INSTITUTE OF MECHANICAL ENGINEERING
AUTOMATIC CONTROL LABORATORY

MASTER PROJECT

Modelling and Identification of the Human Balancing System

Student: Raphaëlle Peyraud
Supervisors: Prof. Dr. Moritz Diehl, Dr. Lorenz Assländer, Dr. Alireza Karimi

August 16, 2019

No one trusts a model except the man who wrote it; everyone trusts an observation, except the man who made it.

Harlow Shapley

Declaration

I hereby declare that I am the sole author and composer of my thesis and that no other sources or learning aids, other than those listed, have been used. Furthermore, I declare that I have acknowledged the work of others by providing detailed references of said work. I hereby also declare that my thesis has not been prepared for another examination or assignment, either wholly or excerpts thereof.

Place, Date

Signature

Acknowledgements

I would like to thank Prof. Dr. Moritz Diehl and Dr. Lorenz Assländer my both main supervisors, for giving me the great opportunity to work on this exciting project, halfway between their two fields of expertise. This enriched considerably my knowledge in both domains, and permitted me to combine two of my great interests. I also want to thank Dr. Karimi, for having aroused my interest in system identification and control system theory in general, throughout his lessons, and for having accepted to take the place of my EPFL supervisor.

Individually, I thank Prof. Dr Moritz Diehl for welcoming me in his group in Freiburg, sharing his expertise in the domains of numerical optimization and modelling, and even more, for arising with fruitful ideas. I also would like to warmly thank Dr. Assländer, for sharing and teaching me his knowledge, motivating me, and for always being available to discuss and share ideas.

I gratefully acknowledge the people in the syscop office, for making me feel part of the team, and being kind and friendly with me. I specifically express my gratitude to Jochem De Schutter, who always answered my questions with interest and patience, and for giving me useful hints that significantly contributed to this work. A special thanks also to Andrea Zanelli, for his time, his expertise, and sharing opinions that helped me throughout the realization of this thesis. I will also not forget to express my sincere gratitude to Tommaso, Florian, Tobias, and Matilde, for their support, their help with software, and the constructive discussions. A special thank also goes to Benjamin Michaud, for helping providing a personal and artistic touch to the layout of the written part.

I also want to express to my parents and my sister, my infinite gratitude, for their unconditional support, their wisdom, their advices and their love.

Last, but not least, I want to thank all my friends, who gave me rest, enjoyable moments, and precious help even through the physical distance.

Abstract

Humans maintain stable balance easily in a lot of different environments, and against different external perturbations. Ageing or diseases can affect the control mechanism of the human upright stance, which results in a fail or difficulties of maintaining balance for the affected people. In order to better understand this mechanism, models have been developed, and parameters estimation based on experimental data has been carried out, because they represent the state of the system. Even if effort was put to keep those models simple, they still contain nonlinearities. Until now, the optimization fitting was realized in the frequency domain, but it remains a challenge to implement the models. To overcome this limitation, two provided models (Independent Channels, IC) and (Disturbance Estimation and Compensation, DEC) were formulated in a state-space formulation, and numerical optimization methods were implemented to estimate their parameter values.

The objective to minimize was derived following a maximum-likelihood estimation approach. A Nonlinear Program (NLP) was written, and discretized using the multiple shooting method. Moreover, because of the high dimensionality of the addressed problem, the inherent sampling time of the experimental data was increased by a factor of 25, and a Runge-Kutta method of order four with enough steps to ensure stability of the dynamics, was chosen.

The derived models were validated against the provided ones. The optimization problem converged to an optimum in every considered case. The optimization was run on both averaged and non-averaged data, and different experiment duration. The fitting of the input and output measurements was at least 88.21%. The histogram plots of the optimal residuals showed that some assumptions were not fulfilled. No confidence bounds were provided for the parameter values, but some tendency were noticed. The dynamical reweighting of the sensory weights was observed, and the parameter set for the DEC model was reproducing the sway response across different stimulus amplitudes.

A "clean" formulation in time domain, including the nonlinearities of both models, is available, and the derived methodology permits to broaden the range of possibilities of tests and analyzes. The method still needs to be further developed and fully validated for multiple data sets. The method should be extended to compute the confidence bounds on the identified parameters, and the computational time can be improved by exploiting even more the problem characteristics.

Contents

List of Abbreviations

AD	Algorithmic Differentiation
CAS	Computer Algebra Systems
CNS	Central Nervous System
COM	Center Of Mass
CPU	Central Processing Unit
DEC	Disturbance Estimation Compensation
GL	Gauss-Legendre
IC	Independent Channels
I/O	Input/Output
IVP	Initial Value Problem
KKT	Karush-Kuhn-Tucker
LTI	Linear Time Invariant
MISO	Multi-Input Single-Output
NLP	Nonlinear Program
ODE	Ordinary Differential Equations
PCL	Posture Control Lab
PD	Proportional Derivative
PDF	Probability Density Function
PID	Proportional Integral Derivative
PRTS	Pseudo-Random Ternary Sequence
RK	Runge-Kutta
SISO	Single-Input Single Output
SYSCOP	System Control and Optimization Lab
TFs	Transfer Functions

1 Introduction

How do humans maintain balance? This subconscious task that is trivial for most of us, and does not need apparent efforts, is actually a difficult mission realized by the human sensorimotor control system. Ageing, or specific diseases, can affect this system, and make it fail in maintaining the postural control. For people affected, it means less well-being, less autonomy, and it also increases the health costs. Thus, one needs to identify and quantify the characteristics underlying this complex mechanism.

The central nervous system (CNS) is responsible for interpreting the different sensory inputs acting on the body, and then directs the movements execution. Indeed, when the body posture is perturbed by external disturbances, it reacts by producing a torque helping stabilizing it. The CNS uses the sensory information from three different subsystems to maintain balance during upright stance. Those subsystems are the proprioceptive, the graviceptive, and the visual subsystem. This sensory information contributes to the postural control and is fused by the CNS. It is nowadays still not well known how this "sensory fusion" is realized [26], and many models were developed in order to answer this question. Among others, two models that try to describe how humans maintain balance, two models are investigated. The first one named (IC) model developed by Peterka (2002/2018), and the second one named (DEC) model, developed by Mergner (2006) and Assländer et al. (2015).

In order to investigate the postural control mechanism, stimuli are used as a perturbation of the system. Those perturbations are fully controlled, and can be multiple: (support)-surface tilt, visual tilt, electrical stimulation, push/pull. In this thesis, one will focus on the (support)-surface tilt. To apply the tilt surface perturbation, the patient is placed vertically on a tilting platform (tilt around the ankle joint in the sagittal plane), and its body sway response is then measured thanks to opto-electronic markers placed on him. The Center of Mass (COM), is then computed taking those measurements into account, and is considered as an output value of the system.

One of the most widely implemented postural control model referred to as the "Independent Channels" (IC) model. This model primarily proposed by Peterka (2002) showed that the sway response of the human body to tilts stimuli, can be reproduced by a feedback mechanism [4, 6, 23, 30]. Thus, one was able to use allocated methods to estimate control parameters of the model, such as the different gains, time constants. This IC model, has the particularity to fuse the sensory information arising from the different sensory subsystems, in a linear way. In fact, it weights each sensory channel independently depending on the sensory contribution, availability and reliability of the considered channel, and then, sum all the weighted sensory information together.

This model is able to reproduce body sway response to an external stimuli [23], but is not able to reproduce some nonlinearities observed in the data, without requiring parameters changes, when the stimuli or the environmental conditions change. To overcome these limitations, a new model called "Disturbance Estimation Compensation" (DEC) model is provided and implemented. Indeed, it has already shown good abilities to express experimental data [6, 23, 30]. It is based on the assumption that the body is internally reconstructing the applied physical disturbances by an internal processing mechanism [21], and uses this information to then generate a proper stabilizing torque.

By identifying the parameters of a model, one already noticed that they take different values, if the available source signals change (eyes closed condition, loss of vestibular functions,...etc), and that they can even change for same conditions, between patients showing CNS disorder or not [3, 4, 23, 29]. Thus, looking at those parameters values would be useful and could be used as a diagnostic tool. This could provide useful tools for clinical research, and be used as a diagnostic tool to evaluate treatments and training programs for affected patients.

To estimate the balance control parameters, until now, the models were always represented in a frequency domain by transfer functions (TFs), and then the parameters were obtained from a curve fit of the model's TFs equations to the experimental TFs data [4, 6, 23]. A fitting criterion in the frequency domain was defined, and Matlab[1] functions (fminsearch, fmincom) were used to solve the optimization problem. The models involve some nonlinearities that were difficultly formulated in the frequency domain. To overcome this limitation, a new approach is proposed.

The aim of this thesis, is to reformulate the existing models in a state-space formulation (time domain), to validate them, and then to implement them in an optimization procedure, in order to estimate the models parameters. To do so, the data and the models are provided by the Posture Control Lab (PCL) of the university of Konstanz, and the optimization methods and software used, are provided by the System Control and Optimization Laboratory (SYSCOP) of the university of Freiburg im Breisgau.

As a first step in this thesis, the underlying biological aspects of both models (IC and DEC) are described to give a better understanding of the human balance control mechanism. Both models are provided in the form of block diagrams, and will then be transposed into a state-space formulation. Afterwards, the different methods used to implement them in an optimization framework are detailed, and a Nonlinear Program (NLP) is proposed. Finally, some validation steps of the models are presented, and the results of the parameters estimation are discussed.

[1] The MathWorks, Natick, MA, USA

2 Model description

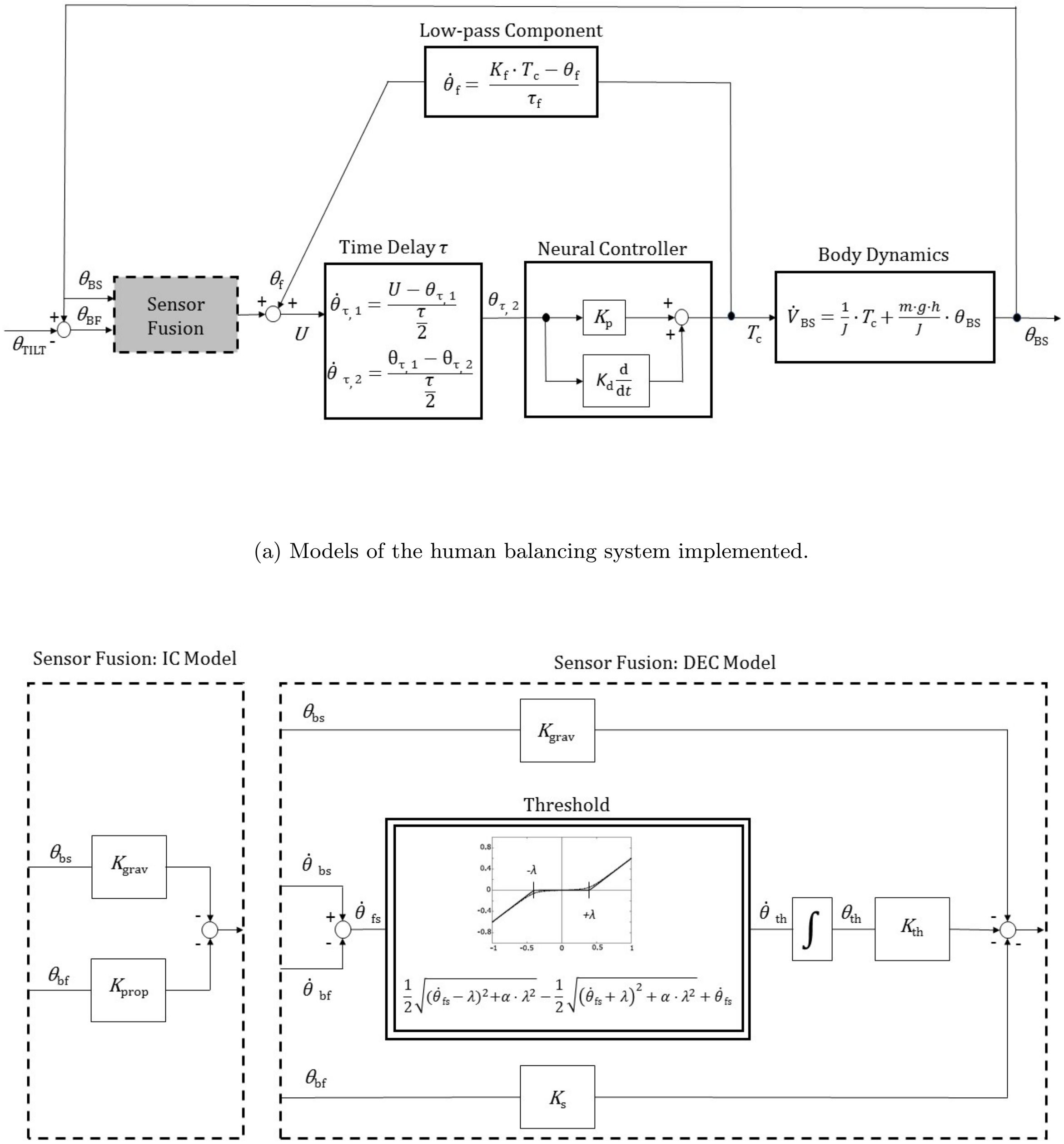
In this section, the two dynamical models of the human balancing system implemented in this thesis, are presented in details. The models are given in the form of a block diagram. Each constituent parts of this diagram are described. The biomechanics background underlying each of those elements is discussed.

The particularity of the block diagram representation, is that it presents two models. The difference between the two models, looking at the figure 2.1, lies in the "Sensory Fusion" subsystems. The first one is called the "Sensory Fusion IC Model", and the second one is called the "Sensor Fusion DEC Model". Those two models are considered here, because the first one (IC model), is a model that is broadly used but quite simple, while the second one (DEC model) contains a more detailed view and is modelling the human stance control using an innovative approach.

2.1 Model Overview

When the human body posture deviates from the vertical, because it is inherently unstable due to its high center of mass, and also because of perturbations (surface tilt, visual environment), it reacts in order to stabilize itself and keep an upright position. It is commonly accepted, that this mechanism of human stance control, can be explained by a feedback mechanism [4, 6, 24, 30]. The body-sway is sensed, and this information is integrated in the sensory system "Sensor Fusion" to generate context dependent motor commands. This sensory fusion mechanism uses three different sources of information to generate a signal, informing on the state in space of the body. Those different source signals used and fused together, are generated by the visual, the graviceptive, and the proprioceptive/somatoceptive subsystems. Thus, this sensory information changes with the environmental conditions and/or with the available source signals (eyes closed, loss of vestibular functions). In order to maintain balance, the body uses this information to generate a corrective torque T_c around the ankle joint axis, to counteract the destabilizing action of the gravity on it. In fact, the "Neural Controller", uses the sensory information to generate this corrective torque. The process encounters a "Time Delay τ", that accounts for various delays present in the whole process.

How the different signals of the sensory subsystems are processed and combined in the "Sensor Fusion" is nowadays still not well known [23]. The most implemented model to describe it, is the "Sensor Fusion IC Model" (**b**) in the figure 2.1, but this one is not able to reproduce and explain all the different observations made until now. The Sensor Fusion DEC Model (**b**) is an alternative model that has already shown better results than the other one, in mimicking the human evoked sway response across stimulus amplitudes [4]. The difference between these two models will be explained in more detail later in this work (subsections 2.4, 2.5). Both of these models are implemented in this thesis.


Low-pass Component
$\dot{\theta}_{\mathrm{f}} = \frac{K_{\mathrm{f}} \cdot T_{\mathrm{c}} - \theta_{\mathrm{f}}}{\tau_{\mathrm{f}}}$
Time Delay τ
$\dot{\theta}_{\tau,1} = \frac{U - \theta_{\tau,1}}{\frac{\tau}{2}}$
$\dot{\theta}_{\tau,2} = \frac{\theta_{\tau,1} - \theta_{\tau,2}}{\frac{\tau}{2}}$
Neural Controller
K_{p}
$K_{\mathrm{d}} \frac{\mathrm{d}}{\mathrm{d}t}$
Body Dynamics
$\dot{V}_{\mathrm{BS}} = \frac{1}{J} \cdot T_{\mathrm{c}} + \frac{m \cdot g \cdot h}{J} \cdot \theta_{\mathrm{BS}}$
Sensor Fusion
θ_{BS}
θ_{BF}
θ_{TILT}
θ_{f}
U
$\theta_{\tau,2}$
T_{c}
θ_{BS}
Sensor Fusion: IC Model
θ_{bs}
K_{grav}
θ_{bf}
K_{prop}
Sensor Fusion: DEC Model
θ_{bs}
K_{grav}
Threshold
$-\lambda$
$+\lambda$
$\dot{\theta}_{\mathrm{bs}}$
$\dot{\theta}_{\mathrm{bf}}$
$\dot{\theta}_{\mathrm{fs}}$
$\frac{1}{2}\sqrt{(\dot{\theta}_{\mathrm{fs}} - \lambda)^2 + \alpha \cdot \lambda^2} - \frac{1}{2}\sqrt{(\dot{\theta}_{\mathrm{fs}} + \lambda)^2 + \alpha \cdot \lambda^2} + \dot{\theta}_{\mathrm{fs}}$
$\dot{\theta}_{\mathrm{th}}$
$\int$
θ_{th}
K_{th}
θ_{bf}
K_{s}

(a) Models of the human balancing system implemented.

(b) Detailed "Sensor Fusion" block, with on the left, the representation of the IC model, and on the right, the representation of the DEC model.

Figure 2.1: (a) Block diagram of the human balancing system model, (b) including the both variants of "Sensor Fusion" subsystems, "Sensor Fusion IC model", and "Sensor Fusion DEC model".

The models input is the (support)-surface tilt angle in anterior-posterior direction θ_{TILT}, with ankle joint axis as the rotation axis [4].

The main output considered is the body center of mass COM sway angle θ_{BS} with respect to the vertical. To obtained this angle, opto-electronic markers were placed on the hip and the shoulder of the patient. Then, using those markers measurements and anthropometric data [32], the center of mass COM position was derived [4]. For this thesis, the data were already pre-processed, and just the COM position, with the surface tilt angle were provided.

2.2 Body Dynamics

The dynamics of a human body can be represented by a multilink inverted pendulum. In the model used in this work, the body dynamics is simplified to a single-link inverted pendulum, with body rotation in the sagittal plane, around the ankle joint axis.

This simplification is in fact motivated, and is still able to explain the body sway response [30]. Firstly, as a model-based approach is applied to estimate the model parameters, complexifying the body dynamics would also increased the number of parameters to be estimated and makes the interpretation of results more difficult. Moreover, a study compared data of patients allowed to stand freely ('free standing') and patients whose balancing was restricted to 1 DoF by tying them to a board [23]. This comparison showed that the difference in results was negligible. It was also showed that for small angle amplitudes, the equations governing a simple inverted pendulum were the same as those modelling more complex dynamics [33]. Secondly, the interest here is the postural orientation in space, and not the inter-segmental coordination. Finally, from a physical point of view, the analysis focuses on the COM motion, and it has been observed that the upper and the lower parts of the body tend to move in phase when the applied input signal frequency is below 1.5 Hz [26], which is the case for the data provided here. Moreover, another study [12] revealed that the contribution of the knee joint is low when a moderate disturbance acts on the system, and that thus just the ankle joint is evolved in the stabilizing strategy.

Moreover, the small angle approximation is used for the equation of motion. Thus the body dynamics becomes linear. In fact, small angle amplitudes are taken as inputs, and for those applied amplitudes, the maximal COM angle measured was of 5.46 °. Additionally, while running the experiments on patients, care is taken to not make them fall, do a step forward/backward, or make arm/knee movements [4].

The body dynamics considered is illustrated in the figure 2.2, where the notation for the different angles that are used, is also introduced. As a convention throughout this work, angle notation in capital letters stands for the physical values of those angles, and in lowercase letters, it stands for the body sensory representation of those physical angles.

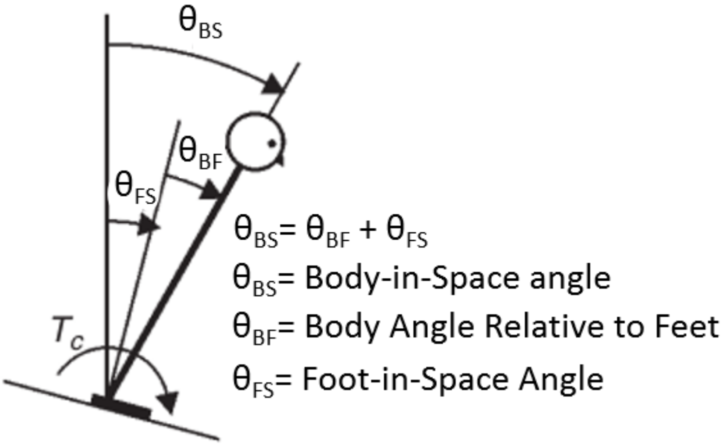

θ_BS
θ_BF
θ_FS
T_c
θ_BS= θ_BF + θ_FS
θ_BS= Body-in-Space angle
θ_BF= Body Angle Relative to Feet
θ_FS= Foot-in-Space Angle

Figure 2.2: Schema of the body in space, simplified as a single inverted pendulum, with the different angles defining its relative position with respect to the horizontal, and the generative corrective torque T_c ©2003 IEEE [23].

2.3 Sensory Fusion

The different body angles present in the model, are estimated by the body through three different sensory subsystems.

The visual subsystem detects the body orientation with respect to the visual environment. The graviceptive subsystem detects the body orientation with respect to the earth vertical. It was noticed that the vestibular system brings a massive contribution to the graviceptive system [23, 30]. The information from those both subsystems (visual and graviceptive) allows to estimate the joint angle θ_{BS} and its derivative $\dot{\theta}_{\mathrm{BS}}$.

The model (figure 2.1) assumes that joint proprioception is used to sense the angle θ_{BF} and its derivative $\dot{\theta}_{\mathrm{BS}}$, which are involved in the human balance control mechanism [4]. Those different sensed signals come from muscles, tendons, skins and structures around joints [19]. Indeed, the proprioceptive subsystem detects the body tilt relative to the support surface. It gives a sense of position and movement of the body with respect to the feet, θ_{BF}=θ_{BS}-θ_{FS}, where θ_{FS} is the surface tilt.

The vestibular system, is a sensory organ providing a sense of balance and spatial orientation. It is able to derive the head movements without the need of a reference. It is thus similar to an IMU (Inertial Measurement Unit). This system is fusing different sensory signals into an electric signal that is then sent to the neural system. The vestibular system is located in the inner ear. It is mainly composed of three semi-circular canals and two cavities. These three semi-circular canals are oriented in the three space direction, and are orthogonal to each other. They are full of an endolymph fluid. Then, when the head moves, so the fluid, and its inertia

applies a pressure and make cilia that are located on the surface of the vestibular cells, move. This is caught by neural receptors, and transformed into a neural signal. This allows to detect the rotation and the angular accelerations. The two cavities (otolithic organs), the saccule (vertical) and the utricle (horizontal), "sense" the gravity and detect the linear accelerations, also thanks to cilia movements. They permit to know the absolute head orientation with respect to the vertical. These ones allow body to have the notion of up and down [5].
The goal of the sensorymotor unit, is the integration of the different sensory signals, in order to produce a sufficient amount of corrective torque, to ensure the body stability [25]. The use of each sensory cue is represented by a sensory weight K_{i}. It has been shown and accepted that a dynamic reweighting is used in the balance postural control mechanism [3, 23, 25]. This reweighting informs on which sensory information is taken into account to maintain balance. This process is influenced and depends on the environmental changing conditions, and on the available sensory information [25]. It has been shown that the sensory channels weights show large systematic changes depending on the environment, and the applied stimulus [25]. Then, it can be stated that the nervous system regulates the sensory orientation information needed for the stance control, depending on the environmental, biomechanical, and pathological conditions [6]. This reweighting mechanism allows the body to stabilize without changing all the balance control system dynamic [3].

A simplification that is often made on the different sensory subsystems, is that their own dynamics is not taken into account. So the complex dynamic of each individual sensory receptor and subsystem is not included in the model [4, 6, 23].

2.4 Sensor Fusion IC Model

One possibility to combine the different sensory information, is to fuse them linearly. This very simplified model, was still able to explain experimental results on previous studies [6, 23, 30]. As the stabilization of the balance control mechanism is achieved by the relative contribution of each sensory subsystems, it is thus common to restrict the weights values between 0 and 1 [3, 23, 24]. Then the convention, that in steady-state conditions, the sum of all the sensory weights is equal to one, is applied ($\sum K_{\mathrm{i}} = 1$) [3, 23]. This assumption was verified in multiple works [23, 30]. Indeed, it was observed that the transfer functions do not change much with changing the stimulus amplitude. It was noticed, that when the stimulus amplitude was increasing, the proprioceptive weight K_{prop} was decreasing. But the generated corrective torque was not changing greatly with this decrease of K_{prop}. It was then assumed that there should be another signal responsible to the generation of this torque. This signal, in accordance with their experimental data, was identified as the graviceptive one, thus corresponding weight K_{grav} [23]. This means that when one weight is decreasing, another one is increasing. This leads to the name of this model "Independent Channels", where the idea is that every sensory channel (information) is weighted by its own specific weight K_{i} (gain). This process allows not only the body to maintain balance, but also to ensure a proper dynamic behavior. Indeed, if insufficient corrective torque is produced, the body falls because it cannot encounter the gravity. And in contrary, if this torque is too high, the dynamics exhibit a resonance peak that deteriorates the stance stability [6, 23].

2.5 Sensor Fusion DEC Model

The acronym DEC stands for "Disturbances Estimation and Compensation". In fact, in order to reproduce the human body sway response to external stimulus, this other model of the "Sensor Fusion" has already shown very good abilities to express experimental data [4, 12, 16, 20, 21]. In this model based approach, parameters of disturbance estimators are identified. It is based on the hypothesis that the body reconstructs the external stimulus by internal processing [21] (estimation (**E**)). This estimation is realized by body sensors and sensory integration. Then the mechanism of maintaining balance would use this disturbance estimation to then feed the control feedback loop, rather than directly use the sensory signals. After this estimation process, the compensation (**C**) would be made by the body space kinematics [21]. This model is based on psychophysical work that showed that the perception of the human self-motion is based on estimates of multisensory signals, rather than on the direct sensory signals [18].

