10, October 2000

[11] *Feedback Regulation of Hand Grasp Opening and Contact During Stimulation of Paralyzed Muscle*, P. E. Crago, R. J. Nakai and H. J. Chizeck. IEEE Transactions on Biomedical Engineering, vol. 38, N°. 1, January 1991

[12] *Estimation of Intrinsic and Reflex Contributions to Muscle Dynamics: A Modeling Study*, E. J. Perreault, P. E. Crago, R. F. Kirsch, IEEE Transactions on Biomedical Engineering, vol. 47, N°. 11, November 2000

[13] *Precise Onset Detection of Human Motor Responses Using a Whitening Filter and the Log-Likelihood-Ratio Test*, G. H. Staude, IEEE Transactions on Biomedical Engineering, vol. 48, N°. 11, November 2001

[14] *Simulating Closed and Open Loop Voluntary Movement: A NonLinear Control-Systems Approach*, P.R. Davidson, R. D. Jones, J. H. Andreae, H. R. Sirisena, IEEE Transactions on Biomedical Engineering, vol. 49, N°. 11, November 2002

[15] *Features based upon curve fitting and System Identification*, Multitel ASBL, WP4 Internal Note.

[16] J. J. Verbeek, N. Vlassis, and B. J. A. Krose. *Efficient greedy learning of Gaussian mixture models*. Neural Computation 15(2):469--485, 2003

[17] Silverman, B. (1986). *Density Estimation for Statistics and Data analysis*. Chapman & Hall.

[18] Mark J. van der Laan, Sandrine Dudoit, and Sunduz Keles (2004) "Asymptotic Optimality of Likelihood-Based Cross-Validation," *Statistical Applications in Genetics and Molecular Biology*: Vol. 3 : Iss. 1, Article 4

[19] L. Couvreur*, Benchmarking automatic speech recognition algorithms: Statistical. issues*, Tech. report, TCTS Lab., Faculte Polytechnique de Mons, 2002.

[20] B. Rohrer, S. Fasoli, H.I. Krebs, R. Hughes, B.Volpe, W.R. Frontera, J. Stein, N. Hogan, *"Movement Smoothness Changes during Stroke Recovery",* The Journal of Neuroscience, September 15, 2002, 22(18):8297–8304.