Thèse de doctorat

Présentée en vue d'obtenir le grade de

Docteur de l'Université Polytechnique Hauts-de-France

En **Automatique**

Présentée et soutenue par Guoxi FENG

Le 04/11/2019, à Valenciennes

Mobility aid for the disabled using unknown input observers and reinforcement learning

Aide à la mobilité des PMR : une approche basée sur les observateurs à entrée inconnue et l'apprentissage par renforcement

JURY:

Président du jury

Jean-Philippe Lauffenburger. Professeur, Université de Haute-Alsace.

Rapporteurs

Luc Dugard. Directeur de Recherche CNRS GIPSA-Lab Grenoble, France.

Ann Nowé. Professeur, Université Libre de Bruxelles, Belgique.

Examinateurs

Bruno Scherrer. Maitre de conférence, Université de Lorraine, France.

Sami Mohammad, Docteur, président Autonamad Mobility

Directeur de thèse

Thierry-Marie Guerra. Professeur, Université Polytechnique de Hauts-de-France, France.

Lucian Busoniu. Professeur, Technical University of Cluj-Napoca, Romania.

Thèse préparée dans le Laboratoire LAMIH (UMR CNRS 8201)

Ecole doctorale : Science Pour l'Ingénieur (SPI 072)

Le projet de cette thèse bénéfice le soutien financier de la région Hauts-de-France

Abstract

In aging societies, improving the mobility of disabled persons is a key challenge for this century. With an elderly population estimated at over 2 billion in 2050 (OMS 2012), the heterogeneity of disabilities is becoming more important to address. In addition, assistive devices remain quite expensive and some disabled persons are not able to purchase such devices. In this context, we propose an innovative idea using model-based automatic control approaches and model-free reinforcement learning for a Power-Assisted wheelchair (PAW) design. The proposed idea aims to provide a personalized assistance to different user without using expensive sensors, such as torque sensors. In order to evaluate the feasibility of such ideas in practice, we carry out two preliminary designs.

The first one is a model-based design, where we need to exploit as much as possible the prior knowledge on the human-wheelchair system to not use torque sensors. Via an observer and a mechanical model of the wheelchair, human pushing frequencies and direction are reconstructed from the available velocity measurements provided by incremental encoders. Based on the reconstructed pushing frequencies and direction, we estimate the human intention and a robust observer-based assistive control is designed. Both simulation and experimental results are presented to show the performance of the proposed model-based assistive algorithm. The objective of this first design is to illustrate that the need of expensive torque sensors can be removed for a PAW design.

A second design developed in this work is to see the capabilities of learning techniques to adapt to the high heterogeneity of human behaviours. This design results in a proof-of-concept study that aims to adapt heterogeneous human behaviours using a model-free algorithm. The case study is based on trying to provide the assistance according to the user's state-of-fatigue. To confirm this proof-of-concept, simulation results and experimental result are performed.

Finally, we propose perspectives to these two designs and especially propose a framework to combine automatic control and reinforcement learning for the PAW application.

Keywords: Observer, reinforcement learning, disabled persons, mobility, Power-assistive wheelchair, assistive control.

RESUME

Dans les sociétés vieillissantes, l'amélioration de la mobilité des personnes handicapées est un défi majeur pour ce siècle. Avec une population âgée estimée à plus de 2 milliard d'habitants en 2050 (OMS 2012), l'hétérogénéité des handicaps devient de plus en plus importante. En outre, les appareils fonctionnels restent assez coûteux et certaines personnes handicapées ne sont pas en mesure de les acheter. Dans ce contexte, nous proposons une innovante utilisant des approches de contrôle automatique basées sur un modèle et des approches d'apprentissage par renforcement sans modèle pour notre conception de fauteuil roulant assisté. L'idée proposée vise à fournir une assistance personnalisée à un utilisateur particulier sans utiliser de capteurs coûteux, tels que des capteurs de couple. Afin de pré-évaluer la faisabilité de telles idées dans la pratique, nous effectuons deux études préliminaires.

Le premier concerne une conception basée sur modèle, où nous devons exploiter au maximum les connaissances préalables du système de fauteuil roulant humain. Via un observateur et un modèle mécanique du fauteuil roulant, les fréquences et la direction de poussée humane sont reconstruites à partir des mesures de vitesse disponibles fournies par les encodeurs incrémentaux. Sur la base fréquences et de la direction de poussée reconstituées, nous estimons l'intention de l'homme et un contrôle assisté robuste basé sur un observateur a été conçu la simulation et les résultats expérimentaux sont présentés pour montrer les performances de l'algorithme d'assistance proposé basé sur un modèle. L'objectif de la première conception est d'illustrer que le besoin de capteurs de couple coûteux peut être supprimé pour une conception PAW.

Une deuxième idée développée dans ce travail est de voir les capacités des techniques d'apprentissage à s'adapter à la grande hétérogénéité des comportements humains. Il en résulte une étude de validation de concept visant à adapter les comportements humain hétérogènes à l'aide d'un algorithme sans modèle. Le cas d'étude est basé sur l'essai de fournir une assistance en fonction de l'état de fatigue de l'utilisateur. Les preuves de convergence de tels algorithmes d'assistances sont également des questions importantes abordées dans cette thèse. Pour confirmer cette validation de concept, des résultats de simulation et des résultats expérimentaux sont effectués.

Enfin, nous proposons des perspectives pour ces travaux et, en particulier, un cadre combinant contrôle automatique et apprentissage pour l'application PAW.

Mots-clés : observateur, apprentissage par renforcement, personnes handicapées, mobilité fauteuil roulant à assistance électrique.

RERMERCIEREMENT

Je remercie tout d'abord Professeur Luc Dugard et Professeur Ann Nowé, qui m'ont fait l'honneur d'accepter d'être les rapporteurs de cette thèse. Vos remarques et questions m'ont beaucoup aidé à améliorer ce manuscrit. Merci également à Professeur Jean-Philippe Lauffenburger, Maitre de conférence Bruno Scherrer et Docteur Sami Mohammad, les examinateurs de mon jury de thèse pour l'intérêt qu'ils ont porté à mon travail.

Je remercie également la Région Hauts-de-France qui m'a attribué une bourse de recherche durant ces trois années de thèse. Ce financement m'a permis de me concentrer entièrement à mon sujet de recherche. Cette thèse n'aurait jamais pu aboutir sans cette aide.

Je tiens à exprimer toute ma reconnaissance à mes Directeurs de thèse Thierry-Marie Guerra et Lucian Busoniu, pour m'avoir encadré et m'avoir guidé dans le monde de recherche pendant ces trois années enrichissantes. Je les remercie également pour leurs conseils constructifs qui m'ont permis d'améliorer la qualité de mes travaux de recherche.

Un immense merci à tous mes collègues du laboratoire : Isabelle, Mélanie, Mikael, Braulio, Tariq, Juan Carlos, Walid, Camila, Jimmy, Jérémie, Mathias et tous ceux que j'oublie. Merci pour votre soutien administrative et technique. Je voudrais remercier particulièrement Anh-Tu pour les discussions sur la recherche pendant le travail, les repas au cantine et les trajets en tram.

Merci également à Authony qui m'a accompagné depuis le premier jour de mon arrivée au campus universitaire. Sans qui, je n'aurais pas pu choisir le cursus automatique et aboutir cette thèse en automatique.

Finalement, je voudrais remercier ma famille et ma copine pour votre précieux soutien mental qui m'a beaucoup aidé pendant la phase de rédaction.

Acknowledgements

First of all, I would like to thank Professor Luc Dugard and Professor Ann Nowé, who accept to be the rapporteurs of this thesis. Your comments and questions helped me a lot to improve this manuscript. Special thanks go to Professor Jean-Philippe Lauffenburger, Associate Professor Bruno Scherrer and Dr. Sami Mohammad for being the examiners of my thesis and their interest in my work.

I would like to thank the Region Hauts-de-France who awarded me a research grant during three years of thesis. This funding allowed me to focus entirely on my research topic. This thesis could never have succeeded without this help.

I wish to express my gratitude to my Thesis Directors Thierry-Marie Guerra and Lucian Busoniu, for having supervised and guided me in the world of research during these unforgettable three years. I also thank them for their constructive advice which allowed me to improve the quality of my research work.

Huge thanks go to all my colleagues in the laboratory: Isabelle, Mélanie, Mikael, Braulio, Tariq, Juan Carlos, Walid, Camila, Jimmy, Jérémie, Mathias all those I forget.

Thank you for your administrative and technical support. I would like to especially thank Anh-Tu for the discussions on research during work, canteen meals and tram journeys.

Many thanks go also to Authony who has accompanied me since the first day of my arrival at the university campus. Without you, I could not have chosen the course of automated control and begin a thesis in this field.

Finally, I would like to thank my family and my girlfriend for your valuable mental support which helped me a lot during the redaction of the thesis.

Table content

List of Figures

List of Tables

Chapter 1. Introduction

1.1 Context & motivation of the thesis

The 2011 world report of the World Health Organization (WHO) states that "*About 15% of the world's population lives with some form of disability, of whom 2-4% experience significant difficulties in functioning*". This global disability prevalence is higher than previous WHO estimates, which date from the 1970s and suggested a figure of around 10%. Global disability is on the rise due to population ageing and the rapid spread of chronic diseases. With an elderly population estimated at over 2 billion in 2050 (OMS 2012), the heterogeneity of disabilities is becoming more important to address and the issue of mobility is fundamental.

For developed countries, the mobility of disabled persons is therefore a key challenge for this century. Today's existing solutions (Faure 2009), for example manual, electric wheelchair and/or assistance tools; are neither suited to ageing not address the highly heterogeneous human factors i.e. human fatigue dynamics, human pushing strategies (Poletti 2008).

To solve the issue of mobility, advanced work in assistive technologies, such as exoskeleton robotic suits, power-assisted wheelchair, etc., is deeply committed in recent years. In addition, mobility aid is increasingly democratized with more affordable technologies. However, assistive devices remain quite expensive and some disabled persons are not able to purchase such devices. Therefore, reducing the cost of assistive devices provides a better access to mobility for disabled people.

Heterogeneous human behaviours are common, e.g. important differences (extra individual) of propelling according to the physical power of the PRM, possible dissymmetry, decrease of abilities due to ageing; intra individual behaviour modifications are also to be considered, they appear over a long trip or after an intensive physical exercise or are due to particular physical conditions (fatigue, stress). Thus, it is necessary to build assistances that can manage these various kind of states, resulting in very different human propulsion ability. These assistances should be based on limited real-time measurements and propose solutions to an optimal mobility seamlessly to the users.

From this perspective, we seek new innovative solutions that:

- Replace expensive sensors by “software” sensors to reduce the cost of assistive devices.
- Adapt to the disability level of each person based on software strategies (extra individual component);
- For a given user (intra-individual component), adapt the strategies according to his/her behaviour, both in the long term (for example degeneration) and in the short term (e.g. fatigue);
- Are robust and efficient: via minimal information (weight of the disabled person, size of the wheelchair…) the assistance adapts itself transparently to the users without changing any hardware;

1.2 Thesis scope

This work aims to design an “intelligent” (understood as software adaptive solution with no extra sensors) assistive control for a power-assisted wheelchair (PAW) application. From a scientific point of view, we are faced to a problem with highly heterogeneous human and wheelchair dynamics, including signals with various frequencies and powers (human propelling torques) that are not directly measured etc. Therefore, the use of classical model-based approaches of automatic control to deal with such heterogeneous systems appears difficult. Effectively, if these approaches need such a precision that they require the modelling of the human fatigue + wheelchair, there is lillte chance that the solutions would be interesting (generalizable, robust, performant) in view of the heterogeneity discussed.

One way to avoid a precise modelling is to use model-free reinforcement learning techniques (Modares et al. 2014) and see their potential. Therefore, one originality of this work is to use multidisciplinary knowledge, such as model-based automatic control and model-free reinforcement learning, to test their capabilities and limits.

In order to pre-evaluate the feasibility of such ideas in practice, we carry out two preliminary designs in collaboration with SME Autonomad Mobility, which has significant expertise in the mobility of disabled people.

The first one is concerned with a conventional automatic control design, where we need to exploit as much as possible the prior knowledge of the human-wheelchair system. It must be kept in mind that a precise model is definitively unrealistic to propose, as explained previously. The challenge is to know if a rather simplified model of the wheelchair and human would be enough to propose some solutions. Based on this simplified model, an unknown input observer (Koenig 2005, Estrada-Manzo et al. 2015) has been designed. Via this observer, human torque signals are estimated from the available velocity measurements provided by encoders. Of course, due to the simplicity of the model, the reconstructed signals are not fully reliable, especially in amplitude. Nevertheless, from these signals the propelling frequency as well as the direction are satisfactorily reconstructed. Based on these variables, reference signals are computed (center and yaw velocities of the wheelchair) via a generation module, that are expected to estimate the user's intention. The tracking of reference velocities is intended to work in presence of uncertainties such as mass (user and wheelchair) and road conditions (viscous friction, slope); therefore, a robust observer-based tracking controller has been designed. Finally, both simulation and experimental results are presented to show the performance of the proposed model-based assistive algorithm. The first study show the possibility to remove the need of expensive torque sensors.

A second idea developed in this work is to see the capabilities of learning techniques to adapt to the high heterogeneity mentioned previously. It results in a proof-of-concept study that aims to adapt heterogeneous human behaviours using a model-free algorithm. The study case is based on trying to provide the assistance according to the user's state-of-fatigue. This state-of-fatigue may vary for the user through time (intra individual variation) or be different according to the user under consideration (extra individual variation). The proposed model-free assistive algorithm aims to obtain a (near-)optimal assistance for a particular user. Proofs of convergence of such algorithms are also important issues that are provided in this work. To confirm this proof-of-concept, simulation results and experimental result are performed.

Interestingly, the two approaches give results that can be seen as complementary. Instead of using a kind of black-box learning, a grey-box learning could be an interesting solution to explore. It could combine the advantages of both techniques. For example, in the former solution developed, a robust and performant control has been derived that allows following predefined trajectories. "Learning" the way to compute these trajectories from the user would be an interesting challenge. This would deliver, a more "personalized" assistance, it could be the right place for learning. This idea is developed as a perspective of the work.

1.3 Structure of the thesis

The manuscript is decomposed in seven chapters:

Chapter 2 provides a literature review on the mechanical model of a wheelchair and both the model-based control and the model-free control techniques, that will be applied to the PAW design. The prototype used for experimental validations is also introduced.

Chapter 3 introduces a model-based assistive control, which consists of an unknown input observer, a reference generation module and finally a robust observer-based tracking controller. Simulation results are carried out to validate the design of each part. In addition, the proposal of an observer design using time-varying sampling rate is also given.

Chapter 4 provides experimental results, which aims to validate the whole model-based assistive control under a constant sampling rate of Chapter 3.

Chapter 5 proposes a completely different point of view and intends to give a proof-of-concept study to show that the adaptability to heterogeneous human behaviours, such as human fatigue evolution, is possible using a model-free reinforcement learning method. Real-time experiments are carried out to support this proof-of-concept.

Chapter 6 concludes the work and proposes perspectives to this work and especially proposes to combine control and learning for the PAW application.

1.4 Publications

The research carried out within this thesis has already led to several published contributions in both theory and application:

International Journals

- **G. Feng**, L. Buşoniu, T.M. Guerra, S. Mohammad (2019) – Data-Efficient Reinforcement Learning for Energy Optimization Under Human Fatigue Constraints of Power-Assisted Wheelchairs – IEEE Transactions on Industrial Electronics, Special Section on: Artificial Intelligence in Industrial System, 66 (12), 9734-9744 (IF 7.05)

International Conferences

- **Feng, G.**, Guerra, T. M., Nguyen A. T., Busoniu, L., & Mohammad, S. "Robust Observer-Based Tracking Control Design for Power-Assisted Wheelchairs". 5th IFAC Conference on Intelligent Control and Automation Sciences 21-23 August 2019, Belfast, Northern Ireland
- **Feng, G.**, Buşoniu, L., Guerra, T. M., & Mohammad, S. (2018, June). Reinforcement Learning for Energy Optimization Under Human Fatigue Constraints of Power-Assisted Wheelchairs. Annual American Control Conference (ACC) 27-29 June 2018 (pp. 4117-4122). IEEE.
- **Feng, G.**, Guerra, T. M., Mohammad, S., & Busoniu, L. "Observer-Based Assistive Control Design Under Time-Varying Sampling for Power-Assisted Wheelchairs". The 3rd IFAC Conference on Embedded Systems, Computational Intelligence and Telematics in Control June.6-8, 2018, Faro, Portugal IFAC-PapersOnLine, 51(10), 151-156.
- **Feng, G.**, Guerra, T. M., Busoniu, L., & Mohammad, S. "Unknown input observer in descriptor form via LMIs for power-assisted wheelchairs". In *2017 36th Chinese Control Conference (CCC)* (pp. 6299-6304). IEEE.

Workshop

- Guerra, T. M., **Feng, G.**, Buşoniu, L., & Mohammad, S. "An example on trying to mix control and learning: power assisted wheelchair". *2nd Workshop Machine Learning Control (wMLC-2)*, Valenciennes, France, janvier 20.

Chapter 2. Background and state of the art

2.1 Introduction

This chapter provides a brief state of the art on Power-Assisted Wheelchair designs, model-based control designs and model-free control designs. In addition, the wheelchair prototype and its corresponding dynamic model are also introduced.

2.2 Power-assisted wheelchairs

Since disabled people and elderly persons who lose the ability to walk occupy a significant percentage of the population in modern societies (World Health Organization 2011), mobility aids, such as, manual wheelchairs, electric powered wheelchair, and Power-Assisted Wheelchairs (PAW), are available to satisfy some of their mobility requests. The manual wheelchair is a common mean to improve accessibility and mobility for such disabled persons. However, the majority of them may have difficulty to propel a manual wheelchair, due to some physical constraints or difficult road conditions (Cooper et al. 2001). This poor efficiency of manual wheelchairs also causes on the long term, injuries such as joint degradation (Algood et al. 2004). A solution is the use of electric wheelchairs (De La Cruz et al. 2011; M. Tsai and Hsueh 2012), which have been commercialized in the 1950s (BA et al. 2003). Nevertheless, this solution has also poor capabilities according to road conditions, and an unexpected drawback is linked to the pure electrical propelling, resulting in a high decrease of physical activity, pointed out by specialists (Giesbrecht et al. 2009).

An intermediate solution is the so-called power-assisted wheelchairs (PAW), that can provide an alternative choice to the users. Having an electrically powered motor, PAW assistance strategy is designed to reduce the user's physical workload, ideally taking into account his/her physical condition. The medical investigations by Fay and Boninger (2002) Giesbrecht et al. (2009) show the physical and physiological advantages derived from the PAW rather than fully manual or electrical solutions (e.g. moderate metabolic demands of propulsion and maintaining participation in community-based activities among others). In

contrast with manual wheelchairs and electric wheelchairs, PAW combines human and electrical powers and therefore can give a good compromise between rest and exercise for users. Several PAWs are available on the market, amongst which the motorisation kits Duo designed by AutoNomad Mobility, Wheeldrive from Sunrise Medical and MAX Mobility provided by Smartdrive. Figure 2.2.1 presents these kits, which can be installed on most manual wheelchairs and offer good manoeuvrability.

The three PAWs shown in Figure 2.2.1 use different technologies. For the Duo kit, the user can select between two assistance modes that suit his/her wishes and driving conditions. The first mode, called Electric Propulsion Assistance, amplifies human torques which are estimated by an observer, called a software sensor (US20170151109A1 - Method and device assisting with the electric propulsion of a rolling system, wheelchair kit comprising such a device and wheelchair equipped with such a device - Mohammad et al. 2015). The second mode, Single Push, keeps a constant velocity and is convenient for covering long distances. Smartdrive estimates the human intention using a smart watch. With the help from the electric motor, the user combines pushing and different arm gestures (detected by the smart watch) to manipulate the wheelchair. Wheeldrive uses a dual rim concept to deliver the assistive torque. The assist rim (the big one) is used to generate a power assistance; the drive rime (the small one) is used for a continuous drive. For more information on various commercial PAWs, the reader can refer to detailed literature reviews (D. Ding and Cooper 2005; Simpson 2005). Thanks to a wide range of choices, disabled persons should be able to select a suitable assistive device according to their needs.

Figure 2.2.1. Duo kit, Smartdrive and Wheeldrive commercialized respectively by Autonomad Mobility, MAX Mobility and Sunrise Medical (From left to right)

PAWs research activities are also increasing in the recent years. References Seki et al. (2009), H. Seki and Kiso (2011), Seki and Tadakuma (2006) and Seki et al. (2005) have analysed the impact of different road conditions on the human-wheelchair system. A corresponding control scheme has been implemented to assist the user for each road condition. In (R. A. Cooper et al. 2002; Seki and Tadakuma 2004), the human behaviour and the interaction with the device are studied. Leaman and La (2017) give a complete overview of this field.

Unfortunately, most of the current PAW researches do little to address the highly heterogeneous population of the disabled persons. Adaptability to the person is a key point for PAW assistance design, especially thinking to various intra and extra individual variations, including non-measurable features such as level of disability, fatigue, pain… Combined with the fact that current commercial PAWs are usually expensive; designing an adaptable and affordable PAW is a challenging research project.

2.2.1 PAW prototype

Several prototypes have been designed and built by the Autonomad Mobility company (Start-up created in 2015 by S. Mohamad Doctor from LAMIH UMR CNRS and UPHF laboratory of Valenciennes) to evaluate the validity of PAW assistance designs. The prototype used for experimental validations is shown Figure 2.2.2.

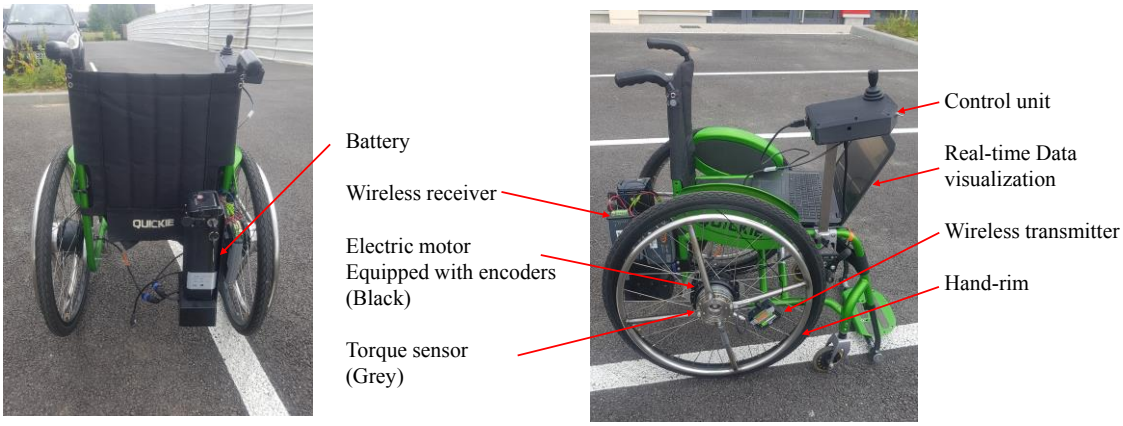

Figure 2.2.2. Wheelchair prototype and its components

The mechatronics structure of the prototype is described Figure 2.2.3. The wheelchair prototype is equipped with two brushless DC motors powered by a DC battery (about 15Km

autonomy range). The maximum torque delivered by DC motors is around 40 Nm. The motors receive control signals (Voltage or current) via a Texas Instruments C2000 real time micro-controller. Using the software Code Composer Studio, C/C++, a code generated by Simulink can be compiled and executed on the microcontroller. Therefore, the algorithms are directly coded using Matlab/Simulink and directly embedded in the microcontroller for the experimental validations. The data acquisition is done using the same microcontroller connected to a laptop. In addition, the data received can be stored in the laptop and/or can be displayed in real time.

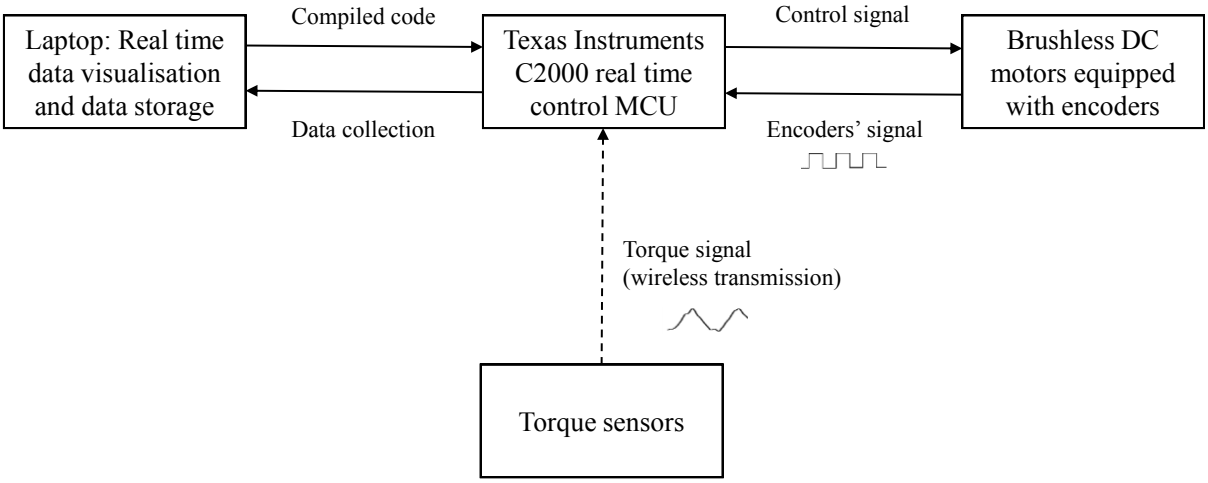

Figure 2.2.3. Mechatronics structure of the wheelchair prototype

The prototype is equipped with the following sensors:

- Two incremental encoders to measure the angular velocity of each motor; outputs are pulse signals. The number of pulses is counted for a given time interval (sampling time) in order to determine the relative position between two consecutive measurements.

- Two torque sensors with wireless transmissions are supplied by CapInstrumentation. Figure 2.2.2 shows their installation on the wheel axis to measure the human torques exerted on the push-rims using strain gauges located in rotating shafts. When the user exerts a torque on the push-rims or rotating shafts, strain gauges are deformed and cause their electrical resistances to change. To avoid any cable connection between the moving wheels and the seat, the transmitters of the torque sensors are placed on the wheels and their receivers are installed on the back of the seat.

Remark 1. In order to be clear for the reader, the torque sensors are not available for the Duo kit sold, they would render the kit too expensive. Nevertheless, they are very important for our work as they will be used to validate the designs, showing that the methodologies used, especially the observation part, are suitable without these extra sensors.

2.2.2 Dynamical modeling of PAWs

To design model-based assistive controls, a model of the wheelchair is needed. In the literature, several dynamic models have been developed for different control purposes. The dynamic model of Shung et al. (1983) describes the wheelchair motion on a sloping surface and is used to design a velocity feedback controller. The model of De La Cruz et al. (2011) takes into account the casters dynamic. Based on this model, an adaptive control law has been proposed for trajectory tracking. The 3D dynamic model of Aula et al. (2015) has been used for stabilizing the wheelchair in a two wheel balancing mode.