This model was able to describe important aspects of the human balancing system [2, 4, 12], and it was also of a remarkable flexibility, when changing the stimulus amplitude for example [21]. It has been noticed, that an amplitude non-linearity is present in the gain curves of the transfer functions between the input surface tilt and the body sway COM angle. Indeed, as the amplitude of the applied stimulus was increasing, the gain was decreasing [16, 20, 23]. This non-linear phenomenon can be seen in the figure 2.3. The DEC model was able to reproduce this non-linearity.

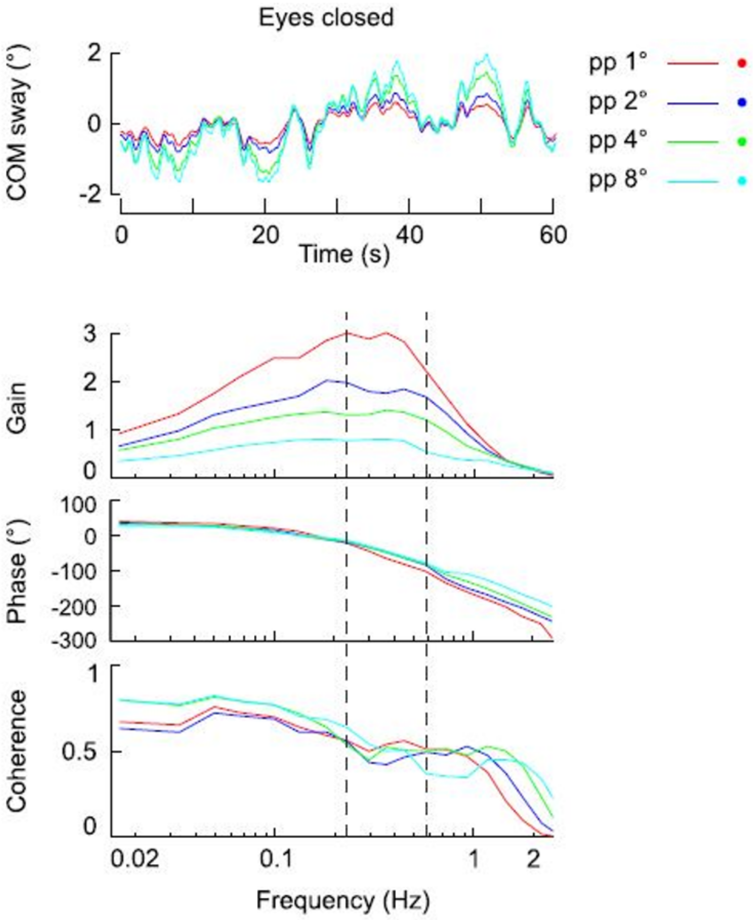


Figure 2.3: Illustration of: (up) stimulus evoked sway for four different (support)-surface tilt amplitudes (color coding) and (down) stimulus evoked sway in terms of Bode diagrams (gain and phase) across frequency. The measurements were averaged across 7 subjects and 10 cycles per subject, and taken with eyes closed condition. ©2015 [4].

The model assumes that the body is able to derive the body in space θ_{BS} angle from the vestibular receptors (by eyes closed conditions), and body sway relative to feet θ_{BF} angle from proprioceptive receptors [16, 18, 20]. It is common to decompose the disturbances that can destabilize the human balancing system in four groups. One can identify the support tilt, the tilt translation, force fields, and force as potential disturbances. In the model implemented in this work, only two of them are considered. As a force field disturbance, one considers the gravity acting on the body. The second one considered, is the support tilt [4]. To build estimators of the support surface tilt and of the gravitational torque, it was hypothesized [20] and then confirmed [2], that the body uses two mechanisms. A velocity mechanism for the tilt compensation, and a position mechanism for the gravity compensation. The vestibular subsystem is involved on the estimation of the θ_{BS} angle, and both vestibular, and proprioceptive velocity cues are involved on the estimation of the support surface tilt.

Using the DEC model, the main effects observed on the amplitude non-linearities, were reproduced by the weights and thresholds present in the model. Indeed, the change of those parameters were sufficient to reproduced the sway response across different tilt amplitudes [4, 16, 20]. It has been hypothesized that there is a relation between the estimator threshold, and the internal noise of the body[4]. In fact, the sensory signals produced by the body are noisy. And moreover, depending on which sensory subsystem is generating the signal, the noise level can vary. An example to see the effect of this internal noise, is the spontaneous sway. This is the fact that the body sways during quiet upright stance when there is an absence of external perturbations. It was shown [30], that this spontaneous sway can be reproduced by internal noise sources. Thus, the action of adding thresholds to the model is beneficial, because it takes also this aspect into account. The action of the threshold, is that below it, no signal, whether contributing to maintaining balance, or noisy, can pass. If the thresholds level is lowered, then lower signals can cross it, and then, as more information is provided, a better compensation of the disturbance is assumed. Because it was reasonably hypothesized that lowering thresholds is related to lower noise levels [4].

The decreasing response gain with the increasing stimulus amplitude or velocity described previously and referred as "amplitude non-linearity", is well caught by the use of thresholds [16, 21]. These thresholds account for the fact that small body deviations are less compensated than larger ones.

In the considered model of this work, the threshold is integrated in the tilt estimator loop (figure 2.1(b)). The foot (support)-in-space angular velocity $\dot{\theta}_{\mathrm{fs}}$ is fed through a threshold, which value should be bigger than the vestibular noise in order to be useful [16]. The $\dot{\theta}_{\mathrm{fs}}$ angular velocity is constructed with the body-in-space and the body-to-feet velocities ($\dot{\theta}_{\mathrm{fs}} = \dot{\theta}_{\mathrm{bs}} - \dot{\theta}_{\mathrm{bf}}$). It is present to integrate the velocity signal, because then the position signal is used to be combined and weighted with other position signals. There are three main effects that make the use of a velocity threshold in the $\dot{\theta}_{\mathrm{fs}}$ pathway relevant :

- The idea that thresholds have a noise reduction function [17]. At rest it blocks the vestibular sensory noise from producing illusory perception [4].
- The fact that the vestibular sensory signal is much more noisy than the proprioceptive one [21, 23].
- The vestibular and the proprioceptive signals are combined together to estimate the angular velocity [18].

Both vestibular (eyes closed conditions) and proprioceptive signals are fused and used together to stabilize the entire body sway response. Indeed, either if the vestibular subsystem is located in the inner ear, and thus in the head, it is not only needed for the head stabilization, but also for the stabilization of the entire body. The vestibular sensory signal plays then a role in the coordination of all body parts [18].

In this project, a smoothed threshold function is used, in order for it to be differentiable. The threshold function used in the model is :

$$y = \frac{1}{2}\sqrt{(x-\lambda)^2 + \alpha \cdot \lambda^2} - \sqrt{(x+\lambda)^2 + \alpha \cdot \lambda^2} + x, \tag{2.1}$$

where a plot example can be seen in the figure 2.4. The α value defines the smoothness of the function, the smaller it is, the more "abrupt" the function will be. And the λ value is defining the range of the threshold.

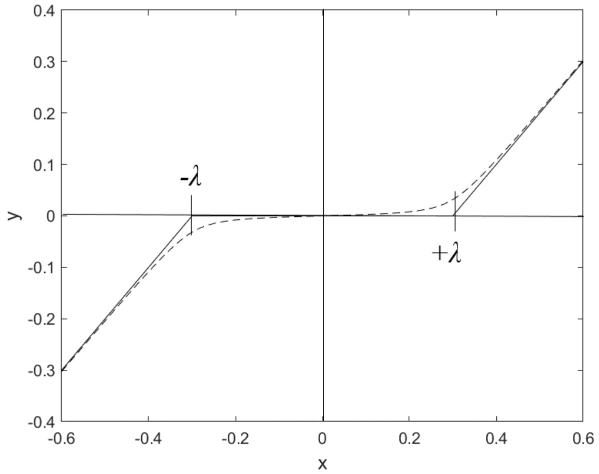


Figure 2.4: Smoothed threshold function with $\lambda = 0.3$ and $\alpha = 0.05$

To have a better representation to what is happening to a signal passing through the threshold function, let us refer to the figure 2.5.

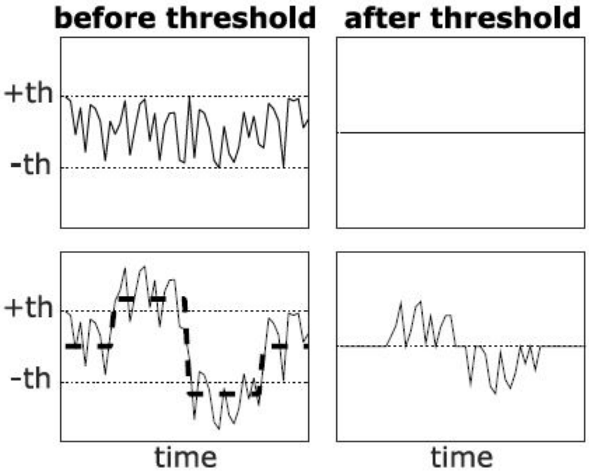


Figure 2.5: Idea of a signal threshold interaction. The signal containing noise is represented by the solid line. The dashed line in the bottom-left represents an underlying *true* signal. The +th/-th indicates the threshold level.

2.6 Neural Controller

The "Neural Controller" block of the model is producing the corrective torque T_c needed to stabilize all the system, using the processed sensory information. In a study, different models were derived with different kind of Neural Controllers [24]. The conclusion was, that a PD (Proportional Derivative) Neural Controller with a positive force feedback, was better expressing the data for low-frequencies, than a PID (Proportional Integral Derivative) Neural Controller. This can be seen in the figure 2.6.

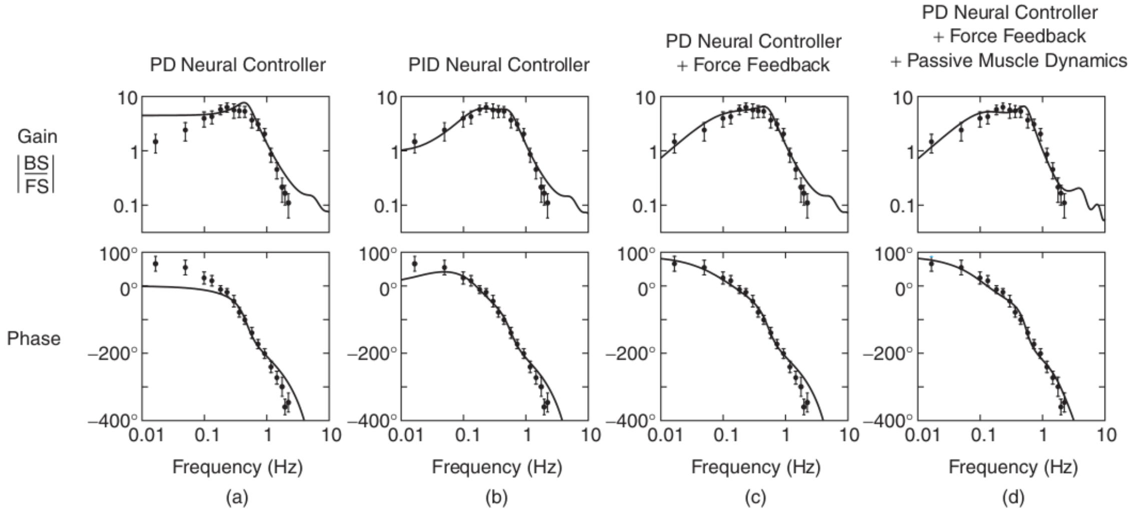


Figure 2.6: Curve fits model equations (solid lines) to experimental transfer functions (points showing mean with 95% confidence limits) obtained from a bilateral vestibular loss subject standing with eyes closed. Posture was perturbed by a pseudorandom rotation of the support surface (1° peak-to-peak amplitude) about an axis aligned with the ankle joint axis. Panels (a)-(d) show different model fits to the same set of experimental gain and phase data ©2003 IEEE [23].

An other study also favors a PD Neural Controller with force feedback, because it is able to account for low-frequencies gain declines and phase advances [6].

In the model considered in this work, a PD controller is implemented. A PD controller is elaborating a signal proportional to its input, and proportional to the derivative of this one. It includes two components, the gain constant K_p producing the proportional part, and the gain constant K_d producing the part proportional to the rate of change of the input. The model equation of a PD controller is given below, where $e(t)$ is the input to the controller at time t, and $u(t)$ is the command produced by the controller at time t.

$$u(t) = K_\mathrm{p} e(t) + K_\mathrm{d} \frac{\mathrm{d}e}{\mathrm{d}t}(t), \tag{2.2}$$

where $K_\mathrm{p} > 0$ and $K_\mathrm{d} > 0$.

2.7 Time Delay and Low-pass Component

Low-pass component

One particularity of this human postural control model, is that the torque feedback is implemented through a positive feedback loop. Even though positive feedback is usually thought to have a destabilizing effect on a control system, here it helps stabilizing, and improves the performance of the postural control mechanism. In fact, a positive torque is necessary to sustain a forward lean, and prevent a backward fall [6, 24]. It helps correct the deviations of the body orientation with the vertical.

This torque feedback loop contains a low-pass component. This component is a first order low-pass filter (also described as a leaky integrator) with a gain constant K_{f} and a time constant τ_{f}. It was previously noticed, that this time constant was pretty long, meaning that low-frequency data are needed to estimate it. This low-pass component is accounting for the low-frequency phase lead and gain declined observed in previous studies [6, 24]. This behaviour can be seen in the figure 2.6. Indeed the estimates showed that this component has only a small contribution to the sensory signal error in comparison to the contribution of the proprioceptive and graviceptive subsystems, that influence it over the entire bandwidth [6].

Time delay

The time delay block "Time Delay τ" combines all the delays of the various subsystems involved in the postural control mechanism. These delays come from the sensory transduction, the neural transmission, the nervous system processing, the muscle activation, and the force development.

This delay was found to be around 150-200 ms [23], which is long in comparison to the stretch reflex. But this value is consistent with the large and increasing phase lag while increasing the frequency, present in the curves of the transfer functions between the input surface tilt and the body sway COM angle, with results from an IC model. This value also suggests that it takes time for the nervous system to extract and combine the sensory information, and then generate a motor command [25].

It was observed that this time delay shows minimal variations to the environmental conditions, but has a strong influence on resonant properties, when too little or too much corrective torque is produced. In fact, when simulating the system with the IC model, for specific conditions (increase of stiffness K_{p}), a resonance peak is present in the transfer function. But in the experimental data, there are no such resonant peak, and it was observed that this increase of stiffness was accompanied by an increase in the overall damping. And this damping can be produced by an adaptation of the time delay [23].

3 Methods

3.1 State-space Formulation

The models, IC or DEC, are written in a state-space formulation. The formulation of a model in a state-space form has many advantages in comparison to a transfer functions representation. Indeed, transfer functions exist only for linear systems, and the DEC model results in non-linear dynamics. Moreover, as transfer functions are I/O models, they provide no description of internal variables, which may be desirable. For example in this work, it may be interesting to get values for the produced corrective torque T_c. Also in case of a MISO (Multi-Input Single-Output) system , a state-space formulation is more convenient to analyze and manipulate. This can be the case here, in case a visual world tilt is also applied, in addition to the surface support tilt.

State-space models are also broadly used with optimization methods. Any continuous dynamical system can be characterized by the following dynamics :

$$\dot{x}(t) = f(x(t), u(t), t) \tag{3.1a}$$

$$y(t) = g(x(t), u(t), t) \tag{3.1b}$$

where $x(t) \in \mathbb{R}^{n_x}$ is the state vector, $u(t) \in \mathbb{R}^{n_u}$ is the input vector, and $y(t) \in \mathbb{R}^{n_y}$ is the output vector. Here, the functions $f(t)$ and $g(t)$ can be either linear or nonlinear. The equation (3.1a) is called the state equation, while (3.1b) is the output equation. And if an initial state $x(t_0)$ with the sequence of control inputs $u(t)$ is provided, one can simulate the system dynamics. This is known as *forward simulation*, and will be explained in more detail in the following subsection.

What is referred to as state of the system $x(t)$, summarizes the effect of the past inputs on the future outputs. It can be seen as memory of the system.

Let us define the *state* of a system.

Definition 3.1. [State] The state of a system at a given time, is the required minimal information, that permits if the inputs are known at the present and past time, to determine completely the behavior of the system at the present and past time.

Thus the state of a system is the information summarizing perfectly the past of the system, because it fixes all the evolution if the inputs are specified.

The state-space formulation is represented by the dynamics of an n_x^{th} order system as a first order differential equation in a vector composed of states. The representation (3.1) is not unique, but if the state-space realization is fixed, so is the representation.

In this subsection, the model equations, both for the IC and the DEC model are derived in detail.

3.2 Sensor Fusion IC Model

Because this model is a Linear Time-Invariant (LTI) system, the state-space formulation (3.1) can be written in the following form:

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{3.2a}$$

$$y(t) = Cx(t), \tag{3.2b}$$

where A, B, C are matrices of proper size.

For this model, the state vector is :

$$x(t) := \begin{bmatrix} \theta_{\mathrm{BS}}(t) \\ V_{\mathrm{BS}}(t) \\ \theta_{\mathrm{f}}(t) \\ \theta_{\tau,1}(t) \\ \theta_{\tau,2}(t) \end{bmatrix}, \tag{3.3}$$

and the input is defined as:

$$u(t) := \theta_{\mathrm{TILT}}(t).$$

The different matrices describing the dynamics are:

$$A := \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \frac{mgh}{J} & 0 & 0 & \frac{2 \cdot K_{\mathrm{d}}}{J \cdot \tau} & \frac{1}{J}\left(K_{\mathrm{p}} - \frac{2 \cdot K_{\mathrm{d}}}{\tau}\right) \\ 0 & 0 & -\frac{1}{\tau_{\mathrm{f}}} & \frac{2 \cdot K_{\mathrm{f}} \cdot K_{\mathrm{d}}}{\tau \cdot \tau_{\mathrm{f}}} & \frac{K_{\mathrm{f}}}{\tau_{\mathrm{f}}}\left(K_{\mathrm{p}} - \frac{2 \cdot K_{\mathrm{d}}}{\tau}\right) \\ 0 & 0 & \frac{2}{\tau} & -\frac{2}{\tau} & 0 \\ 0 & 0 & 0 & \frac{2}{\tau} & -\frac{2}{\tau} \end{bmatrix}, \tag{3.4a}$$

$$B := \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{2 \cdot (1 - K_{\mathrm{grav}})}{\tau} \\ 0 \end{bmatrix}, \tag{3.4b}$$

$$C := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.4c}$$

IC model parameters					
K_{grav}	K_{d}	K_{p}	τ	K_{f}	τ_{f}

Table 3.1: Parameters to be estimated for the IC model.

IC model dimensions		
n_{x}	n_{u}	n_{param}
5	1	6

Table 3.2: Dimensions of the IC model, where n_{x} is the state vector dimension, n_{u} the input dimension, and n_{param} the unknown parameters vector dimension.

The derivation of the model in described into detail in the coming paragraphs.

Body dynamics: The dynamics of a single inverted pendulum is given by

$$J \cdot \ddot{\theta}_{\mathrm{BS}}(t) = T_{\mathrm{c}}(t) + mgh \cdot \sin(\theta_{\mathrm{BS}}(t)) \tag{3.5}$$

where J is the moment of inertia around the ankle joint, m the body mass (not including the feet), g the gravitational accelerations, and h the height of the COM above the ankle joint. T_{c} is the generated corrective torque, and θ_{BS} the measured body-in-space angle. Making the small angle approximation, it becomes

$$J \cdot \ddot{\theta}_{\mathrm{BS}}(t) = T_c(t) + mgh \cdot \theta_{\mathrm{BS}}(t). \tag{3.6}$$

The state-space formulation of the considered body dynamics can be then described by:

$$\dot{\theta}_{\mathrm{BS}}(t) = V_{\mathrm{BS}}(t) \tag{3.7a}$$

$$\dot{V}_{\mathrm{BS}}(t) = \frac{1}{J} \cdot T_{\mathrm{c}}(t) + \frac{mgh}{J} \cdot \theta_{\mathrm{BS}}(t). \tag{3.7b}$$

Low-pass component: The low-pass component introduces a new state θ_{f}, and with the generated torque $T_{\mathrm{c}}(t)$ as an input to it, this feedback loop can be modelled in a state-space form as follows :

$$\dot{\theta}_{\mathrm{f}}(t) = \frac{K_{\mathrm{f}} \cdot T_{\mathrm{c}}(t) - \theta_{\mathrm{f}}(t)}{\tau_{\mathrm{f}}}, \tag{3.8}$$

where K_{f} is the gain and τ_{f} is the time constant of the low-pass component.

Sensor Fusion: Considering the Sensor Fusion block, its upper loop gives:

$$U_1(t) := K_{\mathrm{grav}} \cdot \theta_{\mathrm{BS}}(t), \tag{3.9}$$

and its lower loop gives:

$$U_2(t) := K_{\mathrm{prop}} \cdot (\theta_{\mathrm{BS}}(t) - \theta_{\mathrm{TILT}}(t)). \tag{3.10}$$

Moreover, the (support)-surface tilt stimulus is defined as

$$u(t) := \theta_{\mathrm{TILT}}(t).$$

The parameters K_{prop} and K_{grav} are the gains that represent the weights applied on the proprioceptive and the gravitational sensory channels.

To simplify the notation, an internal variable $U(t)$ is introduced, and represents the result of the summation node before the Time Delay block:

$$U(t) = -(U_1(t) + U_2(t)) + \theta_{\mathrm{f}}(t). \tag{3.11}$$

Time Delay: Taking into account a time delay τ, the internal variable $U(t)$ becomes then

$$U(t-\tau) = -U_1(t-\tau) - U_2(t-\tau) + \theta_{\mathrm{f}}(t-\tau). \tag{3.12}$$

In order to model this time delay, two states are introduced. The time delay will then be approximated by two new states $\theta_{\tau,1}$ and $\theta_{\tau,2}$ [9]. The previous equation becomes then:

$$\dot{\theta}_{\tau,1}(t) = \frac{U(t) - \theta_{\tau,1}(t)}{\frac{\tau}{2}} \tag{3.13}$$

$$\dot{\theta}_{\tau,2}(t) = \frac{\theta_{\tau,1}(t) - \theta_{\tau,2}(t)}{\frac{\tau}{2}} \tag{3.14}$$

where $\theta_{\tau,2}(t)$ is the input to the Neural Controller.

Neural Controller: The generated torque $T_{\mathrm{c}}(t)$ at the output of the Neural controller, taking into account that the input to this controller is $\theta_{\tau,2}(t)$, and that the Neural Controller is approximated by a PD controller, is expressed by:

$$T_{\mathrm{c}}(t) = K_{\mathrm{p}} \cdot \theta_{\tau,2}(t) + K_{\mathrm{d}} \cdot \dot{\theta}_{\tau,2}(t). \tag{3.15}$$

Substituting 3.14 into 3.15

$$T_{\mathrm{c}}(t) = K_{\mathrm{p}} \cdot \theta_{\tau,2}(t) + K_{\mathrm{d}} \cdot \frac{\theta_{\tau,1}(t) - \theta_{\tau,2}(t)}{\frac{\tau}{2}}. \tag{3.16}$$

This expression for the $T_{\mathrm{c}}(t)$, can then be substituted in 3.7b and 3.8.

The parameter K_{prop} does not need to be estimated through the optimization procedure, because the convention on the sum of the sensory weights is used. And as the experiment is realized in eyes closed conditions, it follows: $K_{\mathrm{prop}} = 1 - K_{\mathrm{grav}}$.

3.2.1 Sensor Fusion DEC Model

The DEC model, in comparison to the IC model, is no more linear in states, because of the smoothed function (2.1) approximating the threshold. It is thus represented in the same form as (3.1), where f is a nonlinear function.

For this model, the state vector is:

$$x(t) := \begin{bmatrix} \theta_{\mathrm{BS}}(t) \\ V_{\mathrm{BS}}(t) \\ \theta_{\mathrm{f}}(t) \\ \theta_{\tau,1}(t) \\ \theta_{\tau,2}(t) \\ \theta_{\mathrm{f}}(t) \\ \bar{u}(t) \end{bmatrix}, \tag{3.17}$$

and the input is the same for both model:

$$u(t) := \theta_{\mathrm{TILT}}(t).$$

The state-space formulation of the model dynamics is described by:

$$\dot{x}(t) = \phi^{\mathrm{DEC}}(u(t), x(t), t), \tag{3.18a}$$

$$y(t) = x_1(t). \tag{3.18b}$$

where ϕ^{DEC} is a nonlinear function such that $\phi^{\mathrm{DEC}} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$.

DEC model parameters							
K_{grav}	K_{th}	λ	K_{s}	K_{d}	τ	K_{f}	τ_{f}

Table 3.3: Parameters to be estimated for the DEC model.

DEC model dimensions		
n_{x}	n_{u}	n_{param}
7	1	8

Table 3.4: Dimensions of the DEC model, where n_{x} is the state vector dimension, n_{u} the input dimension, and n_{param} the unknown parameters vector dimension.

The derivation of the model is described into detail in the following paragraphs.

Sensor Fusion: As already explained before, the difference of this model in comparison to the last one, lies in the Sensor Fusion block. This means, that for the state-space formulation,

and with the previous introduced notation, only the information before the time delay node is affected, and thus the states $\theta_{\tau,1}(t)$ and $\theta_{\tau,2}(t)$. The state vector is also augmented by two states, in order to be able to model the threshold function.
This block takes 4 inputs,

$$\begin{cases} \theta_{\text{BS}} \\ \theta_{\text{BF}} = \theta_{\text{BS}} - \theta_{\text{TILT}} = \theta_{\text{BS}} - u \\ \dot{\theta}_{\text{BS}} = V_{\text{BS}} \\ \dot{\theta}_{\text{BF}} = V_{\text{BF}} = V_{\text{BS}} - \dot{u}, \end{cases}$$

and returns one output. The upper loop, representing the estimation of the gravity gives:

$$U_1(t) := K_{\text{grav}} \cdot \theta_{\text{BS}}(t). \tag{3.20}$$

The mid loop that represents the estimation of the tilt, introduces an additional state θ_{th}. The output of the threshold function block is equal to this additional state derivative and is given by:

$$\dot{\theta}_{\text{th}} = \frac{1}{2}\sqrt{(\dot{u} - \lambda)^2 + \alpha \cdot \lambda^2} - \frac{1}{2}\sqrt{(\dot{u} + \lambda)^2 + \alpha \cdot \lambda^2} + \dot{u}. \tag{3.21}$$

Since $\dot{u}(t)$ in (3.21) is not available, it needs to be approximated, and this is realized by adding an additional state $\bar{u}(t)$ to the model,

$$\dot{\bar{u}}(t) = \frac{1}{T}(u(t) - \bar{u}(t)), \tag{3.22}$$

where T is a time variable that needs to be tuned.