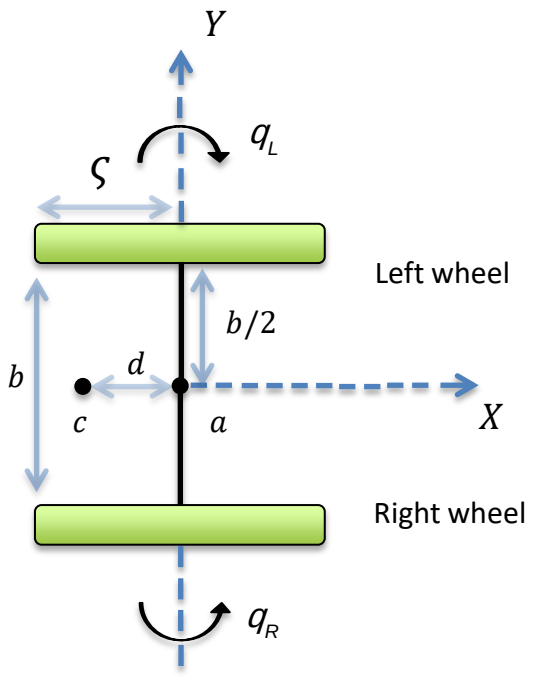

Figure 2.2.4. Simplified top view of the wheelchair

Table I. SYSTEM PARAMETERS

Symbol	Description	Value

ς	Wheel radius [m]	0.33
b	Distance between two wheels [m]	0.6
d	Distance between the point a and the point c [m]	0.4
c	centre of gravity of the wheelchair with the human	-
a	Middle point between two wheels	-
m	Mass of wheelchair including the human [kg]	100
$\mathcal{K}$	Viscous friction coefficient [$\mathrm{N}\cdot\mathrm{m}\cdot\mathrm{s}$]	5
I_a	Inertia of the wheelchair with respect to the vertical axis through the point a [$\mathrm{kg}\cdot\mathrm{m}^2$]	16
I_0	Inertia of each driving wheel around the wheel axis [$\mathrm{kg}\cdot\mathrm{m}^2$]	0.25
T_e	Sampling time [s]	0.05

The wheelchair studied is modelled as a two-wheeled transporter, see Figure 2.2.4. The physical parameters of the prototype used in this work are given Table I. The two-wheeled PAW is described by the dynamics (M. Tsai and Hsueh 2012; Tsuyoshi et al. 2008):

$$\begin{aligned}\boldsymbol{\alpha}\dot{\boldsymbol{\theta}}_R+\boldsymbol{\beta}\dot{\boldsymbol{\theta}}_L&=T_{mr}+T_{hr}-\mathrm{K}\boldsymbol{\theta}_R\\ \boldsymbol{\alpha}\dot{\boldsymbol{\theta}}_L+\boldsymbol{\beta}\dot{\boldsymbol{\theta}}_R&=T_{ml}+T_{hl}-\mathrm{K}\boldsymbol{\theta}_L\end{aligned} \tag{2.2.1}$$

where

$$\begin{aligned}\boldsymbol{\alpha}&=\frac{mr^2}{4}+\frac{\left(I_a+md^2\right)\varsigma^2}{b^2}+I_0\\ \boldsymbol{\beta}&=\frac{mr^2}{4}-\frac{\left(I_a+md^2\right)\varsigma^2}{b^2}\end{aligned} \tag{2.2.2}$$

The total torques consist of the human torques T_{hr}, T_{hl} and the assistive torques T_{mr}, T_{ml} given by the electrical motors. The left angular velocity and the right angular velocity are respectively $\boldsymbol{\theta}_L$ and $\boldsymbol{\theta}_R$. Using Euler's approximation with $\dot{\boldsymbol{\theta}}_R(t)=\left(\boldsymbol{\theta}_R{}^{+}-\boldsymbol{\theta}_R\right)/T_e$ and $\dot{\boldsymbol{\theta}}_L(t)=\left(\boldsymbol{\theta}_L{}^{+}-\boldsymbol{\theta}_L\right)/T_e$, a discrete-time model of the mechanical system (2.2.1) can be obtained and written in state-space representation as follows:

$$\begin{bmatrix} \boldsymbol{\alpha} & \boldsymbol{\beta} \\ \boldsymbol{\beta} & \boldsymbol{\alpha} \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_R^{+} \\ \boldsymbol{\theta}_L^{+} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\alpha} - T_e \mathrm{K} & \boldsymbol{\beta} \\ \boldsymbol{\beta} & \boldsymbol{\alpha} - T_e \mathrm{K} \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_R \\ \boldsymbol{\theta}_L \end{bmatrix} + T_e \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \left(\begin{bmatrix} T_{mr} \\ T_{ml} \end{bmatrix} + \begin{bmatrix} T_{hr} \\ T_{hl} \end{bmatrix} \right) \tag{2.2.3}$$

Note that $\boldsymbol{\theta}_R^{+}$ and $\boldsymbol{\theta}_L^{+}$ stands for $\boldsymbol{\theta}_R(k+1)$ and $\boldsymbol{\theta}_L(k+1)$ respectively.

In particular, the velocity ω of the centre of gravity and the yaw velocity φ of the wheelchair are the two basic motions naturally and implicitly used by an individual as controlled variables for a desired trajectory. These variables can be computed from the angular velocity $\boldsymbol{\theta}_L$ and $\boldsymbol{\theta}_R$ as follow:

$$\begin{bmatrix} \omega \\ \varphi \end{bmatrix} = \begin{bmatrix} \frac{\varsigma}{2} & \frac{\varsigma}{2} \\ \frac{\varsigma}{b} & -\frac{\varsigma}{b} \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_R \\ \boldsymbol{\theta}_L \end{bmatrix} \tag{2.2.4}$$

Using the transformation (2.2.4), the mechanical system (2.2.3) can be rewritten in the following discrete-time descriptor form:

$$\begin{aligned} Ex^{+} &= Ax + Bu_h + Bu_m \\ y &= Cx \end{aligned} \tag{2.2.5}$$

with the state vector $x = [\omega, \varphi]^T$, the human torques $u_h = [T_{hr}, T_{hl}]^T$, the motor torques $u_m = [T_{mr}, T_{ml}]^T$ and the outputs $y = [\boldsymbol{\theta}_R, \boldsymbol{\theta}_L]^T$. As usual, x^{+} stands for $x(k+1)$. The corresponding matrices are:

$$E = \begin{bmatrix} \boldsymbol{\alpha} & \boldsymbol{\beta} \\ \boldsymbol{\beta} & \boldsymbol{\alpha} \end{bmatrix} \begin{bmatrix} \varsigma/2 & \varsigma/2 \\ \varsigma/b & -\varsigma/b \end{bmatrix}, A = \begin{bmatrix} \boldsymbol{\alpha} - T_e \mathrm{K} & \boldsymbol{\beta} \\ \boldsymbol{\beta} & \boldsymbol{\alpha} - T_e \mathrm{K} \end{bmatrix} \begin{bmatrix} \varsigma/2 & \varsigma/2 \\ \varsigma/b & -\varsigma/b \end{bmatrix},$$
$$B = T_e I_2, C = \begin{bmatrix} 1/\varsigma & b/(2\varsigma) \\ 1/\varsigma & -b/(2\varsigma) \end{bmatrix}.$$

For system (2.2.5), the number of states $n_x = 2$, the number of control inputs $n_u = 2$ and the number of outputs $n_y = 2$.

Remark 2. In the descriptor system (2.2.5), all the inertial parameters are on the left hand-side of the equation. Compared to the conventional state-space form, the descriptor form preserves the physical interpretation of mechanical systems. Due to the "natural" descriptor form of the wheelchair, this form is kept for most control designs in this thesis.

Remark 3. The mechanical model (2.2.1) does not take into account the casters dynamic, the road conditions (change in the viscous friction), or the users variability (mass, inertia). We have to keep in mind that these non-modelled dynamics and uncertainties change the behaviour of the wheelchair. However, we expect that the two-wheeled model (2.2.1) is enough to capture the main behaviour for motion control designs.

2.3 Nonlinear control

Control of nonlinear systems has been deeply investigated. Significant theoretical progress provides powerful control techniques to solve nonlinear problems, such as model predictive control (Mayne et al. 2000), linear parameter-varying control (C. W. Scherer 2001) or sliding mode control (Levant 1993) etc. This part gives a quick review of the control techniques used thereafter in this work.

2.3.1 Unknown input observer

Unknown variables, including inputs such as driver torque (Nguyen et al. 2018) or fouling in a heat exchanger (Delrot et al. 2012), are common in industrial applications and make automatic control designs more challenging. Unknown inputs can be non-measurable, for example human body torques produced would need invasive sensors (Blandeau et al. 2018) or are expensive to measure with commercial sensors. Removing these sensors reduce the costs and can give a competitive advantage. However, the real-time information of unknown inputs is crucial for controller design and high level strategies. To overcome this problem, unknown input observers (UIO) can be applied, as an alternative solution, to estimate jointly the state of the system and the unknown inputs. In the literature, different classes of unknown input observers exist and for a detailed state-of-the-art the reader can refer to the overview (Chen et al. 2016).

In the works of Chadli et al. (2013), Chibani et al. (2016), the authors decouple the influence of unknown inputs on the state estimation such that the dynamic of the estimation error asymptotically converges (Darouach et al. 1994). This decoupling technique is extensively used for fault detection. Note that a perfect unknown-input decoupling is not always possible. In this case, (Marx et al. 2007) minimise the L_2-norm from the unknown input to the

estimation error. However, the human torque u_h, considered as an unknown input, acts on the system (2.2.5) in the same place as the control input u_m. Therefore, the decoupling technique may not be applicable for the model used for the PAW application.

The second framework is the frequency domain UIO design which was initially proposed by Ohishi et al. (1987) for a DC motor application. The simplified diagram of this approach is depicted in Figure 2.3.1, where the linear transfer function $\mathcal{G}(s)$ represents the real system dynamics and $\mathcal{G}_n(s)$ is the mathematical model available for the control design. For the consistency of the notation, u_m and u_h denote the control input and the unknown input respectively. Then, the estimated unknown input can be expressed as follows:

$$\hat{u}_h(s) = \left[\mathrm{G}(s)^{-1} - \mathrm{G}_n(s)^{-1} \right] y(s) + u_h(s) \tag{2.3.1}$$

In the absence of measurement noise, the estimated unknown input $\hat{u}_h$ captures together the modelling error and the unknown input. If we have the exact model of the physical system i.e $\mathrm{G}(s)^{-1} - \mathrm{G}_n(s)^{-1} = 0$, the unknown input can be perfectly reconstruct. In addition, a filter can be used to reduce measurement noise. Applying the filtered estimation to a feedback control, the modelling error and the unknown input can be attenuated efficiently in real-time applications (Tsai and Hsueh 2013; Umeno et al 1993).

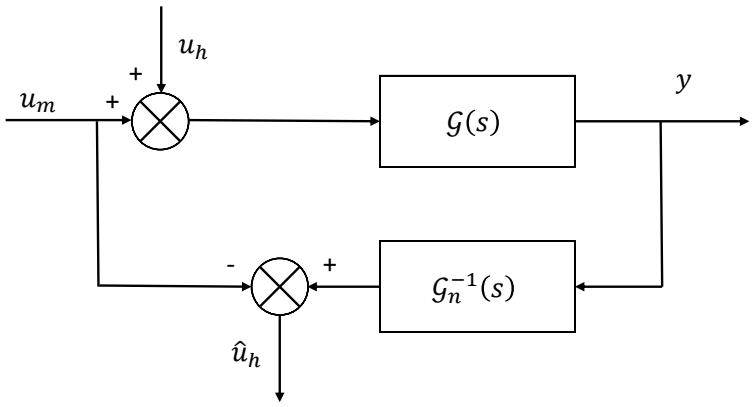

Figure 2.3.1. Frequency domain UIO design (Chen et al. 2016)

The third framework is based on the Luenberger observer (Luenberger 1971) and the so-called unknown input PI-observer (Ichalal et al. 2009). It assumes that the dynamics of the unknown input u_h can be captured with a cascade of integrators, its n_p^{th} variation can be

considered null, i.e. $\frac{d^{n_p} u_h}{dt^{n_p}} \approx 0$. Therefore the unknown input u_h and its derivatives $u_h^{(i)}$, $i \in \{1,\ldots,n_p - 1\}$ are part of an extended state vector that is integrated in the PI-observer. This technique has been successfully applied to real-time applications (Blandeau et al. 2018; Han et al. 2017; Thieffry et al. 2019).

To reconstruct unknown inputs, a fourth framework is based on the sliding mode concept. For the detailed design procedure, the reader can refer to (Floquet et al. 2007) and (Kalsi et al. 2010). The drawback of this approach is the chattering effect on the estimated information which deteriorates the precision of the controller. In the presence of measurement noise, the chattering effect can have a bad impact for real-time applications and filters have to be added.

2.3.2 Takagi-Sugeno model and Polytopic representation

Linear Parameter Varying (LPV) or quasi-LPV or the so-called Takagi-Sugeno (T-S) fuzzy models have attracted numerous researches. When required, the framework thereafter will refer to T-S models that use a polytopic representation. They were initially proposed by Takagi and Sugeno (Takagi et al. 1993). It is proved that the convex structure of T-S model can exactly represent any smooth nonlinear system (Fantuzzi et al, 1996). Thanks to its convex structure, a systematic methodology based Lyapunov function has been established for nonlinear state feedback/output feedback controllers and for observer designs. Generally, the goal is to write the problems as Linear-Matrix Inequality (LMI) constraints problem that can be solved efficiently by existing mathematical toolbox, such as LMI Matlab toolbox and Yalmip.

The following nonlinear system is considered:

$$\begin{aligned} E(z)x^{+} &= A(z)x + B(z)u \\ y &= C(z)x \end{aligned} \tag{2.3.2}$$

where the matrices have the corresponding dimensions. In the linear parameter varying (LPV) control framework, the variable z is not state-dependent and can be partly measurable or not. For q-LPV z can be state-dependent and for the robust control framework, it can represent uncertain time-varying parameters, generally not accessible. A T-S model of the nonlinear system (2.3.2) which is an exact representation in a compact set of the state space, is thus a polytopic representation as:

$$\begin{aligned} \sum_{i=1}^{\mathbf{r}} h_i(z) E_i x^+ &= \sum_{i=1}^{\mathbf{r}} h_i(z)(A_i x + B_i u) \\ y &= \sum_{i=1}^{\mathbf{r}} h_i(z) C_i x \end{aligned} \tag{2.3.3}$$

The matrices A_i, B_i, C_i, E_i, represent **r** linear models. The number of linear models increases exponentially with the number of nonlinearities. (Guerra et al. 2015) give a detailed insight on this computational complexity problem. The nonlinear membership functions $h_i(z)$ can be determined by the sector nonlinearity approach (Taniguchi et al. 2001). Moreover, the membership functions satisfy the convex-sum property i.e. $\sum_{i=1}^{\mathbf{r}} h_i(z) = 1$.

The nonlinear system (2.3.3) is represented by the interpolation of **r** linear models via nonlinear membership functions $h_i(z)$. This property gives the possibility to reuse some linear concepts for stability analysis, LPV control designs and robust control designs.

2.3.3 Lyapunov stability and LMI-based synthesis

Thereafter, both the observer and the controller designs are principally based on Lyapunov framework (Pai 1981). In this framework, a Lyapunov function candidate is required in order to prove the stability of the closed-loop (global or local), the convergence of the estimation and also taking into account some performances (H_2 property, H_∞ attenuation, decay rate and so on). To exhibit this very classical way of doing, we recall the case of a discrete state feedback stabilization. Consider a discrete system with a linear control:

$$\begin{aligned} x^+ &= Ax + Bu \\ u &= -Kx \end{aligned} \tag{2.3.4}$$

together with a quadratic Lyapunov function:

$$V(x) = x^T P x \tag{2.3.5}$$

where $P = P^T \in \mathrm{R}^{n_x}$ is a positive definite matrix, The convergence of x to the origin is ensured if the variation of the Lyapunov function is negative, i.e.:

$$\Delta V(x) = V(x^+) - V(x) < 0 \tag{2.3.6}$$

which means the quadratic Lyapunov function strictly decreases towards zero. With the equalities (2.3.5) and (2.3.4), the inequality (2.3.6) is transformed as the following matrix inequality:

$$(A-BK)^T P(A-BK)-P<0 \tag{2.3.7}$$

Thereafter, the stability analysis is formulated as a LMI constraint optimization problem. Hence, existing powerful LMI tool can be applied for both control and observer designs. The reader can refer to numerous publication in the field and especially the textbooks (BOYD 1994; C. Scherer and Weiland 2015).

Notice that a quadratic Lyapunov function can introduce an important conservativeness, therefore reducing the area of possible solutions. To overcome this drawback, different sophisticated structures for the Lyapunov function, such as delayed non-quadratic Lyapunov functions (Guerra et al. 2012; Lendek, Guerra, and Lauber 2015), can be considered.

Thereafter, the following technical lemmas will be useful for obtaining LMI constraints.

Lemma 1. (Congruence property) given two matrices P and Q, if $P>0$ and Q is a non-singular matrix, the matrix QPQ^T is positive definite.

Lemma 2. (Schur complement) Given two symmetric matrices $P\in \mathrm{R}^{m\times m}$, $Q\in \mathrm{R}^{n\times n}$ and a matrix $X\in \mathrm{R}^{n\times m}$. The following statements are equivalent:

$$\begin{bmatrix} Q & X^T \\ X & P \end{bmatrix}>0 \tag{2.3.8}$$

$$\begin{cases} Q>0 \\ P-XQ^{-1}X^T>0 \end{cases} \Leftrightarrow \begin{cases} P>0 \\ Q-X^TP^{-1}X>0 \end{cases} \tag{2.3.9}$$

Lemma 3. (De Oliveira et al. 2001). Let $X\epsilon\mathbb{R}^n$, $Q=Q^T\epsilon\mathbb{R}^{n\times n}$, and $W\epsilon\mathbb{R}^{m\times n}$ such that $\mathrm{rank}(W)<n$; the following expressions are equivalent:

a) $X^YQX<0, \forall X\epsilon\{X\epsilon\mathbb{R}^n: X\neq 0, WX=0\}$
b) $\exists M\epsilon\mathbb{R}^{n\times m}: MW+W^TM^T+Q<0$

This section focused on classical model-based tools used for the design of controllers. The work proposed thereafter also relies on learning techniques due to the inherent heterogeneity of the problem. Next section recalls the basis of these techniques.

2.4 Reinforcement learning optimal control

Reinforcement learning (RL) searches for an optimal decision by trial and error in an unknown environment. The general framework of RL is depicted in Figure 2.4.1, where an agent learns autonomously to make decisions (take actions) by interacting with the environment. The learning objective is to obtain as much cumulative reward as possible. For an overview, the textbook (Sutton and Barto 2018) gives a complete introduction to RL.

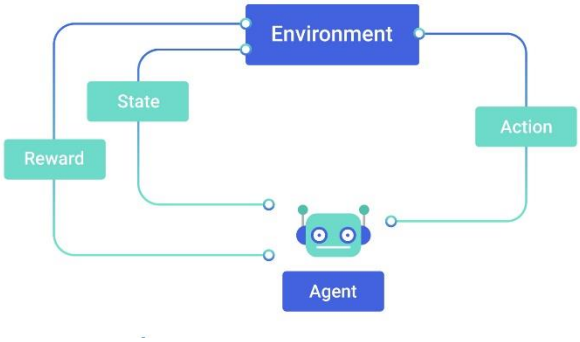

Figure 2.4.1. The conventional framework of Reinforcement learning (Edwards and Fenwick 2016)

The field of RL has exploded in recent years. People from many different backgrounds have started using this framework to solve a wide range of new tasks. The success of AlphaGo (Silver et al. 2016) and AlphaGo Zero (Silver et al. 2017) is considered as a key milestone in the world of reinforcement learning. Besides achievements in artificial intelligence (AI), many research works have been carried out by the control system community to solve optimal control problems by RL techniques. From the viewpoint of control theory, the works of Buşoniu et al. (2018) and Lewis and Vrabie (2009) provide an overview. In addition, more and more research works in robotics focus on RL techniques. Impressive robotic applications using RL can be found in the survey of Kober et al. (2013). The experimental demonstrations such as Lampert and Peters (2012), Maeda et al. (2016), Nair et al. (2018), and Vecerik et al. (2019) show conventional robotic arms are able to perform different tasks i.e. playing table tennis and imitating human behaviours. These practical results show that most existing robots are physically capable of performing a wide range of useful tasks. In most cases, building "intelligent" robots is a software challenge rather than a hardware problem. The successful applications in optimal control and in robotics presented above show that reinforcement learning is one of the most promising approaches to design "intelligent" control software.

Since we apply RL techniques to control in this thesis, next sections provide a quick review of model-free RL from a control engineering viewpoint. Specially, we focus on Policy Search approaches using parametric approximators, since these methods are able to efficiently handle the continuous actions needed for the PAW application.

2.4.1 Basics of reinforcement learning

In the RL framework, a discrete-time optimal control problem is generally formalized as a Markov decision process (MDP) (Howard, 1960), where the next state x_{k+1} is derived from the current state x_k, according to transition function and a chosen action u_k. A MDP is in general a discrete-time stochastic control process. However, we focus here on the deterministic case. The deterministic state transition function can be expressed as follows:

$$x_{k+1} = \xi\left(x_k, u_k\right) \tag{2.4.1}$$

The quality of each chosen action is represented by a stage reward $r(x_k, u_k)$. For example, the stage reward $r(x_k, u_k)$ is a quadratic function of the state x_k and the action u_k. The way to generate this reward depends on the control objective. For a finite-horizon problem, the accumulated reward along a trajectory $\tau = \left(x_0, u_0, x_1, u_1, \ldots x_{K-1}, u_{K-1}, x_K\right)$ is then denoted by:

$$R\left(\tau\right) = \sum_{k=0}^{K-1} \gamma^k r\left(x_k, u_k\right) + \gamma^K T\left(x_K\right) \tag{2.4.2}$$

where $T(x_K)$ is a terminal reward. The term $T(x_K)$ is used to cope with soft constraints on the terminal state. For example, the system is expected to achieve to the desired terminal state. A discount factor $\gamma \in (0, 1]$ may be used; in the finite-horizon case, γ is often taken equal to 1. An infinite-horizon problem can be also considered and its corresponding reward is defined as follows:

$$R\left(\tau\right) = \sum_{k=0}^{K=\infty} \gamma^k r\left(x_k, u_k\right) \tag{2.4.3}$$

with $\gamma \in \left]0,1\right[$, in order that the value of the accumulated reward is finite when the horizon K tends to infinite. The optimal control problem consists of finding a sequence of actions to maximize the accumulated reward (2.4.2) or (2.4.3).

To characterize policies, two value functions, the Q-function and the V-function, are usually defined. Under a policy π, e.g. $u_k = \pi_k(x_k)$, the finite-horizon case with the reward (2.4.2) leads to a time-varying Q-function as follows:

$$\begin{aligned} &Q_k^{\pi}(x_k,u_k) = \gamma^k r(x_k,u_k) + \gamma^{k+1}(x_{k+1},u_{k+1}) + \ldots + \gamma^{K-1}(x_{K-1},u_{K-1}) + \gamma^K T(x_K) \\ &\text{for } k = K-1,\ldots,0 \text{ and } \forall x \in X,\ \forall u \in U \end{aligned} \tag{2.4.4}$$

When $k = K$, $Q_K = \gamma^K T(x_K)$. The Q-function characterizes how good is an action taken in a given state. According to the Bellman optimality principle, the optimal Q-function Q^* is defined as follows:

$$\begin{aligned} &Q_{K-1}^*(x_{K-1},u_{K-1}) = r(x_{K-1},u_{K-1}) + \gamma T\left(\xi(x_{K-1},u_{K-1})\right) \\ &Q_k^*(x_k,u_k) = r(x_k,u_k) + \gamma \max_{u_{k+1}} Q_{k+1}^*\left(\xi(x_k,u_k),u_{k+1}\right), \\ &\text{for } k = K-2,\ldots,0 \text{ and } \forall x \in X,\ \forall u \in U \end{aligned} \tag{2.4.5}$$

where the optimal Q-value is equal to the sum of the immediate reward and the discounted optimal Q-value of the next step obtained by the best action. From the optimal Q-function (2.4.5), the time-varying optimal policy is computed as:

$$\pi^*(x_k,k) = \arg\max_{u_k} Q_k^*(x_k,u_k) \tag{2.4.6}$$

The V-function characterizes how good is to achieve a given state. For the finite-horizon case with the reward (2.4.2), the time-varying V-function is defined for a given policy π as follows:

$$V_k^{\pi}(x_k) = Q_k^{\pi}(x_k,u_k) \tag{2.4.7}$$

where the control action $u_k = \pi_k(x_k)$. The optimal V-function V^* is defined as follows:

$$V_k^*(x_k) = \max_{u_k} Q_k^*(x_k,u_k) \tag{2.4.8}$$

The time-varying optimal policy is computed from V^* as:

$$\pi^*(x_k,k) = \arg\max_{u_k}\left[r(x_k,u_k) + \gamma V_{k+1}^*\left(\xi(x_k,u_k)\right)\right] \tag{2.4.9}$$

Using this MDP formulation, online or offline RL methods solve the problem without model of the system. A taxonomy of model-free RL algorithms is given in Figure 2.4.2.

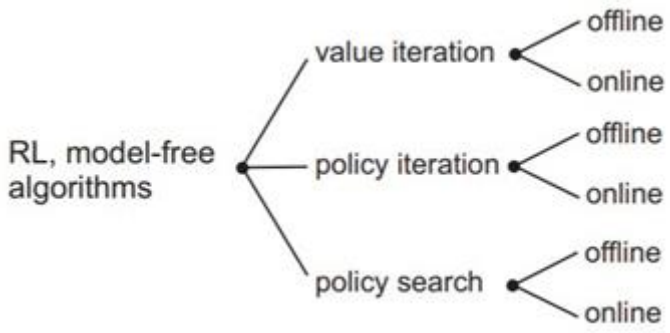

Figure 2.4.2. Taxonomy of model-free RL algorithms (Busoniu et al. 2010)

Depending different ways to compute a new policy, these model-free RL algorithms can be classified into three categories i.e. Value Iteration, Policy Iteration and Policy Search in Figure 2.4.2.

The concepts of Value Iteration, Policy Iteration and Policy Search are given hereafter:

- Value Iteration, such as (Bradtke and Barto 1996) and (Rummery and Niranjan 1994), computes an optimal value function (namely the V-function) or action-value function (namely the Q-function which evaluates the quality of a state-action pair), from which the optimal actions can be derived. These approaches provide the possibility to solve the Bellman optimality (Bellman 1966) using data measured from the system (2.4.1).
- Policy Iteration, such as (Lagoudakis and Parr 2003) and (Tesauro 1995), consists of two steps: policy evaluation and policy improvement. To evaluate current policies, algorithms compute their corresponding V-functions or Q-function which are then used to obtain improved policies. This two-step procedure is stopped when policies converge.
- Policy Search, such as (Sutton et al. 2000), differs from the two previous approaches as it searches directly for an optimal policy without necessarily computing any value function. To achieve an optimal solution, different optimization techniques are available to integrat in this approach, for example expectation-maximization, gradient descent, cross-entropy optimization etc.

Computing an exact optimal solution is computationally feasible only in low-dimensional domains with discrete states and discrete actions. When the states and actions are continuous,

the number of state values or action values is uncountable. The number of discrete state values increases dramatically when the dimension of the system increases. This phenomenon is called the curse of dimensionality. Therefore, an exact V-function, Q-function, or policy in general becomes difficult or even impossible to obtain.

Tackling this issue is crucial for real-time control applications, since the state and control actions are generally continuous in such applications. One of the efficient methods is Approximate Reinforcement Learning (ARL) (Bertsekas et al. 1995; Busoniu et al. 2010; Sutton and Barto 2018; Szepesvári 2010). Instead of exactly representing value functions or policies, ARL uses function approximators and aims to derive a (near-)optimal solution. Two classes of function approximators can be distinguished: parametric approximators having a fixed number of parameters; and non-parametric approximators having a flexible number of parameters depending on the collected data.

Since human behaviours and states, such as human fatigue dynamic, stress… are considered unknown in this thesis, the next sections focus on model-free RL algorithms. Algorithms, such as PoWER (Kober and Peters 2009) and REPS (Peters et al. 2010), show that the Policy Search framework is able to learn a (sub-)optimal solution with a reduced set of data. This data-efficiency feature is extremely important for a real-time application, which requires a satisfactory performance early in the learning. Therefore, the Policy Search framework has been chosen for the PAW design. In particular, the approximate version of Policy Search with parametric approximators is used for a finite-horizon problem hereafter.

2.4.2 Policy search using parametric approximators

Rather than learning a value function, Policy Search methods aim to find directly optimal parameters for a given parameterized policy. In addition, parameterized policies allow learning algorithms to operate directly in continuous action spaces.

Deterministic policies are typically represented by a linear basis function approximation as follows:

$$\bar{\pi}_{\lambda}\left(x_k\right) = \lambda^T \varphi\left(x_k\right) \tag{2.4.10}$$

where λ is a parameter vector and φ is a basis function vector. The basis function vector can be configured using Gaussian radial basis functions, polynomial functions, etc. Nonlinear approximation techniques are also possible (Mnih et al. 2015). The structure of the policy

parametrization is very important for the learning performance. More basis functions generally provide a more precise solution at the end of learning; but, of course the more basis functions, the more parameters are to learn, which impacts directly the learning time. A compromise must be found between a refined solution and a reasonably fast learning speed. Designers can choose a structure for (2.4.10) depending on the particular application.

In the literature, there exist different Policy Search algorithms which provide various performances in terms of learning speed, computation and complexity etc. In this work, we select two algorithms: Gradient Partially Observable Markovian Decision Processes (GPOMDP) (Baxter and Bartlett 2000) and Policy Learning by Weighting Exploration with the Returns (PoWER) (Kober and Peters 2009). The reasons for this choice are the simplicity of these two model-free methods and their implementability into a microcontroller, necessary condition for an application such as PAW. Beside of these two chosen algorithms, there are other powerful Policy Search and Policy Gradient approaches, such as Deep Reinforcement Learning (Duan et al. 2016; Schulman et al. 2015). However, Deep Reinforcement Learning uses approximating functions with multiple hidden layers. Such approximation implies a considerable number of parameters to learn. Therefore, this framework may need important memory and computation which are not desirable for our PAW application.

2.4.2.1 GPOMDP

Like other PG methods, GPOMDP estimates the gradient of the expected reward with respect to the parameters of the policy. Based on this gradient, the parameters are updated such that the received expected reward progressively increases. GPOMDP is different from actor-critic algorithms, e.g. (Grondman et al. 2012), (Peters and Schaal 2008), which estimate the gradient with the help of an approximate value function. Since an approximate value function is not needed, GPOMDP provides computational advantages. Therefore, this approach may be more easily embedded due to limited CPU power. Thus, we apply first GPOMDP in this work to verify if the Policy Search framework is suitable for a PAW control design.

Notice that learning algorithms require exploration, which is carried out by a random noise added to control actions. A policy exploration allows model-free algorithms to discovery new control actions such that a (near-)optimal control sequence is found. Therefore, the deterministic policy (2.4.10) becomes stochastic.

In GPOMDP, the parameters λ are updated as follows:

$$\lambda \leftarrow \lambda + \bar{\alpha} \nabla_{\lambda} \bar{R}_{\lambda} \tag{2.4.11}$$

where $\bar{R}_{\lambda}$ is the expected reward under the stochastic parametrized policy with the parameter vector λ. A stochastic policy distribution with the parameter vector is denoted by the notation $\tilde{\pi}_{\lambda}(u_k|x_k,k)$. To obtain the gradient $\nabla_{\lambda}\bar{R}_{\lambda}$ without knowing the model of the system, the Likelihood Ratio Estimator is typically used. Since we are in the setting of a deterministic MDP, the probability distribution $p_{\lambda}(\tau)$ over trajectories τ depends only on the initial state distribution $p(x_0)$, the stochastic policy distribution $\tilde{\pi}_{\lambda}(u_k|x_k,k)$, and the distribution of the transition function ξ. Then, $p_{\lambda}(\tau)$ can be expressed in the following way:

$$p_{\lambda}(\tau) = p(x_0)\prod_{k=0}^{K-1}\tilde{\pi}_{\lambda}(u_k|x_k,k) \tag{2.4.12}$$

The expected return under the random trajectories τ generated by $\tilde{\pi}_{\lambda}$ is:

$$\bar{R}_{\lambda} = \int p_{\lambda}(\tau) R(\tau) d\tau \tag{2.4.13}$$

The gradient $\nabla_{\lambda}\bar{R}_{\lambda}$ can be expressed as:

$$\nabla_{\lambda}\bar{R}_{\lambda} = \int \nabla_{\lambda} p_{\lambda}(\tau) R(\tau) d\tau \tag{2.4.14}$$

Since $\nabla_{\lambda} p_{\lambda}(\tau) = p_{\lambda}(\tau)\nabla_{\lambda}\log p_{\lambda}(\tau)$, we have:

$$\nabla_{\lambda}\bar{R}_{\lambda} = \int p_{\lambda}(\tau) R(\tau) \nabla_{\lambda}\log p_{\lambda}(\tau) d\tau \tag{2.4.15}$$

By replacing $p_{\lambda}(\tau)$ by (2.4.12), we obtain:

$$\nabla_{\lambda}\bar{R}_{\lambda} = \int p_{\lambda}(\tau)\nabla_{\lambda}\left[\log p(x_0)\prod_{k=0}^{K-1}\tilde{\pi}_{\lambda}(u_k|x_k,k)\right] R(\tau) d\tau \tag{2.4.16}$$

Finally, by replacing the integral with the equivalent expected value notation, the Reinforce gradient (Williams 1992) can be computed by:

$$\nabla_{\lambda}\bar{R}_{\lambda} = E_{\tau}\left[\nabla_{\lambda}\left[\log p(x_0)\prod_{k=0}^{K-1}\tilde{\pi}_{\lambda}(u_k|x_k,k)\right] R(\tau) d\tau\right] \tag{2.4.17}$$

Since the current rewards are only correlated with past actions, $\left[\nabla_\lambda \log \tilde{\pi}_\lambda\left(u_h^\varsigma | x_h^\varsigma, k\right)\right] r_k^\varsigma = 0$ for $h > k$. Thus, the gradient can be simplified as follows:

$$\nabla_\lambda \bar{R}_\lambda = \frac{1}{N_\tau} \sum_{\varsigma=1}^{N_\tau} \left[\sum_{k=0}^{K-1} \sum_{h=0}^{k} \left[\nabla_\lambda \log \tilde{\pi}_\lambda\left(u_h^\varsigma | x_h^\varsigma, k\right)\right] r_k^\varsigma \right] \tag{2.4.18}$$

where N_τ is the total number of trajectories used to compute the gradient and ς is the index of trajectories. Based on the gradient (2.4.18), the model-free algorithm GPOMDP updates the parameter vector λ with (2.4.11).

The entire procedure is given in the following table, where Γ is the total trials.

GPOMDP

1 Initialize λ_0

2 **for** $l = 0, 1, 2, \ldots \Gamma$

3 Generate N_τ trajectories τ of length K using λ_l

4 $\nabla_\lambda \bar{R}_\lambda = \frac{1}{N_\tau} \sum_{\varsigma=1}^{N_\tau} \left[\sum_{k=0}^{K-1} \sum_{h=0}^{k} \left[\nabla_\lambda \log \tilde{\pi}_\lambda\left(u_h^\varsigma | x_h^\varsigma, k\right)\right] r_k^\varsigma \right]$

5 $\lambda_{l+1} = \lambda_l + \alpha \cdot \nabla_\lambda \bar{R}_\lambda$ with the learning rate $\alpha > 0$

6 **end for beurk police**

Using the Likelihood Ratio Estimator, the policy update (2.4.11) leads to a local optimum. An intuitive example is shown in Figure 2.4.3, where the red colour means a high expected return and the blue colour indicates a low expected return. There are two parameters to learn. The stars indicate each parameter update. As shown in the figure, two different initial parameter vectors λ increase gradually their rewards and converge to the same solution. However, the obtained solution may be a local optimal, since a better combination of $\left(\lambda_1, \lambda_2\right)$ may exist.

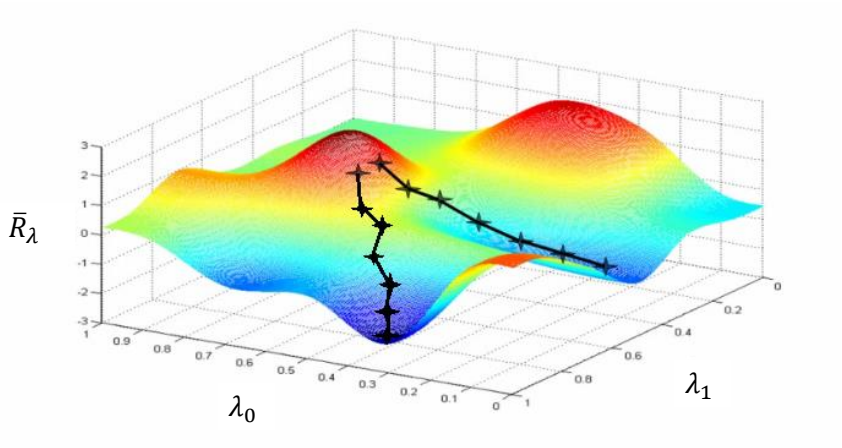

Figure 2.4.3. Model-free policy gradient example

2.4.2.2 PoWER

Another powerful Policy Search algorithm, successfully applied in robotics, is PoWER. Rather than computing the gradient $\nabla_\lambda \bar{R}_\lambda$ as (2.4.18), PoWER maximizes a lower bound on the expected return. This maximization guarantees that the performance of the new policy is improved. As shown by Kober and Peters (2009), a lower bound of the expected rewards under the latest parameter λ is given as follows:

$$L_\lambda(\lambda') = \int p_\lambda(\tau) R(\tau) \log\left(\frac{p_{\lambda'}(\tau)}{p_\lambda(\tau) R(\tau)}\right) d\tau \tag{2.4.19}$$

This can be furthermore expressed as:

$$L_\lambda(\lambda') = -D\left(p_\lambda(\tau) R(\tau) \| p_{\lambda'}(\tau)\right) \tag{2.4.20}$$

where D is the Kullback-Leibler divergence operator. In fact, $L_\lambda(\lambda')$ is the negative of the Kullback-Leibler divergence between the new path distribution $p_{\lambda'}(\tau)$ and the reward-weighted distribution $p_\lambda(\tau) R(\tau)$. Maximizing $L_\lambda(\lambda')$ is equivalent to minimizing the distance between the two distributions $p_{\lambda'}(\tau)$ and $p_\lambda(\tau) R(\tau)$.

The idea behind this minimization is that the new parameter vector λ' will increase the expected reward. An illustrative example is given in Figure 2.4.4, where the red line is the reward as a function of trajectories. The blue and the green lines (Figure 2.4.4 bottom) represent respectively the current and the new path distributions. Under the current policy

with the parameters $_{\lambda}$, high reward trajectories may have a low probability to occur (left side of Figure 2.4.4). However, these high reward trajectories are emphasized by the reward-weighted distribution $p_{\lambda}(\tau)R(\tau)$, which is used as a target distribution for updating the policy. Since the optimization step reduces the distance between $p_{\lambda'}(\tau)$ and $p_{\lambda}(\tau)R(\tau)$, the new policy will put more probability mass on the trajectories with higher rewards.

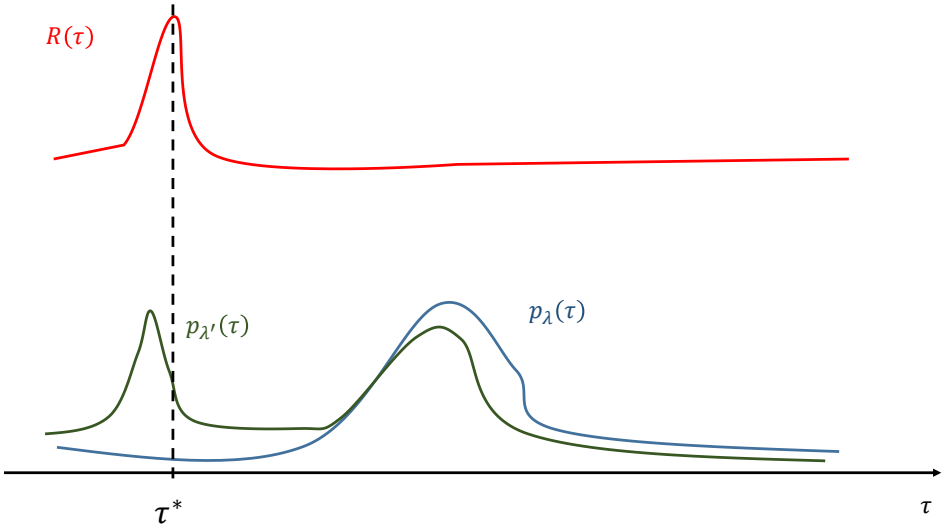

Figure 2.4.4. Illustration of the policy improvement

The policy update is done by the following optimization:

$$\lambda \leftarrow \arg\max_{\lambda'} L_{\lambda}(\lambda') \tag{2.4.21}$$

An analytical proof (Dayan and Hinton 1997) shows that the optimization (2.4.21) guarantees the improvement of the expected reward. Moreover, the derivative of (2.4.19) is:

$$\nabla_{\lambda'} L_{\lambda}(\lambda') = \int p_{\lambda}(\tau) R(\tau) \nabla_{\lambda'} \log p_{\lambda'}(\tau) d\tau \tag{2.4.22}$$

Since the considered dynamic is deterministic, after replacing the trajectory distribution $p_{\lambda'}$ by the policy distribution $\tilde{\pi}_{\lambda'}(u_k|x_k,k)$, we obtain:

$$\nabla_{\lambda'} L_{\lambda}(\lambda') = E_{\tau}\left\{\sum_{k=0}^{K-1} \nabla_{\lambda'} \log \tilde{\pi}_{\lambda'}(u_k|x_k,k) R(\tau)\right\} \tag{2.4.23}$$

Notice that $\tilde{\pi}_{\lambda'}(u_k|x_k,k)$ is an exponential family function. Therefore, the lower bound is a convex function, and the policy update (2.4.21) is equivalent to setting (2.4.23) to zero, i.e.:

$$E_{\tau}\left\{\sum_{k=0}^{K-1}\nabla_{\lambda'}\log\tilde{\pi}_{\lambda'}\left(u_k|x_k,k\right)R\left(\tau\right)\right\}=0 \tag{2.4.24}$$

To increase the learning speed, PoWER avoids policy exploration directly in the action-space. Since an exploration at each control action could introduce a high variance in the obtained data, the policy exploration is performed by adding a random noise to the control parameters. Thus, the stochastic control action u_k is expressed as follows:

$$u_k=\left(\lambda+z\right)\varphi\left(x_k,k\right) \tag{2.4.25}$$

where z is a zero mean Gaussian noise vector and φ are general basis functions. Under the exploration in the parameter space (2.4.25), by solving (2.4.24), the parameter vector λ is updated as follows:

$$\lambda_{l+1}=\lambda_l+\frac{\sum_{s=1}^{N_s}\left(\lambda_s-\lambda_l\right)R\left(\tau_s\right)}{\sum_{s=1}^{N_s}R\left(\tau_s\right)} \tag{2.4.26}$$

The whole PoWER algorithm is given as follows:

PoWER

1 Initialize λ_0

2 **for** $l=0,1,2,\ldots\ \Gamma$

3 Generate a new trajectory τ of length K using λ_l

4 Sort the performed trials decreasingly by return

5 Select the N_s trials with the highest return

6 Update parameters $\lambda_{l+1}=\lambda_l+\frac{\sum_{s=1}^{N_s}(\lambda_s-\lambda_l)R(\tau_s)}{\sum_{s=1}^{N_s}R(\tau_s)}$

7 **end for**

2.5 Summary

This chapter has provided the background on the PAW prototype and its modelling. We have also introduced relevant control techniques, such as nonlinear control and reinforcement learning optimal control.

After reviewing several commercial PAWs, we find out that these assistive devices are usually expensive and do little to address the highly heterogeneous population of the disabled persons. In addition, various intra and extra individual variations, including non-measurable features such as level of disability, fatigue, and pain, are not ignorable for a PAW application. In this context, the following problems will be addressed in the next chapters.

- In order to reduce the hardware cost, an unknown input PI-observer is designed to estimate the human torques, avoiding the use of torque sensors (see Remark 1). Based on the estimated information, a robust assistive control algorithm is designed.
- To deal with heterogeneous individuals dynamic, including human fatigue, stress, etc. We apply reinforcement learning optimal control techniques to design an assistive control. The objective is to provide an "intelligent" assistive strategy which is able to adapt itself to the PAW user.
- At last, we propose ideas to combine the model-based and the model-free approaches to design a PAW which is affordable to disabled population and is able to provide adaptive behaviours.

Chapter 3. Model-based design subject to PAWs

3.1 Introduction

A push-rim sensor, such an electromyography sensor or a torque sensor, is typically used to detect the users' intention in PAW applications. However, such sensors considerably increase the hardware complexity and the system cost. In this chapter, the objective is to design an observer-based assistive control using only encoder sensors. An unknown input observer (UIO) is first designed to estimate the human torques produced. Then, the estimated variables are used to determine the frequency of the signals and to propose a reference trajectory.

The difficulty in designing an observer-based assistive control relies in the fact that users control the velocity and yaw rotation of the PAW depending on their own will and perception of the environment. To exemplify, a user may want to go fast to a destination (unknown from the designer) implying a desired velocity and may suddenly have to turn because of an obstacle (unknown environment from the designer). The assistive torques have to be generated according to the human torque profiles, which are estimated by UIO via the angular velocity $\dot{\theta}$. In this framework, shown in Figure 3.1.1, the user plays two important roles.

The first role is to act as a metabolic energy storage unit. This metabolic energy storage may be driven by the state of fatigue which would influence the performance of the human propelling. The second role is to act as a human controller that perceives the environment to generate control signals (human torques). In such context, the user can be considered as an extra "sensor". The user gets information about the surrounding environment to take a decision. The future trajectory of the PAW is derived from this information.

The advantage of this setup is that the user can perceive naturally the information where conventional sensors would have high difficulties if not failing to compute and treat the information in the perspective of an autonomous framework. Therefore, in accordance with the user personal perception (velocity, yaw rotation, environment and also his/her state of fatigue, stress…), the assistive system should be as "transparent" as possible, just helping the user to accomplish his/her will.

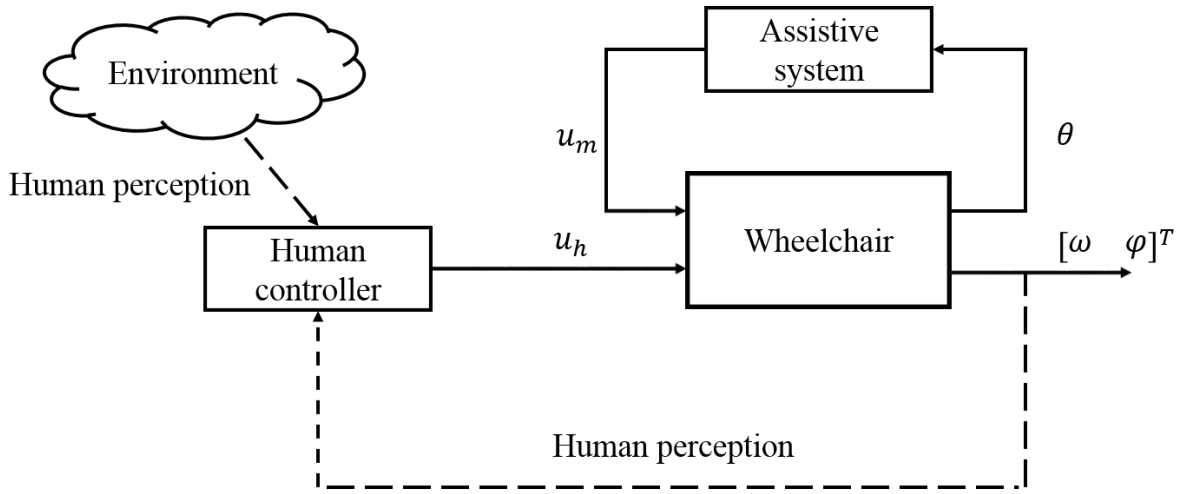

Figure 3.1.1. Power-assistance framework

To achieve these objectives, we are facing the challenging problems: human torque observation and human intention estimation. Moreover, these problems are coupled with system uncertainties, such as the mass (different wheelchairs and users) and the changing environment conditions (viscous friction coefficient, slope) or unmodelled issues (casters wheels). In addition, due to the limited torques of the electrical motors, actuator saturations also have to be taken into account. Thus, a very important issue is the stability in presence of system uncertainties and actuator saturations, since an unstable situation could damage the wheelchair and possibly injure the user. Furthermore, the quantification of encoders could degrade seriously the quality of measurement and thus the performance of the assistive control.

Various driver assistance systems for PAWs have been studied in the literature over the last decade. They aim to improve the driving comfort and the efficiency of users' pushing. R. A. Cooper et al. (2002) proposes a pushrim-actived power-assisted wheelchair (PAPAW) which takes into account the human behaviour, via the pushing torque measured by sensors and human interaction with the device. The obtained results show that the proposed PAPAW can reduce considerably the strain on the upper arm compared with manual wheelchairs. Instead of measuring human torques, Oonishi et al. (2010) uses an electromyogram sensor and a disturbance observer (or unknown input observer) to estimate the drivers' intention. According to the estimated human intention, assistive torques are generated to help users propelling the wheelchair. Compared with these two studies, hereafter we aim to design the assistive algorithm using only the incremental encoders without additional sensors such as for

torque or EMG. Since limited measurements are available and due to the numerous unknown and/or non-modelled uncertainties, the amplitude of human torques cannot be reconstructed perfectly, and this issue will be discussed later in the chapter. The user's intentions, such as accelerating, turning and braking, are designed from the estimated signals and especially considering the propelling frequencies. The resulting reference accelerations, deceleration and turning speed are tracked by a robust PI-like controller.

To deal with the model uncertainties, i.e. the mass and the viscous friction coefficient, a polytopic representation will be used to represent the uncertain model. Based on this representation, the robust PI-like controller is obtained by solving an LMI constraints problem. The influence of unmodelled dynamics, such as the dynamic of the casters, can be considered as a part of the unknown input (Chen et al. 2016). The control action, based on the estimated unknown input, can attenuate this influence and enhance the tracking performance. Moreover, the actuator saturations are taken into account for the control design, and the stability analysis of the entire mechanical system is provided. Finally, simulation results validate the effectiveness of the proposed observer, the reference generation, and observer-based robust tracking controller.

In Section 3.2, based on a nominal wheelchair model, the observer design using a constant sampling time is introduced. To validate the proposed observer, two control strategies, such as a low pass filter and a PI controller, are provided.

Section 3.3 presents a human torque observer under time-varying sampling. The same estimation design introduced in Section 3.2 is used. However, the sampling time is not anymore constant. The sampling time depending on the angular speed aims to reduce the quantification of encoders. In addition, the observer-based assistive control system based on a reference generation algorithm is introduced. The reference tracking is accomplished by a PI controller. Simulations are also provided to validate the approach.

Since the actual acquisition card of the prototype does not support a time-varying sampling, stability analysis, Section 3.4 and Section 3.5 is derived only for a constant sampling time. Section 3.4 provides a stability analysis of the observer-based control with system uncertainties, whereas Section 3.5 also includes the constraints on the inputs.

3.2 Human torque estimation

In this section, we focus on the human torque estimation problem. The human torques exerted on the wheels are estimated by using a so-called unknown input PI-observer (Guan et al. 1991). In addition, a structure of the observer using a descriptor form model (Estrada-Manzo et al. 2015) is applied to obtain observer gains by solving LMI constraint problems. The observer design obtained in this section has been published in (Feng et al. 2017).

3.2.1 Approximation of human torques

The behaviour of the human torques T_{hr} and T_{hl} exerted are approximated by a n_pth degree derivative in time to zero. Therefore, we can rewrite the model considering the inputs, and their derivatives, as state variables. In continuous-time for the right wheel, this corresponds to $d^{n_p}T_{hr}/dt^{n_p}=0$. In discrete-time, the input torques are assumed to satisfy:

$$\left(1-z^{-1}\right)^{n_p}T_{hr}(k)=0 \tag{3.2.1}$$

Further, the equality (3.2.1) can be expressed as:

$$T_{hr}(k)=-\sum_{i=1}^{n_p}\binom{n_p}{i}(-1)^{n_p}T_{hr}(k-i) \tag{3.2.2}$$

where $\binom{n_p}{i}$ corresponds to the binomial coefficient. Consider the unknown input vector $T_{hr}^{n_p}(k)=\left[T_{hr}(k),T_{hr}(k-1),\ldots,T_{hr}\left(k-n_p+1\right)\right]^T\epsilon\mathbb{R}^{n_p}$. The dynamics (3.2.2) of the vector $T_{hr}^{n_p}$ can be written as:

$$T_{hr}^{n_p}(k+1)=\Gamma_{n_p}T_{hr}^{n_p}(k) \tag{3.2.3}$$

where

$$\Gamma_{n_p}=\begin{bmatrix}-(-1)^1\binom{n_p}{1} & -(-1)^2\binom{n_p}{2} & \ldots & -(-1)^{n_p}\binom{n_p}{n_p}\\ I_{n_p-1} & & & 0_{(n_p-1)\times 1}\end{bmatrix}$$

The same reasoning is applied for the left wheel, so the dynamic of the vector $T_{hl}^{n_p}(k) = \left[T_{hl}(k), T_{hl}(k-1), \ldots, T_{hl}\left(k-n_p+1\right)\right]^T \epsilon \mathbb{R}^{n_p}$ is:

$$T_{hl}^{n_p}\left(k+1\right) = \Gamma_{n_p} T_{hl}^{n_p}\left(k\right) \tag{3.2.4}$$

The observability property is given as the following constraint $rank\left(B\right) \leq p$ (with p the number of outputs) is satisfied (Ichalal et al. 2009). The human input vector can be expressed as follows:

$$u_h = \begin{bmatrix} B_r & B_l \end{bmatrix} \begin{bmatrix} T_{hr}^{n_p} \\ T_{hl}^{n_p} \end{bmatrix} \tag{3.2.5}$$

with $B_r = \left[\begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad 0_{2\times\left(n_p-1\right)}\right], B_l = \left[\begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad 0_{2\times\left(n_p-1\right)}\right].$

by defining an extended state vector as $x_o = \left[x, T_{hr}^{n_p T}, T_{hl}^{n_p T}\right]^T \in \mathrm{R}^{2n_p+n_x}$, the discrete-time system (2.2.5) can be rewritten as:

$$\begin{aligned} E_o x_o^+ &= A_o x_o + B_o u_m \\ y &= C_o x_o \end{aligned} \tag{3.2.6}$$

where the matrices are:

$$E_o = \begin{bmatrix} E & 0_{2\times 2n_p} \\ 0_{2n_p\times 2} & I_{2n_p} \end{bmatrix}, A_o = \begin{bmatrix} A & BB_r & BB_l \\ 0_{n_p\times 2} & \Gamma_{n_p} & 0_{n_p\times n_p} \\ 0_{n_p\times 2} & 0_{n_p\times n_p} & \Gamma_{n_p} \end{bmatrix},$$

$$B_o = \begin{bmatrix} B \\ 0_{2n_p\times 2} \end{bmatrix}, C_o = \begin{bmatrix} C & 0_{n_y\times 2n_p} \end{bmatrix}.$$

Note that the problem is well posed as E_o is always invertible.