Thus, the mid loop produces:

$$U_2(t) := K_{\text{th}} \cdot \theta_{\text{th}}(t). \tag{3.23}$$

The lowest loop often referred as "servo loop" gives:

$$U_3(t) := K_{\text{s}} \cdot (\theta_{\text{BS}}(t) - u(t)). \tag{3.24}$$

Considering this Sensor Fusion block, the internal variable $U(t)$ is then:

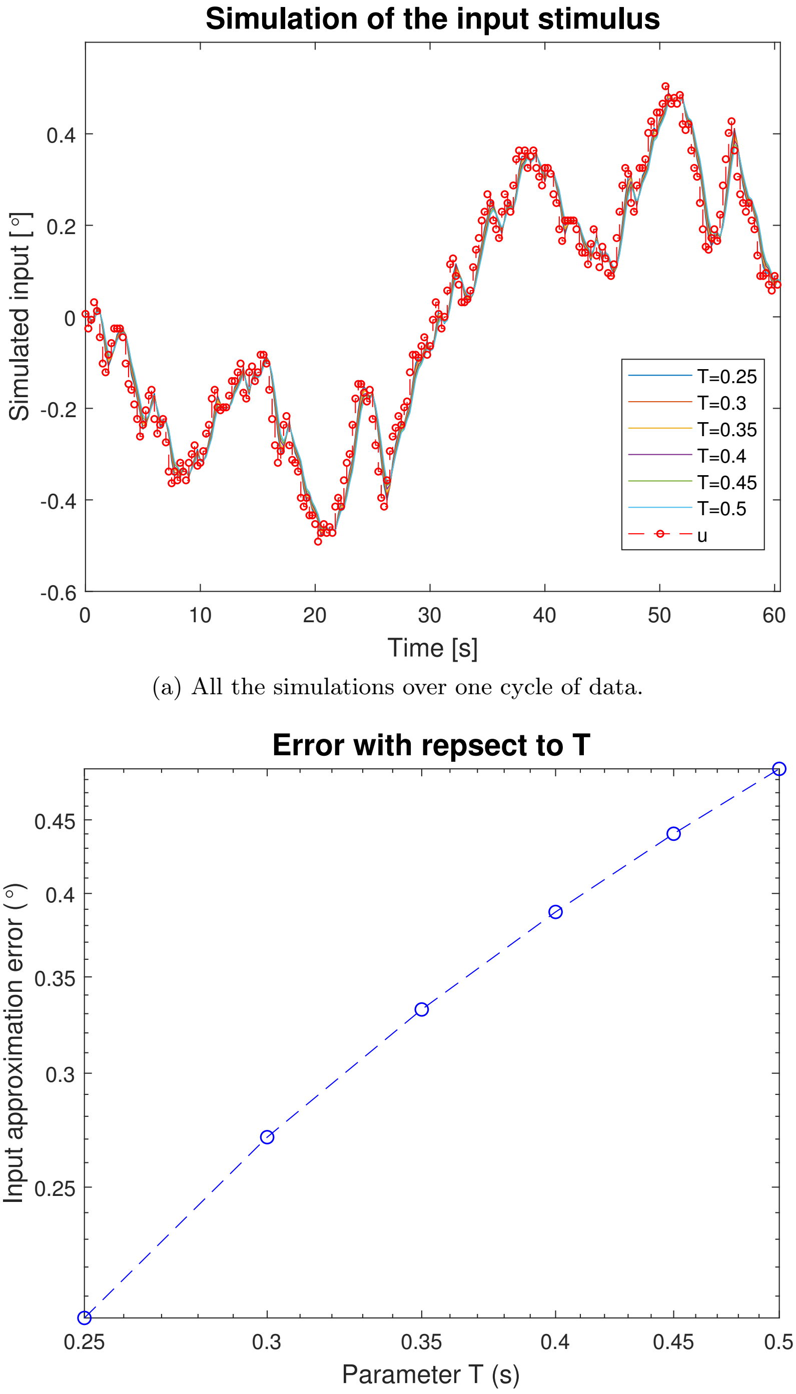
$$U(t) = -(U_1(t) + U_2(t) + U_3(t)) + \theta_f(t). \tag{3.25}$$

Then, by combining previous developed equations, ones obtains the system dynamics. The first two state derivatives $\dot{\theta}_{\text{BS}}$ and $\dot{V}_{\text{BS}}$ are given by (3.7) and (3.16). $\dot{\theta}_{\text{f}}$ is obtained by substituting the torque equation (3.16) into (3.8). The derivative of the first time delay state

$\dot{\theta}_{\tau,1}$ is obtained by substituting (3.25) into (3.13). The second time delay state $\dot{\theta}_{\tau,2}$ is given by (3.14). And the last to states derivatives $\dot{\theta}_{\mathrm{f}}$ and $\dot{\tilde{u}}$ are given by (3.21) and (3.22).

The parameters α and K_{p} need to be fixed. Indeed, the α is modulating the sharpness of the threshold function, and that makes it difficult to estimate, but moreover, on a biological point of view, one is more interested to know the range of the threshold, which is modelled by the λ value. The gain K_{p} needs to be fixed, because otherwise two gain factors would be in serie, and would have to be estimated, since no convention is applied on the sensory weights. Its value is fixed to: $K_{\mathrm{p}} = mgh$ [4].

Choice of the parameter *T*: The parameter T present in the approximation of the input derivative $\dot{u}(t)$ in (3.22), is tuned in order to achieve a good approximation. This parameter must be at least equal or bigger than the sampling time. A forward simulation using a Runge-Kutta of Order Four method (explained in more details in the next subsection), and varying the sampling time in the interval $[t, 2t]$ where $t = 0.25s$ the actual sampling time, is realized, and compared to the measurements u_{k}. The result of this simulation can be seen in the figure 3.1.



(a) All the simulations over one cycle of data.

(b) Two-norm error made on the input approximation.

Figure 3.1: Forward simulation of the input using RK4 with values for the parameter T between $[t, 2t]$ (color coding). The dashed red line being the provided measurements. The simulation uses one cycle of input data with pp°1.

Regarding those results, the forward simulation with the lowest error, is the one using $T = 0.25s$. Thus, this value was kept throughout this work.

3.3 Numerical Simulation

In order to check whether the derived models are able to represent the provided data, and moreover, to validate them, the system dynamics can be simulated. This task is realized through a numerical simulation. In this subsection, the different methods used to simulate systems are presented, with a focus on the ones that were applied on the models.

A numerical simulation, is providing a solution to the *initial value problem* (IVP) [8]. Let us first bring some definitions to better apprehend what does it means.

Definition 3.2. Ordinary Differential Equations (ODE) A controlled dynamic system expressed in continuous time, can be described by an Ordinary Differential Equation (ODE) on the time interval $t \in [t_0, T]$, by:

$$\dot{x}(t) = f(x(t), u(t), t) \quad t \in [t_0, T], \tag{3.26}$$

where t is the time, $u(t) \in \mathbb{R}^{n_u}$ the controls, $x(t) \in \mathbb{R}^{n_x}$ the state, and f a function such that $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times [t_0, T] \to \mathbb{R}^{n_x}$.

Definition 3.3. [Initial Value Problem (IVP)] An ODE like (3.26) with an initial condition $x(t_0) = x_0$ and given controls $u(t)$ is called an *initial value problem.*

The numerical simulation of an ODE is often called *numerical integration*, and can be used to simulate the system dynamics. All numerical integration methods give approximations of the solution of the IVP, $x(t)$. If one denotes this approximation by $\tilde{x}(t)$, the aim of a numerical integration method is to have the smallest error $\mid \tilde{x}(T) - x(T) \mid$ regarding computational effort for the final state at time T.

Thus if one knows the initial state x_0 and the controls u_k for $k = 0, ...N - 1$, one can recursively solve the IVP for the entire time horizon, and obtain the states $x_1, ..., x_N$. This is called a *forward simulation* [8].

Definition 3.4. [Forward simulation] The forward simulation is the map

$$\begin{aligned} f : \mathbb{R}^{n_x + Nn_u} &\to \mathbb{R}^{(N+1)n_x} \\ (x_0; u_0, u_1, ..., u_{N-1}) &\mapsto (x_0, x_1, ..., x_N) \end{aligned}$$

that is defined by solving recursively

$$x_{k+1} = f_k(x_k, u_k), \quad k = 0, 1, ..., N - 1.$$

3.3.1 Runge-Kutta Methods

Runge-Kutta (RK) methods are a class of explicit and implicit integration methods of differential equations [27]. In those methods, assuming that the steps are equidistant, the time horizon is divided into N intervals, each of length h, and the input u is kept constant into each of those intervals. In the case of an explicit scheme, the approximated solution of the IVP for a discrete system is:

$$\tilde{x}(t+h) = \tilde{x}(t) + \phi(t, \tilde{x}(t), u(t), h),$$

where ϕ is a map that approximated the integral $\int_t^{t+h} f(\tau, x(\tau), u(\tau))\, \mathrm{d}\tau$. Here, the function f is evaluated once per time step. But one can also divide the time interval into M subintervals, each of size $h = \Delta/M$, and evaluated the function f also at those intermediate points. The number of evaluations of the function f per step, denoted s, is referred to the *stage* of the method. One then speaks of *s-stage Runge-Kutta* methods. By evaluating the function f in intermediate points, one increases the order of the method used. The order of an integration method is defined as the order (in term of h) of the total accumulated error in the final step. If the computational effort of a method is measured in term of evaluations of the function f, higher stages methods are more costly, but on the over side, to achieve the same accuracy level, one needs less subintervals with an higher order method. The RK methods proceed as follow:

$$\phi = \tilde{x} + h \sum_{i=1}^{s} b_i k_i \tag{3.27a}$$

$$k_i = f(t + c_i h, \tilde{x} + h \sum_{j=1}^{s} a_{ij} k_j) \qquad i = 1, ..., s \tag{3.27b}$$

where s is the number of stages, $k_i \in \mathbb{R}^{n_x}$ are the intermediate quantities, a_{ij}, b_i, and c_i are coefficients. It is common to represent them through a *Butcher tableau*:

$$\begin{array}{c|c} c & A \\ \hline & b^\top \end{array} \qquad = \qquad \begin{array}{c|c|c|c} c_1 & a_{11} & \cdots & a_{1s} \\ \hline \vdots & \cdots & & \vdots \\ \hline c_s & a_{s1} & \cdots & a_{ss} \\ \hline & b_1 & \cdots & b_s \end{array}$$

For explicit integration methods, this tableau will end up to be lower triangular. Below are some well known examples, the simplest integrator, the explicit Euler method, and the explicit method of fourth order, the Rung-Kutta 4-stage, oftenly referred as RK4:

3.3.2 Implicit Runge-Kutta Methods

For some systems, their differential equations cannot be solved by an explicit method, because the integration method is not stable when applied to such systems. This can happen, even if

$$
\begin{array}{c|c}
0 & \\
\hline
 & 1
\end{array}
\qquad\qquad
\begin{array}{c|cccc}
0 & & & & \\
\frac{1}{2} & \frac{1}{2} & & & \\
\frac{1}{2} & 0 & \frac{1}{2} & & \\
1 & 0 & 0 & 1 & \\
\hline
 & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6}
\end{array}
$$

Table 3.5: Butcher tableau: Explicit Euler (left) and RK4 (right).

the ODE of the system is stable. In fact, this is the case when the ODE is extremely stable. Some systems, that exhibit both fast and slow dynamic (modes), required in one hand, a long simulation time, and on the other hand, a very small step size, to guarantee the stability of the integrator. This kind of ODE are called *stiff equations [27].* Those systems are extremely stable, because some of their Jacobian $\frac{\partial f}{\partial x}$ eigenvalues λ, have large negative real part, and some of them have a small magnitude referring to the slow dynamic part. This causes the ODE to converge very fast to zero, but causes the integrator to be unstable and oscillate, if the step size is not small enough. In the figure 3.2, this phenomenon is illustrated.

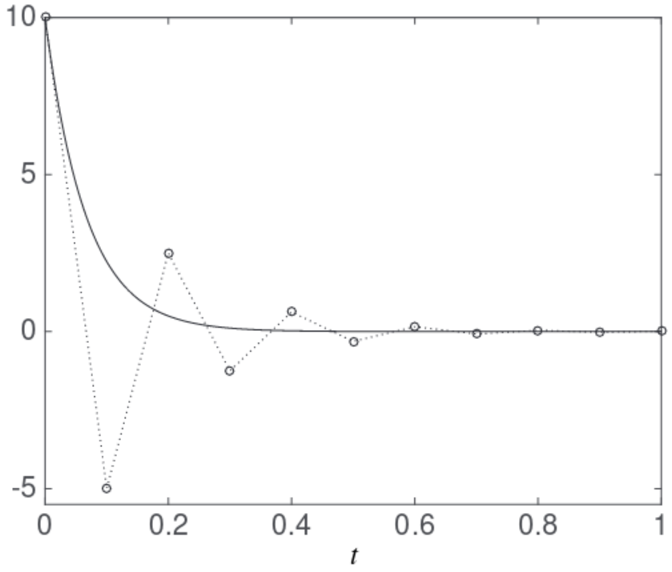


Figure 3.2: Numerical simulation of the system dynamic $\dot{x}(t) = -15x(t)$, using an explicit Euler as integrator, with a step size $\Delta t = 0.1$ and starting with the initial condition $x(0) = 10$. The plain curve represents the exact solution of the ODE, and the dashed one, the numerical simulation, with the circles being the computation points. One remarks that the integrator overshoots the exact solution. Figure taken from [10].

In order to solve such systems of equations, implicit integration methods can be used. The drawback to pay, is that now, to compute the solution of the next step, one needs to solve a nonlinear root-finding problem of size $s \cdot n_x$, increasing the computational complexity. As it can be seen in (3.28)

$$\tilde{x}(t+h) = \tilde{x}(t) + \phi(t, \tilde{x}(t+h), u(t), h), \tag{3.28}$$

the desired value $\tilde{x}(t+h)$ is presented on both side of the equation. Such equations are often solved by a Newton-type method, see next subsection.

3.3.3 Collocation Methods

Collocation methods are a subclass of implicit Runge-Kutta methods. The key feature of those methods, is that they are not providing a discrete, but a continuous approximation of the solution of the IVP. The trajectory is approximated by a polynomial on the collocation interval, that satisfies the ODE at some defined points called collocation points, and denoted by c_i (can be read in the Butcher tableau). Those collocation points are defined on a unit interval and are distinct real numbers such that $0 \leq c_1 < c_2 < ... \leq c_s \leq 1$, where s refers to the number of collocation points in this interval, but also to the *stage* of the method, and thus the degree of the collocation polynomial. The points c_i defined the time points $(t + hc_i)$ at which the polynomial should satisfy the system dynamic [27]. This collocation polynomial $\tilde{x}(\tau)$ for $\tau \in [t, t+h]$ of degree s is satisfying

$$\tilde{x}(t) = x_0 \tag{3.29a}$$

$$\dot{\tilde{x}}(t + c_i h) = f(t + c_i h, \tilde{x}(t + c_i h)), \qquad i = 1, ..., s, \tag{3.29b}$$

and the numerical solution is defined as $\tilde{x}_1 = \tilde{x}(t+h)$. The system defined by (3.29), determinates root-finding problems having (s+1) conditions. Some examples of some well known collocation methods are, the midpoint rule, where $c_1 = \frac{1}{2}$, or the implicit trapezoidal rule, where $c_1 = 0$ and $c_2 = 1$.

The polynomial can be parametrized with the x and the derivative values k_i at the collocation points. Because all collocation points are different, the polynomial is thus well defined, and can be represented by Lagrange basis functions: where $L_i(\sigma) := \prod_{1 \leq j \leq s, j \neq i} \frac{(\sigma - c_j)}{(c_i - c_j)}$ is the Lagrange basis polynomial. Thus $\dot{x}(\tau)$ is approximated by:

$$\tilde{x}(t + c_i h; x, k_1, ..., k_s) = x + \int_t^{t + c_i h} \dot{\tilde{x}}(\tau_1; k_1, ..., k_s) \mathrm{d}\tau_1, \tag{3.30a}$$

$$= x + \sum_{j=1}^{s} k_j h \int_0^{c_i} L_j(\sigma) \mathrm{d}\sigma, \tag{3.30b}$$

and the final state on the interval

$$\tilde{x}(t + h; x, k_1, ..., k_s) = x + \sum_{i=1}^{s} k_i h \int_0^1 L_i(\sigma) \mathrm{d}\sigma. \tag{3.31}$$

The collocation method (3.29), is equivalent to an s-stage Runge-Kutta method (3.27) with:

$$a_{ij} := \int_0^{c_i} L_j(\sigma)\mathrm{d}\sigma \quad b_i := \int_0^1 L_i(\sigma)\mathrm{d}\sigma.$$

Below in the figure 3.3 is an example of the use of a collocation method for a nonlinear and stiff system.

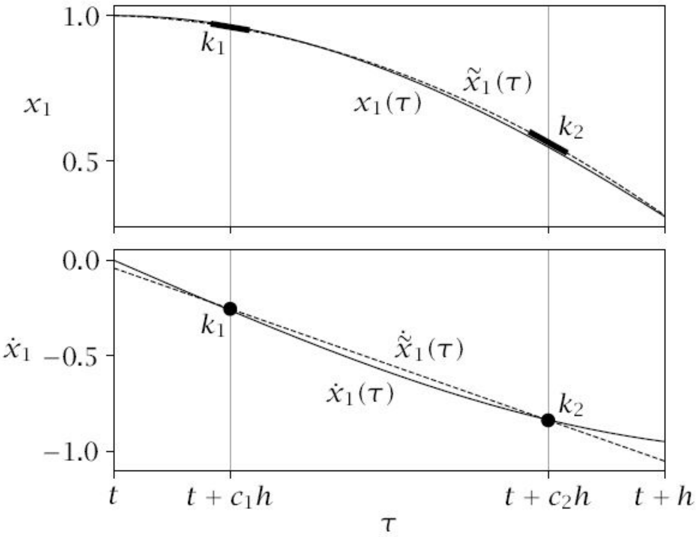


Figure 3.3: Polynomial approximation $\tilde{x}_1(t)$ and true trajectory $x_1(t)$ of the first state and its derivative, computed at the first integration step of a GL4 collocation method applied to the stiff ODE system: $\dot{x} = Ax - 500x(|x|^2 - 1)^2$, with $x_0 = [1, 0]'$ and $A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$. Figure taken from [27].

Because those concepts are used in the following paragraphs, let us define what is a A-stable and a L-stable system.

First, the test problem $\dot{x}(t) = \lambda x(t)$ with initial value $x(0) = x_0$ is considered, where its exact solution is $x = x_0 \exp(\lambda t)$. By using a RK method to solve this problem one gets $x_{n+1} = \phi(h\lambda) \cdot x_n$, where h is the integration time. By induction, one obtains $x_{n+1} = (\phi(h\lambda))^n \cdot x_0$. Then $x_n \to 0$ as $n \to \infty$ is equivalent as $|\phi(h\lambda)| < 1$. Here $\phi(h\lambda)$ is the stability function of the method. For simpler notation, define $z = h\lambda$, and the *region of absolute stability*, which is the set $\{z \in \mathbb{C} \mid |\phi(z) < 1|\}$.

Definition 3.5. [A-stable] An integration method is A-stable, if its region of absolute stability contains the set $\{z \in \mathbb{C} \mid |\mathrm{Re}(z)| < 0\}$ (left half-plane).

Definition 3.6. [L-stable] An integration method is L-stable, if it is A-stable, and if $\phi(z) \to 0$ as $z \to \infty$.

Gauss-Legendre collocation method In order to have a high order collocation method, care should be taken to the choice of the collocation points. Indeed, the higher order that an implicit Runge-Kutta method can achieve is $2s$. This is realized by Gauss-Legendre (GL) methods. Those methods take the collocation points, as the roots of the orthogonal Legendre polynomials on the collocation interval. This is based on the principle of Gauss-quadrature. Those methods have the particularity to be A-stable. For example, the implicit midpoint rule is a Gauss-Legendre method of order two (GL2).

Gauss-Radau IIA collocation method Gauss-Radau methods are fully implicit integration methods that achieve order $2s - 1$ as s being the stage. Those methods include the end point of the interval in the collocation points $c_s = 1$. One of those methods is the Gauss-Radau IIA collocation method, which is very useful to deal with stiff problems, because it is A-stable and L-stable.
The Gauss-Radau IIA method of order three is defined in the Butcher tableau below.

$$\begin{array}{c|cc} \frac{1}{3} & 5/12 & \text{-}1/12 \\ 1 & 3/4 & 1/4 \\ \hline & 3/4 & 1/4 \end{array}$$

Table 3.6: Butcher tableau: Gauss-Radau IIA of order five.

3.4 Nonlinear Optimization

Now that the dynamic models are available, one needs to find the parameters that are able to express as good as possible the given data. In order to compute those parameters values, an optimization problem is set and solved. To do so, a brief introduction on optimization is given, and then, different mathematical and numerical methods are developed, to finally be able to write the needed optimization problem. Afterwards, a description of the software used to formulate and solve the optimization problem is given.

Indeed, sometimes, some problems do not exhibit an analytic solution, and in order to find a solution to them, it is necessary to dispose of proper methods that are able to find a local or global minimum (or maximum) of the problem.

Assume that one wants to find the argument x, that minimizes a function $f : \mathbb{R}^b \to \mathbb{R}$:

$$\arg \min_{x \in \mathbb{R}^b} f(x) \tag{3.32}$$

let us denote the solution to this problem x^*, and in order to be a solution, it needs to fulfill to necessary conditions, that are further specified below [8].

Definition 3.7. [FONC] First Order Necessary Condition: if x^* minimizes (3.32), then $\nabla f(x^*) = 0$.

Definition 3.8. [SONC] Second Order Necessary Condition: if x^* minimizes (3.32), then $\nabla^2 f(x^*) \succeq 0$.

Where ∇ denotes the Jacobian, and ∇^2 the Hessian matrix of $f(x)$, and the inequality symbol $\succeq$ stands for semi-positive definite.

Let us now consider a standard *Nonlinear Program* (NLP), that can be stated as follows :

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g(x) = 0, \\ & h(x) \leq 0, \end{aligned} \tag{3.33}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is the *objective function*, $g : \mathbb{R}^n \to \mathbb{R}^{n_g}$ is the vector of *equality constraints*, and $h : \mathbb{R}^n \to \mathbb{R}^{n_h}$ the vector of *inequality constraints.* All the functions are assumed to be twice continuously differentiable.
The FONC can be reformulate in the form of the well-known Karush-Kuhn-Tucker (KKT) conditions.

Definition 3.9. [KKT point] If x^* is a local minimizer of (3.32), then there exist $\lambda^* \in \mathbb{R}^{n_g}$ and $\mu^* \in \mathbb{R}^{n_h}$ that are called *Lagrange multipliers*, such that the triple (x^*, λ^*, μ^*) is a *KKT point* and thus satisfies the following KKT conditions:

$$\nabla f(x^*) + \nabla g(x^*)\lambda^* + \nabla h(x^*)\mu^* = 0 \tag{3.34a}$$

$$g(x^*) = 0 \tag{3.34b}$$

$$h(x^*) \leq 0 \tag{3.34c}$$

$$\mu_i^* \geq 0 \qquad \forall i = 1, ..., n_h. \tag{3.34d}$$

$$\mu_i^* h_i(x^*) = 0, \qquad \forall i = 1, ..., n_h. \tag{3.34e}$$

Definition 3.10. Lagrange function The Lagrange function is defined as:

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \lambda^\top g(x) + \mu^\top h(x). \tag{3.35}$$

This definition allows one to define the KKT conditions in a short-hand $(3.34a) \Leftrightarrow \nabla_x \mathcal{L}(x^*, \lambda^*, \mu^*) = 0$. And in the case of no inequality constraints, which will be the case in this work, the KKT conditions will then lead to $\nabla_x \mathcal{L}(x, \lambda) = 0$, $g(x) = 0$ [10].

3.4.1 Parameter Identification

As explained before, one of the goal of this work, is to estimate parameters of the models regarding some provided measurements of the system. Thus this task enters in the domain of identification of parametric models. The aim of parameter estimation problem, is to compute an estimator $\hat{\theta} \in \mathbb{R}^d$ of the unknown parameters vector $\theta \in \mathbb{R}^d$, that depends on the measured data. Indeed, in this case, the measured data are the input and output signals. And an estimate of the *true* parameters should be found. The estimator is computed by using an optimization algorithm. An advantage of such methods with compare to nonparametric identification methods, is that the number of parameters to estimate is lower than number of data. But one requirement, is a priori knowledge on the model structure. Another non negligible advantage, is that a stochastic approach can be considered, where the measured data are considered as realization of random variables, and thus it permits to assess the statistical performance of the model structure.

Here, the identification of the model parameters is realized by the minimization of the error between the measured, and the model output. Three main things have to be considered in order to be able to do that [14].

- First, one needs to be provided with a set of experimental data Z^N:

  $$Z^N = \{(y(k), u(k)) \mid k = 1, ..., N\},$$

  where the choice of the input (excitation) signal is of crucial importance for the identification method.

- Secondly, a model structure should be available, and should map the past inputs and outputs into the model outputs. The two models used (IC and DEC), were already described and given previously in this work.

- Finally, a fit criterion has to be minimized. It can be described by:

  $$R(\theta) = \sum_{k=1}^{N} \varepsilon^2(k, \theta), \tag{3.36}$$

  where the function $\varepsilon(k, \theta)$ that is the prediction of the error, can be any functions (linear, nonlinear,...), even if generally, one considers the difference between the measured and the model output data $\varepsilon(k, \theta) = y(k) - \hat{y}(k, \theta)$. The function $R(\theta)$ that maps $\mathbb{R}^d \to \mathbb{R}^N$ is called the *residual function R*.

For the following let us write y_k and u_k instead of $y(k)$ and $u(k)$.

Experimental data For identification methods, because one uses only input and output signals to find a model, or parameters, the choice of the input signal is of great importance. Indeed, the applied input signal is the one that will excite the physical system, and depending on its ability to do it, the quality of the identified model or parameters will be affected.

In order to characterize the "goodness" of the input signal, let us introduce some definitions [14].

Definition 3.11. [Spectral density function or spectrum] The spectral density function of a signal (or spectrum), is defined as the square of the Fourier transform of this signal, and can be written as follow:

$$\varPhi_{\mathrm{uu}}(w) = \left|U(e^{(jw)}\right|^2$$

where the signal considered is $u(k)$, and its Fourier transform $U(e^{jw})$. The spectrum of a signal can also be computed with the use of its autocorrelation function:

$$\Phi_{\mathrm{uu}}(w) = \sum_{h=-\infty}^{\infty} R_{\mathrm{uu}}(h)e^{-jwhT},$$

where T is the sampling period and $R_{\mathrm{uu}}(h)$ the autocorrelation function.

Definition 3.12. [Energy of a signal]

$$\frac{1}{2\pi}\int_0^{2\pi} \Phi_{\mathrm{uu}}(w) = \sum_{u=-\infty}^{\infty} u^2(k)$$

In fact, if a signal has no bounded energy, then its power should be used. It is to say, that it can happen that a sygnal has neither bounded energy, nor bounded power.

Definition 3.13. [Power of a signal]

$$\frac{1}{2\pi}\int_0^{2\pi} \Phi_{\mathrm{uu}}(w) = \lim_{N\to\infty}\frac{1}{2N+1}\sum_{k=-N}^{N} u^2(k)$$

Definition 3.14. [Autocorrelation function] For a deterministic discrete-time energy signal, the autocorrelation function is:

$$R_{\mathrm{uu}}(h) = \lim_{N\to\infty}\sum_{k=-N}^{N} u(k)u(k-h)$$

if the signal has a bounded energy. And in case of a signal with bounded power, its autocorrelation function is :

$$R_{\mathrm{uu}}(h) = \lim_{N\to\infty}\frac{1}{2N+1}\sum_{k=-N}^{N} u(k)u(k-h).$$

As said, the quality of a signal is related to its spectrum. And the characteristics of its spectrum are related to what is called the richness of a signal. In fact, the richer is a signal in a frequency domain, the more it is able to excite a system, and thus better is the quality of the model or the identified parameters in this frequency range. The richness of a signal is giving an information on how much parameters can be estimated by the use of it. As a measure of the richness of a signal, one defines the number of non-zero values of its spectrum $\phi_{\mathrm{uu}(w)}$ between the frequency domain $[0, 2\pi[$, in a normalized frequency scale.

To identify the parameters of the provided models of the PCL, a random signal as input signal is used. Such signals have the ability to excite all frequencies and their amplitude can be controlled. For this experiment, a pseudo-random ternary sequence (PRTS) was used as exciting signal. It is a deterministic signal that shifts between three levels by following specific switching rules. This signal consists of three states $(-v, 0, +v)$ that occur in a pseudo-randomized order. The level v is the velocity that scale with the stimulus amplitude (pp°1, pp°2, pp°4, pp°8). The sequence used contains 242 steps, each of duration 250 μs, which makes the sequence duration equal to 60500 μs. A sequence of data is referred as a cycle in our context.

A PRTS signal is generated by a shift register of length m. The shift register is initialized, and then at each clock pulse Δt, all the content of the shift register is shifted to the right side, and a new value is entering on the left side. This new value results from a modulo-3 addition of values of the register, and is taken as output of the register. Then, the generated sequence is transformed into a velocity sequence with values (-v, 0, +v). A velocity waveform is constructed by holding each velocity value during the clock time Δt $= 0.25s$. Afterwards, this waveform is integrated to obtain a position waveform. And this position waveform has a period of 60.5 s.

This way, a periodic signal is generated, and its maximum period is the number of all the possible combinations $M = 3^m - 1$. The degree of excitation of a PRTS is equal to M. It can be shown that using a PRTS is beneficial in comparison to a sum of sinusoids for example, because one advantage, is that one can fit its amplitude to the maximum allowable amplitude of the system.

The generation of a PRTS signal through a shift register is illustrated in the figure 3.4.

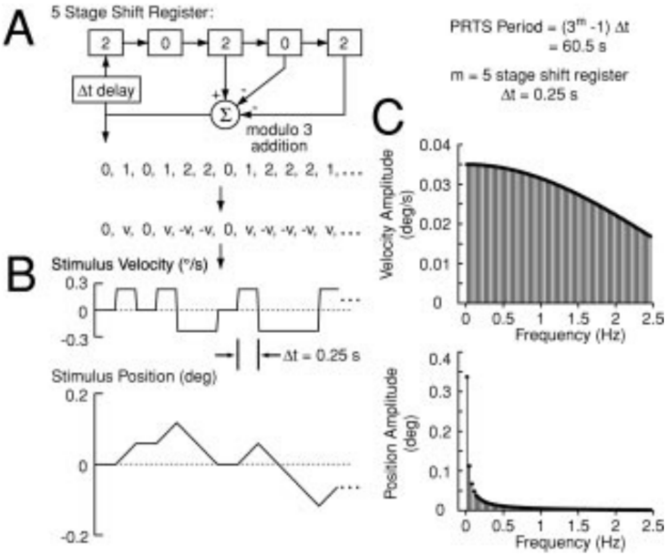
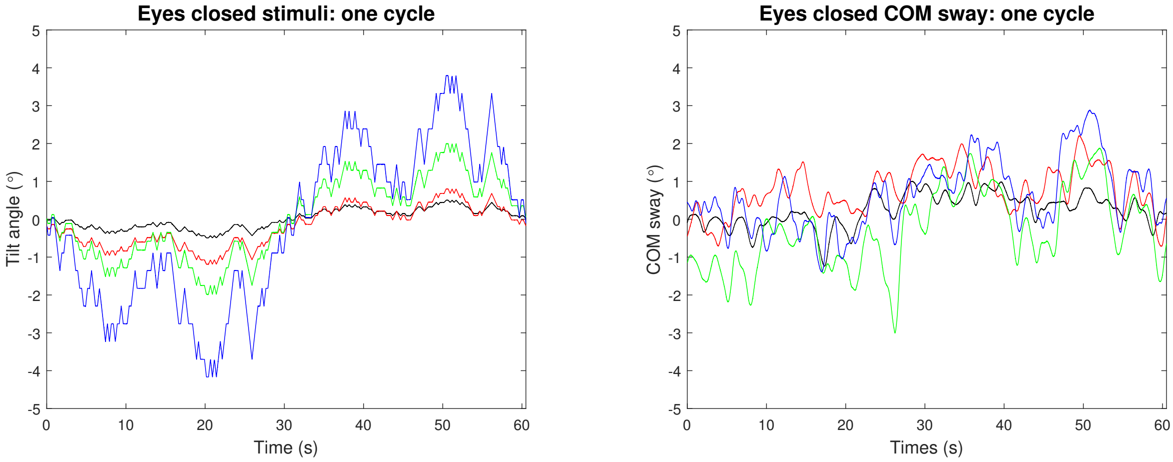


Figure 3.4: Illustration of the different steps involved in the generation of a PRTS stimulus with $\mathit{\Delta}$t = 0.25 s. (A) Shift register used to generate the random sequence. 5 stages are used for the shift register, and the 12 first output values got from the shift register initialization that is displayed, are shown below it. (B) Velocity waveform and position waveform obtained by transforming the value of the shift register and integrating them. (C) Spectral composition of the velocity and position waveforms for a stimulus of pp°1 ©2003 IEEE [23].

The data provided by the PCL to realized the parameters identification, are both input and output signals, for eyes closed conditions. The input is the PRTS tilt stimulus of the platform θ_{TILT}. Inputs of four different experiments, where four different tilt amplitudes peak-to-peak (pp°1, pp°2, pp°4, pp°8), are provided, four a number of 7 patients. And each single experiment is composed of 10 cycles of data, were the first cycle was ignored in the optimization, because it usually contains transient behavior. The output, as it was described in the previous section, is the angle θ_{BS}. In order to have a better idea of these signals, an example is plotted in the figure 3.5.



(a) PRTS tilt stimuli

(b) Stimulus evoked sway

Figure 3.5: Provided data in the time domain for eyes closed conditions and four stimulus amplitudes (color coding), for patient 1 and cycle 4.

Fit criterion A traditional technique that is used to find a model, is the well known least squares estimation method. This method is often and widely used, because it only requires data generation model, and it is simple. The idea underlying it, is to fit the model output to the measurements data, and thus find the parameters of the model, that express the better the data. This fitting process can be defined as the minimization of the sum of the squares of the error, between the measurements, and the predicted output: $\varepsilon(k,\theta) = y(k) - \hat{y}(k,\theta)$. Subsequently, if one makes the assumption on the data, that the measurements are realizations of random variables, then the parameter estimate $\hat{\theta}$ is also a random variable, and its statistics can be described by its probability density function (PDF).

Maximum likelihood estimation: The Maximum likelihood estimation method, that estimates the vector of unknown parameters by maximizing a likelihood function, can be applied when the model dynamic is described by a nonlinear function with respect to the parameters θ, which is the case here [8].

Some definitions are introduced to describe this method.

Definition 3.15. [Likelihood] The likelihood function $\mathcal{L}(\theta)$ describes how likely, in a statistical sense, it is, that the measurements y_k would have been produced by the considered model, with parameters values θ. This is defined by the probability that given θ, one gets $y \in \mathbb{R}^N$ the measurements, $\mathcal{L}(\theta) = p(y \mid \theta)$, where p is the Probability Density Function (PDF) of y given θ.

Definition 3.16. [Maximum-Likelihood Estimate] The maximum-likelihood estimate, is the parameter $\hat{\theta}$ that maximizes the probability of getting the measurements, given a model.

Then, assuming that the measurements produced by the model M are expressed by:

$$y_i = M_i(\theta) + \epsilon_i, \tag{3.37}$$

where $M(\theta)$ can be a nonlinear function, and $\epsilon \in \mathbb{R}^N$ is the noise, and by making the further assumptions, that, ϵ is a Gaussian noise with the expectation $\mathbb{E}\left\{\epsilon\right\} = 0$ and the variance $\mathbb{E}\left\{\epsilon \cdot \epsilon\right\} = \sigma^2$, and ϵ_i, ϵ_j independent for $i \neq j$, then:

$$\begin{aligned} p(y \mid \theta) &= \prod_{i=1}^{N} p(y_i \mid \theta) \\ &= C \prod_{i=1}^{N} \exp \frac{-(y_i - M_i(\theta))^2}{2\sigma_i^2}, \end{aligned} \tag{3.38}$$

with $C = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma_i^2}}$. By taking the logarithm:

$$\log(p \mid \theta) = \log(C) + \sum_{i=1}^{N} -\frac{(y_i - M_i(\theta))^2}{2\sigma_i^2}, \tag{3.39}$$

where C is a constant, and because the logarithmic function is monotone, taking the maximum is equivalent as taking the minimum of the negative expression, then it leads:

$$\begin{aligned} \arg\max_{\theta \in \mathbb{R}^d} p(y \mid \theta) &= \arg\min_{\theta \in \mathbb{R}^d} -\log(p(y \mid \theta)) \\ &= \arg\min_{\theta \in \mathbb{R}^d} \sum_{i=1}^{m} \frac{(y_i - M_i(\theta))^2}{2\sigma_i^2} \\ &= \arg\min_{\theta \in \mathbb{R}^d} \frac{1}{2} \left\| S^{-1}(y - M(\theta)) \right\|_2^2, \end{aligned} \tag{3.40}$$

where S is a matrix that contains all the standard deviations of the measurements. It is not necessary that all its entries are the same. Indeed, each measurement can have an other standard deviation as the others:

$$S = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_N \end{bmatrix}. \tag{3.41}$$

Let us now remind that the models are stochastic, meaning that they are affected by some noise terms $\epsilon(k)$. These noise terms can enter the model at different stages. It is assumed as before, that the noise terms are i.i.d Gaussian noises.

It is known from the PCL, that both input and output provided measurements are affected by noise. Moreover, as derived previously in this work, the models are nonlinear in parameters, justifying the use of a nonlinear least-squares estimation method. Assume that the input noise is described by ϵ^u and the output noise by ϵ^y. The output of the model is then given by [8]:

$$y_k = M(x_0, u_k + \epsilon_k^u, \theta) + \epsilon_k^y, \tag{3.42}$$

where x_0 are the initial conditions, and θ the vector of unknown parameters.

Assume that the input and output variances are given by σ_u^2 and σ_y^2, then the nonlinear least squares problem leads to:

$$\min_{x_0,\theta,\epsilon^u} \sum_{K=1}^{N} \frac{1}{\sigma_y^2}(y_k - M(x_0, u_k + \epsilon_k^u, \theta))^2 + \frac{1}{\sigma_u^2}(\epsilon_k^u)^2. \tag{3.43}$$

By considering $\tilde{u}_k = u_k + \epsilon_k^u$, (3.43) can be expressed by:

$$\min_{x_0,\tilde{u},\theta} \sum_{K=1}^{N} \frac{1}{\sigma_y^2}(y_k - M(x_0, \tilde{u}_k, \theta))^2 + \frac{1}{\sigma_u^2}(u_k - \tilde{u}_k)^2. \tag{3.44}$$

Thus, because the residual R is nonlinear in parameters, one needs a numerical iterative method to find a solution to the nonlinear least squares problem.

3.4.2 Newton-type Methods

Now that the system is modelled, and that the integration was set out, one needs to solve systems of nonlinear equations that are denoted $R(z)$, where R is a function, $R : \mathbb{R}^n \to \mathbb{R}^{n_z}$, $z \mapsto R(z)$. Such nonlinear systems are encountered in the nonlinear least squares problem that needs to be solved, or at each step of the implicit Runge-Kutta method. The methods that solve the problem $R(z) = 0$ are called *Newton's methods* [10].

In order to solve those equations systems, one first assumes that $R(z)$ is continuously differentiable, one starts with an initial guess z_0 and iterates to get the sequence $\{z_k\}_{k=0}^{\infty}$ where the nonlinear equation is linearized at each iterate. Indeed, the nonlinear function is approximated locally by a first order Taylor series, and gives:

$$R(z_k) + \frac{\partial R}{\partial z}(z_k)(z - z_k) = 0, \tag{3.45}$$

where the solution for the next iterate is:

$$z_{k+1} = z_k - J(z_k)^{-1}R(z), \tag{3.46}$$

if the Jacobian $J(z_k) := \frac{\partial R}{\partial z}(z_k)$ is invertible. In practise, the inverse of the Jacobian $J(z_k)^{-1}$ is never computed at each step, but is obtained through a matrix decomposition. It is also in many cases, more convenient to use an approximation of the Jacobian $J(z_k)^{-1}$. Indeed, sometimes J is not invertible, or using an approximation is cheaper in terms of computation or factorization. The problem to solve is then:

$$z_{k+1} = z_k - M_k^{-1} R(z_k), \tag{3.47}$$

where M_k is the approximation of $J(z_k)$, and depending on the choice of M_k, the convergence of the algorithm to the solution, will be fast/slow and converge/not converge. Those kind of methods using an approximation of the Jacobian, are called *Newton-types methods.*

Generalized Gauss-Newton method This method that is a subclass of Newton-type methods, can be applied to solve nonlinear least-squares problems, where the vector of unknown parameters θ is wanted. Let us consider the fitting criterion to minimize $R(\theta)$ (3.36), that is a function $R : \mathbb{R}^d \to \mathbb{R}^N$, where d is the number of unknown parameters, and N the number of measurements. If one uses a Newton-type algorithm, at iteration $k+1$, the vector of parameters is given by the following equation [14]:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - (\nabla^2 R(\hat{\theta}_k))^{-1} \nabla R(\hat{\theta}_k) \tag{3.48}$$

where $\nabla R(\hat{\theta}_k)$ is the transposed Jacobian and $\nabla^2 R(\hat{\theta}_k))$ is the Hessian of the fitting criterion, evaluated at the current estimate $\hat{\theta}_k$. By developing the expression (3.48), one can write:

$$\nabla R(\theta) = \frac{\partial R}{\partial \theta} = -2 \sum_{k=1}^{N} \frac{\partial \hat{y}}{\partial \theta} \epsilon(k, \theta) = -2 \sum_{k=1}^{N} \psi(k, \theta) \epsilon(k, \theta), \tag{3.49}$$

where $\psi(k, \theta)$ is expressed as:

$$\psi(k, \theta) := \frac{\partial \hat{y}}{\partial \theta}. \tag{3.50}$$

The Hessian is then computed as follow:

$$\begin{aligned} \nabla^2 R(\theta) = \frac{\partial^2 R}{\partial \theta \partial \theta^\top} &= 2 \sum_{k=1}^{N} \left[\psi(k, \theta) \psi^\top(k, \theta) - \frac{\partial \psi}{\partial \theta} \epsilon(k, \theta) \right] \\ &\approx 2 \sum_{k=1}^{N} \psi(k, \theta) \psi^\top(k, \theta) = B_{GN}(\theta). \end{aligned} \tag{3.51}$$

Let us now consider that the problem is subjected to some equality constraints, define by the function g [10]:

$$g(k, \hat{\theta}) = 0. \tag{3.52}$$

When this constraint is linearized at iteration $k+1$, the following approximation is obtained:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - \nabla g(k, \hat{\theta})^{-\top} g(k, \hat{\theta}). \tag{3.53}$$

Including this constraint in the problem, and using the previously introduced Lagrangian function, now the exact Hessian is described by:

$$\nabla^2\mathcal{L}(\theta,\lambda) = 2\sum_{k=1}^{N}\left[\psi(k,\theta)\psi^\top(k,\theta) - \frac{\partial\psi}{\partial\theta}\epsilon(k,\theta)\right] + \sum_{k=1}^{N}\lambda_k\nabla^2 g(k,\theta) \tag{3.54a}$$

$$= B_{\mathrm{GN}}(\theta) + \mathcal{O}(\|R(\theta)\|) + \mathcal{O}(\|\lambda\|) \tag{3.54b}$$

Here, in this method, the second and the third terms in the Hessian computation are omitted. The Hessian is thus approximated by first order derivatives. The matrix $B_{\mathrm{GN}}(\theta)$ is the so called *Gauss-Newton Hessian approximation*, and is a positive semidefinite matrix. When another approximation for the Hessian matrix is used, one ends up with an other type of Newton-type method. One famous other type of those methods is the *Newton method* that takes the real Hessian as the matrix $B_{\mathrm{GN}}(\theta)$.

It can be shown, that near the optimal solution, $\|\lambda^*\| = \mathcal{O}(\|R(\theta^*)\|)$ holds. Thus, the difference between the approximated Hessian B_{GN} and the exact Hessian, is $\mathcal{O}(\|R(\theta^*)\|)$. Especially when solving a least-squares problem, the residual function R evaluated at the optimal solution is generally small, making the use of this approximated Hessian reasonable.

3.4.3 Direct Multiple Shooting

The direct multiple shooting method is part of the class of direct optimal control methods. These methods turn the continuous time original problem, into a finite dimensional nonlinear program (NLP). The problem time $[0,T]$ is discretized into N time intervals $[t_k, t_{k+1}] \in [0,T]$ called *shooting intervals*. It is common to use a uniform time grid, so that $t_k = k\frac{T}{N}$. A widespread parametrization for the control variable $u(t)$ is to keep it constant over each shooting interval, and is then denoted $\tilde{u}_k$. The state trajectories are also discretized following the same time intervals. Each interval has, a start state $\tilde{x}_k$ and a control signal $\tilde{u}_k$. Then an integration step of the ODE is realized using a Newton-type method, and provides the state values x_{k+1}, which are the prediction of the state of the system at the end of the interval. This method allows to decouple the numerical integration on each interval. All states can be initialized independently, but here they will be initialized to the same values. But then, the integrator prediction $\phi(\tilde{x}_k, \tilde{u}_k)$ (end state) is not necessarily equal to the next start state $\tilde{x}_{k+1}$. In order to close this gap, and to ensure continuity of the dynamics, dynamic constraints are added to the NLP, at each shooting nodes [27].

$$\tilde{x}_{k+1} = \phi(\tilde{x}_k, \tilde{u}_k) \quad k = 0,..,N-1 \qquad \text{dynamic constrains - gap closing} \tag{3.55}$$

It has to be pointed out, that an integration method is applied to each shooting interval, and the whole parametrization is kept as optimization variables, $\tilde{x}_0, ..., \tilde{x}_N$, $\tilde{u}_0, ..., \tilde{u}_{N-1}$. This increases significantly the originally number of variables, which produces also a large Jacobian matrix in the Newton step. But fortunately, it is sparse and structured, and those features can then be smartly exploited by an adequate solver.

This method is defined as a *simultaneous approach*, in the sense, that it solves both the simulation and the optimization problem in one Newton-type iteration. Another simultaneous method is the *Direct collocation method*, which will be briefly described later. An other type of methods should be mentioned, and is the *sequential approach*. Instead of keeping all the variables as optimization variables, some of them (ex: intermediate state variables) are hidden to the solver but implicitly computed in the optimization. An example of such a method is the *Single shooting* one.

The benefit of using the multiple shooting method, is that any integrator can be used in (3.55). Moreover, the contraction rate of the Newton-type iterations is often faster for this method than for the single shooting for example, in the case of nonlinear and unstable systems.

An illustration of the direct multiple shooting method is provided in the figure 3.6 below, for a better understanding.

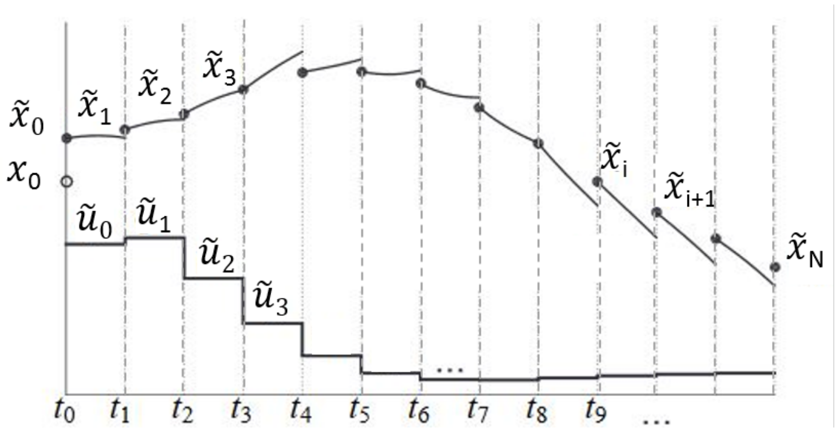


Figure 3.6: Illustration of the direct multiple shooting method, with piecewise constant control input $\tilde{u}_i$, and discrete state $\tilde{x}_i$, on a time grid $t_0, ..., t_N$. The states $\tilde{x}_i$ are computed thanks to a numerical integrator. Figure taken from [10].

Now that all the used methods were described, the solved NLP considered in this work can be written.

NLP formulation

$$\underset{\tilde{x},\tilde{u},\tilde{w},p}{\text{minimize}} \quad \sum_{i=0}^{N-1} \frac{1}{2}\|V\tilde{x}_i - y_i\|^2_{Q_{\tilde{x}}} + \sum_{i=0}^{N-1}\frac{1}{2}\|\tilde{u}_i-u_i\|^2_{Q_{\tilde{u}}} + \sum_{i=0}^{N-1}\frac{1}{2}\tilde{w}_i^\top Q_{\tilde{w}}\tilde{w}_i \tag{3.56a}$$

subject to

$$\{ \; \tilde{x}_{k+1} = M(\tilde{x}_k,\tilde{u}_k,\tilde{w}_k,p) \quad k=0,\dots,N-1 \tag{3.56b}$$

$$* \tag{3.56c}$$

where M is the model dynamics, either IC (3.4) or DEC (3.18), and p the vector of unknown parameters. The matrix V is selecting the first state from the state vector $\tilde{x}$. The vector $\tilde{w}$ represents the process noise. $Q_{\tilde{x}_i}$, $Q_{\tilde{u}_i}$, and $Q_{\tilde{w}_i}$ are the respective inverse covariance matrices, and y_i and u_i are the provided measurements.

*** Bounds**

IC model:

$$\begin{cases} 0 \leq K_{\mathrm{grav}} \leq 1 & (\text{-}) \\ 0 \leq K_{\mathrm{d}} & \left(\frac{\mathrm{deg}}{\mathrm{Nm}}\right) \\ mgh \leq K_{\mathrm{p}} \leq 2 \cdot mgh & \left(\frac{\mathrm{Nm}}{(\mathrm{deg/s})}\right) \\ 0 \leq K_{\mathrm{f}} & \left(\frac{\mathrm{Nm}}{\mathrm{deg}}\right) \\ 0 \leq \tau \leq 60.5 & (\mathrm{s}) \\ 0 \leq \tau_{\mathrm{f}} \leq 60.5 & (\mathrm{s}) \end{cases}$$

$$p = [K_{\mathrm{grav}}, K_{\mathrm{d}}, K_{\mathrm{p}}, K_{\mathrm{f}}, \tau, \tau_{\mathrm{f}}]^{\top}$$
$$V = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

DEC model:

$$\begin{cases} 0 \leq K_{\mathrm{grav}} \leq 1 & (\text{-}) \\ 0 \leq K_{\mathrm{th}} \leq 1 & (\text{-}) \\ 0 \leq \lambda & (\text{-}) \\ 0 \leq K_{\mathrm{s}} & (\text{-}) \\ 0 \leq K_{\mathrm{d}} & \left(\frac{\mathrm{deg}}{\mathrm{Nm}}\right) \\ 0 \leq \tau & (\mathrm{s}) \\ 0 \leq K_{\mathrm{f}} & \left(\frac{\mathrm{Nm}}{\mathrm{deg}}\right) \\ 0 \leq \tau_{\mathrm{f}} & (\mathrm{s}) \end{cases}$$

$$p = [K_{\mathrm{grav}}, K_{\mathrm{th}}, \lambda, K_{\mathrm{s}}, K_{\mathrm{d}}, \tau, K_{\mathrm{f}}, \tau_{\mathrm{f}}]^{\top}$$
$$V = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

3.4.4 Software and Solver

In this subsection, a small description of the software and solver used to solve and formulate the nonlinear optimization problem is given.