3.2.2 Unknown input observer design

The aim is to estimate the unknown input torques T_{hr}, T_{hl}. The observer considered for the descriptor model (3.2.6) is (Estrada-Manzo et al. 2016):

$$\begin{aligned} E_o\hat{x}_o^+ &= A_o\hat{x}_o + B_o u_m + G_o^{-1}K_o\left(y-\hat{y}\right) \\ \hat{y} &= C_o\hat{x}_o \end{aligned} \tag{3.2.7}$$

The estimation error is $e_o(k) = x_o(k) - \hat{x}_o(k)$, and its dynamics resulting issued from (3.2.6) and (3.2.7), is given by:

$$E_o e_o^+ = \left(A_o - G_o^{-1}K_o C_o\right)e_o \tag{3.2.8}$$

where the matrices G_o and K_o to design have to guarantee the convergence of the state estimation error ℓ_0. In order to design these matrices, we consider the following Lyapunov function candidate:

$$V\left(e_o\right) = e_o^T P_o e_o \tag{3.2.9}$$

Theorem 1. *The estimation error dynamics (3.2.8) are asymptotically stable if there exist* $P_o = P_o^T \in \mathrm{R}^{2n_p+n_x}$, $G_o \in \mathrm{R}^{2n_p+n_x}$, $K_o \in \mathrm{R}^{(2n_p+n_x)\times n_y}$ *and a scalar decay rate* η *with* $0 < \eta \leq 1$ *such that*

$$\begin{bmatrix} -\eta P_o & * \\ G_o A_o - K_o C_o & P_o - G_o E_o - E_o{}^T G_o{}^T \end{bmatrix} < 0 \tag{3.2.10}$$

Thereinafter, the asterisk $(*)$ represents a transpose quantity in the symmetric position.

Proof. The variation $\triangle V(e_o) = V(e_o^+) - V(e_o)$ of the Lyapunov function (3.2.9) including the decay rate $\overline{\eta}$, using an extended vector $\begin{bmatrix} e_o \\ e_o^+ \end{bmatrix}$, is:

$$\Delta V\left(e_o\right) = \begin{bmatrix} e_o \\ e_o^+ \end{bmatrix}^T \begin{bmatrix} -\overline{\eta} P_o & 0_{2n_p+n_x} \\ 0_{2n_p+n_x} & P_o \end{bmatrix} \begin{bmatrix} e_o \\ e_o^+ \end{bmatrix} < 0 \tag{3.2.11}$$

The estimation error dynamic (3.2.8) can be rewritten as an equality constraint:

$$\begin{bmatrix} A_o - G_o^{-1}K_o C & -E_o \end{bmatrix} \begin{bmatrix} e_o \\ e_o^+ \end{bmatrix} = 0 \tag{3.2.12}$$

From Lemma 3, the inequality (3.2.11) under constraint (3.2.12) is equivalent to the following inequality:

$$\begin{bmatrix} 0_{2n_p+n_x} \\ G_o \end{bmatrix}\begin{bmatrix} A_o - G_o^{-1}K_oC & -E_o \end{bmatrix} + (*) + \begin{bmatrix} -\bar{\eta}P_o & 0_{2n_p+n_x} \\ 0_{2n_p+n_x} & P_o \end{bmatrix} < 0 \qquad (3.2.13)$$

Expanding (3.2.13), we obtain directly the linear matrix inequality (3.2.10). ■

Remark 4. Applying the observer gain obtained by solving (3.2.10), the Lyapunov function candidate (3.2.9) decreases exponentially as follows:

$$V_{k+1} < \bar{\eta}V_k < \bar{\eta}^2 V_{k-1} \cdots < \bar{\eta}^{k+1}V_0 \qquad (3.2.14)$$

In this way, the convergence of the estimation error e_o can be tuned via the decay rate $\bar{\eta}$.

3.2.3 Simulation results

In order to carry out the numerical simulations, we use the parameters in Table I of the chapter 2. Considering that the human torques are slow dynamic signals and after some initial tests, $n_p = 2$ was chosen in (3.2.1). It represents a perfect compromise between complexity of the design and accuracy of the estimations. Therefore, solving the LMI conditions in Theorem 1, the following observer gains are obtained:

$$G_o^{-1}K_o = \begin{bmatrix} 18.76 & -10.56 \\ -10.56 & 18.76 \\ 143.61 & -81.53 \\ 160.06 & -90.92 \\ -81.53 & 143.61 \\ -90.92 & 160.06 \end{bmatrix} \qquad (3.2.15)$$

- **UIO without power-assistance**

The PAW is assumed to move on a flat surface and the human input torque is represented by the positive half cycle of a sinusoidal. The human input torque and the velocity are successfully reconstructed, see Figure 3.2.1. Note that there is a delay of two sampling time units induced by the observer between the real input torque signal and the estimated one. This delay effect is due to 2nd degree polynomial approximation (3.2.4).

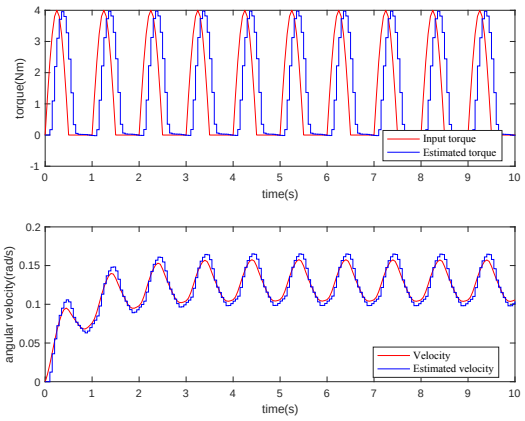

Figure 3.2.1. Driving simulation on a flat road without assistance (torque/velocity)

- **UIO with power-assistance**

A power-assisted system is added to help the user to propel the wheelchair on a flat road. We have to keep in mind that the following simulations do not serve to validate the assistive control. The objective is to apply different input torques to check the performance of the proposed observer under two proposed power-assisted algorithms. A Gaussian white noise is added to the inputs to simulate small road irregularities. For the first trial, the assistive torque u_m is generated by a low pass filter (H. Seki and Kiso 2011) as follows:

$$u_m(z) = \frac{\psi}{z - e^{-\frac{T_e}{\tau}}} \hat{u}_h(z) \tag{3.2.16}$$

where T_e is the sample time, ψ is the assistance ratio and τ is a time parameter related to the response time of the assistive system. The parameter ψ determines the amplification ratio between u_m and $\hat{u}_h$. The time parameter τ determines the inertial dynamics of the assistive torque. These two parameters should be configured correctly to have a good compromise between smooth driving and rapid torque assistance. Different parameter setting strategies can be found in (H. Seki and Kiso 2011; H. Seki and Tadakuma 2006), e.g. adaptive driving control using parameter adjustment. In the present study, only constant parameters $(\psi = 0.89 \text{ and } \tau = 0.2)$ are used to validate our torque-sensorless PAW design.

As depicted in Figure 3.2.2, the observer provides a good estimate of the human torque, the centre and the yaw velocities. Small road irregularities are filtered by the inertia of the

wheelchair dynamic. Moreover, the assistive torques are amplified with respect to the torques estimated by the observer (3.2.7).

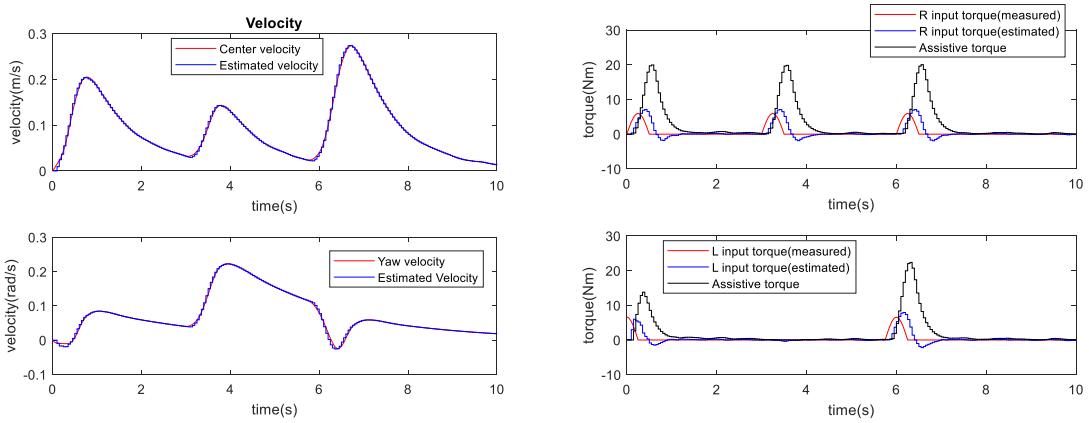

Figure 3.2.2. Driving simulation on a flat road with the proposed proportional power-assistance system

For the second trial, a PI controller is applied to track a reference velocity. Human torques are again represented by the positive half cycle of a sinusoidal. Figure 3.2.3 illustrates that the unknown input observer adequately estimates the input torques. Moreover, the velocity tracking objectives are satisfactorily met.

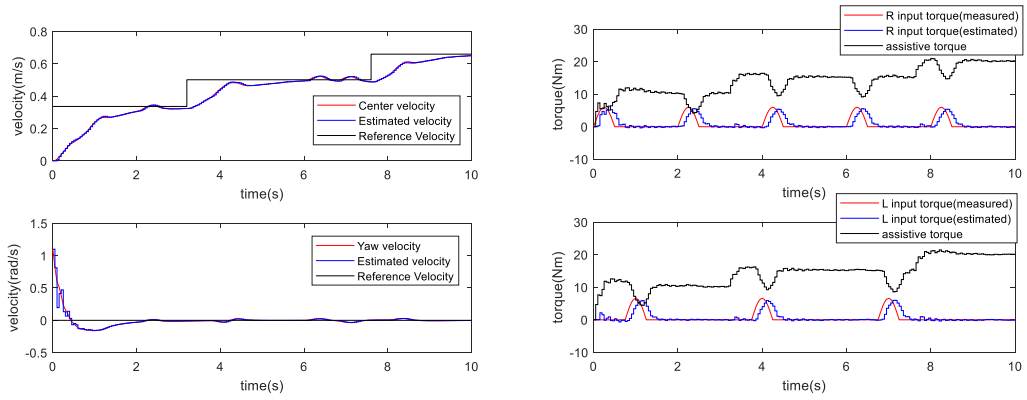

Figure 3.2.3. Driving simulation on a flat road with the proposed PI velocity controller

3.2.4 Summary

The design of human torque estimations has been presented in this section. The main objective of applying the proposed observer is to estimate human torques without using torque sensors. This torque-sensorless design could significantly reduce not only hardware

complexity but also system cost. The following sections are devoted to designing assistive controls for PAWs based on the estimation technique presented.

3.3 Observer-based assistive framework under time-varying sampling

An observer-based approach (Feng et al. 2017) has been used in the previous section to reconstruct the human torques using a constant sampling. However, the encoder sensors only provide a new measurement at a fixed angular position interval (Phillips et al. 1995). In other words, the sampling time is time-varying depending on the angular velocity. This time-varying sampling can be tackled using a discrete-time Linear Parameter-Varying (LPV) model for the wheelchair. We use the so-called Takagi-Sugeno (TS) form to represent the discrete-time LPV model (Precup and Hellendoorn 2011; Takagi and Sugeno 1993). Moreover, the observer design will be written as a Linear Matrix Inequality (LMI) constraints problem (BOYD 1994; Estrada-Manzo et al 2016). Compared to the observer in the previous section, the design considers a time-varying sampling rate together with delayed non-quadratic Lyapunov function to guarantee the convergence of the observer. Moreover, the tracking of longitudinal velocities is achieved by a conventional PI controller. The contribution of this section has been published in (Feng et al. 2018).

Based on the estimated torques, the question that arises is how to detect the human intention? Accelerating the wheelchair, resumes for the users in propelling it more frequently, which can be detected via signal treatment such as Fast Fourier Transform (FFT). Once running, if no action is detected, the velocity is maintained or slowly decreased. For turning, the user has just to brake the right or left wheel to turn right or left respectively. To stop or slow down the wheelchair, the user has to brake both wheels. These four rules govern the assistive system. Once the desired references ω_{ref} and φ_{ref} are generated, the tracking is achieved by a PI controller. The whole assistive system is presented Figure 3.3.1. The design of each function, acceleration, turning, and braking, will be depicted.

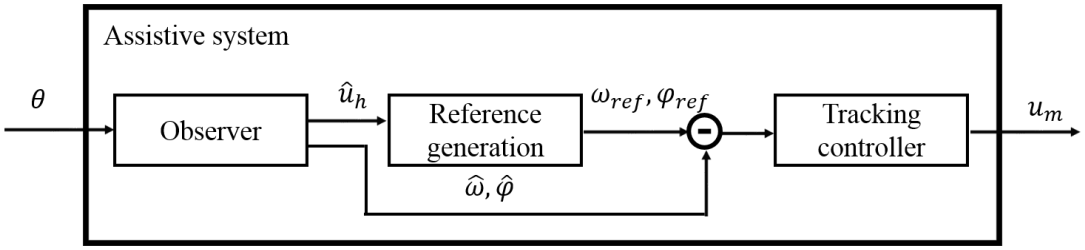

Figure 3.3.1. Assistive system overview

3.3.1 Time-varying sampling

Instead of using a predetermined sampling time, this sampling approach updates the state information as soon as a new measurement is received by the system. This approach corresponds to a sampling-in-angle domain instead of a sampling-in-time domain. Compared to the conventional fixed sampling rate, several works, IC engine (Kerkeni et al. 2010) crankshaft torque estimation (Losero et al. 2016; Losero et al. 2015), show that for these kind of measurements, it simplifies the design while giving persuasive results. For this approach, a measurement is taken when detecting a rising edge. Between two consecutive measurements, Figure 3.3.2, the relative position is known which is equal to the distance between two neighbouring teeth. With a constant sampling rate T_s', the relative position is computed by counting the rising edges during the constant sampling period. For example, in Figure 3.3.2, one rising edge is detected between t_k' and t_{k+1}'. In other words, the computer obtains the same relative position for two cases. However, as shown in Figure 3.3.2, the relative position between t_k' and t_{k+1}' is obviously smaller than the distance between two neighbouring teeth. Intuitively, the sampling technique used in this section would provide a better measurement than the conventional constant sampling. Thanks to this advantage of the time-varying sampling, the observer in the following section will be designed using a linear parameter-varying model.

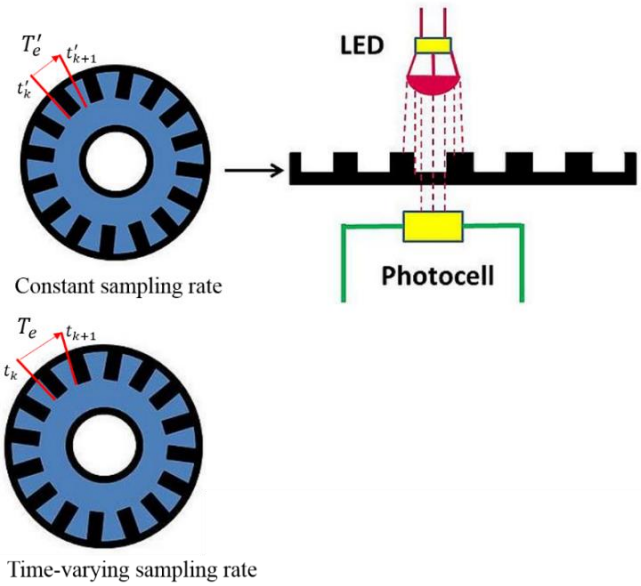

Figure 3.3.2. Working principles of the incremental encoder, constant sampling and time-varying sampling (Pogorzelski and Hillenbrand n.d.)

Due to the way the two incremental encoders receive the signals, the sampling period of the angular positions is time varying with the angular velocity. After detecting a rising edge from one of the two angular position sensors, see Figure 3.3.3, the system updates the state of the discrete system with the new measurement (bottom of Figure 3.3.3). The angular velocities are assumed to be constant between two updates. Therefore, the sampling time T_e depends on the angular velocities of both wheels.

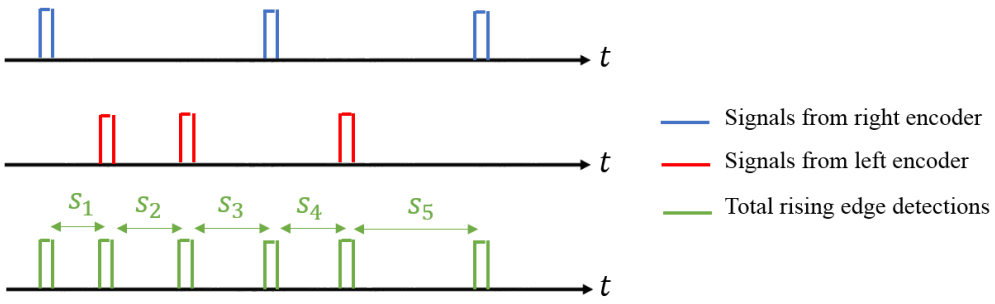

Figure 3.3.3. Data time-varying sampling example

Under time-varying sampling, the descriptor system (2.2.5) can be rewritten as the following discrete-time LPV model using Euler discretization:

$$\begin{aligned} x^{+} &= \tilde{A}(T_e)x + \tilde{B}(T_e)[u_h + u_m] \\ y &= Cx \end{aligned} \tag{3.3.1}$$

with the following matrices:

$$\tilde{A}(T_e) = T_e \begin{bmatrix} -\mathrm{K}/(\boldsymbol{\alpha}+\boldsymbol{\beta}) & 0 \\ 0 & -\mathrm{K}/(\boldsymbol{\alpha}-\boldsymbol{\beta}) \end{bmatrix} + I_2,$$
$$\tilde{B}(T_e) = T_e \begin{bmatrix} \varsigma/(2(\boldsymbol{\alpha}+\boldsymbol{\beta})) & \varsigma/(2(\boldsymbol{\alpha}+\boldsymbol{\beta})) \\ \varsigma/(b(\boldsymbol{\alpha}-\boldsymbol{\beta})) & -\varsigma/(b(\boldsymbol{\alpha}-\boldsymbol{\beta})) \end{bmatrix}.$$

Remark 5. Since the descriptor matrix E_o is constant and there is only one nonlinearity Γ_t in the LPV system (3.3.1), there is no need for the descriptor form and we can return to the conventional state-space form for the observer design in this section.

3.3.2 Observer design under time-varying sampling

Based on the model (3.3.1), an unknown input observer for discrete-time LPV system (3.3.1) is designed using LMI techniques and recent results on non-quadratic Lyapunov functions (B. Ding 2010; Guerra and Vermeiren 2004) and delayed Lyapunov functions (Guerra et al. 2012).

Using again the polynomial approximation (3.2.1) for the human torques, the discrete-time LPV system (3.3.1) can be expressed as:

$$\begin{aligned} x_o^{+} &= \tilde{A}_o(T_e)x_o + \tilde{B}_o(T_e)u_m \\ y &= \tilde{C}_o x_o \end{aligned} \tag{3.3.2}$$

where the extended state vector is $x_o = \left[x, T_{hr}^{n_p T}, T_{hl}^{n_p T}\right]^T \in \mathrm{R}^{2n_p+n_x}$ and the corresponding extended matrices are:

$$\tilde{A}_o(T_e)=\begin{bmatrix}\tilde{A}(T_e) & \tilde{B}(T_e)B_r & 0_{n_x\times(n_p-1)} & \tilde{B}(T_e)B_l & 0_{n_x\times(n_p-1)}\\ 0_{n_p\times n_x} & \Gamma_{n_P} & & 0_{n_p} & \\ 0_{n_p\times n_x} & 0_{n_p} & & \Gamma_{n_P} & \end{bmatrix},$$

$$\tilde{B}_o(T_e)=\begin{bmatrix}\tilde{B}(T_e) & 0_{n_x\times 2n_p}\end{bmatrix}^T, \tilde{C}_o=\begin{bmatrix}C_d & 0_{n_y\times 2n_p}\end{bmatrix}.$$

For the sake of simplicity, we adopt the notations:

$$T_e=T_e(k);\ T_e^-=T_e(k-1) \tag{3.3.3}$$

The nonlinear term T_e in (3.3.1) is the time difference between two consecutive rising edges produced by the two encoders. This information can be easily obtained during data acquisition. As the sampling time is bounded (assuming that the angular velocities are not zero), the nonlinear term can be expressed using the classical Sector Nonlinearity Approach (Taniguchi et al. 2001):

$$T_e(\boldsymbol{\theta}_R,\boldsymbol{\theta}_L)=\frac{\overline{T_e}-T_e(\boldsymbol{\theta}_R,\boldsymbol{\theta}_L)}{\overline{T_e}-\underline{T_e}}\underline{s}+\frac{T_e(\boldsymbol{\theta}_R,\boldsymbol{\theta}_L)-\underline{T_e}}{\overline{T_e}-\underline{T_e}}\overline{T_e} \tag{3.3.4}$$

where $\underline{T_e}$ and $\overline{T_e}$ are the bounds on the sampling time $T_e(\boldsymbol{\theta}_R,\boldsymbol{\theta}_L)$, i.e. $T_e(\boldsymbol{\theta}_R,\boldsymbol{\theta}_L)\in\left[\underline{T_e},\overline{T_e}\right]$. Therefore, we can rewrite the nonlinear model (3.3.2) as the following TS model:

$$\begin{aligned}\tilde{x}_o^{\,+}&=\sum_{i=1}^{2}h_i(T_e)\left[\tilde{A}_{o_i}\tilde{x}_o+\tilde{B}_{o_i}u_m\right]\\ y&=\tilde{C}_o\tilde{x}_o\end{aligned} \tag{3.3.5}$$

with the following matrices:

$$\tilde{A}_{o_1}=\tilde{A}_o\left(\underline{T_e}\right),\tilde{A}_{o_2}=\tilde{A}_o\left(\overline{T_e}\right),\tilde{B}_{o_1}=\tilde{B}_o\left(\underline{T_e}\right),\tilde{B}_{o_2}=\tilde{B}_o\left(\overline{T_e}\right).$$

The membership functions are:

$$h_1\left(T_e(\boldsymbol{\theta}_R,\boldsymbol{\theta}_L)\right)=\frac{\overline{T_e}-T_e(\boldsymbol{\theta}_R,\boldsymbol{\theta}_L)}{\overline{T_e}-\underline{T_e}},h_2\left(T_e(\boldsymbol{\theta}_R,\boldsymbol{\theta}_L)\right)=1-h_1\left(T_e(\boldsymbol{\theta}_R,\boldsymbol{\theta}_L)\right).$$

Based on the TS model (3.3.5), the observer considered is:

$$\begin{aligned}\hat{x}_o^+ &= \sum_{i=1}^{2} h_i\left(T_e\right)\left[\tilde{A}_{o_i}\hat{x}_o + \tilde{B}_{o_i}u_m\right] + \tilde{G}_{T_e^-}^{-1}\tilde{K}_{T_eT_e^-}\left(y-\hat{y}\right)\\ \hat{y} &= \tilde{C}_o\hat{x}_o\end{aligned} \tag{3.3.6}$$

For the delayed state the notation:

$$\tilde{A}_{T_e} = \sum_{i=1}^{2} h_i\left(T_e\right)\tilde{A}_{o_i}, \tilde{G}_{T_e^-}^{-1} = \left(\sum_{j=1}^{2} h_j\left(T_e^-\right)\tilde{G}_j\right)^{-1}, \tilde{K}_{T_eT_e^-} = \sum_{i=1}^{2}\sum_{j=1}^{2} h_i\left(T_e\right)h_j\left(T_e^-\right)\tilde{K}_{ij}.$$

Here, $\tilde{G}_j$ and $\tilde{K}_{ij}$, $i, j = 1,2$ are free matrices to be derived from the LMI constraint problem. The existence of $\tilde{G}_{T_e^-}^{-1}$ and the delayed parts of the observer will be discussed later. The estimation error is $\tilde{e}_o = \tilde{x}_o - \tilde{x}_o$. Its dynamic is derived as:

$$e_o^+ = \left(\tilde{A}_{T_e} - \tilde{G}_{T_e^-}^{-1}\tilde{K}_{T_eT_e^-}\tilde{C}_o\right)e_o \tag{3.3.7}$$

We define a delayed non-quadratic Lyapunov function given by (Guerra et al. 2012):

$$V\left(e_o, T_e\right) = {e_o}^T\tilde{P}_{T_e^-}e_o \tag{3.3.8}$$

where the matrix $\tilde{P}_{T_e^-}$ is symmetric positive definite and writes:

$$\tilde{P}_{T_e^-} = \tilde{P}_{T_e^-}^T = \sum_{j=1}^{2} h_j\left(T_e^-\right)\tilde{P}_j > 0$$

In order to guarantee the convergence of the estimation error, the Lyapunov function (3.3.8) must decrease along the trajectories of (3.3.7). The variation of (3.3.8) is negative if the following inequality holds:

$$\begin{bmatrix} -\tilde{P}_{T_e^-} & (*) \\ \tilde{G}_{T_e^-}\tilde{A}_{T_e} - \tilde{K}_{T_eT_e^-}\tilde{C}_o & -\tilde{G}_{T_e^-} - {\tilde{G}_{T_e^-}}^T + \tilde{P}_{T_e} \end{bmatrix} < 0 \tag{3.3.9}$$

We define the following LMI term:

$$\boldsymbol{\gamma}_{ij} = \begin{bmatrix} -\tilde{P}_j & (*) \\ \tilde{G}_j\tilde{A}_{o_i} - \tilde{K}_{ij}\tilde{C}_o & -\tilde{G}_j - \tilde{G}_j^T + \tilde{P}_i \end{bmatrix} < 0 \tag{3.3.10}$$

Theorem 2. *(Guerra et al. 2012): The estimation error (3.3.7) is globally asymptotically stable if there exist some matrices* $\tilde{P}_j \in \mathrm{R}^{(2n_p+n_x)\times(2n_p+n_x)}$, $\tilde{G}_j \in \mathrm{R}^{(2n_p+n_x)\times(2n_p+n_x)}$ *and* $\tilde{K}_{ij} \in \mathrm{R}^{n_y\times(2n_p+n_x)}$ *for all* $i,j \in \{1,2\}$ *such that the LMI conditions* Υ_{ij} *in (3.3.10) hold.*

The complete proof and more details can be found in (Guerra et al. 2012). By applying Theorem 2, the observer gains in (3.3.7) can be found by solving the LMIs (3.3.10). Notice that due to the last term of (3.3.9), if theorem 2 conditions are satisfied then: $\tilde{G}_{T_e^-} + \tilde{G}_{T_e^-}{}^T \geq \tilde{P}_{T_e} > 0$ which ensure the existence of $\tilde{G}_{T_e^-}^{-1}$. The condition (3.3.9) also shows the way the delay parts were chosen: the goal is to avoid increasing the number of LMI constraints. Therefore, as a double sum was considered in our case sufficient, i.e. $\tilde{K}_{T_eT_e^-}$, $\tilde{G}_{T_e^-}$ multiplying $\tilde{A}_{T_e}$ is the only solution without increasing the number of sums, as well as $\tilde{P}_{T_e^-}$ that introduces one sample after $\tilde{P}_{T_e}$.

The simulation results of the proposed observer is provided along with the reference trajectory generation in the next section.

3.3.3 Reference trajectory generation

Based on the estimated human torques, a reference trajectory generation algorithm is introduced in this section. The proposed power assisted control method aims to make the wheelchair more manoeuvrable for the user. More precisely, the longitudinal velocity, yaw rotation of the wheelchair should be efficiently controlled by human torques. Since the goal is neither to use a torque sensor, nor to have a precise wheelchair and human model, the reconstruction of a precise amplitude of the human torques is difficult to achieve. Our reference generation algorithm is based merely on the direction and the frequencies of the human torques estimated from angular positions.

We consider that the frequencies of the human torque range between 0.2-2 Hz, which is a representative range for the frequency of propulsion performed by normal users (Boninger et al. 2000). Consequently, the undesired high frequency components in the estimated signals are filtered. Then, the main frequencies over a predefined time interval are determined by using the real-time fast Fourier transform (FFT) function. The system performs a FFT over a predefined time interval by using a windowing technique. This technique provides a "view" of data through a time interval called window (Heydt et al. 1999).