CasADi CasADi is an open-source software written in C++ that is used for numerical optimization and optimal control. It is available for C++, Python, MATLAB/Octave. It derives its name, from the fact that at the beginning, it started to be a tool for Algorithmic Differentiation (AD), and is using syntax from Computer Algebra Systems (CAS). Later a support for ODE/DAE integration and sensitivity analysis, nonlinear programming, and interfaces to other tools were added. One should notify, that the AD is realized on user-defined symbolic expressions [1, 27].

An important thing to point out, is that CasADi is not a solver. In fact, it is computing the necessary derivatives (of f, h) and then is passing those information to a Newton-type optimization solver, like IPOPT for example, that will return the optimal solution.

IPOPT IPOPT is an Interior Point Optimizer for large-scale nonlinear optimization, written in C++, that is provided with the standard CasADi installation[31][27]. This made it a good choice to solve (3.58).

This solver is performing an interior point line-search approach, in order to find a local solution to the following nonlinear programming problem:

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g(x_0, x) = 0 \\ & h(x) \leq 0, \end{aligned} \tag{3.59}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function, $g : \mathbb{R}^n \to \mathbb{R}^{n_g}$ the vector of equality constraints, and $h : \mathbb{R}^n \to \mathbb{R}^{n_h}$ the vector of inequality constraints. $x_0 \in \mathbb{R}^n$ is a parameter, and x the variables. It is also assumed that all the functions are at least twice differentiable.

What the algorithm is doing, is that it removes the nonsmoothness of the KKT conditions by formulating smooth root-finding problems. It replaces all the inequality constraints of (3.59) by equality ones, and to do that, introduces slack variables s, that need to be positive $s \geq 0$, and those inequalities are the only one remaining. Those inequalities are then eliminated by adding a barrier term of the form $-\tau \sum_{i=1}^{n_h} \log s_i$, $\tau > 0$ in the objective function.

The NLP can then be reformulated, without inequality constraints, using the interior point (barrier) method:

$$\begin{aligned} \underset{x,s}{\text{minimize}} \quad & f(x) - \tau \sum_{i=1}^{n_h} \log s_i \\ \text{subject to} \quad & g(x_0, x) = 0 \\ & h(x) + s = 0 \end{aligned} \tag{3.60}$$

If the value of the slack variable s is close to 0, then the objective tends to infinity. The optimal solution is then living in the interior of the region defined by $s \geq 0$. It can be shown that the optimal solution of (3.60) converges to the optimal solution of (3.59), when $\tau \to 0$. To solve the problem, the algorithm first fixes τ to a fixed value (0.1), solves the system, and then iterates while decreasing the τ value and using the last solution as initialization for the next one, until a point satisfies the first order optimality condition up to some tolerances. The problem (3.60) can be equivalently formulated in a Lagrangian form, that is solved by a Newton-type method after having fixed τ for the iteration.

In this work, IPOPT is compiled with a linear solver MA57[2], that can be beneficial to speed up the computation. IPOPT uses a line-search approach, and MA57 is responsible for obtaining the search directions for the step computations. Indeed, when one wants to solve the Lagrangian form of (3.60) with a Newton-type method, in order to ensure the Newton's method to converge locally, a line-search method is used. In this method, one needs to find a search direction, in order to ensure that at each iteration, one is getting closer to the solution.

[2] HSL, a collection of Fortran codes for large-scale scientific computation. See http://wwww.hsl.rl.ac.uk/.

4 Results

4.1 Model Validation

One major goal of this work, was to model the human postural control strategy in the time domain, in order to use optimization methods. Before to optimize, one needs to check if the implemented models are coherent. To do so, first both models derived here, where implemented in Simulink[3] and then compared to the Simulink models supplied by the PCL. And finally, as CasADi is used, and especially its provided integrators, the two models are implemented in the CasADi environment, and compared with the ones implemented in Simulink.

To validate the models, the models output θ_{BS} was considered. Some parameters needed to be fixed for the simulation. As the set of data that is provided was already published [4], one took the physiological values from this study, and got strongly influenced by the values found for the estimated parameters. The parameters values used to validate the models are given in the tables below (4.1, 4.2, 4.3).

Moreover, a special care was taken to use the same measurements provided to compare both Simulink models between them, and with the CasADi integrator results. The solver used was the ode5 (Dormand-Prince) with fixed time step provided with Simulink, that is a method of order fifth using explicit functions to compute the current state value and approximated derivatives at intermediate points. In CasADi, the implicit method Radau IIA was applied.

Body mass kg	COM height m	Moment of inertia $\mathrm{kg} \cdot \mathrm{m}^2$	Gravity $\mathrm{m/s}^2$
69.0	0.93	1.27	9.81

Table 4.1: Physiological data

Gain				Time constant	
$\mathrm{K}_{\mathrm{grav}}$ -	K_{f} deg/(Nm)	K_{d} Nm/(deg/s)	K_{p} Nm/deg	τ s	τ_{f} s
0.55	0.10	3.85	13.7	0.16	20.0

Table 4.2: Fixed parameters for the IC model.

4.1.1 Simulink Validation

Because the models provided were implemented in the software Simulink, the derived models of this work were also implemented in this environment, to first validate the models themself.

[3] The MathWorks, Natick, MA, USA

Gain						Time constant		Other	
K_{grav}	K_{th}	K_s	K_f	K_d	K_p	τ	τ_f	λ	α
-	-	-	deg/(Nm)	Nm/(deg/s)	Nm/deg	s	s	deg/s	$(deg/s)^2$
0.55	0.25	1.00	0.10	3.29	10.9	0.16	20.0	0.30	0.05

Table 4.3: Fixed parameters for the DEC model.

IC model To simulate this model, one cycle of stimulus amplitude pp°1 of one patient was used. The two-norm error between the two Simulink simulations is equal to 1.23 °.

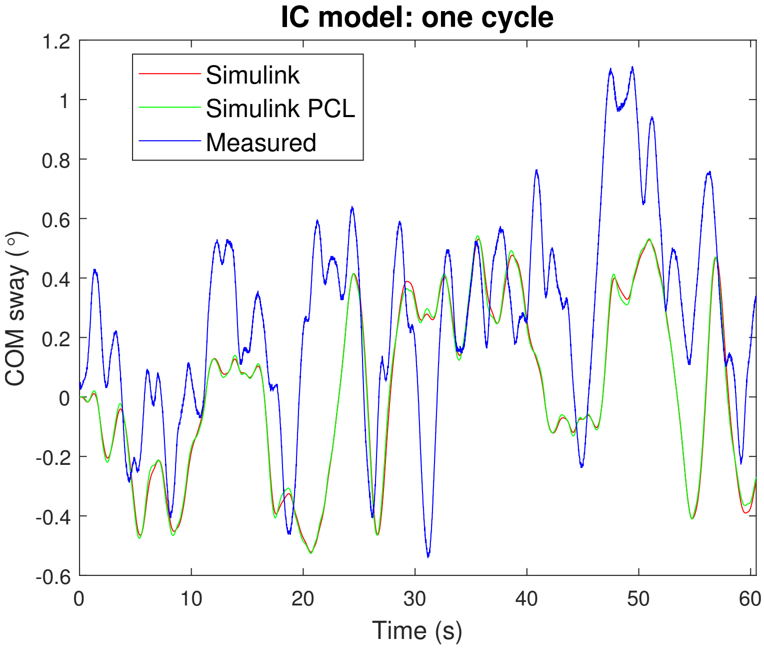


Figure 4.1: Forward simulation of the IC model in Simulink, both with the derived state-space model, and the provided one, using one cycle of data of one patient, with stimulus amplitude pp°1 and given parameters 4.1, 4.2.

DEC model To simulate this model, the data of all the patients and for all the provided cycles for a stimulus amplitude pp°1 were averaged. This was done, because the parameters values used to simulate this system were taken from a previous study [4], were this averaging was done. The two-norm error between the two Simulink simulations was equal to 1.36 °.

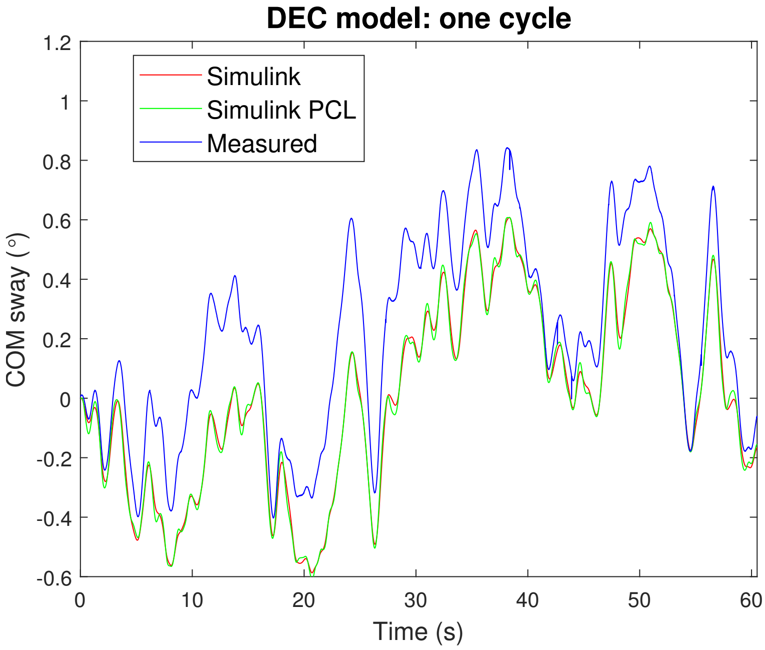


Figure 4.2: Forward simulation of the DEC model in Simulink, both with the derived state-space model, and the provided one. A stimulus amplitude of pp°1 was used, it was averaged over all the patients and all the cycles. The given parameters values 4.1, 4.3 were considered.

4.1.2 Time Delay Approximation

In order to quantify the error made by the approximation of the time delay, the system dynamics was simulated in Simulink using the IC model, and the number of states allocated to the time delay approximation was increased. The two-norm error between the simulated COM body sway using the implemented model and the model of the PCL is plotted in the figure 4.3.

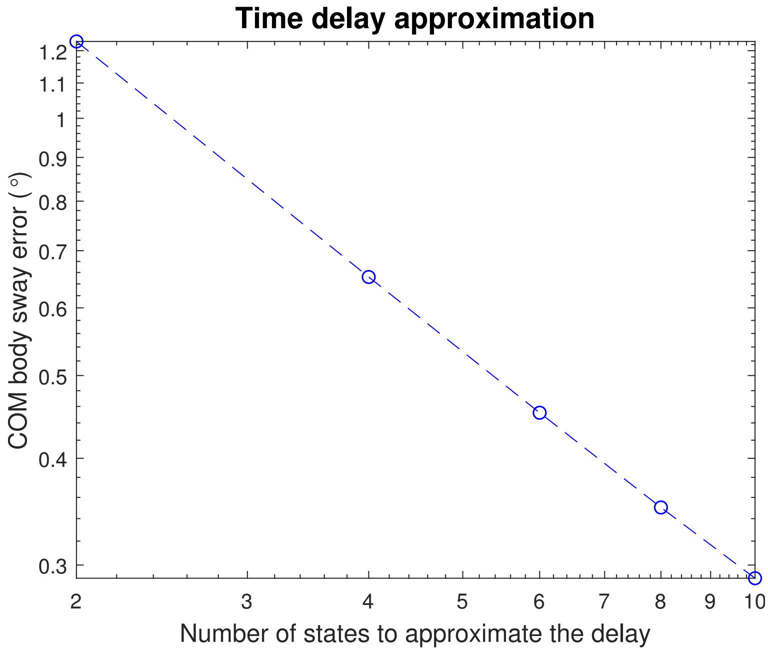


Figure 4.3: Two-norm error made on the COM body sway, while using different number of states to represent the time delay approximation. The simulation was realized using the IC model, with one cycle of data of one patient, with stimulus amplitude pp°1.

4.1.3 CasADi Integrator Validation

CasADi is the software used to implement the optimization problem, and as a multiple shooting method using the provided CasADi integrator is used, one needs to also validate it. As a Radau IIA of order five method is used, the system is simulated using it, and the results are compared with the one obtained with the validated Simulink models.

IC model To simulate this model, one cycle of stimulus amplitude pp°1 of one patient was used. The two-norm error between the simulation using the CasADi integrator, and the Simulink one, was equal to $9.45 \cdot 10^{-2}$.

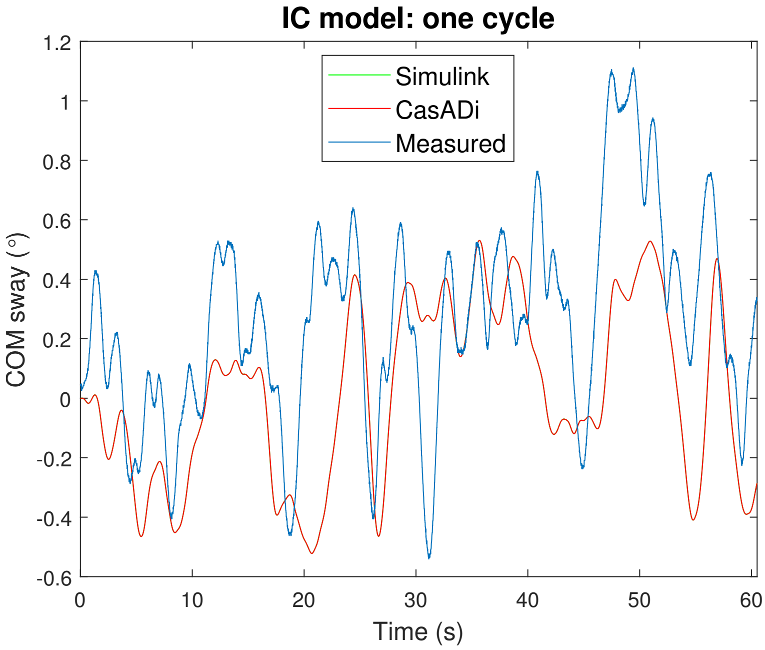


Figure 4.4: Forward simulation of the derived IC model with the Radau IIA integrator of order five of CasADi, and with Simulink with the ode5 solver. A stimulus amplitude of pp°1 was used, and given parameters 4.1, 4.2.

DEC model To simulate this model, the data of all the patients and for all the provided cycles for a stimulus amplitude pp°1 were averaged. The two-norm error between the CasADi integrator and the Simulink one, was equal to $3.85 \cdot 10^{-1}$.

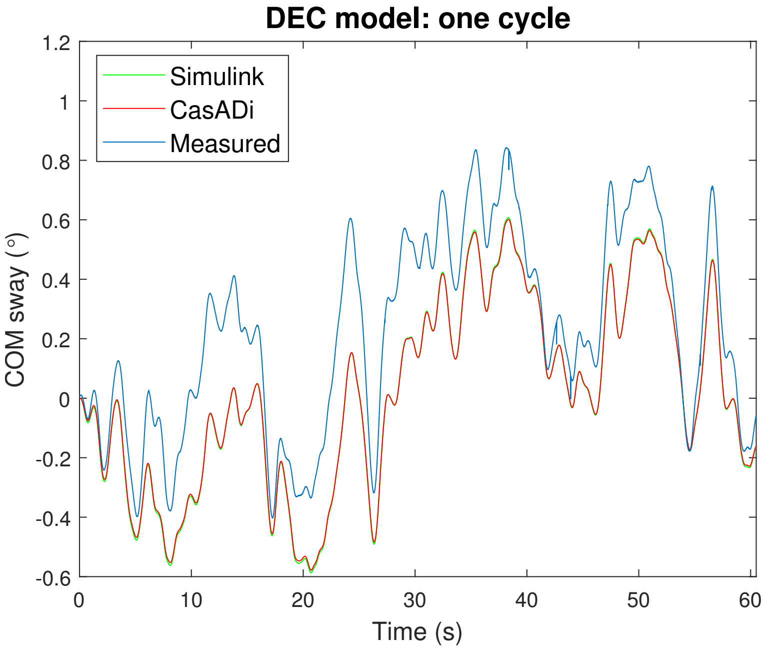
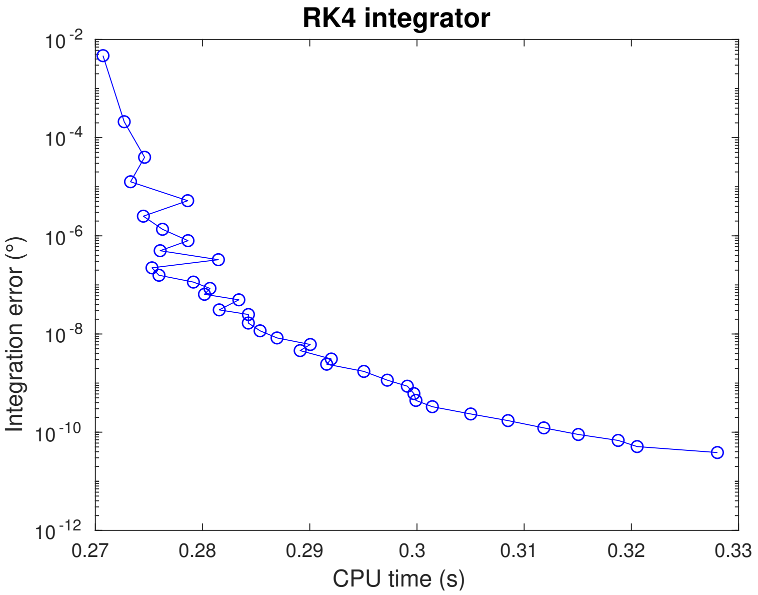


Figure 4.5: Forward simulation of the DEC model with Radau IIA integrator of order five of CasADi, and with Simulink with the ode5 solver. A stimulus amplitude of pp°1 was used, it was averaged over all the patients and all the cycles. The given parameters values 4.1, 4.3 were considered.

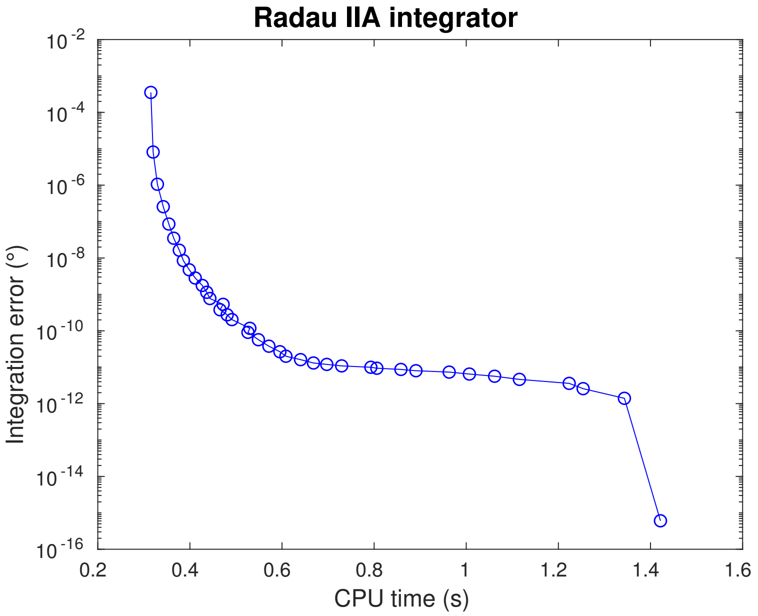
4.1.4 Integration Step

As many time series data are provided and can be analyzed, it would be preferable to try to optimize the computational time as much as possible. Indeed, every provided data cycle contains 242 time points, and there are 10 cycles (9 if discarded the first one because of transient behavior) per stimulus amplitude per patient, which makes 210 cycles in total.

One can try to optimize the computational time passed in the integration step, since the optimization problem to solve is high dimensional, and thus time consuming. The first choice for the integrator, was to use an implicit one, the Radau IIA collocation method of order five. This method is more stable than an explicit one, but is computationally more time consuming. Thus, the idea was to use an explicit integrator, Runge-Kutta of order four (RK4), and to increase the number of finite elements per time step, to make it more stable. Then a test was made on both used integrators, to determine the computational cost with respect to the number of finite elements used. In order to be able to compute an error (criterion) for the result of the integrator, a reference was needed. To compute it, first for the Radau integration method, the number of finite elements used was increased until the error between the last integrator output and the current one, was not decreasing anymore. As an integrator output, θ_{BS} was considered. The found number of finite elements needed for the reference was 100. Then a logarithmic space of 50 values was defined with its upper limit being the reference one. The time needed to realize each forward simulation was recorded, and because it was varying a bit between two realizations, each integration for each finite number of elements was done 10 times, and the result was then averaged. The simulation was realized for both methods using exactly the same set of data, which was one cycle of one patient with stimulus amplitude pp°1. The result is plotted in the figure 4.6.



(a) RK4 integrator with maximal number of finite elements equal to 100.



(b) Radau IIA method of order five with maximal number of finite elements equal to 100.

Figure 4.6: The error versus the Central Processing Unit (CPU) time between integrations of the IC model, varying the number of steps used per integration step is plotted. The integration is realized on one cycle of data with stimulus amplitude pp°1. The integration is computed ten times per each testes number of steps, and the result is then averaged.

It was noticed that to reach an uncertainty of 10^{-8}, for the RK4 integrator, 15 steps whitin one integration step were needed, and for the Radau IIA method, 5 were needed. Even if the Radau IIA method needs fewer steps per integration step to reach the same uncertainty level, it needs more computation time. Indeed, to simulate the system with those number of steps per integration step, the RK4 method needs 0.28s, while the Radau IIA needs 0.35s, which is not a huge difference, but this difference grows significantly if one wants to consider a smaller error.

In the figure 4.6, it was noticed that, the curve behavior was not linear as expected, and it was hypothesized that there should be a minimal computational time for the CasADi integrator that can not be reduced. To investigate that, the RK4 integrator was self-implemented in the CasADi environment, and the computational time was compared. The result obtained with the self-implemented RK4 method can be seen in the figure 4.7.

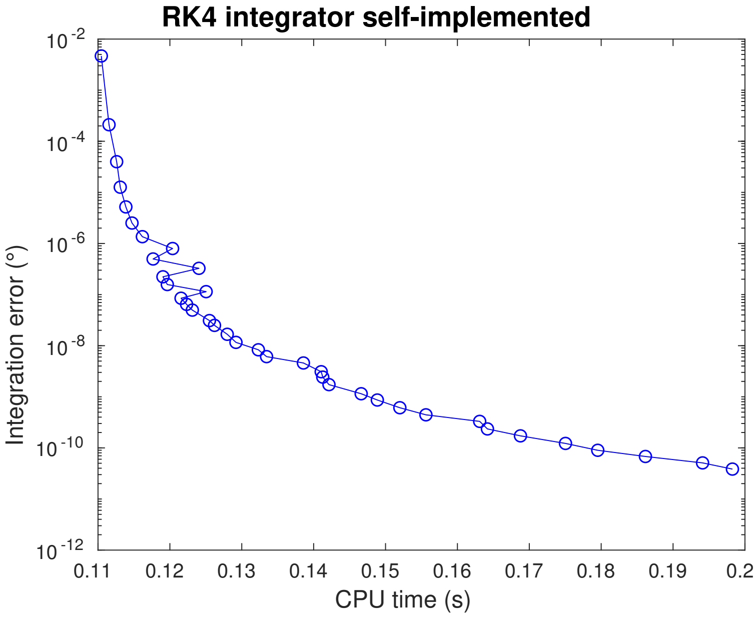


Figure 4.7: The error versus the CPU time between integrations of the IC model with a self-implemented RK4 method, varying the number of finite elements used per integration step is plotted. The integration is realized on one cycle of data with stimulus amplitude pp°1. The integration is computed 10 times per number of finite elements, and the result is then averaged.

First, it was checked that there was no error between the simulation realized with the RK4 integrator provided in CasADi, and the self-implemented one. Then, from the figure 4.7, it can be noticed that to reach an accuracy of 10^{-8}, also 15 steps per integration step are needed, which was expected, since there is no error between the self-implemented and the provided integrator of CasADi. However, there is a huge difference in the computational required time. Indeed, the self-implemented integrator needs only 0.12s against 0.35s for the CasADi one, which makes it tree times faster.

The RK4 method with 15 steps per integration step was then chosen, and implemented in the optimization problem. Optimization procedures were run with both RK4 methods (CasADi and self-implemented). It appears that both methods needed the same number of steps per integration step to converge to the optimal solution, but the one using the CasADi integrator was this time, faster than the other one. Which is exactly the contrary result than the simulation one.

The difference noticed between the simulation and the optimization, can be explained, by the fact that, the native implemented integrator is also propagating sensitivities, that in the case of the forward simulation, is an overhead, but is beneficial and used in the optimization procedure.

4.2 Parameters Estimation

In this section, the achieved results from the optimization procedure are presented. It should be emphasize, that for all the presented results, the optimization converged to an optimum.

The solved NLP was the one described previously by (3.56), using the IC model (3.4), and the DEC model (3.18). The multiple shooting method was applied, with a shooting time interval of 0.25s. The integrator used for the both models, was the native RK4 of CasADi, using a sampling time of 0.25s and 15 steps per integration step.

The used experimental data for the input θ_{TILT} and the output θ_{BS}, were provided and preprocessed by the PCL (especially for the output, the COM described in the models by θ_{BS}) previously. The data represent eyes-closed condition, so that there is just one system input, the (support)-surface tilt stimulus.

For both models, the physiological data given in the table 4.1 where used and fixed. For the DEC model, some parameters were fixed: $K_{\mathrm{p}} = mgh\ \frac{\mathrm{Nm}}{(\mathrm{deg/s})}$, $\alpha = 0.05$, $T = 0.25\mathrm{s}$.

For the initialization of the NLP, the values given in the tables 4.2 and 4.3 were used respectively for the IC, and the DEC model. All the matrices $Q_{\tilde{x}}$, $Q_{\tilde{u}}$, and $Q_{\tilde{w}}$, for both models, were diagonal and of proper size. $Q_{\tilde{x}}$ and $Q_{\tilde{u}}$ having 10 has a value for the diagonal elements, and $Q_{\tilde{w}}$ having 1 as a diagonal elements value.