The assistive algorithm should be simple, user-transparent and efficient enough to give users a natural way to control the wheelchair. Specifically, a higher frequency of users' propulsions leads to a higher velocity ω of the wheelchair. Here, the reference velocity ω is proportional (with a ratio δ) to the highest frequency among the left/right hand propulsions. Notice that even if the user does not push symmetrically, the assistive algorithm makes the wheelchair go straight. To turn the wheelchair, users only need to brake one of the two wheels. The desired angle to turn depends on the length of the braking. When turning is detected, the reference centre velocity is reduced. If the user pushes less frequently or stops pushing, the reference velocity is kept. To brake or stop the wheelchair (excepting on emergency stop provided by a stop button), the user should brake both wheels. This action reduces the reference velocity ω_{ref} with a constant rate η. The whole algorithm is shown in Figure 3.3.4.

This new mechanism enables the users to actively control velocity, braking and rotation by changing the frequencies and direction of their propulsions. Moreover, there are only three parameters ζ, η and δ to tune. These advantages make the algorithm easy to generalize to different types of wheelchairs and users.

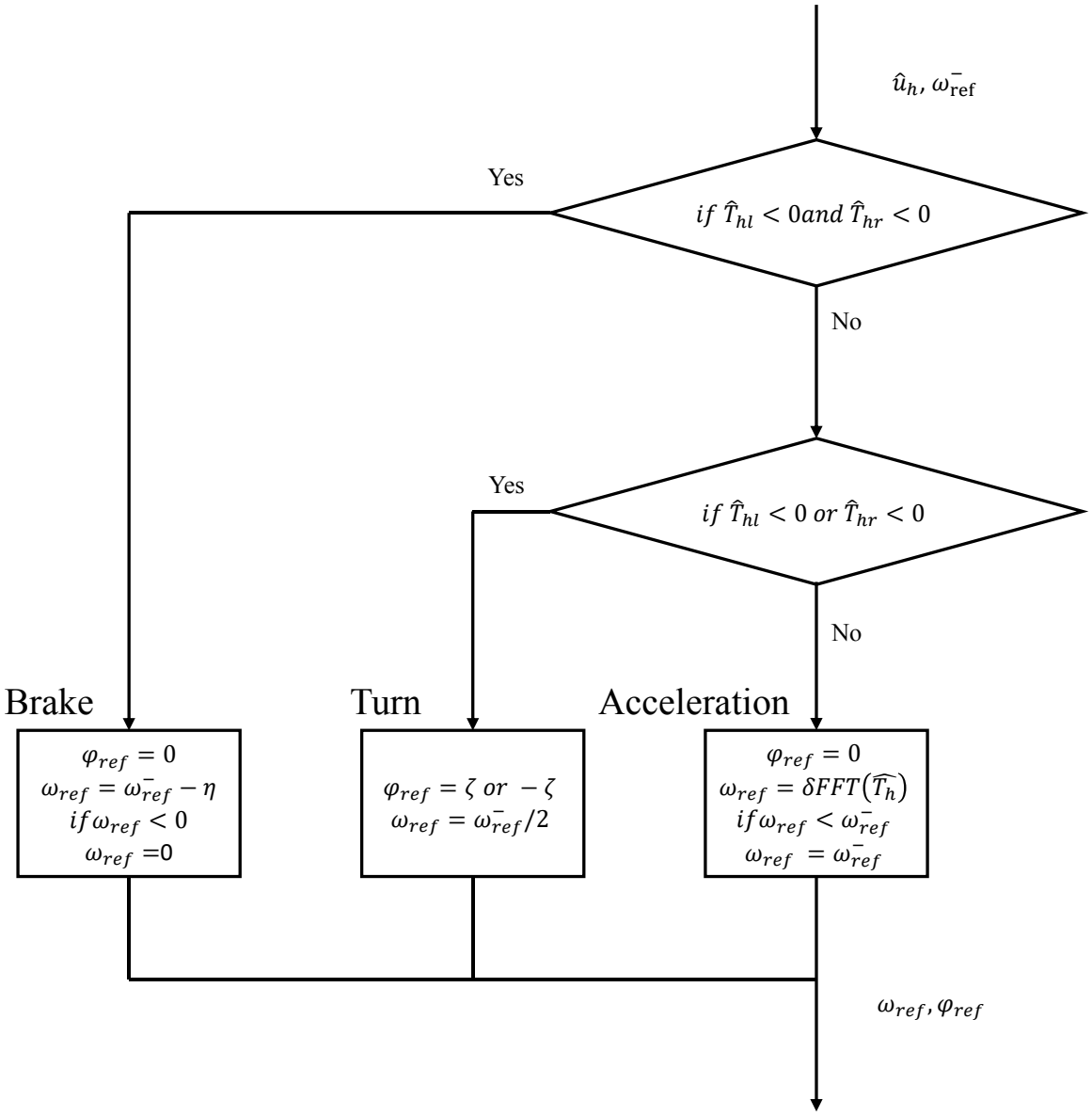

Figure 3.3.4. Reference generation diagram

3.3.4 Driving scenario and simulation results

In this section, the proposed observer and power-assisted algorithm are validated through simulations. The goal is to follow a given reference trajectory (the desired trajectory of the user) under the proposed assistive algorithm and the considered wheelchair dynamic (2.2.5). The human torque control signals are generated by a user. The interaction between the user and the virtual simulation is realised by the keyboard and screen as shown in Figure 3.3.5, where an user is able to manipulate the wheelchair by using the keyboard. The wheelchair is represented virtually by the model (2.2.1). The trajectory of the wheelchair is displayed on the screen such that the user can perform a closed-loop control. The PAW is assumed to

move on a flat surface. The human torques are represented by the positive half cycle of a sinusoidal. To perform the trajectory tracking, the user receives the trajectory of the wheelchair from screen and changes the frequencies and the direction of propulsions through the keyboard. Note that the trajectory is not imposed anymore in this simulation. The desired trajectory is computed from the user's propulsion.

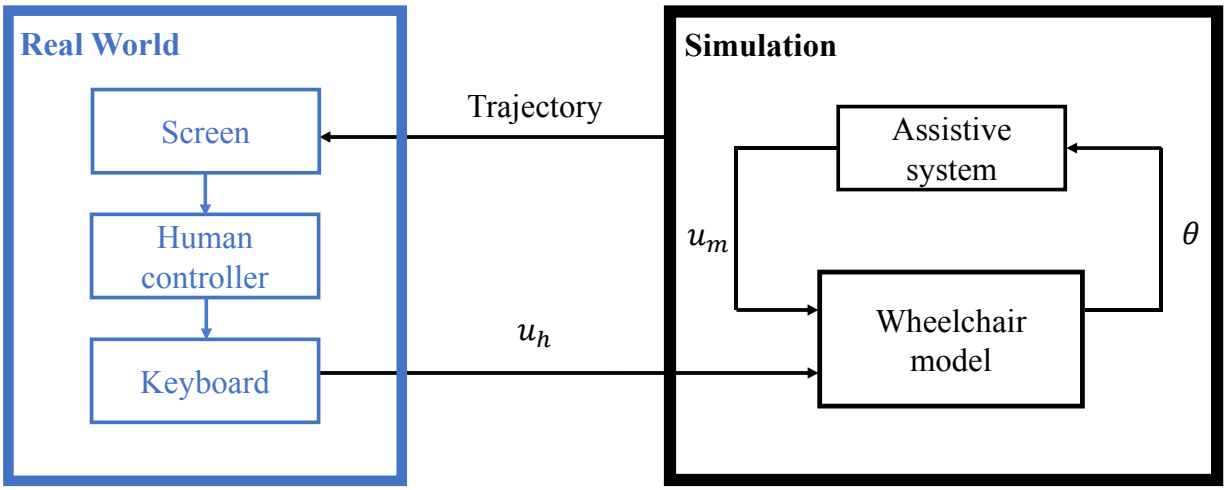

Figure 3.3.5. Wheelchair driving simulation structure

The parameters in Table I of Chapter 2 are used to carry out the simulation. Regarding the observer structure, a second degree polynomial is applied for the approximating function (3.2.1). For the reference trajectory generation in Figure 3.3.4, we use $\zeta = 0.7854$, $\eta = 0.01$ and $\delta = 2$. Regarding the FFT, we choose a time interval of $10s$ for the windowing. Before collecting enough data to compute the frequency in the beginning, we initialize the reference velocity at $\omega_{ref}(0) = 0.2m/s$ for the 10 first seconds. The PI controller gains are obtained via pole placement including, an anti-wind-up structure (Choi and Lee 2009).

- **Observer validation without power-assistance**

Four sequences of human torque are presented in Figure 3.3.6 (green, blue, black and red arrows). They represent respectively a sequence of accelerating, turning right, turning left and braking. The observer (red line) is able to perfectly reconstruct the frequency and amplitude of the signals when the pushing frequency is low enough (before 30s). When the frequency increases, the poles of the observer estimation are not fast enough to recover the correct amplitude. Notice that the poles have been chosen as fast as we could give by the sampling rate of the Autonomad Mobility wheelchair that will be used for the real time experiments.

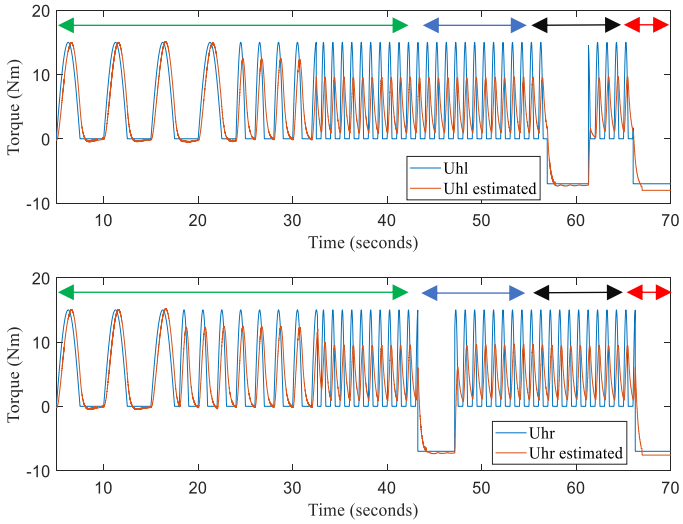

Figure 3.3.6. Human torque reconstruction without assistance (Time-varying sampling results)

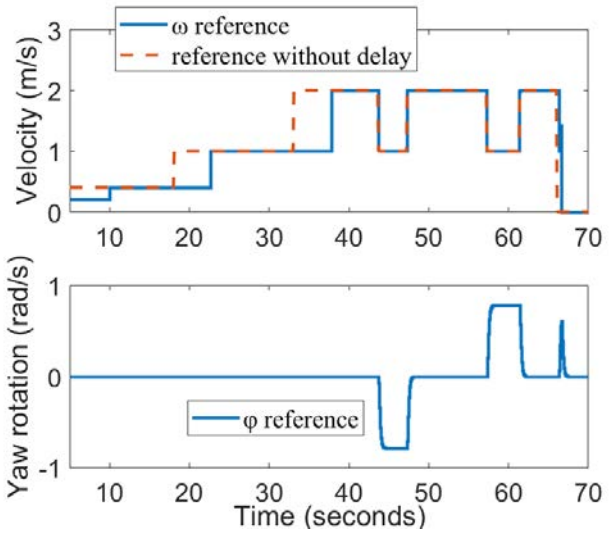

Figure 3.3.7. Reference signals generated from the previous estimated human torques (Time-varying sampling results)

3.3.4.2 Reference trajectory generation validation without power-assistance

We feed the estimated human torques obtained in the previous part to the reference generation bloc. In the green sequence, the frequencies of the human torque are *0.2Hz*, *0.5Hz* and *1Hz*. As mentioned previously, the ratio δ is 2. We notice in Figure 3.3.7 that the reference velocities ω are equal to *0.2m/s*, *0.4m/s*, *1m/s* and *2m/s* and correspond correctly to the frequencies of human torque. However, there is a delay of 5s between the reference velocities ω and the frequencies of human torque. This delay is due to the time interval of 5s

for the FFT. Also, the reference rotation is $\varphi_{ref} = -\zeta$ for $\hat{T}_{hl} < 0$ and $\varphi_{ref} = \zeta$ φ_{ref} for $\hat{T}_{hr} < 0$. In the red sequence, the algorithm detects the need to brake the wheelchair. Accordingly, ω and φ_{ref} are reduced to 0. Thus, overall via the UIO PI-observer and the proposed algorithm, the reference generation block can deliver the reference signals for the ω and φ_{ref} that are compatible with the user's will.

3.3.4.3 Predefined trajectory tracking

For this simulation, a predefined trajectory depicted in Figure 3.3.8 (including the start point and endpoint) is given. The goal is to show that a user can follow "naturally" this trajectory with the help of the proposed assistive system. The wheelchair has an initial velocity $\omega_{ref}(0) = 0.2m/s$.

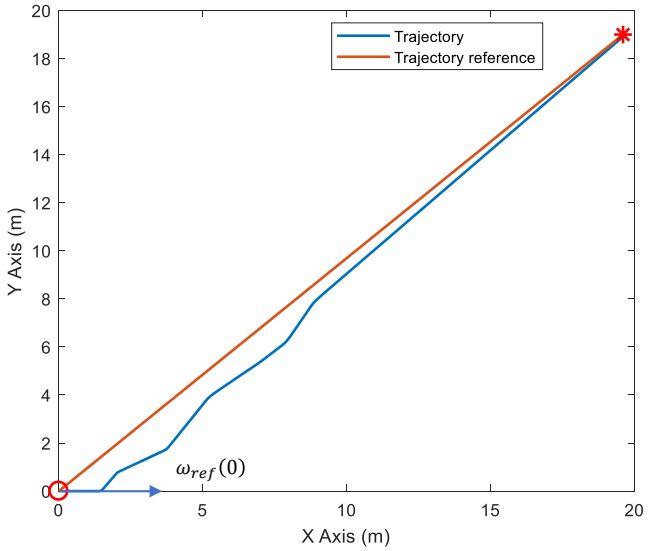

Figure 3.3.8. Predefined trajectory tracking performed by a human controller under the proposed assistive algorithm (Time-varying sampling results)

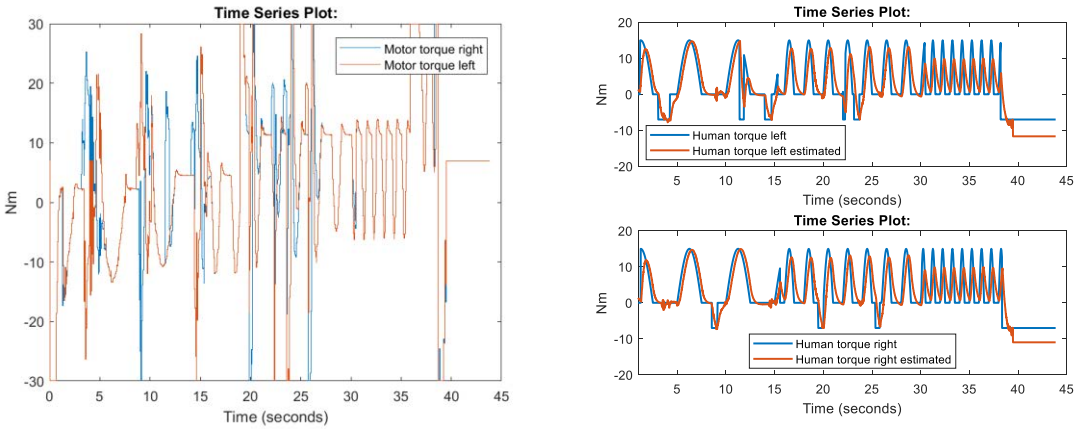

Figure 3.3.9. Assistive motor torques and unknown input estimation with assistive control (Time-varying sampling results)

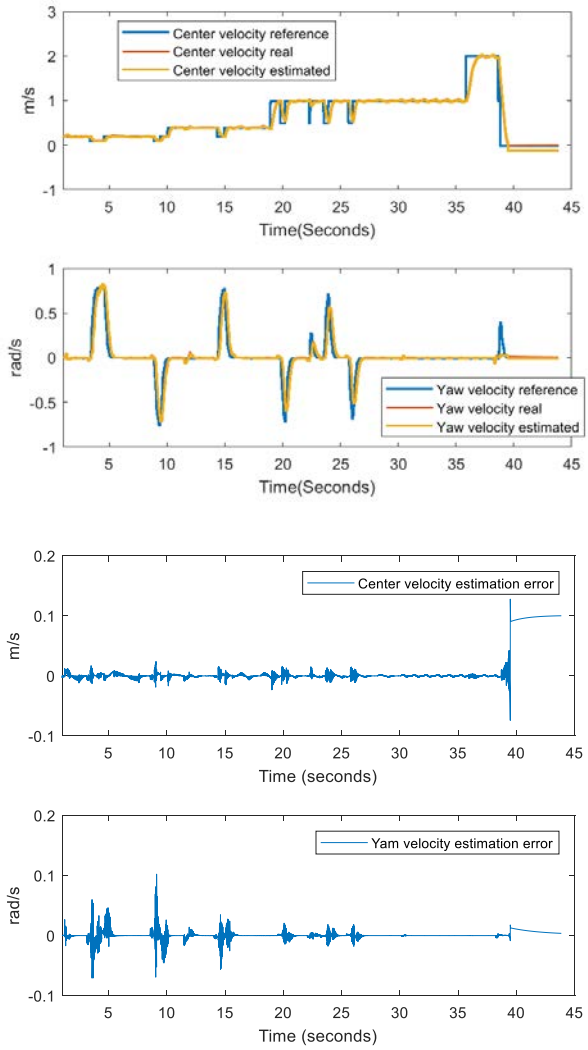

Figure 3.3.10. Reference signals, reference tracking performed by a PI controller and estimation errors (Time-varying sampling results)

As we can see in Figure 3.3.8, the human decides to go to the red-star goal position. For reaching the endpoint from the start point, the shortest trajectory is trivially the red line, Figure 3.3.8. The user aims to follow the red line as the reference trajectory. To this end, the user operates the wheelchair with the help of the assistive system described in Figure 3.3.4. As shown in Figure 3.3.8, the human corrects gradually the direction to point to the goal. In Figure 3.3.10, the observer reconstructs successfully the velocity ω and the rotation velocity φ. Moreover, the estimation errors are given and they are considerably smaller than the actual velocities. When the centre and yaw velocities are close to zero, we notice that the estimation performance is degraded. In addition, the centre and the yaw velocities follow correctly the reference signals which are generated according to the user's propelling. This tracking control is accomplished by the proposed PI controller configured with pole placement.

Remark 6. When the velocity of the wheelchair is equal to zero, the system (3.3.5) loses its observability. Therefore, the observer cannot provide a correct estimation in this condition. To solve this problem, the procedure switches off the assistive control when the velocity is below a given threshold (0.2 m/s for our case).

3.3.5 Summary

An observer-based assistive framework has been introduced is this section. To address the time-varying sampling period of the position encoders, we derived a LPV model for the wheelchair. Next, a nonlinear observer was proposed to reconstruct simultaneously the human torques, the centre velocity of the wheelchair, and the yaw velocity. An advantage of the assistive system is that the manoeuvrability of the wheelchair does not really depend on how strongly the users push. It depends only on the frequency and the direction of propulsion. Simulation results show the validity of the observer and of the reference trajectory generation. Simulations showed that the assistive torque strategy is compatible with trajectories defined by the user (point-to-point for example). However, we have not taken into account the system uncertainties such as the mass of the users or the road profiles, to handle such uncertainties the next section will discuss a robust framework.

3.4 Stability analysis and Robust Observer-based tracking control

In the previous section, an observer-based assistive control has been proposed. However, uncertainties on the mass of the user and on the viscous friction have not been taken into account. In this section, the sampling rate is constant and a robust observer-based tracking control is proposed for the uncertain human-wheelchair system. The mass of the users and the viscous friction coefficient are assumed unknown and bounded in a fixed-interval. The proposed algorithm covers various different situations such as different users for the same PAW and/or a varying ground profile etc. The goal is to guarantee the performance for the whole set of conditions via robust control design. Moreover, the user pushes a PAW depending on his/her will and the pushing techniques may not be stable for the uncertain human-wheelchair system (Oh et al. 2014). Unstable situations are, of course, to be completely avoided in order to prevent user injuries and/or wheelchair damages. Knowing that human propelling cannot be enforced, the proposed controller has to avoid the instable situations created by users' pushing and/or the combination of assistance and user torques.

Using a polytopic Takagi-Sugeno representation, the control design is formulated as a two-step LMI constraint optimization problem. Compared to computing the control gains and the observer gains simultaneously (that requires the use of pessimistic upper bounds to get a LMI formulation (Bennani et al. 2017) shows that a two-steps LMI observer-based control design could reduce the conservatism of the solutions. In this approach, the first step is to design a robust PI (Proportional-Integral) tracking control by temporally considering that the human torques are measured. The control gains calculated at the first step are kept for the second step. Assuming an unknown input observer in a PI form that uses n_p derivatives to reconstruct the human torques see Section3.2, the observer gains are then obtained by solving a second LMI constraint problem. The overall goal is to guarantee the closed-loop stability and an $\boldsymbol{\mathcal{H}}_\infty$ attenuation performance.

Since the human torques are considered as unknown inputs for tracking purposes, the human acts as a high-level controller to generate reference trajectory (Feng et al. 2018). As described in Section 3.3, these reference signals depend on the user's intention derived from the estimated torque signals. Consequently, an accurate torque estimation is not only important for tracking performance but also crucial for the manoeuvrability of the wheelchair. The

results obtained in this section have been partly published in the international conference IFAC ICONS 2019.

3.4.1 Polytopic Representation

Considering that both the mass and the viscous friction coefficient are not well known and possibly time varying, uncertainties are introduced in the nominal system (2.2.5) to get a discrete-time uncertain system, as follows:

$$E(m)x^{+} = A(m,\mathrm{K})x + Bu_h + Bu_m \tag{3.4.1}$$

As usual, the uncertainties are supposed bounded, $\underline{m} < m < \overline{m}$ and $\underline{\mathcal{K}} < \mathcal{K} < \overline{\mathcal{K}}$. The uncertain system (3.4.1) can be represented by the convex sum of linear models whose weights will depend on the unknown premise variables m and K. We rewrite the system as follows:

$$\sum_{j=1}^{2}\zeta_j(m)E_j x^{+} = \sum_{j=1}^{2}\sum_{i=1}^{2}\zeta_j(m)\vartheta_i(\mathrm{K})A_{ij}x + Bu_h + Bu_m \tag{3.4.2}$$

where ζ_i and ϑ_i are membership functions of unknown variables sharing a convex sum property, i.e. $\zeta_j \in [0,1]$, $\vartheta_i \in [0,1]$, $\sum_{j=1}^{2}\zeta_j(m) = 1$ and $\sum_{j=1}^{2}\sum_{i=1}^{2}\zeta_j(m)\vartheta_i(\mathrm{K}) = 1$. The matrices E_j and A_{ij} are known and correspond to the vertices of the polytope such that:

$$E_1 = E(\overline{m}), E_2 = E(\underline{m}),$$
$$A_{11} = A(\overline{m},\overline{\mathrm{K}}), A_{12} = A(\underline{m},\overline{\mathrm{K}}), A_{21} = A(\overline{m},\underline{\mathrm{K}}), A_{22} = A(\underline{m},\underline{\mathrm{K}}).$$

Using this polytopic representation and LMI techniques, we aim to provide a robust design for the proposed observer-based control. Notice that even if the mass m is uncertain, it is considered as constant during driving. However, the viscous friction K coefficient is time-varying. Therefore, a delayed Lyapunov function can be considered.

Note that when using a descriptor form, the uncertainties do not affect the input matrix B and therefore, the form (3.4.2) is kept as it can reduce significantly the pessimism of the results (Chadli and Guerra 2012). Moreover, the inversion of the non-singular matrix $E(m)$ is avoided.

3.4.2 Control Objective

The problem we are faced with, i.e. searching for the Lyapunov function and both the control and the observer gains in an uncertain framework and with unknown inputs, is not convex. Therefore, we decompose the problem into two steps with a guarantee of performances of the whole closed-loop.

Step 1 consists in designing a robust state feedback PI-like controller while temporally assuming that the states and the inputs are perfectly known. Step 2 consists in designing the observer to estimate the human torques and guarantee the closed-loop performance. The observer design uses a LMI constraints problem such that the uncertain system (3.4.1) with the proposed observer-based tracking controller satisfies the following requirements:

- When the reference signal $x_{ref} = 0$, the state of the uncertain system (3.4.1) and the estimation error e_o converge asymptotically to the origin.
- When the reference signal $x_{ref} \neq 0$ and $x_{ref}(\infty) = 0$, under null initial conditions (state and estimation error), the $\mathcal{L}_2$-norm of the estimation error e_o is bounded as follows:

$$\sum_{k=0}^{\infty} e_o{}^T e_o < \gamma_o \sum_{k=0}^{\infty} x_{ref}{}^T x_{ref}$$

3.4.3 Robust PI-like control design

In this section, states and inputs are perfectly known and we propose to design a robust PI-like controller for the uncertain system (3.4.1) via a LMI constraint problem. Considering that the reference is zero, the control law is:

$$\begin{cases} u_m = L_c M_c^{-1} \begin{bmatrix} x \\ e_{\text{int}} \end{bmatrix} - u_h \\ e_{\text{int}}{}^+ = e_{\text{int}} - x \end{cases} \tag{3.4.3}$$

where u_h is human torque and e_{int} corresponds to the integrator state of the integral part. With the controller (3.4.3) and the uncertain system (3.4.1), the closed loop dynamic can be written as:

$$\sum_{j=1}^{2}\zeta_j(m)E_{c_j}x_c^{+}=\left[\sum_{j=1}^{2}\sum_{i=1}^{2}\zeta_j(m)\vartheta_i(\mathrm{K})A_{c_{ij}}+B_cL_cM_c^{-1}\right]x_c \tag{3.4.4}$$

with $x_c=[x \quad e_{\mathrm{int}}]^T$, $j\in\{1,2\}$, $i\in\{1,2\}$ and the matrices:

$$E_{c_j}=\begin{bmatrix}E_j & 0_{n_x}\\ 0_{n_x} & I_{n_x}\end{bmatrix}, A_{c_{ij}}=\begin{bmatrix}A_{ij} & 0_{n_x}\\ -I_{n_x} & I_{n_x}\end{bmatrix}, B_c=\begin{bmatrix}B\\ 0_{n_x}\end{bmatrix}.$$

Moreover, using

$$\begin{aligned}E_c(m)&=\sum_{j=1}^{2}\zeta_j(m)E_{c_j}\\ A_c(m,\mathrm{K})&=\sum_{j=1}^{2}\sum_{i=1}^{2}\zeta_j(m)\vartheta_i(\mathrm{K})A_{c_{ij}}\end{aligned} \tag{3.4.5}$$

The system (3.4.4) can be written as the equality constraint:

$$\begin{bmatrix}A_c(m,\mathrm{K})+B_cL_cM_c^{-1} & -E_c(m)\end{bmatrix}\begin{bmatrix}x_c\\ x_c^{+}\end{bmatrix}=0 \tag{3.4.6}$$

Consider the following Lyapunov function candidate:

$$V(x_c)=x_c^{T}\sum_{j=1}^{2}\sum_{h=1}^{2}\zeta_j(m)\vartheta_i(\mathrm{K})P_{c_{ij}}x_c=x_c^{T}P_cx_c \tag{3.4.7}$$

with $P_{c_{ij}}\in\mathrm{R}^{2n_x}$, $j\in\{1,2\}$, $i\in\{1,2\}$ and:

$$P_c=P_c^{T}=\sum_{j=1}^{2}\sum_{i=1}^{2}\zeta_j(m)\vartheta_i(\mathrm{K})P_{c_{ij}}>0 \tag{3.4.8}$$

From this: $V(x_c^{+})=x_c^{+T}P_c^{+}x_c^{+}$ and $P_c^{+}=P_c^{+T}=\sum_{j=1}^{2}\sum_{h=1}^{2}\zeta_j(m)\vartheta_h(\mathrm{K}^{+})P_{c_{hj}}{}^{+}$. Effectively, as the mass is constant, $\zeta_j(m)$ is the same as (3.4.7), whereas K^{+} is the friction coefficient at the next sample after K. Therefore we use different indices to represent the membership functions $\vartheta_i(\mathrm{K})$ and $\vartheta_h(\mathrm{K}^{+})$ in different moment.