Model validation: In order to assess the validation of the optimization results in the time-domain, a fit criterion is defined [14]. This is useful in order to compare the different approaches. In this method, the optimized output of the model is compared to the measured output, and their difference is defined as $y(k) - y_m(k, \hat{p})$. The fit criterion is defined as:

$$\text{fit} = 100 \cdot (1 - \frac{\|y - y_m\|}{\|y - \bar{y}\|}), \tag{4.1}$$

where

$$\|y - y_m\| = \sqrt{\frac{1}{N} \sum_{k=1}^{N} [y(k) - y_m(k, \hat{p})]^2}. \tag{4.2}$$

Moreover, because only one set of data is available, and because all the data were used to realize the parameters identification, one is not able to test the prediction power of the model. However, one can still investigate, either on a single set of data, if the assumptions made on the models are valid. To do so, the residuals values $R_i(p^*)$ for the optimal parameters, are plotted for the experiment time horizon. The plot should look like a sequence of random numbers, because of the assumptions made on the models. Then, to assess the statistics of those residuals, they are represented in an histogram plot. Thus, regarding its mean and its variance, one can draw conclusions about the correctness of the models assumptions [8].

The PDF plots of the optimal residuals $R(p^*)$ and the process noise w,were obtained using the function *hist* in Matlab with 150 bars. To fit a Gaussian on these data, the mean and the standard deviation were calculated and used with the function *normpdf* in Matlab which computes the pdf for each given value using a normal distribution defined by the mean and the variance of the data. The x-vector is computed using *ksdensity* also in Matlab which returns 100 points x equally-spaced for an estimation of the probability density. The results are plotted on a logarithmic scale along the Y-axis.

4.2.1 IC model

Averaged data: The optimization of the IC model, was run on every single patient, and every single tilt stimulus amplitude. For each case, the data were averaged over 9 cycles of data for the same patient and stimulus amplitude. One cycle of transient data was considered in each case.

The results for the parameter estimation are given in the following tables. The simulation of the model was then run with the optimized found parameters. The patients in red, are the ones for whom, the simulation of the dynamics using the optimized parameters, was unstable.

	Patient 1	Patient 2	Patient 3	Patient 4	Patient 5	Patient 6	Patient 7
K_{grav}	0.00	0.14	0.00	0.00	0.11	0.00	0.00
K_{d}	3.88	4.57	5.06	10.2	3.96	4.78	5.48
K_{p}	18.2	21.9	15.9	10.9	16.8	13.9	15.1
τ	0.07	0.08	0.17	0.11	0.08	0.07	0.08
K_{f}	0.25	0.28	0.14	0.02	0.33	0.08	0.16
τ_{f}	39.0	60.5	60.5	60.5	60.5	5.19	9.39

Table 4.4: Optimized parameters using a (support)-surface tilt stimulus of amplitude pp°1, with physiological data from the table 4.1 and the initial conditions for the NLP from the table 4.2. For each patient, the data are averaged across 9 cycles. The set of parameters for patients in red, results in an unstable dynamics simulation.

	Patient 1	Patient 2	Patient 3	Patient 4	Patient 5	Patient 6	Patient 7
K_{grav}	0.00	0.27	0.00	0.00	0.00	0.00	0.20
K_{d}	7.28	6.47	6.30	6.72	12.5	8.74	4.85
K_{p}	19.4	19.3	21.9	17.5	11.6	13.2	21.9
τ	0.16	0.71	0.08	0.07	0.82	0.06	0.09
K_{f}	0.36	0.18	0.15	0.22	0.12	0.18	0.23
τ_{f}	60.5	10.0	60.5	7.16	4.53	4.28	60.5

Table 4.5: Optimized parameters using a (support)-surface tilt stimulus of amplitude pp°2, with physiological data from the table 4.1 and the initial conditions for the NLP from the table 4.2. For each patient, the data are averaged across 9 cycles.

	Patient 1	Patient 2	Patient 3	Patient 4	Patient 5	Patient 6	Patient 7
K_{grav}	0.26	0.37	0.25	0.56	0.00	0.00	0.39
K_{d}	5.99	5.39	4.34	6.69	25.6	19.6	6.00
K_{p}	20.1	21.9	21.9	21.9	11.8	12.1	21.9
τ	0.05	0.05	0.05	0.12	0.10	0.05	0.08
K_{f}	0.20	0.24	0.20	0.44	0.18	0.18	0.07
τ_{f}	12.2	19.3	60.5	60.5	7.13	5.96	19.9

Table 4.6: Optimized parameters using a (support)-surface tilt stimulus of amplitude pp°4, with physiological data from the table 4.1 and the initial conditions for the NLP from the table 4.2. For each patient, the data are averaged across 9 cycles. The set of parameters for patients in red, results in an unstable dynamics simulation.

	Patient 1	Patient 2	Patient 3	Patient 4	Patient 5	Patient 6	Patient 7
K_{grav}	0.60	0.68	0.49	0.99	0.61	0.42	0.00
K_{d}	5.03	6.28	5.96	0.00	14.4	20.2	0.00
K_{p}	21.0	19.2	21.9	11.2	10.9	10.9	19.6
τ	0.05	0.06	0.07	0.72	0.07	0.07	55.6
K_{f}	0.14	0.14	0.19	0.00	0.13	0.18	0.00
τ_{f}	17.1	8.85	60.5	14.0	3.98	4.63	37.1

Table 4.7: Optimized parameters using a (support)-surface tilt stimulus of amplitude pp°8, with physiological data from the table 4.1 and the initial conditions for the NLP from the table 4.2. For each patient, the data are averaged across 9 cycles. The set of parameters for patients in red, results in an unstable dynamics simulation.

Because lots of data are available, just the plots for one patient and one amplitude are shown.

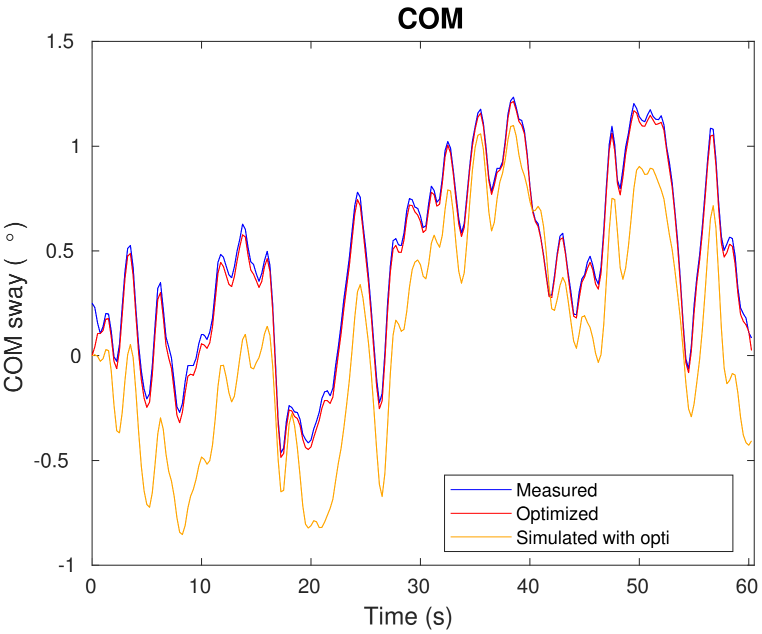


Figure 4.8: COM body sway (output) measurements (blue), optimized (red), and simulated (orange) using the IC model and the optimized parameters. Those results are the ones for the patient 3 using a tilt stimulus of amplitude pp°2. The value of the defined model validation fit criterion is: fit = 91.30%.

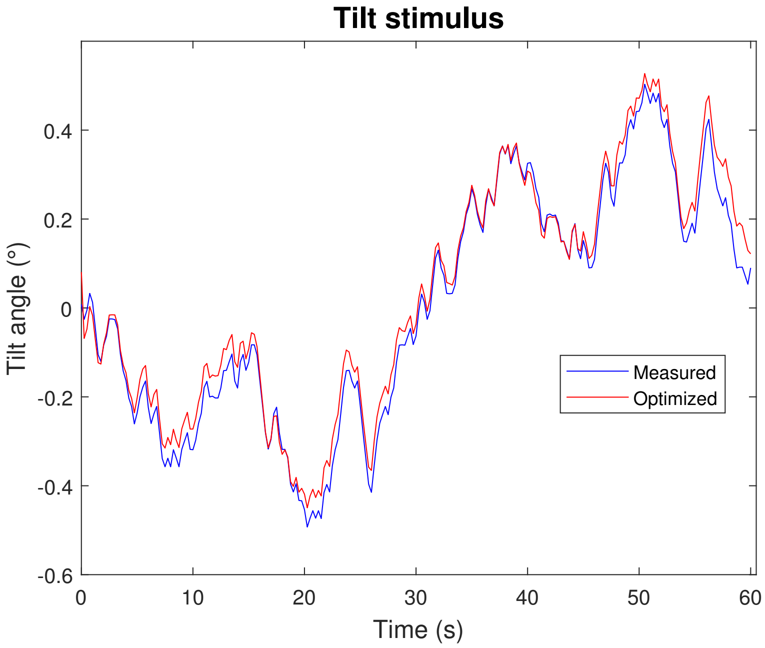


Figure 4.9: Surface tilt stimulus (input) measured (blue) and optimized (red). This result is the one for the patient 3 using a tilt stimulus of amplitude pp°2.

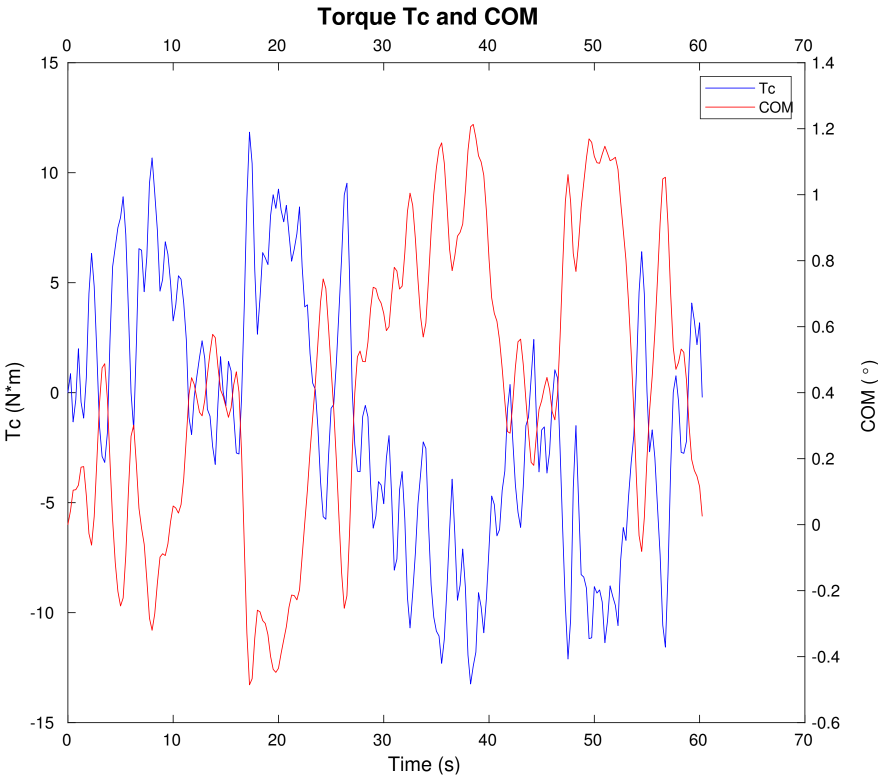
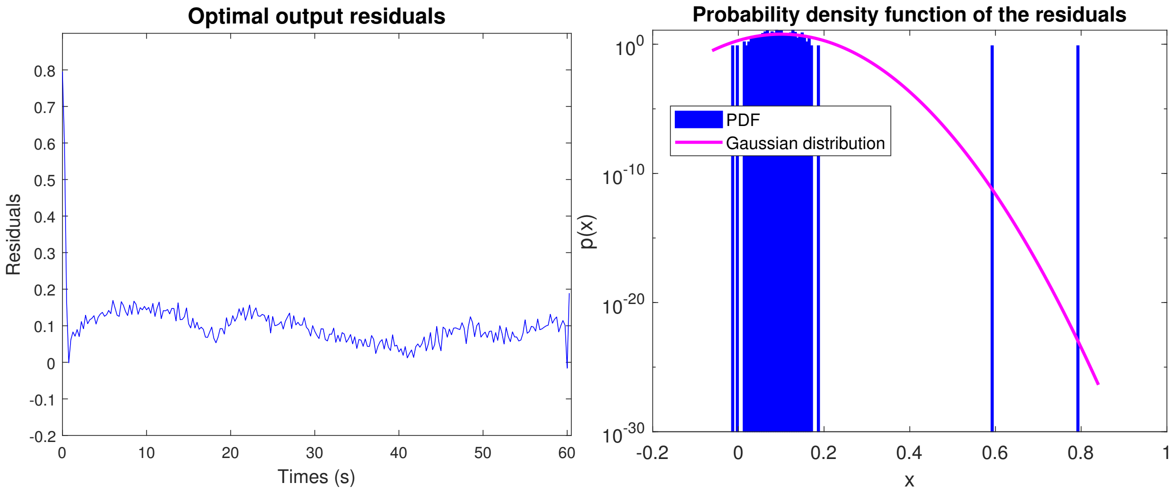


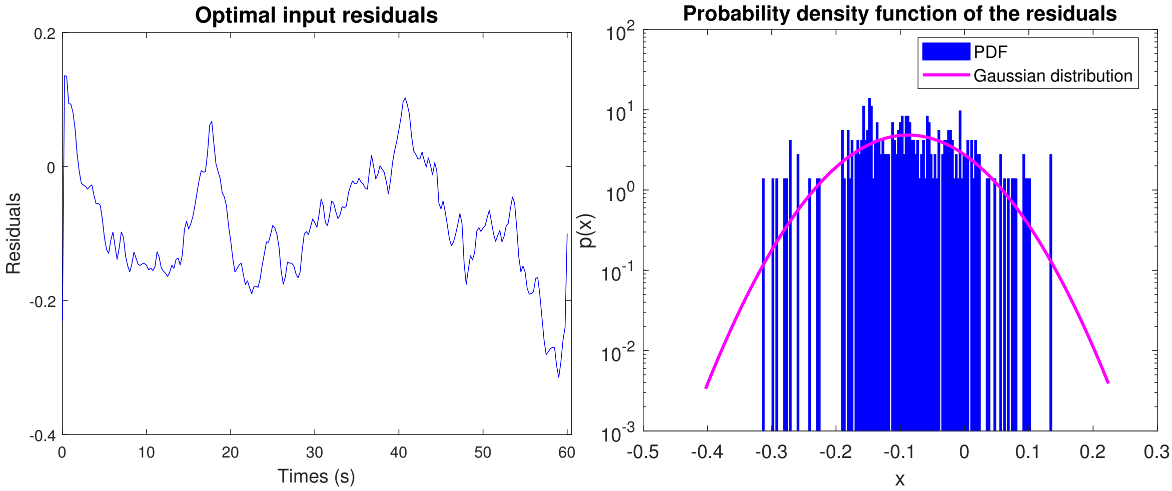
Figure 4.10: Generated corrective torque T_c and COM body sway optimized. Those results are the ones for the patient 3 using a tilt stimulus of amplitude pp°2.



(a) Optimal output residuals over simulation time

(b) PDF of the optimal output residuals (blue) and fitted Gaussian distribution (pink).

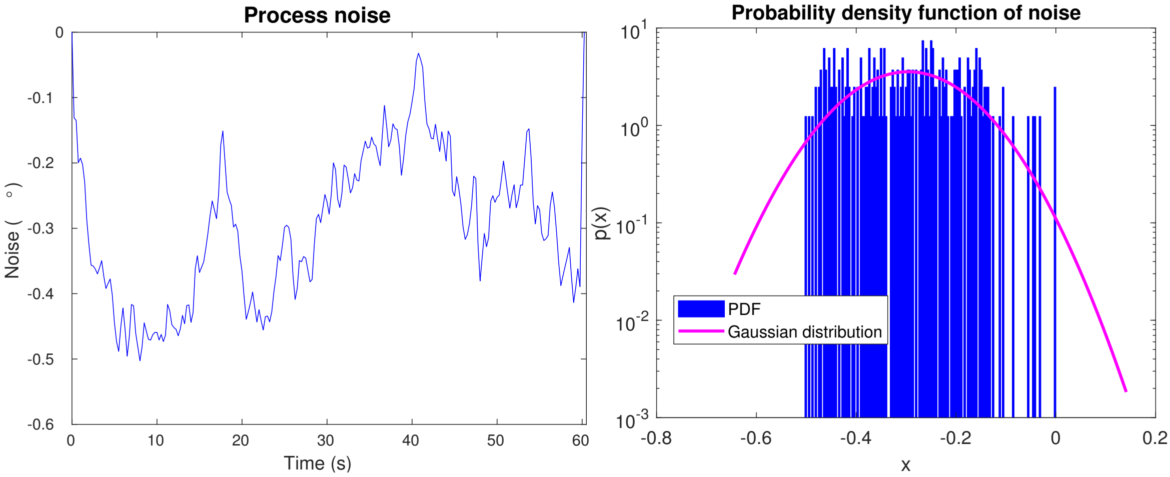
Figure 4.11: Plots of the optimal output residuals. This result is the one for the patient 3 using a tilt stimulus of amplitude pp°2.



(a) Optimal input residuals over simulation time

(b) PDF of the optimal input residuals (blue) and fitted Gaussian distribution (pink).

Figure 4.12: Plots of the optimal input residuals. This result is the one for the patient 3 using a tilt stimulus of amplitude pp°2.



(a) Process noise time serie.

(b) PDF of w_1 (blue) and fitted Gaussian distribution (pink).

Figure 4.13: Process noise of the first state θ_{BS}. This result is the one for the patient 3 using a tilt stimulus of amplitude pp°2.

Non-averaged data: To investigate the difference of results, when using averaged or non-averaged data for one patient, the optimization of the IC model, was then run for every single patients and single amplitude. The 9 allowable cycles where concatenated to form a time serie. The optimization procedure was run on this data serie.

The results for the parameters estimation are given in the following tables. The simulation of the model was then run with the optimized found parameters. The patients in red, are the ones for whom, the simulation of the dynamics using the optimized parameters, was unstable.

	Patient 1	Patient 2	Patient 3	Patient 4	Patient 5	Patient 6	Patient 7
K_{grav}	0.00	0.09	0.00	0.00	0.01	0.00	0.00
K_{d}	4.83	4.26	5.47	9.78	4.12	5.28	1.55
K_{p}	15.0	21.9	13.0	10.9	16.4	13.0	12.3
τ	0.17	0.08	0.24	0.28	0.10	0.15	0.05
K_{f}	0.13	0.33	0.00	0.02	0.36	0.06	0.03
τ_{f}	20.6	60.5	22.9	60.5	60.5	4.75	0.02

Table 4.8: Optimized parameters using a (support)-surface tilt stimulus of amplitude pp°1, with physiological data from the table 4.1 and the initial conditions for the NLP from the table 4.2. For each patient, 9 cycles of provided data were used, one transient cycle was considered. The set of parameters for patients in red, results in bad fit.

	Patient 1	Patient 2	Patient 3	Patient 4	Patient 5	Patient 6	Patient 7
K_{grav}	0.00	0.32	0.29	0.00	0.00	0.00	0.12
K_{d}	5.13	5.67	49.2	6.37	7.63	6.25	5.10
K_{p}	18.1	15.9	10.9	16.9	14.9	13.8	21.9
τ	0.12	0.12	0.07	0.09	0.07	0.08	0.12
K_{f}	0.35	0.15	0.24	0.35	0.09	0.16	0.25
τ_{f}	60.5	8.68	60.5	11.9	5.65	4.71	60.5

Table 4.9: Optimized parameters using a (support)-surface tilt stimulus of amplitude pp°2, with physiological data from the table 4.1 and the initial conditions for the NLP from the table 4.2. For each patient, 9 cycles of provided data were used, one transient cycle was considered.

	Patient 1	Patient 2	Patient 3	Patient 4	Patient 5	Patient 6	Patient 7
K_{grav}	0.38	0.40	0.07	0.54	0.00	0.15	0.29
K_{d}	3.94	4.95	4.90	6.64	18.6	5.84	5.99
K_{p}	17.0	19.8	21.7	21.9	21.2	21.9	21.9
τ	0.07	0.07	0.07	0.15	0.09	0.06	0.10
K_{f}	0.16	0.19	0.10	0.46	0.14	0.12	0.06
τ_{f}	11.9	16.3	16.6	60.5	9.10	8.36	10.9

Table 4.10: Optimized parameters using a (support)-surface tilt stimulus of amplitude pp°4, with physiological data from the table 4.1 and the initial conditions for the NLP from the table 4.2. For each patient, 9 cycles of provided data were used, one transient cycle was considered.

	Patient 1	Patient 2	Patient 3	Patient 4	Patient 5	Patient 6	Patient 7
K_{grav}	0.66	0.75	0.49	0.58	0.83	0.40	0.53
K_{d}	4.76	6.28	5.92	8.16	7.67	19.2	5.02
K_{p}	19.2	15.7	21.9	21.9	12.4	10.9	21.9
τ	0.09	0.17	0.11	0.07	0.26	0.07	0.10
K_{f}	0.14	0.10	0.19	0.13	0.04	0.19	0.21
τ_{f}	21.3	7.31	60.5	5.53	4.05	4.52	60.5

Table 4.11: Optimized parameters using a (support)-surface tilt stimulus of amplitude pp°8, with physiological data from the table 4.1 and the initial conditions for the NLP from the table 4.2. For each patient, 9 cycles of provided data were used, one transient cycle was considered. The set of parameters for patients in red, results in an unstable dynamics simulation.

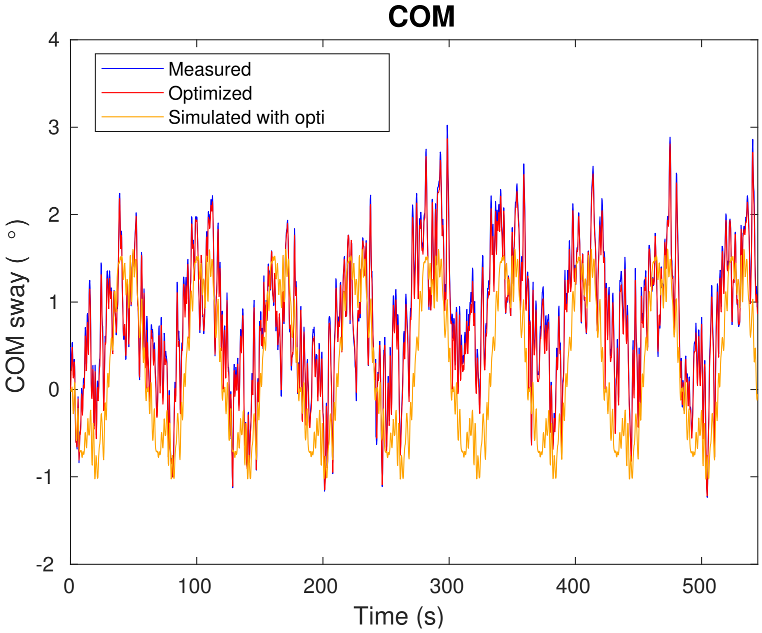


Figure 4.14: COM body sway (output) measurements (blue), optimized (red), and simulated (orange) using the IC model and the optimized parameters. Those results are the ones for the patient 3 using a tilt stimulus of amplitude pp°2. The value of the defined model validation fit criterion is: fit = 88.21%.

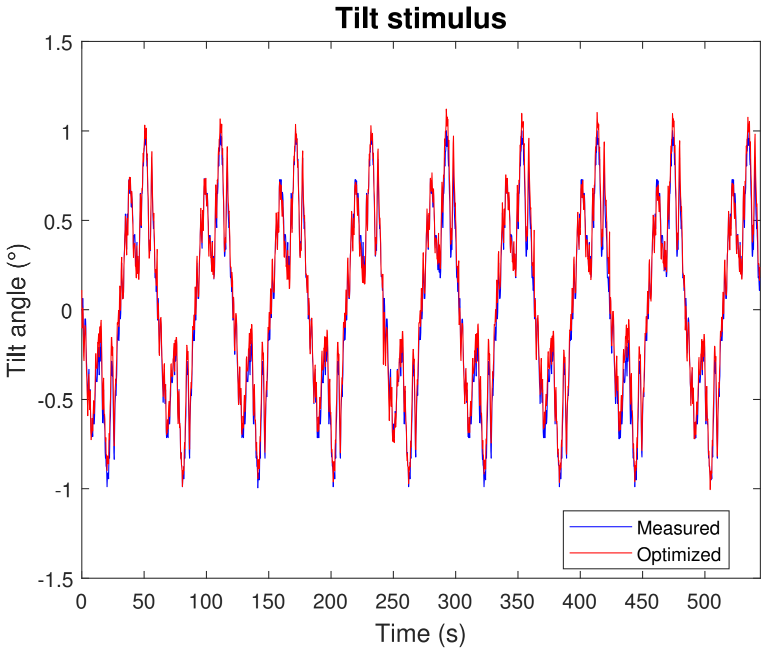


Figure 4.15: Surface tilt stimulus (input) measured (blue) and optimized (red). This result is the one for the patient 3 using a tilt stimulus of amplitude pp°2.

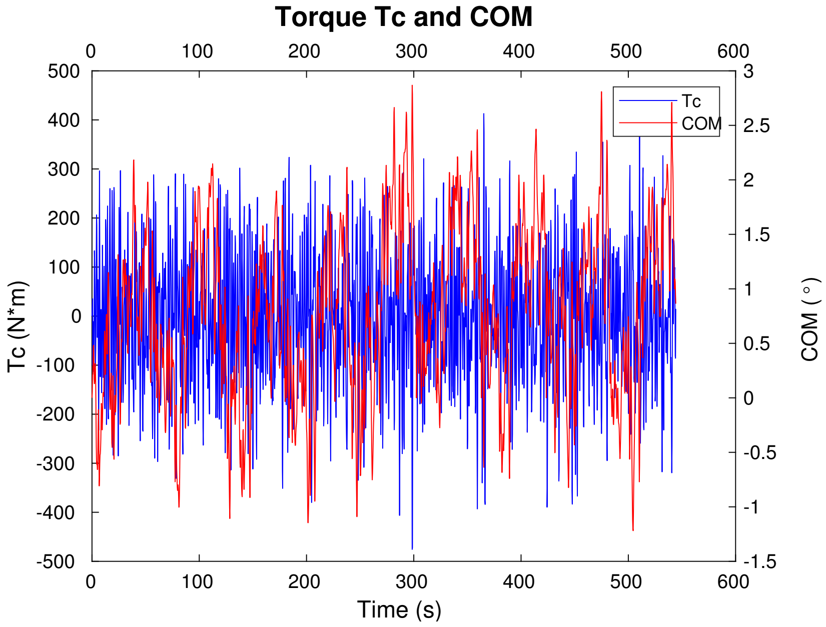
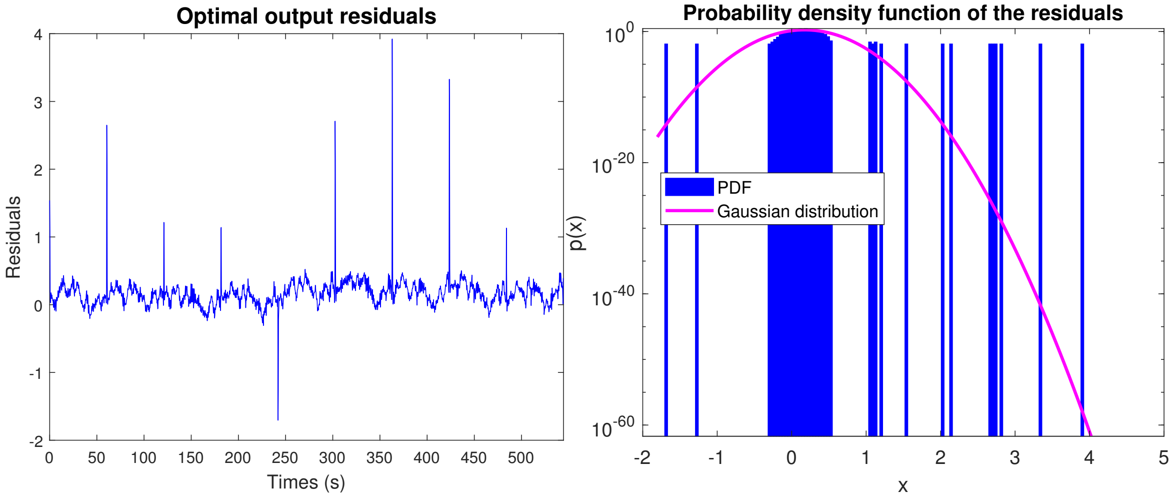
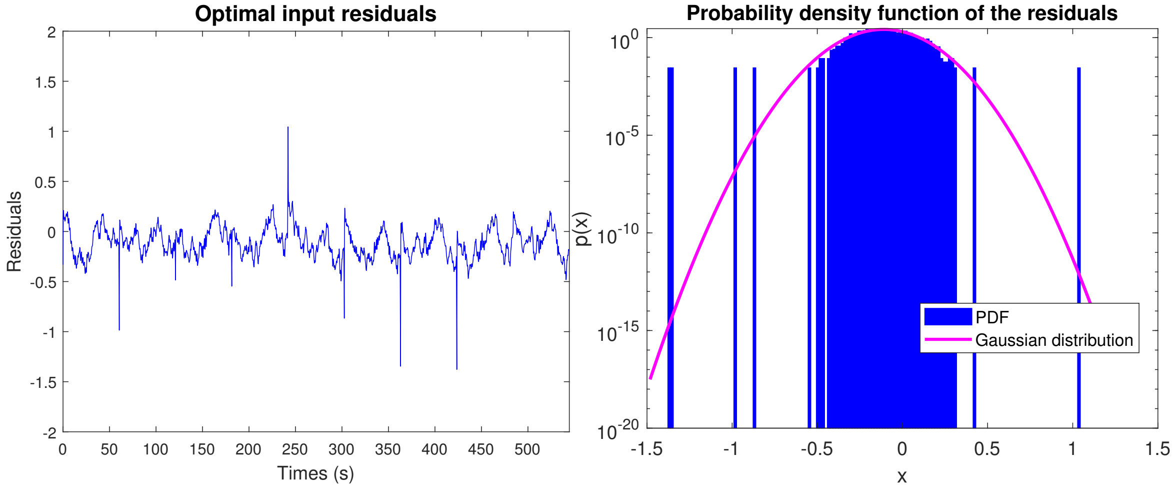


Figure 4.16: Generated corrective torque T_c and COM body sway optimized. Those results are the ones for the patient 3 using a tilt stimulus of amplitude pp°2.



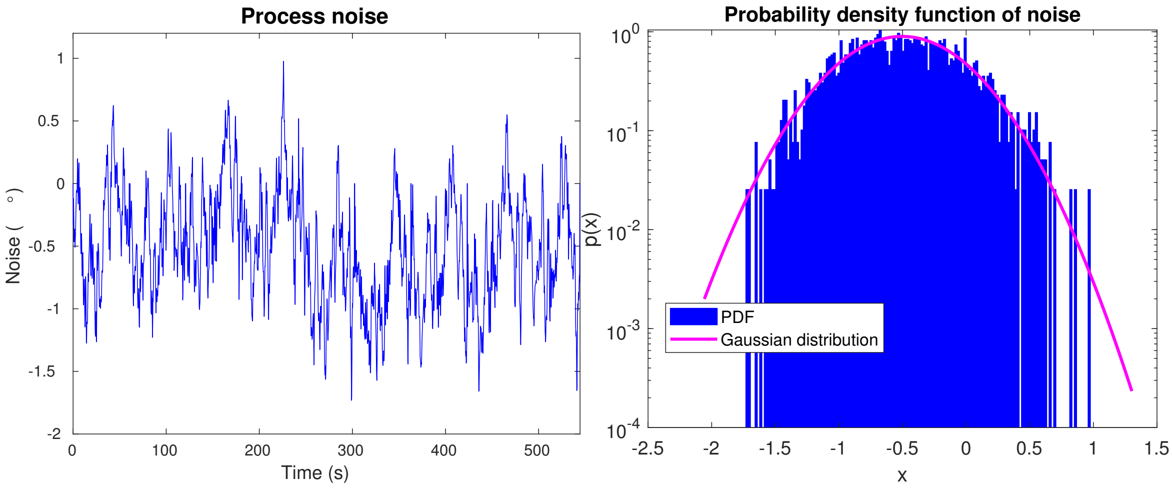
(a) Optimal output residuals over simulation time (b) PDF of the optimal output residuals (blue) and fitted Gaussian distribution (pink).

Figure 4.17: Plots of the optimal output residuals. This result is the one for the patient 3 using a tilt stimulus of amplitude pp°2.



(a) Optimal input residuals over simulation time (b) PDF of the optimal input residuals (blue) and fitted Gaussian distribution (pink).

Figure 4.18: Plots of the optimal input residuals. This result is the one for the patient 3 using a tilt stimulus of amplitude pp°2.



(a) Process noise time serie.

(b) PDF of w_1 (blue) and fitted Gaussian distribution (pink).

Figure 4.19: Process noise of the first state θ_{BS}. This result is the one for the patient 3 using a tilt stimulus of amplitude pp°2.

4.2.2 DEC model

The optimization of the DEC model, was run on every single patient, for four (support)-surface tilt stimulus amplitudes. For each stimulus amplitude and each patient, the data were first averaged across 9 cycles. One cycle of transient was considered. Then, for each patient, each average stimulus amplitude (pp°1, pp°2, pp°4, pp°8) were concatenated together, to form a data serie. The optimization procedure was run on this data serie.

This concatenate of different amplitudes, in order to estimate the parameters model, is necessary for the DEC model, to be able to estimate the threshold value λ. Indeed, the essence of this model, is that its parameters can represent the data over all the considered stimulus amplitudes.

The results for the parameter estimation are given in the following tables. The simulation of the model was then run with the optimized found parameters.

	Patient 1	Patient 2	Patient 3	Patient 4	Patient 5	Patient 6	Patient 7
K_{grav}	0.27	0.28	0.00	0.00	0.00	0.00	0.00
K_{th}	1.00	1.00	1.00	1.00	1.00	1.00	1.00
λ	0.22	0.22	0.38	0.48	0.90	1.33	0.26
K_{s}	0.97	0.96	0.84	0.69	0.40	0.32	0.94
K_{d}	3.58	4.35	5.75	7.60	12.0	24.4	5.79
τ	0.08	0.08	0.08	0.05	0.05	0.05	0.12
K_{f}	0.06	0.08	0.05	0.08	0.08	0.10	0.04
τ_{f}	6.24	5.47	1.10	1.35	1.70	3.20	1.33

Table 4.12: Optimized parameters using four (support)-surface tilt stimulus of amplitude pp°1, pp°2, pp°4 and pp°8. The physiological data are taken from the table 4.1 and the initial conditions for the NLP, from the table 4.3. For each stimulus amplitude, and each patient, the data are averaged across 9 cycles.

Because lots of data are available, just the plots for one patient are shown.

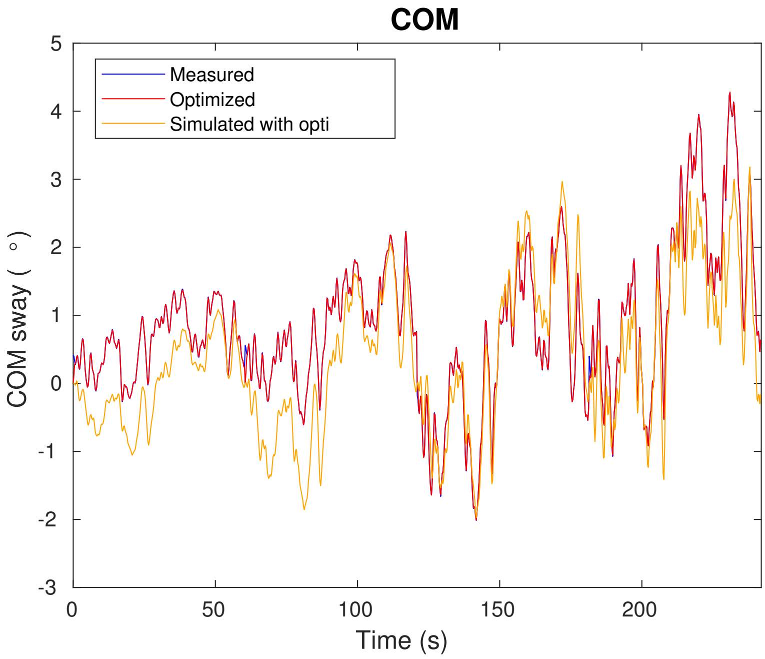


Figure 4.20: COM body sway (output) measured (blue), optimized (red), and simulated (orange) using the DEC model and the optimized parameters. Those results are the ones for the patient 3 using four tilt stimulus of amplitude consecutively pp°1, pp°2, pp°4, pp°8. The value of the defined model validation fit criterion is: fit = 96.98%.

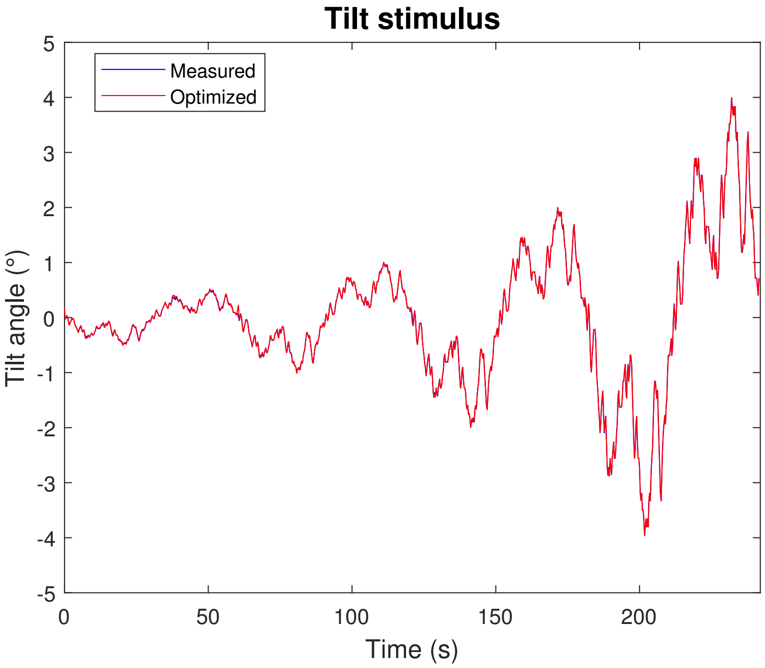


Figure 4.21: Surface tilt stimulus (input) measured (blue) and optimized (red). This result is the one for the patient 3 using four tilt stimulus of amplitude consecutively pp°1, pp°2, pp°4, pp°8.

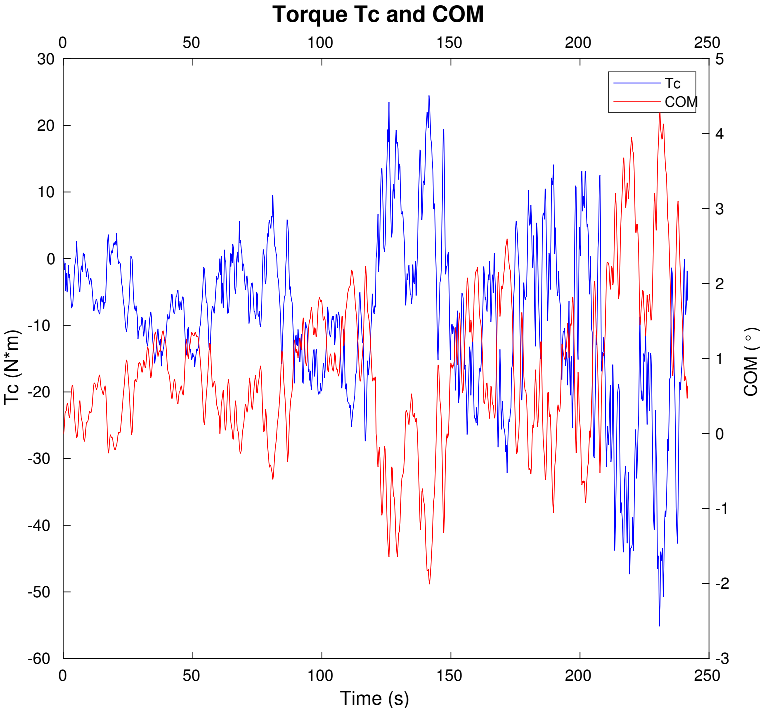
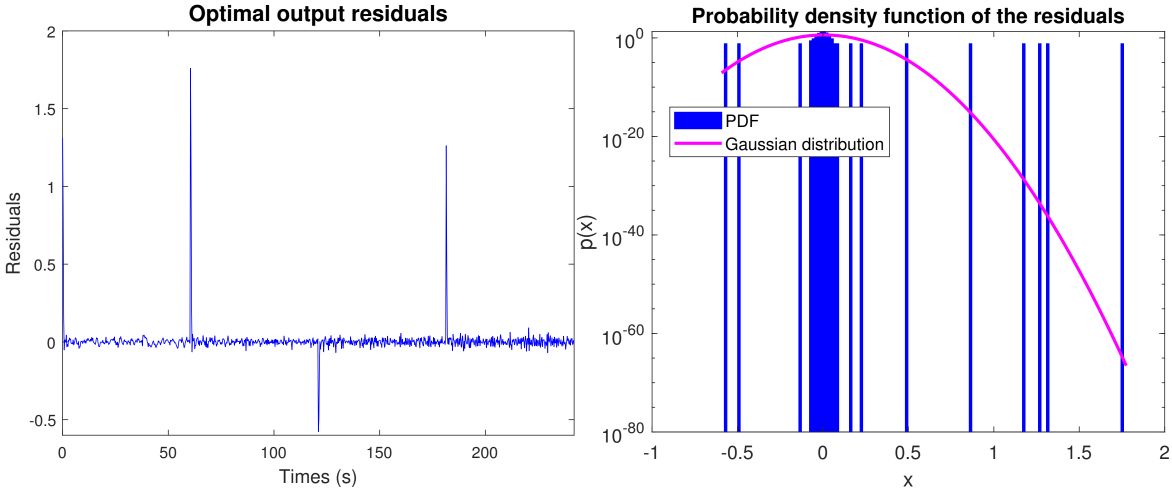


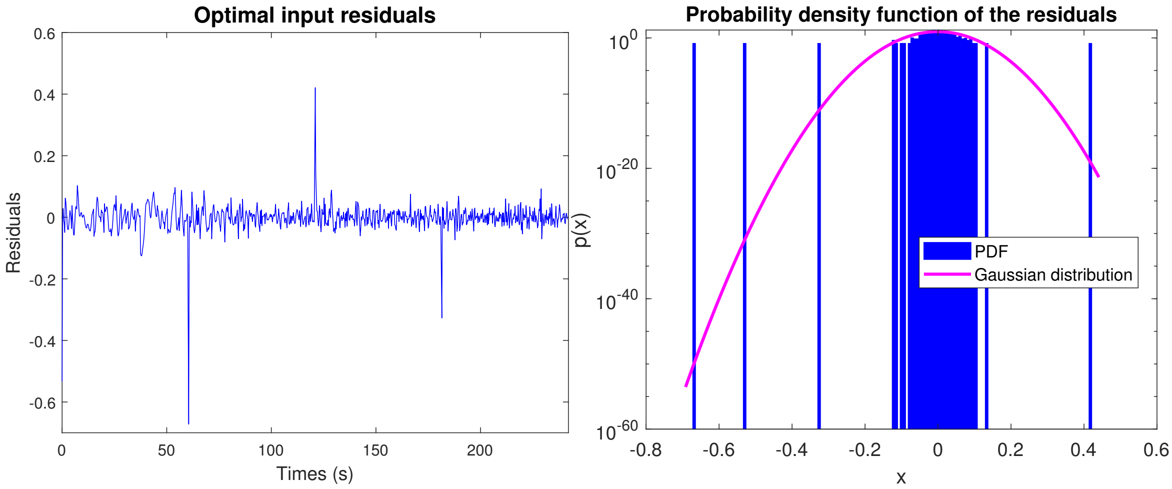
Figure 4.22: Generated corrective torque T_c and COM body sway optimized. Those results are the ones for the patient 3 using four tilt stimulus of amplitude consecutively pp°1, pp°2, pp°4, pp°8.



(a) Optimal output residuals over simulation time

(b) PDF of the optimal output residuals (blue) and fitted Gaussian distribution (pink).

Figure 4.23: Plots of the optimal output residuals. This result is the one for the patient 3 using four tilt stimulus amplitude of consecutively pp°1, pp2°, pp°4, pp°8.



(a) Optimal input residuals over simulation time

(b) PDF of the optimal input residuals (blue) and fitted Gaussian distribution (pink).

Figure 4.24: Plots of the optimal input residuals. This result is the one for the patient 3 using four tilt stimulus amplitude of consecutively pp°1, pp2°, pp°4, pp°8.

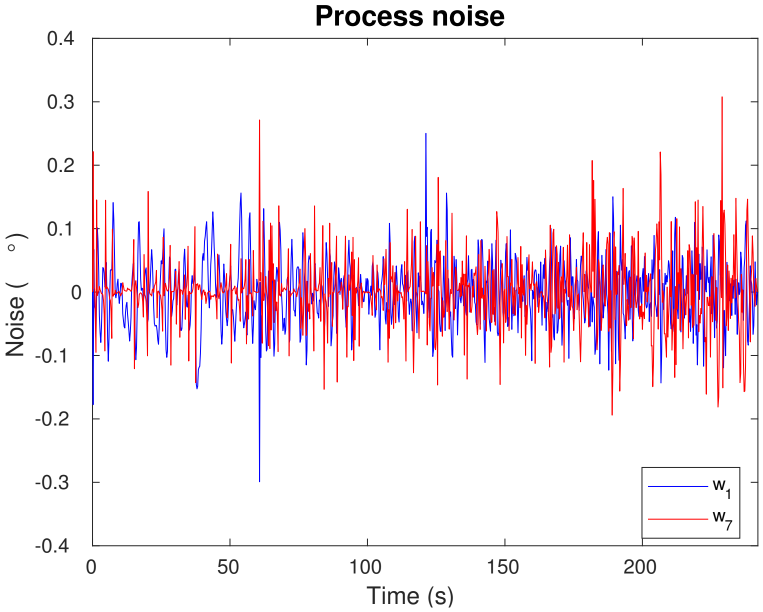
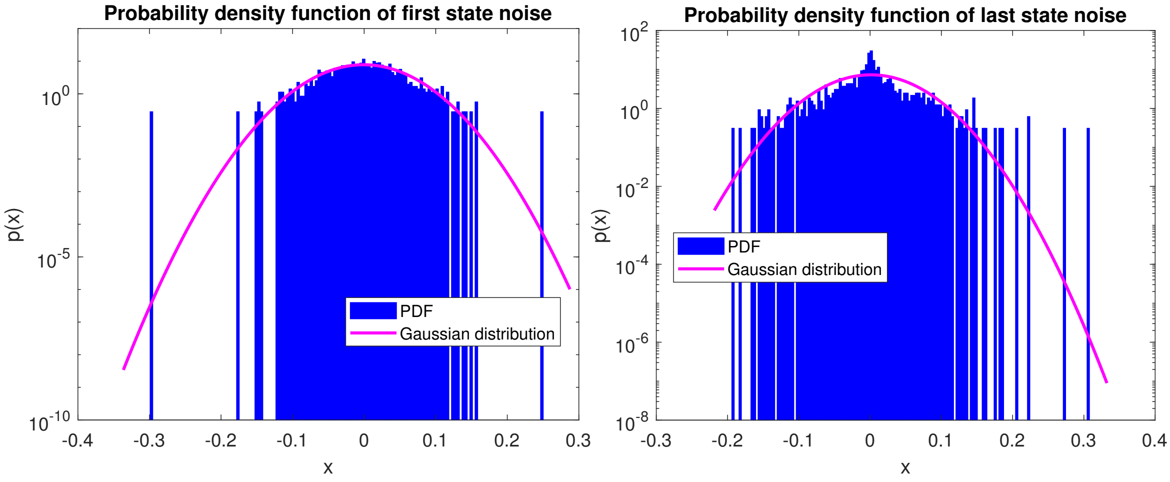


Figure 4.25: Process noise of the first state θ_{BS} (w_1), and on the last one $\dot{\bar{u}}$ (w_7) in a time serie. This result is the one for the patient 3 using four tilt stimulus of amplitude consecutively pp°1, pp°2, pp°4, pp°8.



(a) PDF of w_1 (blue) and fitted Gaussian distribution (pink).

(b) PDF of w_7 (blue) and fitted Gaussian distribution (pink).

Figure 4.26: Process noise of the first state θ_{BS} (w_1), and on the last one $\dot{\bar{u}}$ (w_7). This result is the one for the patient 3 using four tilt stimulus of amplitude consecutively pp°1, pp°2, pp°4, pp°8.

5 Discussion

5.1 Model Validation

In order to write the system in a state-space formulation, three main approximations were made. One on the modelling of the time delay (IC and DEC models), one on the approximation of the input derivative (for the DEC model), and the last one, on the approximation of the threshold (for the DEC model).

Time delay: The time delay was approximated by two states. In order to reduce the discrepancy due to the approximation of the time delay, one could increase the number of states to model it. In the limit of infinite number of states it would perfectly match the time delay experienced by the system. Here, two states were dedicated to the approximation of the time delay, but in order to have an idea on the approximation error, the number of states modelling this delay were increased up to ten. The error then made can be observed in the figure 4.3. Having fewer states to represent the time delay is beneficial, because it will then reduce the number of variables in the optimization problem, and thus the computation time. For the two implemented models, regarding the simulation results, the approximation on the time delay is acceptable. The number of states approximating the time delay could not be reduced to one, since it is expected regarding the literature, that the time delay value should be around 0.16s, and the sampling time of the system is 0.25s.

Threshold function: It was chosen to approximate the action of the threshold by a continuous function described by (3.21). One could have thought of an other way to deal with the modelling of this threshold. In the electronic domain for example, it is often modelled by what is called a "dead zone". Meaning, that the input signal is not transmitted, if it has not passed a certain value. The big disadvantage of using this modelling approach, is that it is described by non continuous dynamics, and it is thus not differentiable for the full time domain. Thus, the continuous formulation of the threshold was preferred. The only disadvantage of the continuous formulation, is that it introduces a non-linearity in the dynamics equations. The smoothed used threshold function is plotted (figure 5.1) with the same values for α and λ used to simulate the dynamics. The maximum deviation between the "true" threshold, and its smoothed approximation, is in this case of value 0.03, which can be considered as small, regarding the physical angles magnitude present in this work. It should be said, the bigger the α value is, the bigger is then the approximation error.

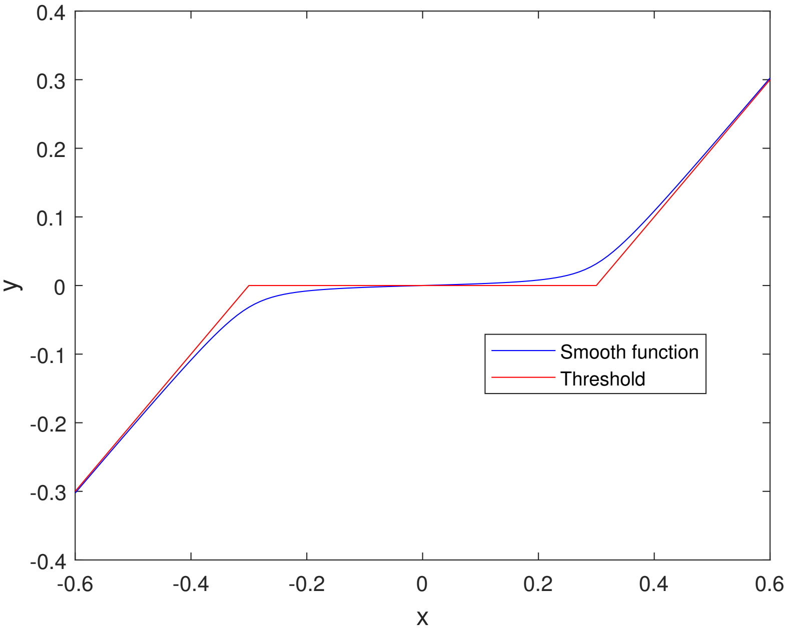


Figure 5.1: Smoothed threshold function with $\lambda = 0.3$ and $\alpha = 0.05$ (blue), and threshold (red).