Theorem 3. *The uncertain system (3.4.1) together with the controller (3.4.3) is asymptotically stable if there exist symmetric positive-definite matrices* $X\in\mathrm{R}^{2n_x}$,

$j \in \{1,2\}$, $i \in \{1,2\}$, $h \in \{1,2\}$, *a matrix* $L_c \in \mathrm{R}^{n_u \times 2n_x}$ *and a regular matrix* $M_c \in \mathrm{R}^{2n_x}$ *such that:*

$$\begin{bmatrix} -X_{ij} & (*) \\ A_{c_{ij}} M_c + B_c L_c & X_{hj} - E_{c_j} M_c - \left(E_{c_j} M_c\right)^T \end{bmatrix} < 0 \tag{3.4.9}$$

Proof*:* The variation of the Lyapunov function (3.4.7) can be written as the following inequality constraint:

$$\Delta V\left(x_c\right) = V\left(x_c^+\right) - V\left(x_c\right) = \begin{bmatrix} x_c \\ x_c^+ \end{bmatrix}^T \begin{bmatrix} -P_c & 0_{2n_x} \\ 0_{2n_x} & P_c^+ \end{bmatrix} \begin{bmatrix} x_c \\ x_c^+ \end{bmatrix} < 0 \tag{3.4.10}$$

Form Lemma 3, the inequality (3.4.10) under the equality constraint (3.4.6) is equivalent to the inequality:

$$\begin{bmatrix} 0_{2n_x} \\ M_c^{-T} \end{bmatrix} \begin{bmatrix} A_c\left(m,\mathrm{K}\right) + B_c L_c M_c^{-1} & -E_c \end{bmatrix} + (*) + \begin{bmatrix} -P_c & 0_{2n_x} \\ 0_{2n_x} & P_c^+ \end{bmatrix} < 0 \tag{3.4.11}$$

Using the property of congruence with $\mathrm{diag}\left(M_c^T, M_c^T\right)$, (3.4.11) is equivalent to:

$$\begin{bmatrix} -M_c^T P_c M_c & (*) \\ A_c\left(m,\mathrm{K}\right) M_c + B_c L & M_c^T P_c^+ M_c - E_c\left(m\right) M_c - \left(E_c\left(m\right) M_c\right)^T \end{bmatrix} < 0 \tag{3.4.12}$$

Since (3.4.5), (3.4.6) hold and $\sum_{j=1}^{2}\sum_{j=1}^{2}\sum_{h=1}^{2} \zeta_j\left(m\right) \vartheta_i\left(K\right) \vartheta_h\left(\mathrm{K}^-\right) = 1$, the inequality (3.4.12) holds if:

$$\begin{bmatrix} -M_c^T P_{c_{ij}} M_c & (*) \\ A_{c_{ij}} M_c + B_c L_c & M_c^T P_{c_{hj}} M_c - E_{c_j} M_c - \left(E_{c_j} M_c\right)^T \end{bmatrix} < 0 \tag{3.4.13}$$

Letting $X_{hj} = M_c^T P_{c\,hj} M_c$ and $X_{ij} = M_c^T P_{c_{ij}} M_c$, we obtain directly the linear matrix inequality (3.4.9).

3.4.4 Stability of observer-based control

In the previous section, we designed the controller (3.4.3) assuming the human input u_h is measured. To get rid of this assumption, we next use an unknown input observer to estimate

the human torque. The objectives of this step are twofold; designing the observer and guaranteeing stability and performance of the whole closed-loop. In this part, the stability analysis focuses on the closed-loop dynamic system which is enclosed by the red frame in Figure 3.4.1.

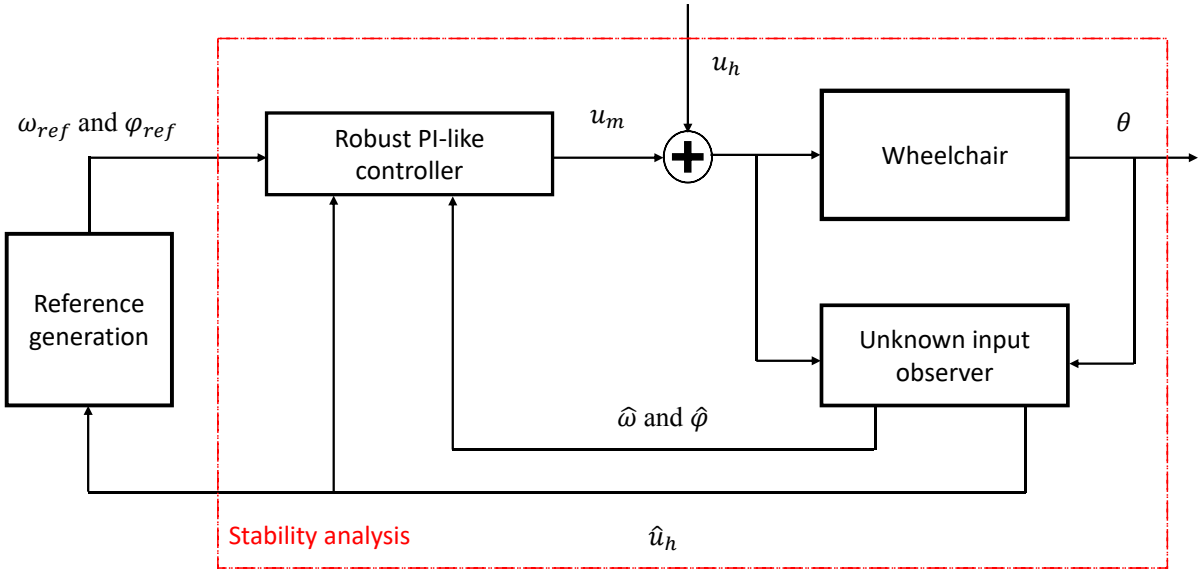

Figure 3.4.1. The closed-loop system with the observer-based tracking control

Using the polynomial approximation (3.2.1), the extended state vector $x_o = \left[x,\ T_{hr}^{n_p T}, T_{hl}^{n_p T} \right]^T \in \mathrm{R}^{2n_p+n_x}$. The uncertain system (3.4.1) can be rewritten as:

$$\begin{aligned} &\sum_{j=1}^{2} \zeta_j(m) E_{o_j} x_o^{+} = \sum_{j=1}^{2}\sum_{i=1}^{2} \zeta_j(m)\, \vartheta_i(\mathrm{K}) A_{o_{ij}} x_o + B_o u_m \\ &y = C_o x_o \end{aligned} \tag{3.4.14}$$

where

$$E_{o_j} = \begin{bmatrix} E_j & 0_{n_x \times 2n_p} \\ 0_{2n_p \times n_x} & I_{2n_p} \end{bmatrix}, A_{o_{ij}} = \begin{bmatrix} A_{ij} & BB_r & BB_l \\ 0_{n_p \times n_x} & \Gamma_{n_p} & 0_{n_p \times n_p} \\ 0_{n_p \times n_x} & 0_{n_p \times n_p} & \Gamma_{n_p} \end{bmatrix}, C_o = \begin{bmatrix} C & 0_{n_y \times 2n_p} \end{bmatrix}, B_o = \begin{bmatrix} B \\ 0_{2n_p \times n_x} \end{bmatrix}.$$

Based on the nominal system (2.2.5), the observer is defined as follows:

$$\begin{aligned} &E_o \hat{x}_o^{+} = A_o \hat{x}_o + B_o u_m + \bar{G}_o^{-1} \bar{K}_o \left(y - \hat{y} \right) \\ &\hat{y} = C_o \hat{x}_o \end{aligned} \tag{3.4.15}$$

where $\bar{G}_o^{-1}\bar{K}_o$ is the observer gain and E_o, A_o, B_o, C_o are the nominal system matrices in (3.2.7). The estimation error is $e_o = x_o - \hat{x}_o$. As the observer is well-posed, we can study the dynamic of $E_o e_o = E_o\left(x_o - \hat{x}_o\right)$ which can be written as:

$$E_o e_o^+ + \begin{bmatrix} \sum_{j=1}^{2} \zeta_j(m) E_j - E \\ 0_{2n_p \times n_x} \end{bmatrix} x^+ = \left(A_o - \bar{G}_o^{-1}\bar{K}_o C_o\right) e_o + \begin{bmatrix} \sum_{j=1}^{2}\sum_{i=1}^{2} \zeta_j(m)\vartheta_i(\mathrm{K}) A_{ij} - A \\ 0_{2n_p \times n_x} \end{bmatrix} x$$

(3.4.16)

The observer-based control law is computed as follows:

$$\begin{cases} u_m = L_c M_c^{-1} \begin{bmatrix} \hat{x} \\ e_{\mathrm{int}} \end{bmatrix} - \hat{u}_h \\ e_{\mathrm{int}}{}^+ = e_{\mathrm{int}} - \hat{x} + x_{ref} \end{cases} \tag{3.4.17}$$

where x_{ref} is the reference velocity, $\hat{x}$ is the estimated state vector and $\hat{u}_h$ are the estimated human torques. Since $\hat{u}_h = u_h - \begin{bmatrix} 0_{n_x} & B_r & B_l \end{bmatrix} e_o$, $\hat{x} = x - \begin{bmatrix} I_{n_x} & 0_{n_x \times n_p} & 0_{n_x \times n_p} \end{bmatrix} e_o$ and $x_c = \begin{bmatrix} x^T & e_{\mathrm{int}}{}^T \end{bmatrix}^T$, the controller (3.4.17) can be expressed as follows:

$$\begin{cases} u_m = \begin{bmatrix} L_c M_c^{-1} & L_o \end{bmatrix} \begin{bmatrix} x_c \\ e_o \end{bmatrix} - u_h \\ e_{\mathrm{int}}{}^+ = e_{\mathrm{int}} - x + \begin{bmatrix} I_{n_x} & 0_{n_x \times n_p} & 0_{n_x \times n_p} \end{bmatrix} e_o + x_{ref} \end{cases} \tag{3.4.18}$$

where $L_o = \begin{bmatrix} -L_c M_c^{-1} \begin{bmatrix} I_{n_x} \\ 0_{n_x} \end{bmatrix} & B_r & B_l \end{bmatrix}$. The uncertain system (3.4.1) together with the observer-based controller (3.4.15)-(3.4.18) give the following closed loop dynamics:

$$\begin{bmatrix} \sum_{j=1}^{2}\sum_{i=1}^{2} \zeta_j(m)\vartheta_i(\mathrm{K}) A_{oc_{ij}} & -\sum_{j=1}^{2} \zeta_j(m) E_{oc_j} & B_{oc} \end{bmatrix} \begin{bmatrix} x_{oc} \\ x_{oc}{}^+ \\ x_{ref} \end{bmatrix} = 0 \tag{3.4.19}$$

where the closed-loop vector $x_{oc} = \begin{bmatrix} x_c^T & e_o{}^T \end{bmatrix}^T$ and $e_o = C_{oc} x_{oc}$ with $C_{oc} = \begin{bmatrix} 0_{(2n_p+n_x)\times 2n_x} & I_{(2n_p+2)} \end{bmatrix}$.

The matrices of (3.4.19) are:

$$E_{oc_j} = \begin{bmatrix} E_{c_j} & 0_{2n_x \times (2n_p+n_x)} \\ \begin{bmatrix} E_j - E & 0_{n_x} \\ 0_{2n_p \times 2n_x} \end{bmatrix} & E_o \end{bmatrix}, A_{oc_{ij}} = \begin{bmatrix} A_{c_{ij}} + B_c L_c M_c^{-1} & B_c L_o \\ \begin{bmatrix} A_{ij} - A & 0_{n_x} \\ 0_{2n_p} \end{bmatrix} & A_o - \bar{G}_o^{-1} \bar{K}_o C_o \end{bmatrix},$$

$$B_{oc} = \begin{bmatrix} 0_{n_x} \\ I_{n_x} \\ 0_{(2n_p+n_x)\times n_x} \end{bmatrix}.$$

Consider the following Lyapunov function candidate:

$$V(x_{oc}) = x_{oc}{}^T \sum_{j=1}^{2} \sum_{h=1}^{2} \zeta_j(m) \vartheta_i(\mathrm{K}) P_{oc_{ij}} x_{oc} = x_{oc}{}^T P_{oc} x_{oc} > 0 \tag{3.4.20}$$

with $P_{oc} \in \mathrm{R}^{(2n_p+3n_x)}$ positive-definite and $P_{oc} = \sum_{j=1}^{2} \sum_{h=1}^{2} \zeta_j(m) \vartheta_i(\mathrm{K}) P_{oc_{ij}}$. At the next sample, the Lyapunov function candidate $V(x_{oc}{}^+) = x_{oc}{}^{+T} \sum_{j=1}^{2} \sum_{i=1}^{2} \zeta_j(m) \vartheta_i(\mathrm{K}^+) P_{oc_{hj}} x_{oc}{}^+ = x_{oc}{}^{+T} P_{oc}{}^+ x_{oc}{}^+$.

Theorem 4. *Given matrices L_c and M_c, if there exist positive definite matrices $P_{oc} \in \mathrm{R}^{(2n_p+3n_x)}$, $j \in \{1,2\}$, $i \in \{1,2\}$, $h \in \{1,2\}$, matrices $\bar{K}_o \in \mathrm{R}^{(2n_p+n_x)\times n_y}$, $\bar{G}_1 \in \mathrm{R}^{2n_x}$, $\bar{G}_2 \in \mathrm{R}^{(2n_p+n_x)\times 2n_x}$, a regular matrix $\bar{G}_o \in \mathrm{R}^{(2n_p+n_x)}$, and a positive scalar γ such that:*

$$\Pi_{ijh}^1 + \Pi_{ijh}^2 + \Pi_{ijh}^{2\,T} < 0 \tag{3.4.21}$$

where

$$\Pi_{ijh}^1 = \begin{bmatrix} -P_{oc_{ij}} + C_{oc}{}^T C_{oc} & 0_{(2n_p+3n_x)} & 0_{(2n_p+6)\times n_x} \\ 0_{(2n_p+3n_x)} & P_{oc_{hj}} & 0_{(2n_p+3n_x)\times n_x} \\ 0_{n_x \times (2n_p+3n_x)} & 0_{n_x \times (2n_p+3n_x)} & -\gamma_o I_{n_x} \end{bmatrix}, \Pi_{ijh}^2 = \begin{bmatrix} \grave{o} G_{oc} \\ G_{oc} \\ 0_{n_x \times (2n_p+3n_x)} \end{bmatrix} \begin{bmatrix} A_{oc_{ij}} & -E_{oc_j} & B_{oc} \end{bmatrix},$$

$$G_{oc} = \begin{bmatrix} \bar{G}_1 & 0_{2n_x \times (2n_p+n_x)} \\ \bar{G}_2 & \bar{G}_o \end{bmatrix}.$$

then the observer-based tracking controller solves the control objective stated in Section 3.4.2.

Proof: The inequality (3.4.21) can be rewritten as follows:

$$\Pi_{ijh}^{1}+\begin{bmatrix} \partial G_{oc} \\ G_{oc} \\ 0_{n_x\times(2n_p+3n_x)} \end{bmatrix}\begin{bmatrix} A_{oc_{ij}} & -E_{oc_j} & B_{oc} \end{bmatrix}+(*)<0 \tag{3.4.22}$$

From (3.4.22) and $\sum_{j=1}^{2}\sum_{j=1}^{2}\sum_{h=1}^{2}\zeta_j(m)\vartheta_i(\mathrm{K})\vartheta_h(\mathrm{K}^+)=1$, we obtain:

$$\Pi_{\zeta\vartheta\vartheta^+}^{1}+\begin{bmatrix} \partial G_{oc} \\ G_{oc} \\ 0_{n_x\times(2n_p+3n_x)} \end{bmatrix}\begin{bmatrix} A_{oc_{\zeta\vartheta}} & -E_{oc_\zeta} & B_{oc} \end{bmatrix}+(*)<0 \tag{3.4.23}$$

with the notation:

$$\Pi_{\zeta\vartheta\vartheta^+}^{1}=\sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{h=1}^{2}\zeta_i(m)\vartheta_j(\mathrm{K})\vartheta_h(\mathrm{K}^+)\Pi_{ijh}^{1},\ A_{c_{\zeta\vartheta}}=\sum_{i=1}^{2}\sum_{j=1}^{2}\zeta_i(m)\vartheta_j(\mathrm{K})A_{c_{ij}},\ E_{c_\zeta}=\sum_{j=1}^{2}\zeta_i(m)E_{c_i}.$$

Using Lemma 1 and the constraint (3.4.19), we have:

$$\begin{bmatrix} -P_{oc}+{C_{oc}}^{T}C_{oc} & 0_{2n_p+3n_x} & 0_{(2n_p+3n_x)\times n_x} \\ 0_{2n_p+3n_x} & {P_{oc}}^{+} & 0_{(2n_p+3n_x)\times n_x} \\ 0_{n_x\times(2n_p+3n_x)} & 0_{n_x\times(2n_p+3n_x)} & -\gamma_o I_{n_x\times n_x} \end{bmatrix}<0 \tag{3.4.24}$$

Pre- and post-multiplying (3.4.24) with the vector $\begin{bmatrix} {x_{oc}}^{T} & x_{oc}^{+T} & x_{ref}^{T} \end{bmatrix}^{T}$, we derive the following inequality:

$$\Delta V(x_{oc})+{e_o}^{T}e_o-\gamma_o{x_{ref}}^{T}x_{ref}<0 \tag{3.4.25}$$

- When $x_{ref}=0$, we can conclude that:

$$\Delta V(x_{oc})<0$$

Thus, the closed loop trajectory x_{oc} converges asymptotically to the origin.

- When $x_{ref}\neq 0$, $x_{ref}(\infty)=0$ and $V(x_{oc}(0))=0$, we obtain:

$$\sum_{k=0}^{\infty} e_o{}^T e_o < \gamma_o \sum_{k=0}^{\infty} x_{ref}{}^T x_{ref}$$

i.e. the $\mathcal{L}_2$-norm of the estimation error is bounded. ■

Remark 7. The matrix product $G_{oc} A_{oc_{hj}}$ is equal to:

$$G_{oc} A_{oc_{hj}} = \begin{bmatrix} \bar{G}_1 \left(A_{c_{ij}} + B_c L_c M_c^{-1} \right) & \bar{G}_1 B_c L_o \\ \bar{G}_2 \left(A_{c_{ij}} + B_c L_c M_c^{-1} \right) + \bar{G}_o \begin{bmatrix} A_{ij} - A & 0_{n_x} \\ 0_{2n_p \times 2n_x} & \end{bmatrix} & \bar{G}_2 B_c L_o + \bar{G}_o A_o - \bar{K}_o C_o \end{bmatrix}$$

The inequalities (3.4.21) are LMIs for a given scalar ϵ. A numerical gridding search for ϵ is carried out in a given interval.

3.4.5 Simulation results

In this section, we will validate first the robust PI control (3.4.3) and then the robust observer-based control presented in (3.4.15)-(3.4.17). All the LMI problems are solved with the Yalmip toolbox (Löfberg 2004). To carry out the simulations, the nominal parameters in Table I are used. A second-degree derivative is applied for the human torque approximation (3.2.1). The mass of users varies between 80kg and 120kg and the viscous friction coefficient changes between $3\mathrm{N}\cdot\mathrm{m}\cdot\mathrm{s}$ and $7\mathrm{N}\cdot\mathrm{m}\cdot\mathrm{s}$. The user pushing profile is represented by a sinusoidal signal.

Solving the LMI conditions in Theorem 3, the following control gains are obtained:

$$L_c M_c^{-1} = \begin{bmatrix} -399.92 & -432.59 & 78.38 & 81.89 \\ -399.92 & 432.59 & 78.38 & -81.89 \end{bmatrix}$$

In Figure 3.4.2, the red line represents the reference trajectory. To follow the reference trajectory, the reference velocities are given in Figure 3.4.3. Then, the proposed PI-like controller needs to track the given reference velocities in presence of uncertainties on the mass of the user and the viscous friction. For the first trial, we reduce the nominal values by 20% (namely $m = 80\mathrm{kg}$ and $\mathrm{K} = 4\mathrm{N}\cdot\mathrm{m}\cdot\mathrm{s}$). For the second trial, we increase the nominal values of these two parameters by 20% (namely $m = 120\mathrm{kg}$ and $\mathrm{K} = 6\mathrm{N}\cdot\mathrm{m}\cdot\mathrm{s}$). As shown in Figure 3.4.3, the proposed PI-like controller is able to track well the given reference

velocities even the uncertainties on the mass of users and the viscous friction coefficient are present. Moreover, the trajectory tracking is also achieved see Figure 3.4.2.

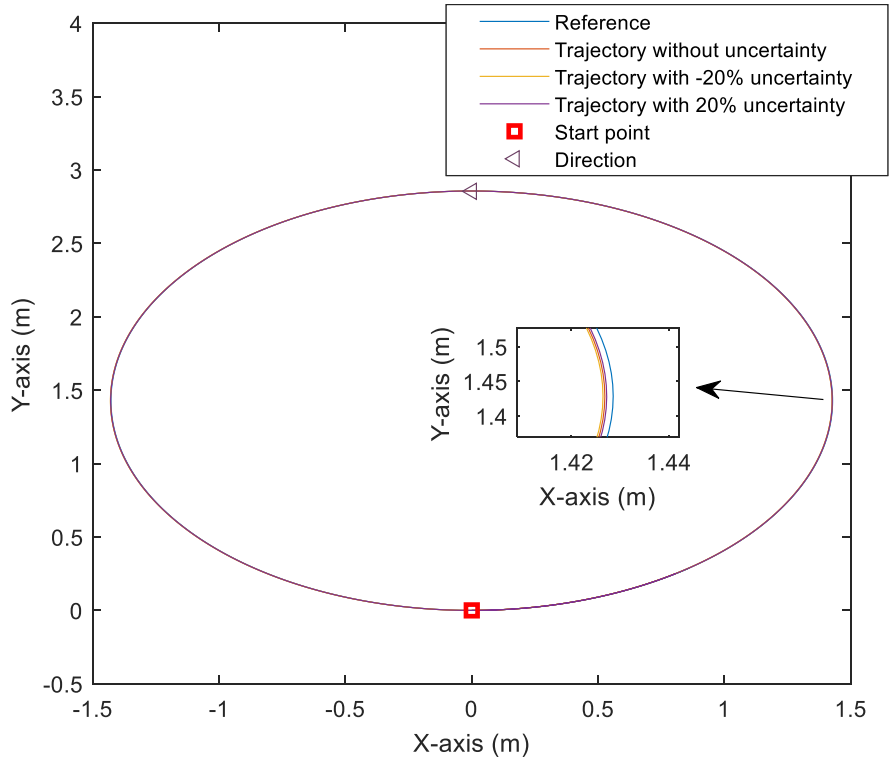

Figure 3.4.2. Obtained trajectories with the proposed robust PI-like controller

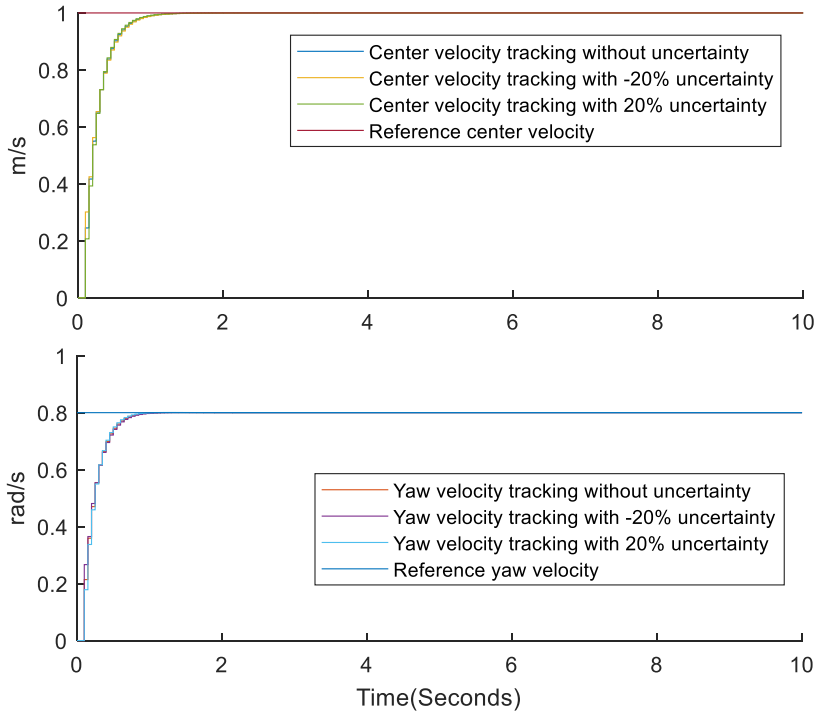

Figure 3.4.3. Obtained velocities with the proposed robust PI-like controller

By solving the LMI conditions in Theorem 4, the following observer gains are obtained:

$$\bar{G}_o^{-1}\bar{K}_o = \begin{bmatrix} 14.54 & -8.08 \\ -8.08 & 14.54 \\ 106.57 & -57.89 \\ 119.06 & -64.61 \\ -57.89 & 106.57 \\ -64.61 & 119.06 \end{bmatrix}$$

We carry out the same scenario as the previous simulation. To simulate uncertainties, the real values of the mass and of the viscous friction have the same variations. The trajectory tracking is still accomplished by following the given reference as depicted in Figure 3.4.5. Thereafter, the measurement of human torques are not available. As shown in Figure 3.4.5, the proposed observer-based controller has nearly the tracking performance as the previous PI-like controller. Since the observer does not reconstruct perfectly human torques see Figure 3.4.6, the tracking performance is degraded see the zoom-in of Figure 3.4.5.