5.2 Optimization Procedure

Computational time: Since the problem solved is high dimensional, and this high dimensionality leads to an increase in the computational time, it was tried to reduce the problem dimensionality, without impacting the results. To achieve this goal, different ways were explored.

First of all, the provided measurements were inherently having a sampling time of 0.1s, and having 60500 measurements per cycle. Looking at how the input signal, that is a PRTS, was constructed (section: 3.4.1), the sampling time was increased to 0.25s. This reduced the number of data points per cycle to 242.

The time required for the integration step is also something that can be optimized. Indeed, integration methods are not equal in terms of computational time. But one should be careful, because they also differ in terms of the accuracy. Thus, the integration methods used was investigated in order to ensure a desired accuracy, but optimizing the required time. This was explained into detail in the subsection 4.1.4. The choice of integration methods is highly related to the desired accuracy.

By winning some time in the integration step, one is also now able to take into account more data (measurements) to reach the same simulation time that was necessary without having this process of optimizing the time done. In the context of parameter identification, it is always preferable to use more data in order to identify the parameters, because this process is based on a statistical interpretation. Thus, the more data are used, the more representative are the parameters.

The method used to discretized the NLP was the multiple shooting method. This method is not the only simultaneous approach available. Indeed, one could have thought to implement a *Direct Collocation* method. This method is a fully simultaneous method, that keeps all the ODE discretization variables as optimization variables. No integration routine is needed, since the numerical simulation equations are taken as equality constraints of the optimization problem [27].

In the direct collocation method, the problem is augmented by all the intermediate state variables. And to ensure continuity at the collocation points, equality dynamical constraints are formulated. Thus this increases the number of optimization variables, and equality constraints, in comparison to a multiple shooting method [27]. Because the dimensionality was of concern here, and because of the integration method used, the multiple shooting method was preferred. This can also be debated, depending on which Runge-Kutta methods is chosen.

NLP bounds: Some specific bounds are given for the parameters in the NLP of both models. Those bounds can be seen as an a priori knowledge on the parameter values. Care was taken to constrain as little as possible the parameter values, in order to be able to not discard some optimal solutions. It should also be reminded, that introducing bounds, introduces a knowledge in the parameter estimation procedure, and thus, one must be sure of them, to not bias the result.

Because all the parameters have an underlying physical/biological meaning, they can not be negative, so all have their lower value were restricted to be positive or equal to zero. Moreover, for the IC model, the parameter K_p was constrained between mgh and $2 \cdot mgh$ [30]. The lowest value is the minimal proportional gain required in order for the system to remain stable. The upper one was fixed regarding the highest observed value in the literature, that is around $1.3 \cdot mgh$, and taking a safety margin. If this gain it too high, this will also destabilize the system. The sensory gain K_{grav} was constrained because of the used convention that $K_{\text{grav}} + K_{\text{prop}} = 1$. This convention is underlying the concept behind the IC model. The upper bounds on the time constants were fixed to the duration of one cycle of the experimental data, because otherwise, the provided data would not have been able to give a proper estimate of it. The lower bound 0.025s comes from the fact, that the fastest reflex known in the human body, is the so called stretch reflex, and is around 50 to 60ms.

It should be said, that looking at the tables of the optimized parameters, it can be seen, that some of the imposed bounds (constraints) are clearly active at the optimal solution. For the IC model, it concerns the constraints on the parameters K_{grav} and τ_{f}, and for the DEC model, it is the constraints on K_{grav} and K_{th} that are mostly active.

As an investigation, it was tried to run the optimization process without any bound constraints. What came out, was that the optimized parameters were taking either very large or very small values for some of them (K_{grav}, K_{th}, K_{f}, τ_f), and thus, simulating the system using them, was leading to an unstable system dynamics. This comes from the fact, that if no bounds are set, then nothing guarantees the stability of the system. Then, in order to try to force the optimizer to find a set of parameters that describe a stable system dynamics, one should impose bounds in the NLP.

Even if bounds were imposed in the IC NLP, it happened for some cases (denoted in red in the tables 4.4, 4.6, 4.7), that the optimized parameters were not producing a stable behavior of the system, when used in the simulation. For those special cases, the bounds were not tightened, because then, one would not be able to justify them anymore, and arbitrary knowledge would be introduced.

Measurement fit: The plots for the both implemented models (figures: 4.8, 4.9, 4.14, 4.15, 4.20, 4.21) show that, both the optimized output θ_{BS} and input θ_{TILT} fit well the measurements. This indicates that the residuals are low at the optimal solution, which is a desired result. Especially if one uses the Gauss-Newton method to solve the nonlinear least-squares problem. Indeed, the lowest value for the model validation fit criterion is 88.21% and was obtained for the non-averaged IC model optimization. Moreover, plots of the optimal residuals over the experiment duration look like a sequence of random numbers (figures: 4.11, 4.12, 4.17, 4.18, 4.23, 4.24). Their probability density functions show that for the IC model using non-averaged data and the DEC model, they follow a Gaussian distribution, and are close to be zero mean but not with unit variance, meaning that model assumptions on the system are not correct, and should be investigated. Moreover, the input residuals for the IC model using averaged data does not seem to follow a Gaussian distribution. This was expected, since the averaging process is reducing the variability present into the data. It should also be pointed out that the residuals for the non-averaged data have bigger values than the ones for averaged data. On the histogram plots, some outliers can be observed. In fact, comparing the histogram plots with their respective optimal residuals plots, one notices that the pic values on the optimal residuals plots correspond to the outliers of the histogram plots. Moreover, those pic values appear at the junction between two cycles of data. This is coherent, since the dynamics constraints were released between two cycles of data, since they do not have common dynamical behavior.

Looking at the COM body sway plots (figures: 4.8, 4.14, 4.20), one notices that the biggest mismatch of the plots, is for both cases, between the results of the simulation using the optimized parameters, and the measurements. This comes from the fact, that the simulation is realized with the "true" input (measurements), and initialized with no process noise. If the system is simulated using the optimized input and the realization of the found process noise, then the plot of the simulation fits perfectly the plot of the optimization.

Generated corrective torque: One huge advantage of the approach proposed in this work, is that it permits to define internal variables, that describe quantities of interest, even though they are not taken as a state of the system. This allows one to assess their value, and can be then useful to interpret the system behavior. In the set-up used to realize the experiment, only the tilt of the platform, and the body sway were measured, but no force sensors were embedded in it, and thus no information on the corrective produced torque was available. In the implemented models, the corrective torque T_c was defined as an internal variable, and thus assessed. Its value for both models can be seen in the figures 4.10 and 4.22. It is plotted with the COM body sway, because it is fairly assumed, that the produced torque, as it is stabilizing the system, should be closely related to the body sway, that the body does want to control. One can notice that the obtained torque is in exact anti-symmetric behavior with the COM body sway. Moreover, as the magnitude of the COM body sway increases, the torque's magnitude does so. Those coherent results tend to validate the obtained results.

Noise: Process noise was only added on the first state for the IC model, and on the first state and on the state accounting for the input derivative, for the DEC model.

As a first try, process noise was added blindly to every state of the system for both models. But then, especially for the IC model, the optimized parameters were not able to produce a stable dynamics. In fact, the third state θ_f was set to zero by the optimizer, and all its values were considered as noise. Thus, this was deleting the positive feedback loop, and the system was not able to stabilize itself anymore. Then, looking at the existing literature, only process noise on the COM body sway was set. In fact, it has been shown, that the body sway is noisy, and one was able to show in a study, that there is a remnant body sway (sway variability) that can account for up to fifty percent of the measured body sway [30]. Moreover, in the same study, they estimated that for eyes closed conditions, the sway variability was mainly originated from the vestibular system, and that the "motor noise", which arises from the torque generation process, was only contributing to ten percent of this remnant sway. Thus, one can assume that the noise on the state taking into account the torque effect, is negligible in comparison to the one affecting the body sway. This thus makes the first state very noisy in comparison to the other ones. By allowing just the first state to be noisy, then optimized parameters found for the IC model were able to produce stable system dynamics.

In order to penalize more the process noise, its variance was set ten times bigger than the one for input and output measurements noises. This was implemented in the NLP through the inverse covariance matrices Q_x, Q_u and Q_w. Reducing the input and output measurements variances with compared to the process noise variance, permits to have a better fit, and thus smaller residuals. At the beginning, it was hypothesized that the output measurements noise was lower than the input measurements one, and consecutively, only the variance of the output measurements was decreased. This was not providing a better fit of the data, and sometimes was also even making it harder the optimization to converge. And once introduced in the simulation, the optimized parameters for such a case where producing a worse simulation result. In fact, increasing the variance of the input measurements, makes that the solver is putting less effort to minimize the input prediction error, and this leads to bad fit in this case. Moreover, there is no clear evidence, or at least quantitative, that shows that the input measurements are more noisy than the output ones.

All the noises were hypothesized to be Gaussian noise with zero mean. Looking at the process noise plots of both models (figures 4.13, 4.19, 4.25), one clearly observes that the hypothesis of zero mean is not fulfilled for the process noise of the IC model, but it is the case for the DEC model process noise. This fact informs that the assumptions made on the process noise are not correct, and must be adapted. For both models, the process noise is of same order of magnitude. Moreover, looking at the histogram plots of the process noise, for the IC model, the one for non-averaged data follows a Gaussian distribution even if it is not zero mean and of proper variance, and the one for averaged data does not follow a Gaussian distribution. This was expected since the averaging process is corrupting the data variability. This finding favors the use of non-averaged data.

5.3 Parameter Value

The methodology developed has the advantage to be flexible, in the sense that results were obtained for averaged and non-averaged data. Even though no confidence bounds are computed for the parameters estimation, which does not allow to draw conclusion on the exact values of the optimized parameters, as optimized parameters were computed for all the patients and all different stimulus amplitudes, one can still identify some tendency regarding those results.

Regarding the IC model optimized parameters, one important behavior, underlying the model itself can be observed. Indeed, sensory reweighting is present. In fact, when the amplitude stimulus is changing, a sensory reweighting is happening in order to ensure the production of sufficient corrective torque to be able to counteract the gravity [30]. One can observe that the graviceptive sensory weight K_{grav} is increasing while increasing the stimulus amplitude, meaning that the proprioceptive weight K_{prop} is decreasing (due to the convention used).

Because eyes closed conditions were identified, the graviceptive weight represents the contribution of the vestibular sensory information. This noticeable change in the sensory weight values was also observed in other studies [6, 23, 30]. As the amplitude stimulus is increasing, the proprioceptive information get compromised, so it is crucial for the system to rely more on the vestibular information to maintain balance, justifying the increase of the graviceptive weight value.

Regarding the DEC model optimized parameters, one noticed that the threshold gain K_{th} is equal to 1 for all patients, meaning that the system fully compensates the disturbance. It has been hypothesized in earlier studies, that this gain value represents either the trust of the estimate, or a voluntary control to use this information for disturbance compensation [4, 22]. A value of 1 is thus unlikely. It is also thought that a lower threshold level λ is related to a lower noise level [4], and this would imply that lower signals can be exploited, and that a better compensation can be achieved. Thus for a threshold gain of 1, low λ values are expected, which is not the case here.

Still concerning the DEC model, the gravity and servo gains mainly contain vestibular and proprioceptive contributions because of the eyes closed condition. The sevo gain K_{s} value is related to the proprioceptive information, since it multiplies the angle between the body and the foot θ_{bf} which is derived from proprioception. One should mention, that there is also the passive contribution mentioned previously that contributes to the torque generation. But the servo gain, already gives an insight. Since the threshold adjusts the sensory contribution of each sensory information, depending on the sensory signal, looking at its value gives a knowledge about which sensory information is used by the system. For large tilt amplitudes, the proprioceptive contribution is decreased in favor to the vestibular one, and the threshold value is thus higher. In the table 4.3, it has been noticed that for the lowest obtained servo gain values, the threshold value λ, was the highest. This is coherent with the previous developed reasoning.

However, for the DEC model, the graviceptive gain is mostly equal to zero, meaning that the action of the gravity on the system is not taken into account, which is unlikely to happen. Moreover, patients with higher λ values were also exhibiting higher derivative gain K_{d} values. Since the proportional gain K_{p} was fixed, the derivative gain is the only adjustable parameter of the neural controller. A higher threshold value means a bigger contribution of the vestibular signal. It has been hypothesized that the vestibular system has a baseline noise level ten times higher than the proprioceptive noise [21, 30], and that it leads to an increase in remnant sway, when the vestibular signal was privileged in comparison to the proprioceptive one. Following this reasoning, if the remnant sway is increased, and thus the sway variability, the neural controller has to adapt, and to allow a faster response of the system in order to maintain balance. This is observed in the results, as the derivative gain was bigger for patients whose threshold value was higher.

The unstable dynamics of some patients can be explained looking at the optimized parameter values. Indeed, for some of them, this fail was due to the low-pass component gain K_f value, which was close to zero. Thus because the low-pass component is filtering the produced corrective torque, the stabilizing action of the torque was deleted. This decrease for this gain value was also accompanied by a higher time delay value. Moreover, for specific patients, the time delay value was reaching 55.6s, and it was noticed in a study, that above a time delay of 0.34s, the system is not able anymore to adapt the gains K_p and K_p to stabilize itself [30].

Some patients, for both implemented models, exhibit a time delay sensitively smaller than other ones for the same conditions. In the implemented models, the passive dynamics that account for the biomechanical of the muscles and tendons properties was not modelled. It is thought that this passive dynamics is also included in the generation of the corrective torque, and that this process is not affected by the time delay [23]. It was hypothesized that patients with large contribution from these passive properties, would have a biased time delay estimate towards lower values [26]. But the dynamics characteristics of the passive properties are very similar to the ones of the neural controller (both produced corrective torque), it was noticed that patients with small time delay value, were also having outlying K_p values. This last fact is not observed in the results here.

5.4 Further Improvements

5.4.1 Modelling

Body dynamics: As the parameter estimation procedure is able to be run on each specific patient, the identified parameters would be closer to their "real" value, if the specific physiological data for the fixed parameters J, m and h of each patient were used in the models.

The provided models were in a block diagram form, and the derived state-space equations were derived following this way of representation of the models. Thus it is simple to change/replace some parts of the existing models. Even if the assumption that the body dynamics can be modelled by a single inverted pendulum is well motivated in this work, one could easily increase its complexity, and implement higher order models for the dynamics. This modification could then account for more movement strategies, like the hip strategy for example [12].

Input tilt: In this work, only eyes-closed conditions were taken into account, meaning, that only one single input (surface tilt) of the system was considered. But the existing formulation of the system that is a SISO (Single Input Single Output) system, can be extended to a MISO (Multiple Inputs Single Output) one. Indeed, the input dimension can be augmented, and then, the visual world tilt could also be considered, which would increase the range of analysis possibilities.

5.4.2 Methods

Confidence bounds: Since the optimization procedure is converging for non-averaged data, it would make sense to compute the covariance of the identified parameters, which can be defined like [8]:

$$\Sigma_{\hat{\theta}} := \frac{\left\|R(\theta^*)\right\|_2^2}{N-d}(J(\theta^*)^\top J(\theta^*))^{-1}, \tag{5.1}$$

where $R(\theta^*)$ is the optimal residual vector, and $J(\theta^*)$ is the Jacobian matrix evaluated at the optimal solution. One should just be aware that a challenge can arise from the high dimensionality of the problem, which would make the Jacobian matrix also high dimensional.

NLP bounds: Instead of using bounds on the parameters, one can think to use a regularization technique. This technique allows to specify some soft constraints on the parameters, by adding a term in the objective function of the NLP. It is another way to impose a priori knowledge of model parameters when dealing with grey box models [13, 15]. In fact, as the parameters of the models have a physical significance, some guesses of their value are available. It has been experienced in this work, that when the parameters of the models do not stay close to the initial guess, the system tends to be close to instability. So then, with this regularization term, one can constrain them to stay close to the initial guess. This would be a way to promote the dynamics stability of the system. Moreover, one can also add other terms in this regularization, that referred to the level of confidence in the initial guess of the parameters. Considering the vector of unknown parameters p, and its initial guess p^*, such a regularization term could be:

$$\frac{\lambda}{2}\left\|p - p^*\right\|_R^2$$

where R and λ represent the confidence on the prior knowledge of the parameters.

For some cases in the IC model, where the optimized parameters were producing an unstable dynamics, it could even be thought to add this regularization term as an addition to the existing bounds.

Computational time: Once the implemented models are accepted, and the optimization procedure used is fully validated for this specific case of parameter identification, one can generate a self-contained C-code. Indeed, the software CasADi allows the generation of such a code. In the author's opinion, this can have a huge impact on the speed of the process. In fact, it was noticed, that most of the time spent for the optimization, was spent to construct all the needed structures of the problem, and not in the solver itself. It is known that doing a numerical evaluation of an auto-generated code could speed up the evaluation time by a factor of 4-10 [1].

Another aspect is, that if this method becomes a diagnostic tool, then it allows one to estimate the parameters faster, and it also allows one to use this tool even if CasADi in not installed on the running machine. And this could even be integrated in an embedded system.

6 Conclusion

The aim of this master thesis was to reformulate existing models of the human balancing system in a state-space formulation, and identify the system parameters, through the use of experimental data and numerical optimization methods. Indeed, the methods currently used, realized the optimization fitting of the measurements, in the frequency domain. But the models are composed of nonlinearities that are difficult to formulate in this domain.

In order to formulate the models using a state-space formulation, different approximations had to be carried out. The system time delay was approximated by two state variables, another state variable was allocated to the input derivative approximation, and the nonlinear threshold was modelled by a continuous time function. Then, the models were simulated and validated against the ones provided. The different approximations realized to represent the models were investigated individually, and they were able to account for the discrepancies between the output simulation of the implemented models, and the provided ones.

Once the models were derived, a NLP was formulated, considering a maximum likelihood estimation criterion. The continuous time dynamics was discretized using the direct optimal control method multiple shooting. This problem was then implemented in CasADi and solved with Ipopt.

In order to achieve a reasonable problem dimensionality and computational time, different aspect of the problem were considered. The integration step was analyzed in term of accuracy versus simulation time, and a RK4 method with enough steps per integration step was chosen. The multiple shooting method was preferred against a direct collocation method, to lower the number of optimization variables, and constraints. Finally, the sampling time was set to 0.25s regarding the input signal generation, while the inherent sampling time of the data was of 0.01s.

The numerical optimization was converging to an optimum for both models, and all the tested configurations. Good fits were achieved between the measurements and the optimized variables. The simulation of the models with the optimized parameters were coherent, even though for some rare cases, the simulated dynamics turned out to be unstable. The optimal residuals look like a random sequence over the optimization time, for one specific realization. But a further investigation using histogram plots, concluded that some model assumptions were not validated. Regarding the histograms of the process noise, the assumptions on the noise were also not fulfilled. Moreover, one can not exclude that the model assumptions were right. So the noise and also maybe the system models should be improved. Even if the accuracy of the estimated parameter values can not be discussed, since no confidence bounds were provided, one is still able to point out some known/accepted behavior of those parameters. Indeed, for both models, even for averaged and non-averaged data, the dynamical reweighting of the sensory weights was observed in the results. Moreover, for the DEC model, the set of identified parameters was able to fit the data and reproduced the sway response across the different stimulus amplitudes. Thanks to the state-space formulation, all the states and internal variables of the models were accessible. This provides a better insight to the internal process. Thus, the produced corrective torque was able to be assessed, and its evolution over the experiment time was coherent.

To the author knowledge, it is the first time that the optimization procedure was able to be applied on non-averaged data for the given models. This broadens the range of possible analysis. Indeed, no more averaging on the data is needed, so now patients that were suffering from a vestibular loss for example, and whose data were not sufficient enough because those patients were not able to sustain a full experiment protocol, can now be analyzed. Further more, the data are no more corrupted by the averaging process, and thus optimized parameters bounds can be computed and the reliability of the parameters value can be assessed.

In the future, more efforts should be put to robustify the developed methodology. Indeed, even if the achieved results are encouraging, the estimated parameter values are too far from the values that were found in the literature, and commonly accepted. To overcome this limitation, it can be thought, to change the objective function of the NLP. But it should be emphasize, that if one is able to compute confidence bounds on the estimated parameters, then it would be easier to compare them to the ones present in the literature.

In summary, a clean formulation of the models and of the optimization problem is now available, and the models' nonlinearities were successfully implemented.

References

[1] Andersson J, Gillis J, Diehl M. *User documentation for CasADi v3.4.4* (May 19, 2018).

[2] Assländer L, Hettich G, Gollhofer A, Mergner T. Contribution of visual velocity and displacement cues to human balancing of support surface tilt. *Exp Brain Res* (2013) 228:297-304. doi: 10.1007/s00221-013-3561-x.

[3] Assländer L, Peterka RJ. Sensory reweighting dynamics in human postural control. *J Neurophysiol* (2014) 111:1852-1864. doi: 10.1152/jn.00669.2013.

[4] Assländer, L., et al. Visual contribution to human standing balance during support surface tilts. *Human Movement Science* (2015), `http://dx.doi.org/10.1016/j.humov.2015.02.010`.

[5] Balavoine, M. (2019, 24 of June). *Le systeme vestibulaire.* Consulted on `https://pulsations.hug-ge.ch/article/le-systeme-vestibulaire`.

[6] Cenciarini M, Peterka RJ. Stimulus-dependent changes in the vestibular contribution to human postural control. *J Neurophysiol* (2006) 95: 2733-2750. doi: 10.1152/jn.00856.2004.

[7] Diehl M. (2014). *Lecture notes on optimal control and estimation.* Freiburg-im-Briesgau : Uni-Freiburg, Albert-Ludwigs-Universität Freiburg.

[8] Diehl M. (2018). *Lecture Notes on Modelling and System Identification.* Freiburg-im-Breisgau : Uni-Freiburg, Albert-Ludwigs-Universität Freiburg.

[9] Diehl M. (2019). *Skript zur Vorlesung "Systemtheorie und Regelungstechnik 1".* Freiburg-im-Breisgau : Uni-Freiburg, Albert-Ludwigs-Universität Freiburg.

[10] Diehl M, Gros S. (2019). *Numerical Optimal Control.* Freiburg-im-Breisgau : Uni-Freiburg, Albert-Ludwigs-Universität Freiburg.

[11] Ernst MO, Banks MS. Humans integrate visual and haptic information in a statistically optimal fashion. *Nature* (2002), 215:429-433.

[12] Hettich G, Assländer L, Gollhofer A, Mergner T. Human hip-ankle coordination emerging from multisensory feedback control. *Human Movement Science* (2014), 37, 123-146.

[13] Johansen T.A. Constrained and regularized system identification. *IFAC Symposium on System Identification, Kitakyushu, Japan* (1997), pp. 1405-1410.

[14] Karimi A. (2015). *System Identification.* Lausanne : EPFL, Ecole Polytechnique Fédérale de Lausanne.

[15] Ljung L., Chen T. What can regularization offer for estimation of dynamical systems? *IFAC International Workshop on Adaptation and Learning in Control and Signal Processing, ALCOSP13, Caen, France* (2013).

[16] Maurer C, Mergner T, Peterka RJ. Multisensory control of human upright stance. *Exp Brain Res* (2006), 171: 231-250. doi: 10.1007/s00221-005-0256-y.

[17] Mergner T, Siebold C, Schweigart G, Becker W. Human perception of horizontal trunk and head rotation in space during vestibular and neck simulation. *Exp Brain Res* (1991), 85:389-404.

[18] Mergner T, Huber W, Beckert W. Vestibular-neck interaction and transormation of sensory coordinates. *J Vest Res* (1997), 7, 119-135.

[19] Mergner T., Maurer C., Peterka R.J. (2002) *Sensory Contributions to the Control of Stance.* In: Gandevia S.C., Proske U., Stuart D.G. (eds) Sensorimotor Control of Movement and Posture. Advances in Experimental Medicine and Biology, vol 508. Springer, Boston, MA

[20] Mergner T, Maurer C, Peterka RJ. A multisensory posture control model of human upright stance. *Progr Brain Res* (2003), 142, 189-201.

[21] Mergner T, Schweigart G, Fennell L. Vestibular humanoid postural control. *Journal of Physiology* (2009), *Paris, 103*(3-5), 178-194.

[22] Mergner T. A neurological view on reactive human stance control. *Annual Reviews in Control* (2010), 34(2), 177-198.

[23] Peterka RJ. Sensory integration in human postural control. *J Neurophysiol* (2002) 88:1097-1118. doi: 101152/jn.00605.2001.

[24] Peterka RJ. Simplifying the complexities of maintaining balance. *IEEE Engineering in Medicine and Biology Magazine* (2003) 22:63-68. doi: 10.1109/MEMB.2003.1195698.

[25] Peterka RJ, Loughlin PJ. Dynamic regulation of sensorimotor integration in human postural control. *J Neurophysiol* (2004) 91:410-423. doi: 10.1152/jn.00516.2003.

[26] Peterka RJ, Murchison CF, Parrington L, Fino PC, King LA. Implementation of a central sensorimotor integration test for characterization of the human balance control during stance. *Front. Neurol.* (2018) 9:1045. doi: 10.3389/fneur.2018.01045.

[27] J. B. Rawlings, D. Q. Mayne, M. M. Diehl. *Model predictive control: theory, computation, and design.* Nob Hill Publishing, Madison, Wisconsin, second edition, 2017.

[28] Smerlser NJ, Baltes PB. *International encyclopedia of social and behavioral sciences.* Pergamon, first edition, 2001.

[29] van der Kooij H, Jacobs R, Koopman B, van der Helm F. An adaptive model of sensory integration in a dynamic environment applied to human stance control. *Biol Cybern* (2001), 84, 103-115.

[30] van der Kooij H, Peterka RJ. Non-linear stimulus-response behavior of the human stance control system is predicted by optimization of a system with sensory and motor noise. *J Comput Neurosci* (2011), 30:759-778. doi: 10.1007/s10827-010-0291-y.

[31] Wächler A. *Short tutorial: getting started with Ipopt.* IBM T.J. Watson Research Center, NY, USA.

[32] DA. Winter. *Biomechanics and motor control of human movement.* Wiley, New York, 1990.

[33] Winter DA, Patla AE, Prince F, Ishac M, Gielo-Perczak K. Stiffness control of balance in quiet standing. *J Neurophysiol* (1998) 80: 1211-1221.