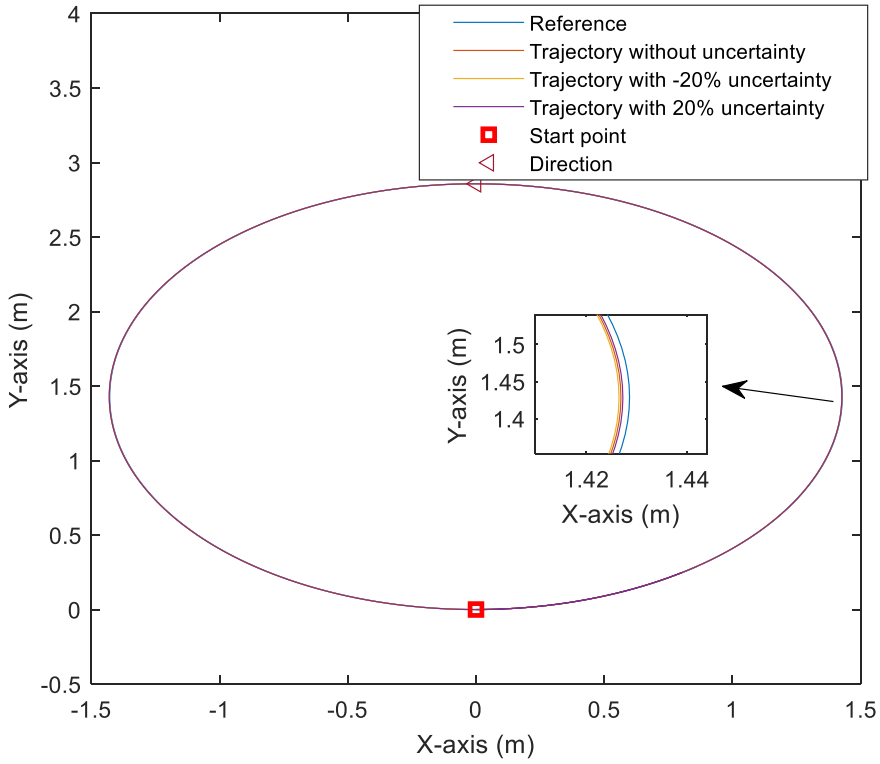

Figure 3.4.4. Simulation results with the proposed observer-based controller

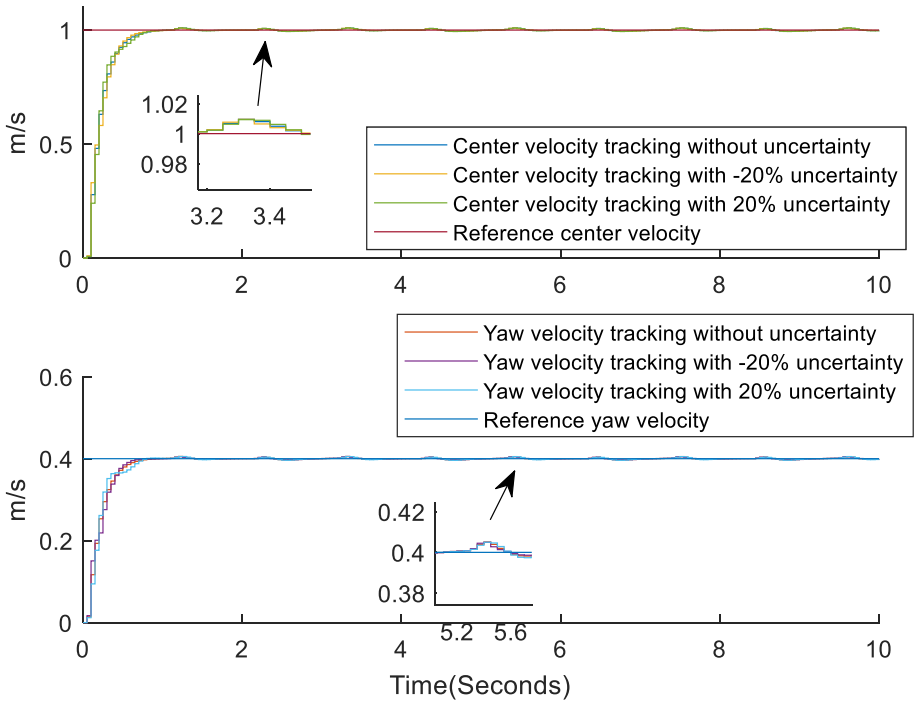

Figure 3.4.5. Obtained velocities with the proposed observer-based controller

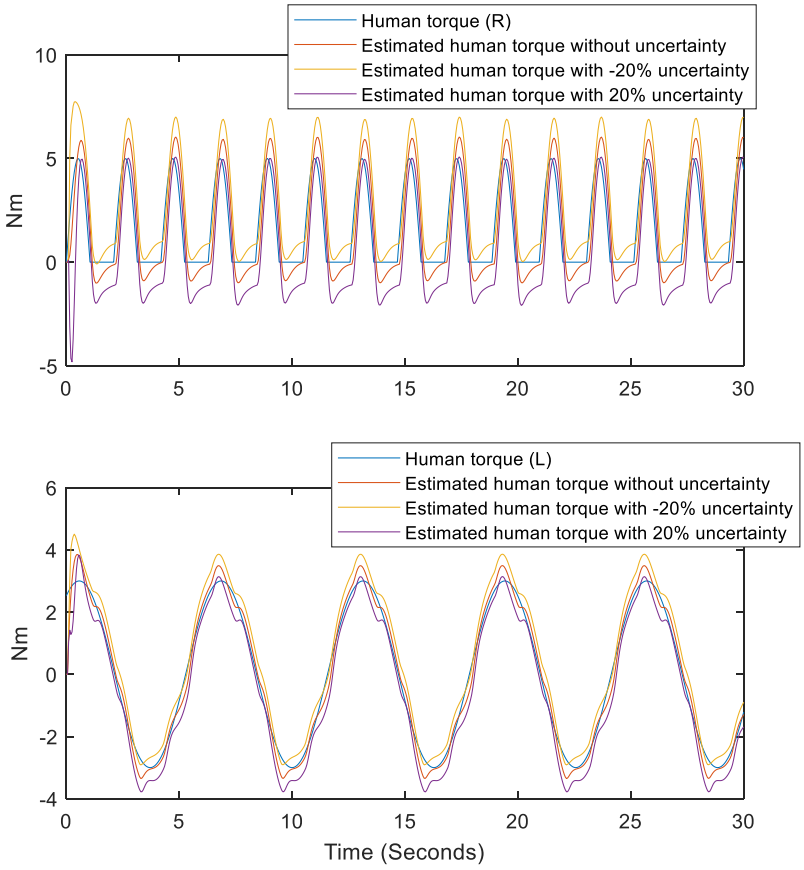

Figure 3.4.6. Obtained estimation of human torques

Remark 8. As shown in Figure 3.4.6, the unknown input estimations $\hat{u}_h$ do not completely match u_h. Of course, the proposed observer cannot distinguish the influence from uncertainties and unmeasurable human torques via the angular velocities. As a result, the observer captures both the influence of system uncertainties and of the unknown inputs u_h into the estimated signals $\hat{u}_h$. We remind that only the frequency of human torques is needed for the trajectory reference generation rather than a perfect match in amplitude.

3.4.6 Summary

In this section, a robust observer-based tracking controller for PAWs was proposed. Using a polytopic representation, the control design is formulated as a two-step LMI optimization problem. The first step is to design PI-like control assuming human torques are measured. Then, using the obtained control gains, the observer gains are derived by solving proposed

LMIs such that the stability of the closed-loop system and the estimation performance are guaranteed. Nevertheless, to guarantee that the control performances are not deteriorated by the actuator saturations, these constraints have to be taken into account in the design of the control. These issues will be treated in the next section.

3.5 Robust Observer-based control with constrained inputs

In PAW systems, the electrical motors have maximum torque limits owing to physical constraints. The performance, such as closed-loop stability, or can be seriously degraded due to actuator saturations. Using the control strategy obtained in Section 3.4, the closed-loop stability may not be guaranteed under saturations. Therefore, this section elaborates a robust observer-based tracking controller under input saturations. In our case, the limited torques provided by two motors. Moreover, an anti-windup is added to deal with the overshooting of the integral state due to actuator saturations. Taking these actuator constraints and an anti-windup component, the control gains and the observer gains are computed by a two-steps algorithm with LMI conditions.

3.5.1 Problem formulation

The uncertain system (3.4.1) under input saturations can be rewritten as:

$$E(m)x^{+} = A(m,\mathrm{K})x + Bu_h + B\mathrm{sat}(u_m) \qquad (3.5.1)$$

where the saturation function $\mathrm{sat}(\square)$ is given by $\mathrm{sat}\left(u_{m(l)}\right) = \mathrm{sign}\left(u_{m(l)}\right)\min\left(\left|u_{m(l)}\right|, u_{\max}\right)$, $l \in \{1, 2..., n_u\}$ and n_u is number of control inputs. In our case, $n_u = 2$. The observer-based control design follows a simple two-step design procedure as presented Section 3.4. After taking into account actuator saturations, the complexity of the control increases; parameters to search are related to the feedforward part, feedback and anti-wind-up parts. It appears that searching first for the control part (that constrains highly the solutions) and second for the observer often ends with unfeasible and/or poor performance solutions. Therefore, the algorithm has been modified, first we search for the UIO PI-observer and second for the control part. In addition, the following Lemma and assumption will be imposed for the control design.

Lemma 4. The inequality on the dead-zone nonlinearity (3.5.16) satisfies for any positive diagonal matrix S:

$$u_m{}^T S^{-1}\phi(u_m) + \phi^T(u_m) S^{-1} u_m - 2\phi^T(u_m) S^{-1}\phi(u_m) > 0 \tag{3.5.2}$$

The complete proof can be found in (Mulder et al. 2009).

Assumption 1. The motor input vector u_m is bounded in amplitude such as:

$$-u_{\max(l)} \le u_{m(l)} \le u_{\max(l)}, l \in \{1, 2, ..., n_x\} \tag{3.5.3}$$

where the maximum motor torque $u_{\max(l)}$ provided by the electrical motor l is known.

Remark 9. The saturation function (3.5.3) is not necessarily symmetric respect to the origin. However, asymmetric actuator saturation can be translated to a symmetric saturation as (3.5.3). Regarding to our PAW application, electrical motors would provide a same maximum torque for both the positive and the negative direction. Consequently, a symmetric saturation function is used here.

Based on the dynamics (3.5.1), we propose a robust observer-based tracking controller using the observer (3.2.7) obtained in Section 3.2.2. Let us consider the observer gains as $G_o^{-1} K_o = \begin{bmatrix} K_x \\ K_{u_h} \end{bmatrix}$. The gains K_x and K_{u_h} are used respectively to estimate the state vector x and reconstruct the unknown inputs. The observer (3.2.7) can be rewritten as:

$$\begin{cases} E\hat{x}^+ = Ax + B\text{sat}(u_m) + B\hat{u}_h + K_x(y - \hat{y}) \\ \hat{u}_h{}^+ = \Gamma_{n_p}\hat{u}_h + K_{u_h}(y - \hat{y}) \\ \hat{y} = C\hat{x} \end{cases} \tag{3.5.4}$$

The estimation error e_{o_x} of the state vector x is:

$$e_{o_x} = x - \hat{x} \tag{3.5.5}$$

with the uncertain system dynamic (3.5.1) and the observer (3.5.4). We obtain the dynamic of e_{o_x} as:

$$\big(E(m) - E\big)x^+ + Ee_{o_x}{}^+ = \big(A(m,\mathrm{K}) - A\big)x + \big(A - K_x C_d\big)e_{o_x} + B\big[u_h - \hat{u}_h\big] \tag{3.5.6}$$

The tracking error e_c is:

$$e_c = x_{ref} - \hat{x} \tag{3.5.7}$$

Then, the state vector x and the integral part e_{int} can be expressed as follows:

$$\begin{aligned} x &= x_{ref} - e_c + e_{o_x} \\ x^+ &= x_{ref}{}^+ - e_c{}^+ + e_{o_x}{}^+ \\ e_{\text{int}}{}^+ &= e_{\text{int}} + e_c \end{aligned} \tag{3.5.8}$$

Replacing the state vector x by the expression (3.5.8), (3.5.6) becomes:

$$\begin{aligned} &\left(-E(m)+E\right)e_c{}^+ + E(m)e_{o_x}{}^+ = \left(-A(m,\mathrm{K})+A\right)e_c + \left(A(m,\mathrm{K})-K_xC_d\right)e_{o_x} \\ &\left(A(m,\mathrm{K})-A\right)x_{ref} + \left(-E(m)+E\right)x_{ref}{}^+ + B\left[u_h - \hat{u}_h\right] \end{aligned} \tag{3.5.9}$$

The complete open loop dynamic from (3.5.1) and (3.5.4) can be written as:

$$\begin{aligned} &\begin{bmatrix} -E(m) & 0 & E(m) \\ 0 & I & 0 \\ -E(m)+E & 0 & E(m) \end{bmatrix}\begin{bmatrix} e_c{}^+ \\ e_{\text{int}}{}^+ \\ e_{o_x}{}^+ \end{bmatrix} = \begin{bmatrix} -A(m,\mathrm{K}) & 0 & A(m,\mathrm{K}) \\ I & I & 0 \\ -A(m,\mathrm{K})+A & 0 & A(m,\mathrm{K})-K_xC \end{bmatrix}\begin{bmatrix} e_c \\ e_{\text{int}} \\ e_{o_x} \end{bmatrix} \\ &+\begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix}\left[\mathrm{sat}(u_m)+u_h\right] + \begin{bmatrix} 0 \\ 0 \\ B\left[u_h-\hat{u}_h\right] \end{bmatrix} + \begin{bmatrix} A(m,\mathrm{K})x_{ref} - E(m)x_{ref}{}^+ \\ 0 \\ \left(-E(m)+E\right)x_{ref}{}^+ + \left(A(m,\mathrm{K})-A\right)x_{ref} \end{bmatrix} \end{aligned} \tag{3.5.10}$$

which is equivalent to:

$$\begin{aligned} &\begin{bmatrix} -E(m) & 0 & E(m) \\ 0 & I & 0 \\ -E(m)+E & 0 & E(m) \end{bmatrix}\begin{bmatrix} e_c \\ e_{\text{int}}{}^+ \\ e_{o_x}{}^+ \end{bmatrix} = \begin{bmatrix} -A(m,\mathrm{K}) & 0 & A(m,\mathrm{K}) \\ I & I & 0 \\ -A(m,\mathrm{K})+A & 0 & A(m,\mathrm{K})-K_xC \end{bmatrix}\begin{bmatrix} e_c \\ e_{\text{int}} \\ e_{o_x} \end{bmatrix} \\ &+\begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix}\mathrm{sat}(u_m) + \begin{bmatrix} B & 0 & A(m,\mathrm{K}) & -E(m) \\ 0 & 0 & 0 & 0 \\ B & -B & A(m,\mathrm{K})-A & -E(m)+E \end{bmatrix}\begin{bmatrix} u_h \\ \hat{u}_h \\ x_{ref} \\ x_{ref}{}^+ \end{bmatrix} \end{aligned} \tag{3.5.11}$$

We define the new signal w signals and the new state vector $\bar{e}_c$ as follows:

$$w = \begin{bmatrix} u_h \\ \hat{u}_h \\ x_{ref} \\ x_{ref}^{+} \end{bmatrix}, \bar{e}_c = \begin{bmatrix} e_c \\ e_{\text{int}} \\ e_{o_x} \end{bmatrix} \tag{3.5.12}$$

Then, the open loop system (3.5.11) is equivalent to:

$$\bar{E}_c(m)\bar{e}_c^{+} = \bar{A}_c(m,\mathrm{K})\bar{e}_c + \bar{B}_c \mathrm{sat}(u_m) + D_w(m,\mathrm{K})w \tag{3.5.13}$$

where the matrices are

$$\bar{E}_c(m) = \begin{bmatrix} -E(m) & 0 & E(m) \\ 0 & I & 0 \\ -E(m)+E & 0 & E(m) \end{bmatrix}, \bar{A}_c(m,\mathrm{K}) = \begin{bmatrix} -A(m,\mathrm{K}) & 0 & A(m,\mathrm{K}) \\ I & I & 0 \\ -A(m,\mathrm{K})+A & 0 & A(m,\mathrm{K})-K_x C \end{bmatrix},$$
$$\bar{B}_c = \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix}, D_w(m,\mathrm{K}) = \begin{bmatrix} B & 0 & A(m,\mathrm{K}) & -E(m) \\ 0 & 0 & 0 & 0 \\ B & -B & A(m,\mathrm{K})-A & -E(m)+E \end{bmatrix}.$$

In order to control the system (3.5.13), we propose the following controller:

$$u_m = \bar{L}_c \bar{M}_c^{-1} \bar{e}_c + \bar{L}_w w \tag{3.5.14}$$

with matrices $\bar{L}_c$ and $\bar{M}_c$ to be determined, and $\bar{L}_w = \begin{bmatrix} 0 & -I & L_{ref} & L_{ref^+} \end{bmatrix}$. As u_h is unknown and not directly measured, the first term of $\bar{L}_w$ is set to 0. Nevertheless, as the unknown input is supposed to be estimated via $\hat{u}_h$, the second term of $\bar{L}_w$ is fixed to $-I$, which acts like a disturbance-observer-based controller (Chen et al. 2016). The terms L_{ref} and L_{ref^+} provide a feedforward control.

The observer-based control closed loop system composed of (3.5.13) together with the controller (3.5.14) is:

$$\bar{E}_c(m)\bar{e}_c^{+} = \left(\bar{A}_c(m,\mathrm{K}) + \bar{B}_c\bar{L}_c\right)\bar{e}_c + \left(D_w(m,\mathrm{K}) + \bar{B}_c\bar{L}_w\right)w - \bar{B}_c\phi(u_m) \tag{3.5.15}$$

where the nonlinear dead-zone function $\phi(u_m)$ is defined as:

$$\phi(u_m) = u_m - \mathrm{sat}(u_m) \tag{3.5.16}$$

Combined with an anti-windup strategy, the integral term of the tracking error is:

$$e_{\text{int}}^{+} = e_{\text{int}} + e_c + L_a S^{-T} \phi(u_m) \tag{3.5.17}$$

The closed loop system (3.5.15) can be rewritten using a polytopic representation:

$$\begin{aligned} \sum_{i=1}^{2} \zeta_i(m) \bar{E}_{c_j} \bar{e}_c^{+} &= \left(\sum_{i=1}^{2} \sum_{j=1}^{2} \zeta_i(m) \vartheta_j(\mathrm{K}) \bar{A}_{c_{ij}} + \bar{B}_c \bar{L}_c \bar{M}_c^{-1} \right) \bar{e}_c \\ &+ \left(\sum_{i=1}^{2} \sum_{j=1}^{2} \zeta_i(m) \vartheta_j(\mathrm{K}) D_{w_{ij}} + \bar{B}_c \bar{L}_w \right) w + \bar{B}_a \phi(u_m) \end{aligned} \tag{3.5.18}$$

where the matrices are:

$$\bar{E}_{c_j} = \begin{bmatrix} -E_j & 0 & E_j \\ 0 & I & 0 \\ -E_j + E & 0 & E_j \end{bmatrix}, \bar{A}_{c_{ij}} = \begin{bmatrix} -A_{ij} & 0 & A_{ij} \\ I & I & 0 \\ -A_{ij} + A & 0 & A_{ij} - K_x C \end{bmatrix},$$

$$\bar{B}_a = \begin{bmatrix} -B \\ L_a S^{-T} \\ 0 \end{bmatrix}, D_{w_{ij}} = \begin{bmatrix} B & 0 & A_{ij} & -E_j \\ 0 & 0 & 0 & 0 \\ B & -B & A_{ij} - A & -E_j + E \end{bmatrix}.$$

The objective is to search for the control gains $\begin{bmatrix} \bar{L}_c \bar{M}_c^{-1} & \bar{L}_w \end{bmatrix}$ and the anti-windup parameters L_a such that the closed-loop PAW system (3.5.18) satisfies the criteria given thereafter.

3.5.2 Control objective

In order to ensure the stability of the closed-loop system shown by the red frame in Figure 3.5.1, to track a given velocity reference in presence of actuator saturations and system uncertainties, we distinguish two different cases:

- First case: When $w^T w = 0$, the vector $\bar{e}_c$ converges asymptotically to the origin.
- Second case: When $w^T w \neq 0$, under null initial conditions ($\bar{e}_c = 0$), the $\mathcal{L}_2$-norm of the tracking errors e_c and the state estimation error e_{0_x} are bounded as follows:

$$\sum_{k=0}^{\infty} \left(\bar{e}_c^{\,T} C_c^{\,T} C_c \bar{e}_c \right) < \gamma_c \sum_{k=0}^{\infty} w^T w \tag{3.5.19}$$

Remark 10. The matrix C_c can be configured to achieve a good compromise between the tracking and the estimation performances. In practice, the human torques u_h and the reference signals x_{ref} are bounded. If the estimated human torques $\hat{u}_h$ are bounded, the amplitude of the vector w is bounded. The velocity vector of the wheelchair is $x = x_{ref} - e_c + e_{o_x}$. When $x_{ref} = 0$ and the vector $\bar{e}_c$ converges asymptotically to the origin (the first case), the velocity x converges asymptotically to the origin. For the second case, the velocity follows the bounded reference value x_{ref} with the tracking errors e_c and the estimation errors e_{o_x} which are bounded by the condition (3.5.19). Consequently, the second case states implicitly that the velocity x remains bounded.

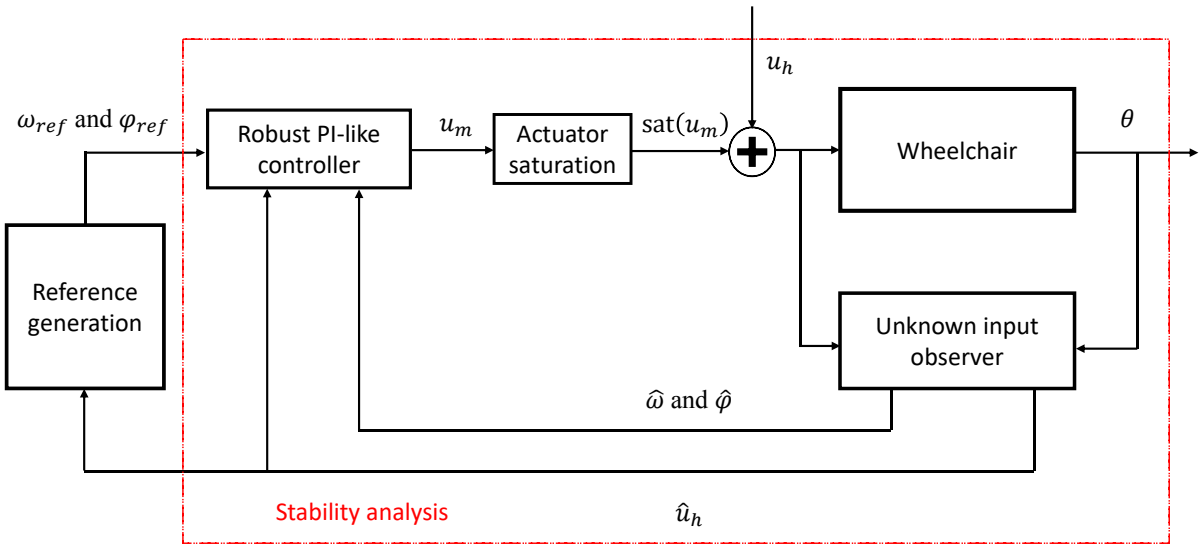

Figure 3.5.1. The closed-loop system with the observer-based tracking control under actuator saturations

3.5.3 Observer-based tracking control design

The dynamics (3.5.18) can be rewritten using the following equality constraint:

$$\begin{aligned}&\left(\sum_{i=1}^{2}\sum_{j=1}^{2}\zeta_i(m)\vartheta_j(\mathrm{K})\bar{A}_{c_{ij}} + \bar{B}_c\bar{L}_c\bar{M}_c^{-1}\right)\bar{e}_c - \sum_{i=1}^{2}\zeta_i(m)\bar{E}_{c_j}\bar{e}_c^{+} \\ &+\left(\sum_{i=1}^{2}\sum_{j=1}^{2}\zeta_i(m)\vartheta_j(\mathrm{K})D_{w_{ij}} + \bar{B}_c\bar{L}_w\right)w + \bar{B}_a\phi(u_m) = 0\end{aligned} \tag{3.5.20}$$

Consider the following non-quadratic Lyapunov function candidate:

$$V(\bar{e}_c) = \bar{e}_c{}^T \bar{P}_c \bar{e}_c = \bar{e}_c{}^T \sum_{i=1}^{2}\sum_{j=1}^{2} \zeta_i(m)\vartheta_j(\mathrm{K})\bar{P}_{c_{ij}} \bar{e}_c > 0 \tag{3.5.21}$$

with $\bar{P}_c = \bar{P}_c^T \in \mathrm{R}^{3n_x+2n_p}$, the Lyapunov function one step ahead writes:

$$V(\bar{e}_c^+) = \bar{e}_c^{+T} \bar{P}_c^+ \bar{e}_c^+ = \bar{e}_c^{+T} \sum_{j=1}^{2}\sum_{h=1}^{2} \zeta_i(m)\vartheta_h(\mathrm{K}^+)\bar{P}_{c_{ih}} \bar{e}_c^+ \tag{3.5.22}$$

Theorem 5. *If there exist positive definite matrices* $\bar{P}_{ij} \in \mathrm{R}^{3n_x+2n_p}$, *matrices* $\bar{L}_c \in \mathrm{R}^{n_u \times 3n_x}$, $\bar{L}_w \in \mathrm{R}^{n_u \times (2n_u+2n_x)}, \bar{L}_a \in \mathrm{R}^{n_u}$, $\overline{M}_c \in \mathrm{R}^{2n_p}$, *a positive diagonal matrix* $S \in \mathrm{R}^{n_u}$ *and a positive scalar* γ_c *such that for* $j \in \{1,2\}$, $i \in \{1,2\}$, $h \in \{1,2\}$:

$$\overline{\Pi}_{ijh}^1 + \overline{\Pi}_{ij}^2 + \overline{\Pi}_{ij}^{2T} < 0 \tag{3.5.23}$$

where

$$\overline{\Pi}_{ij}^2 = \begin{bmatrix} \grave{o}\left(\bar{A}_{c_{ij}}\bar{M}_c + \bar{B}_c\bar{L}_c\right) & 0 & -\grave{o}\bar{E}_{c_i}\bar{M}_c & \grave{o}\bar{B}_a S^T & \grave{o}\left(D_{w_{ij}} + \bar{B}_c\bar{L}_w\right) \\ 0 & 0 & 0 & 0 & 0 \\ \bar{A}_{c_{ij}}\bar{M}_c + \bar{B}_c\bar{L}_c & 0 & -\bar{E}_{c_i}\bar{M}_c & \bar{B}_a S^T & D_{w_{ij}} + \bar{B}_c\bar{L}_w \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\overline{\Pi}_{ijh}^1 = \begin{bmatrix} -\bar{M}_c{}^T\bar{P}_{c_{ij}}\bar{M}_c & * & * & * & * \\ C_c\bar{M}_c & -I & * & * & * \\ 0 & 0 & \bar{M}_c{}^T\bar{P}_{c_{ih}}\bar{M}_c & * & * \\ \bar{L}_c & 0 & 0 & -2S & * \\ 0 & 0 & 0 & \bar{L}_w{}^T & -\gamma_c I \end{bmatrix}, \bar{B}_a S^T = \begin{bmatrix} -B_d S^T \\ L_a \\ 0 \end{bmatrix}.$$

then, the observer-based controller (3.5.14) achieves the control objective defined in Section 3.5.2.

Proof*:* The inequality (3.5.23) can be rewritten as follows:

$$\overline{\Pi}_{\zeta\vartheta\vartheta^+}^1 + \begin{bmatrix} \grave{o}I \\ 0 \\ I \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} \bar{A}_{c_{\zeta\vartheta}}\bar{M}_c + \bar{B}_c\bar{L}_c & 0 & -\bar{E}_{c_\zeta}\bar{M}_c & \bar{B}_a S^T & D_{w_{\zeta\vartheta}} + \bar{B}_c\bar{L}_w \end{bmatrix} + (*) < 0 \tag{3.5.24}$$

with the notation:

$$\overline{\Pi}^{1}_{\zeta\vartheta\vartheta^{+}} = \sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{h=1}^{2}\zeta_i(m)\vartheta_j(\mathrm{K})\vartheta_h(\mathrm{K}^{+})\overline{\Pi}^{1}_{ijh},\ \bar{A}_{c_{\zeta\vartheta}} = \sum_{i=1}^{2}\sum_{j=1}^{2}\zeta_i(m)\vartheta_j(\mathrm{K})\bar{A}_{c_{ij}},$$

$$\bar{E}_{c_\zeta} = \sum_{j=1}^{2}\zeta_i(m)\bar{E}_{c_i},\ D_{w_{\zeta\vartheta}} = \sum_{i=1}^{2}\sum_{j=1}^{2}\zeta_i(m)\vartheta_j(\mathrm{K})D_{w_{ij}}.$$

Using the property of congruence with $diag\left(\bar{M}_c^{-T} \quad I \quad \bar{M}_c^{-T} \quad S^{-1} \quad I\right)$, the inequality (3.5.24) is equivalent to:

$$\begin{bmatrix} -\bar{P}_{c_{\zeta\vartheta}} & * & * & * & * \\ C_c & -I & * & * & * \\ 0 & 0 & \bar{P}_{c_{\zeta\vartheta^{+}}} & * & * \\ S^{-1}\bar{L}_c\bar{M}_c^{-1} & 0 & 0 & -2S^{-1} & * \\ 0 & 0 & 0 & \bar{L}_w^{T}S^{-T} & -\gamma_c I \end{bmatrix} + \begin{bmatrix} \grave{o}\bar{M}_c^{-T} \\ 0 \\ \bar{M}_c^{-T} \\ 0 \\ 0 \end{bmatrix}\left[\bar{A}_{c_{\zeta\vartheta}}\bar{M}_c + \bar{B}_c\bar{L}_c \quad 0 \quad -\bar{E}_{c_\zeta}\bar{M}_c \quad \bar{B}_a \quad D_{w_{\zeta\vartheta}} + \bar{B}_c\bar{L}_w\right] + (*) < 0 \tag{3.5.25}$$

Using Schur complement, we obtain:

$$\begin{bmatrix} -\bar{P}_{c_{\zeta\vartheta}} + C_c^{T}C_c & * & * & * \\ 0 & \bar{P}_{c_{\zeta\vartheta^{+}}} & * & * \\ S^{-1}\bar{L}_c\bar{M}_c^{-1} & 0 & -2S^{-1} & * \\ 0 & 0 & \bar{L}_w^{T}S^{-T} & -\gamma_c I \end{bmatrix} + \begin{bmatrix} \grave{o}\bar{M}_c^{-T} \\ \bar{M}_c^{-T} \\ 0 \\ 0 \end{bmatrix}\left[\bar{A}_{c_{\zeta\vartheta}}\bar{M}_c + \bar{B}_c\bar{L}_c \quad -\bar{E}_{c_\zeta}\bar{M}_c \quad \bar{B}_a \quad D_{w_{\zeta\vartheta}} + \bar{B}_c\bar{L}_w\right] + (*) < 0 \tag{3.5.26}$$

Using Lemma 3 and the slack matrix $\left[\grave{o}\bar{M}_c^{-T} \quad \bar{M}_c^{-T} \quad 0 \quad 0\right]^{T}$ for Finsler's lemma, the inequality (3.5.25) with the equality constraint (3.5.20) is equivalent to:

$$\begin{bmatrix} \bar{e}_c \\ \bar{e}_c^{+} \\ \phi(u_m) \\ w \end{bmatrix}^T \begin{bmatrix} -\bar{P}_{c_{\zeta\vartheta}} + C_c{}^T C_c & * & * & * \\ 0 & \bar{P}_{c_{\zeta\vartheta^+}} & * & * \\ S^{-1}\bar{L}_c\bar{M}_c{}^{-1} & 0 & -2S^{-1} & * \\ 0 & 0 & \bar{L}_w{}^T S^{-T} & -\gamma_c I \end{bmatrix} \begin{bmatrix} \bar{e}_c \\ \bar{e}_c^{+} \\ \phi(u_m) \\ w \end{bmatrix} < 0 \qquad (3.5.27)$$

From (3.5.27) and the controller (3.5.14), we deduce that:

$$\Delta V(\bar{e}_c) + \bar{e}_c{}^T C_c{}^T C_c \bar{e}_c - \gamma_c w^T w + \left[u_m{}^T S^{-1}\phi(u_m) + \phi^T(u_m) S^{-1} u_m - 2\phi^T(u_m) S^{-1}\phi(u_m) \right] < 0 \qquad (3.5.28)$$

where the Lyapunov function $V(\bar{e}_c)$ is defined in (3.5.21). The following two cases can be analysed:

- First case: if the external signals $w = 0$, the following condition can be deduced:

$$\Delta V(\bar{e}_c) = -\bar{e}_c{}^T C_c{}^T C_c \bar{e}_c - \left[u_m{}^T S^{-1}\phi(u_m) + \phi^T(u_m) S^{-1} u_m - 2\phi^T(u_m) S^{-1}\phi(u_m) \right] < 0 \qquad (3.5.29)$$

  which means that the tracking errors converge exponentially to the origin.

- Second case: If $w \neq 0$, the conditions (3.5.2) and (3.5.28) imply that:

$$\Delta V(\bar{e}_c) + \bar{e}_c{}^T C_c{}^T C_c \bar{e}_c - \gamma_c w^T w < 0 \qquad (3.5.30)$$

  Under null initial conditions ($\bar{e}_c = 0$) and the integration of the inequality (3.5.30), we obtain:

$$\sum_{k=0}^{\infty} \left[\bar{e}_c{}^T C_c{}^T C_c \bar{e}_c - \gamma_c w^T w \right] < 0 \qquad (3.5.31)$$

  Then, the inequality (3.5.19) can be derived. Moreover, this implies the following criterion:

$$\| C_c \bar{e}_c \|_2 < \sqrt{\gamma_c}\, \| w \|_2 \qquad (3.5.32)$$

The proof of Theorem 5 is complete. ■

3.5.4 Simulation results

Thereafter, we validate the robust observer-based tracking controller derived from Theorem 5 using some numerical simulations. To carry out these simulations, we keep the nominal parameters of Table I. A second degree derivative is used to approximate the unknown inputs. The mass can take a value between $\underline{m} = 70kg$ and $\bar{m} = 130kg$, and the viscous friction coefficient can vary between $\underline{\mathrm{K}} = 3N.m.s$ and $\overline{\mathrm{K}} = 7N.m.s$. These given intervals would include most cases in practice. When $w \neq 0$, we choose sinusoidal signals for the human pushing profile. The mass of the user, which may be different from the nominal value, is constant during a driving task. However, the viscous friction coefficient may be time-varying in the given interval during a driving task. The maximum motor torque $u_{\max}$ is $30Nm$ for both electrical motors. We choose the matrix C_c as follows:

$$C_c = \begin{bmatrix} I_{n_x} & 0_{n_x} & 0_{n_x} \\ 0_{n_x} & 0_{n_x} & I_{n_x} \end{bmatrix}$$

Remark 11. The yaw velocity tracking is more important than the center velocity tracking, since users regulate the direction of the wheelchair by achieving a desired yaw velocity. In order to reduce the yaw velocity tracking error, its weight in matrix C_c can be increased. The weight matrix C_c acts here similarly to the weighting matrix in Linear-Quadratic Regulator designs. It can be configured such that the observer-based control (3.5.14) ensures first the yaw velocity tracking when actuator saturations occur. Moreover, no constraint on the integral state is needed. Therefore, the terms corresponding to the integral state are set to zero.

The observer gains are derived using the nominal parameters for Section 3.2. The control parameters obtained from Theorem 5 are:

$$\begin{aligned} \bar{L}_c \bar{M}_c^{-1} &= \begin{bmatrix} 170.21 & 661.19 & 128.72 & 176.83 & -26.27 & -519.93 \\ 256.33 & -664.89 & 140.26 & -144.95 & -132.26 & 585.82 \end{bmatrix}, \\ \bar{L}_w &= \begin{bmatrix} 0 & 0 & -1 & 0 & -10.33 & -230.3 & 17.24 & 238.26 \\ 0 & 0 & 0 & -1 & -166.86 & 808.01 & 170.55 & -818.41 \end{bmatrix}, \\ L_a S^{-T} &= \begin{bmatrix} -0.0063 & -0.0061 \\ -0.0014 & 0.0004 \end{bmatrix}, \sqrt{\gamma_c} = 2716. \end{aligned} \tag{3.5.33}$$

To illustrate the robustness of the proposed observer-based controller, the mass and the viscous friction coefficient are taken different from the nominal values in Table I. the two cases stated in Section 3.5.2 are given.

- **Simulation results ($w = 0$)**

In the simulations presented hereafter, the external signal is $w = 0$. The mass is set to $m = 130kg$ and the viscous friction coefficient to $\mathrm{K} = 6.5N.m.s$. The initial velocities of the center and the yaw are 0.16m/s and -0.55rad/s respectively. Notice that the velocities converge to the reference values in Figure 3.5.2 right. In addition, the saturation occurs in the beginning in Figure 3.5.2 left.

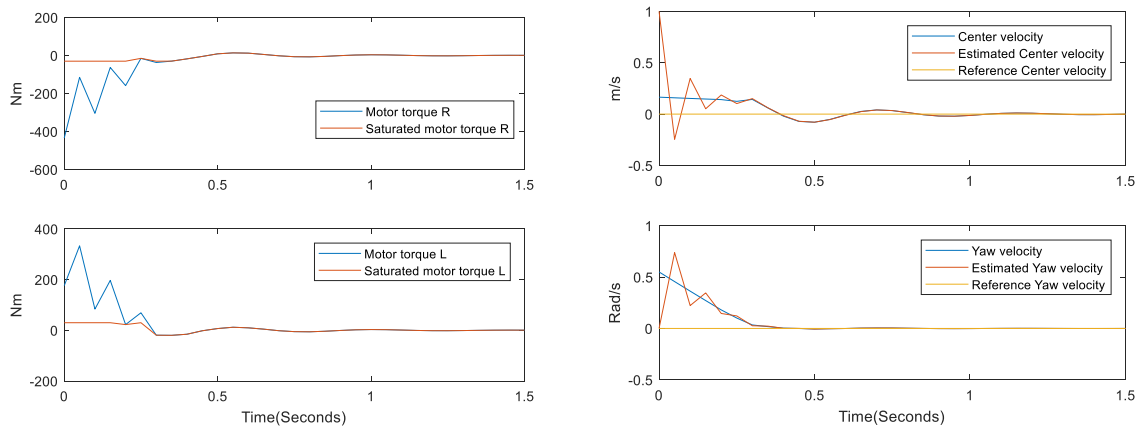

Figure 3.5.2. Assistive motor torques under actuator saturations (Left), Reference velocity and Velocity of the wheelchair (Right) when $w = 0$

- **Simulation results ($w \neq 0$)**

This part illustrates the behavior of the controller in presence of actuator saturations. The nominal parameters, mass and viscous friction coefficient have the same values as the previous case. The initial value of the error vector $\bar{e}_c$ is zero.

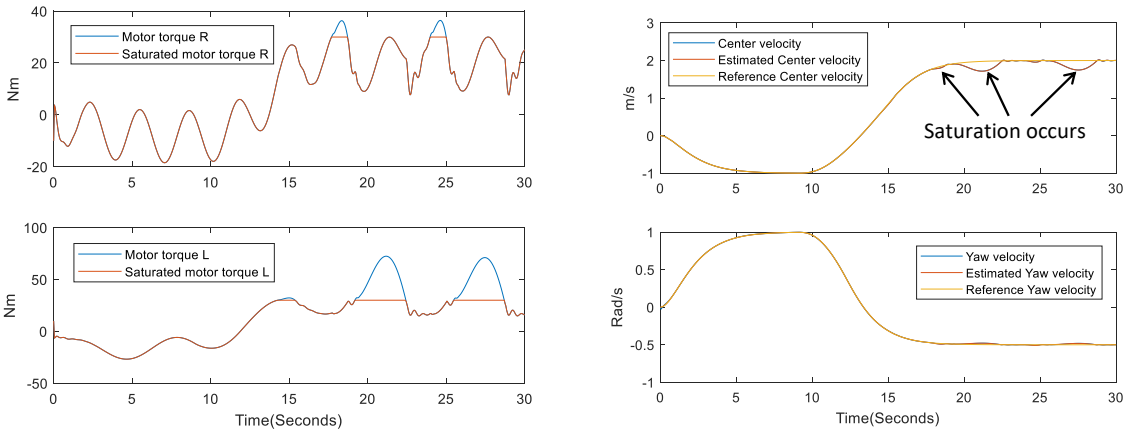

Figure 3.5.3. Assistive motor torques under actuator saturations (Left), Reference velocity and Velocity of the wheelchair (Right) when $w \neq 0$

As shown in Figure 3.5.3, the proposed control law tracks well the given yaw velocity in the presence of uncertainties in the system parameters, of unknown inputs, and of actuator saturations. This can be verified in Figure 3.5.3 right. However, the center velocity tracking performance is degraded when the electrical motors achieve their maximum capacity. Indeed, the most important objective is to help turning. To keep enough available motor torques for the yaw velocity tracking, the controller slows down automatically the center velocity when saturations occur. This driving characteristic has been set up according to the matrix C_c.

An opposite result is provided in Figure 3.5.4, where the matrix C_c is chosen as follows:

$$C_c = \begin{bmatrix} \begin{bmatrix} 20 & 0 \\ 0 & 1 \end{bmatrix} & 0_{n_x} & 0_{n_x} \\ 0_{n_x} & 0_{n_x} & I_{n_x} \end{bmatrix}$$

where the weight for the center velocity tracking is higher than the one for the yaw velocity tracking. As shown in Figure 3.5.4, the controller preferentially tracks the center velocity during actuator saturations. However, the performance of the yaw velocity tracking is degraded. This scenario is undesirable as users must regulate the direction of the wheelchair.

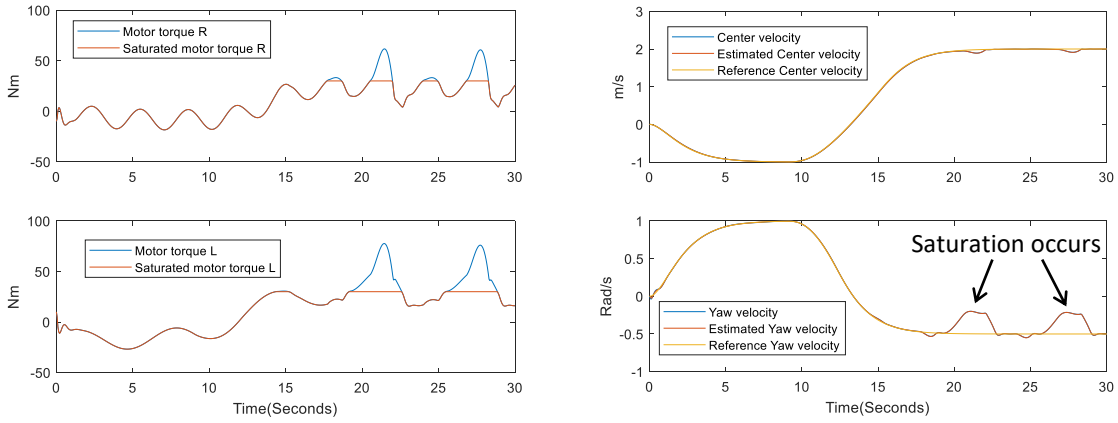

Figure 3.5.4. Center velocity tracking preference: Assistive motor torques under actuator saturations (Left), Reference velocity and Velocity of the wheelchair (Right) when $w \neq 0$

Note that the estimated unknown inputs do not converge completely to the human torques in Figure 3.5.5. As stated in Remark 8, using the approximation technique (3.2.1), the estimated information lumps together the human torques and the non-modelled dynamics due to uncertainties. As depicted in Figure 3.5.4 and Figure 3.5.5, the amplitude of vector w is bounded. This condition implies the stability of the human-wheelchair system. Nevertheless, the estimation without being perfect is able to capture the principal features of the torques and especially the frequency of pushing, which is then used for the reference generation described in Section 3.3.3.

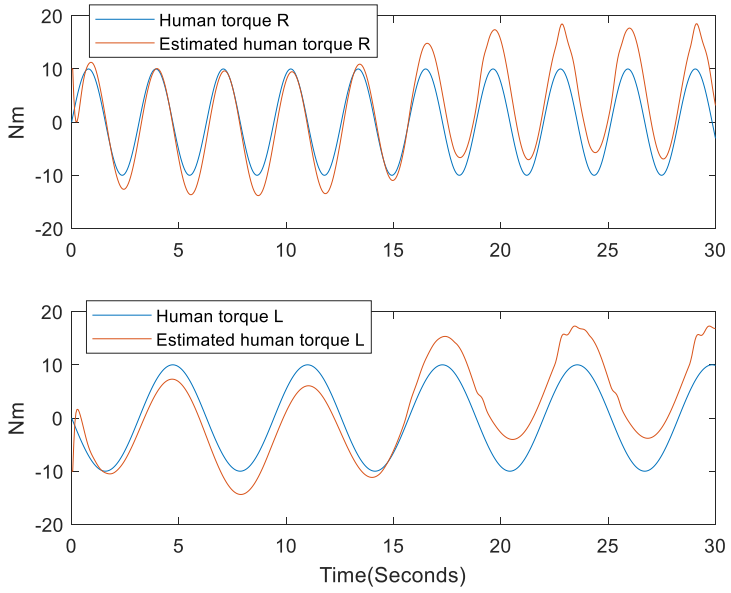

Figure 3.5.5. Reel human torques and estimated human torques

3.6 Conclusion

In this chapter, an unknown input PI-observer was designed to estimate human torques. This human torque estimation was used to generate the reference velocities. For tracking the reference signals generated, a robust observer-based controller under actuator saturations was proposed and validated by simulations for the PAW application. The stability feature of the PAW under the proposed controller has been proven.

Based on the simulation results obtained in this chapter, the next chapter provides experimental validations for the whole assistive control algorithm.

Chapter 4. Experimental validation of model-based approach

In the previous chapter, the simulation results have illustrated the effectiveness of the proposed model-based assistive control algorithms. This chapter is devoted to the experimental tests for validating the assistive algorithm. Since the acquisition card of the prototype does not support a time-varying sampling, the experimental validations in this chapter are based on a constant sampling rate. Several tests are conducted in order to evaluate these three functionalities. The first tests are made to evaluate the capabilities of the PI-observer of Section 3.2. Then, the objective of the second tests is to confirm the performance of the robust observer-based controller presented in Section 3.5. Final tests are devoted to validate the whole assistive algorithm including the reference generation. The prototype presented in Section 2.2.1 is used. The data acquisition is done by a laptop connected to the control module via a USB cable. The tests have been carried out on the athletics tracks of the university stadium and the parking in front of the Autonomad Mobility company. The nominal values of the parameters are given in Table I.

4.1 Unknown input observer validation

First of all, we evaluate the performance of the proposed observer. We have carried out two trials with two different frequencies of human inputs. During these tests, the wheelchair is only driven by users as a manual wheelchair. Using the angular velocity of each wheel, the observer simultaneously estimates the center velocity, the yaw velocity of the wheelchair, and the human torques. To verify the performance of the observer, we compare the estimated human inputs with the real human inputs measured by the two torque sensors. Recall, as stated by remark Remark 1 of Section 2.2.1 that these torques sensors are not available for the Duo kit, they are placed on the prototype in order to be able to validate the proposed approaches.

Figure 4.1.1 shows that the observer estimates the human torques well enough to reconstruct the key features of the user's propelling i.e. pushing frequency, braking, and turning even if the measurements are noisy. As expected, when the speed is around zero the poor quality of

the position encoders does not allow a satisfactory estimation. Effectively, in this case, the number of teeth detected using a constant sampling time is considerately reduced and these measurement error directly influences the torque estimation. Nevertheless, using a sampling in the angle domain could significantly reduce this kind of error as shown in (Losero et al. 2018). Remember also that the model does not take into account non-modeled dynamics (i.e. caster dynamics and roads conditions) and does not include modeling errors that also explains the difference between the measured torque and the estimated torque. At last, recall that due to these limitations, it was not expected to reconstruct perfectly the user's torques, especially in amplitude. Nevertheless, the main goal is reached and the estimation is sufficiently good to proceed to the full control strategy.

Remark 12. As expected, the delay for the torques estimation in Figure 4.1.1 (zoom-in of the delay) is nearly equal to twice the sampling period, namely 0.1 seconds.

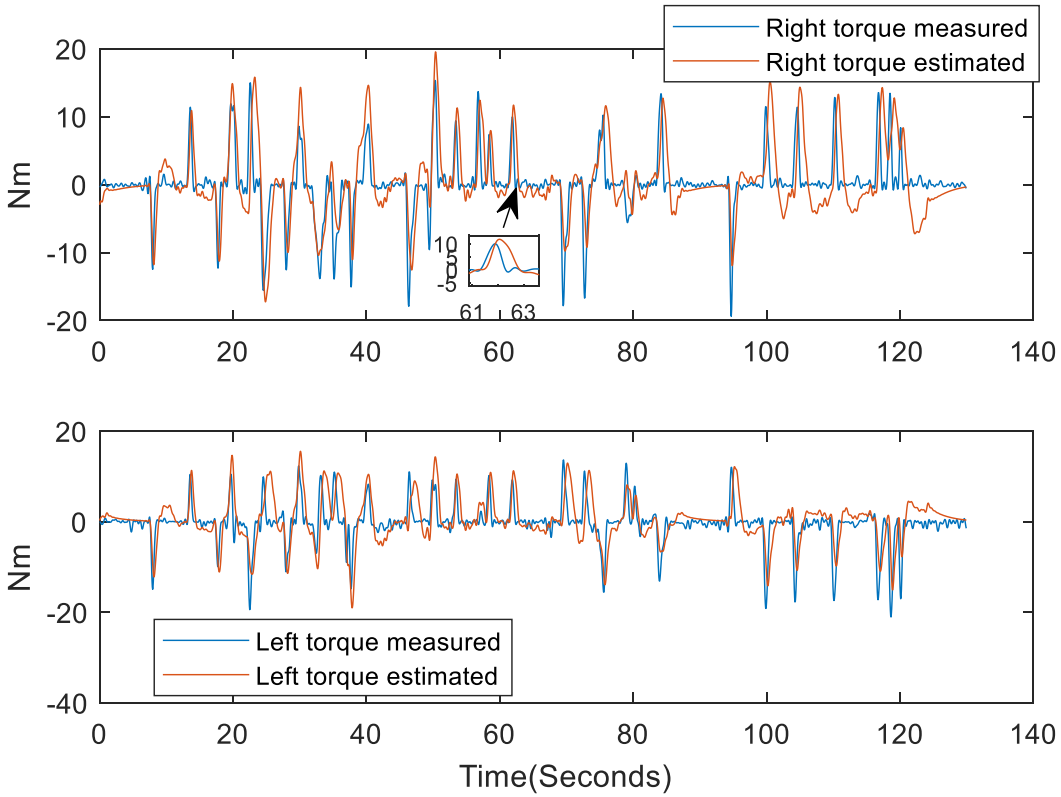

Figure 4.1.1. Human torque estimation (First trial without assistive torques)

The measured angular velocity of each wheel is given in Figure 4.1.2. As well as the center and the yaw velocities are simply obtained via an algebraic transformation.

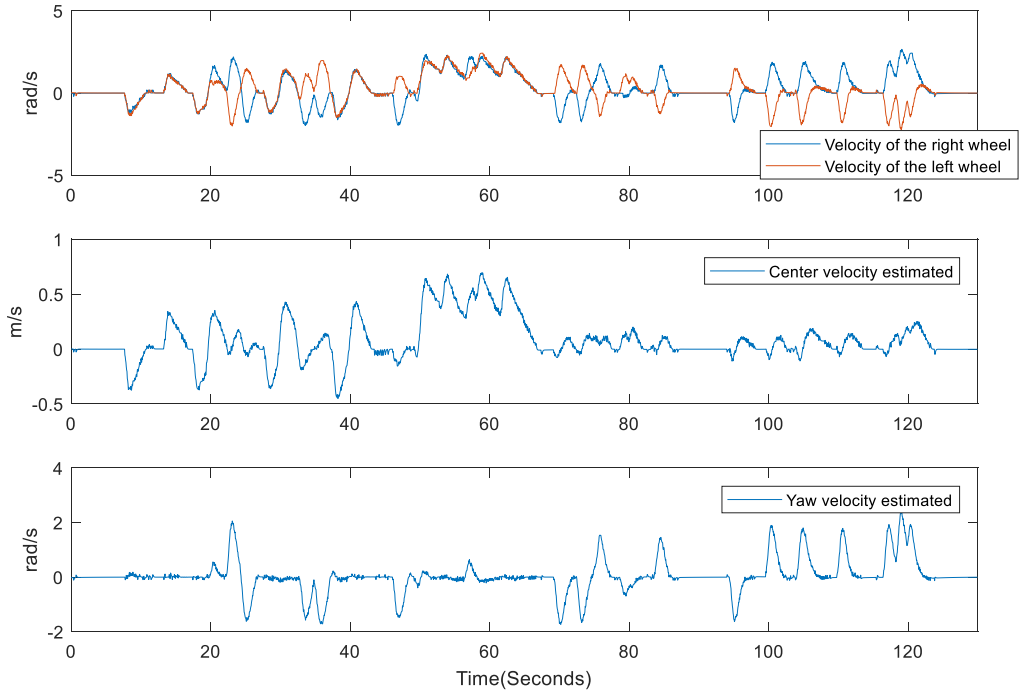

Figure 4.1.2. The measured angular velocity of each wheel, the estimated center and yaw velocities (First trial without assistive torques)

For the second trial, the frequency of the human pushing is increased to near 0.65 Hz. As depicted in Figure 4.1.3, similar performances as the first trial are observed. As previously said, increasing the frequency degrades the amplitude estimation, but since the detection is based on the pushing frequency and direction, these amplitude estimation errors will not influence considerably the estimation of the user's intention. Figure 4.1.4 provides the outputs and the estimated velocities of the wheelchair.

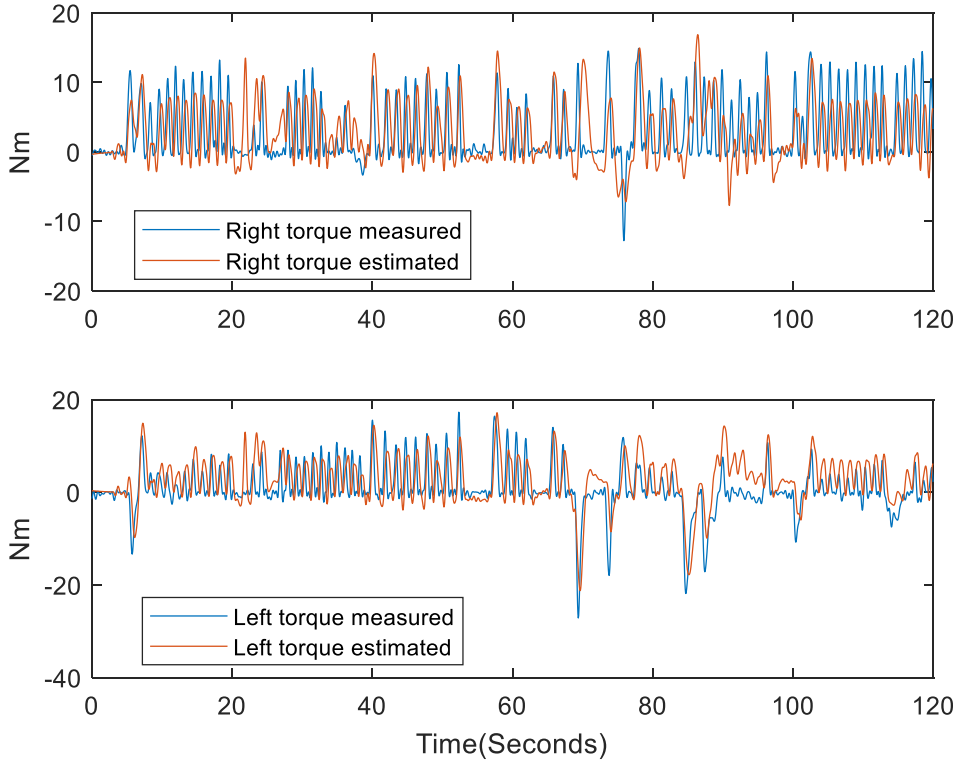

Figure 4.1.3. Human torque estimation (Second trial without assistive torques)

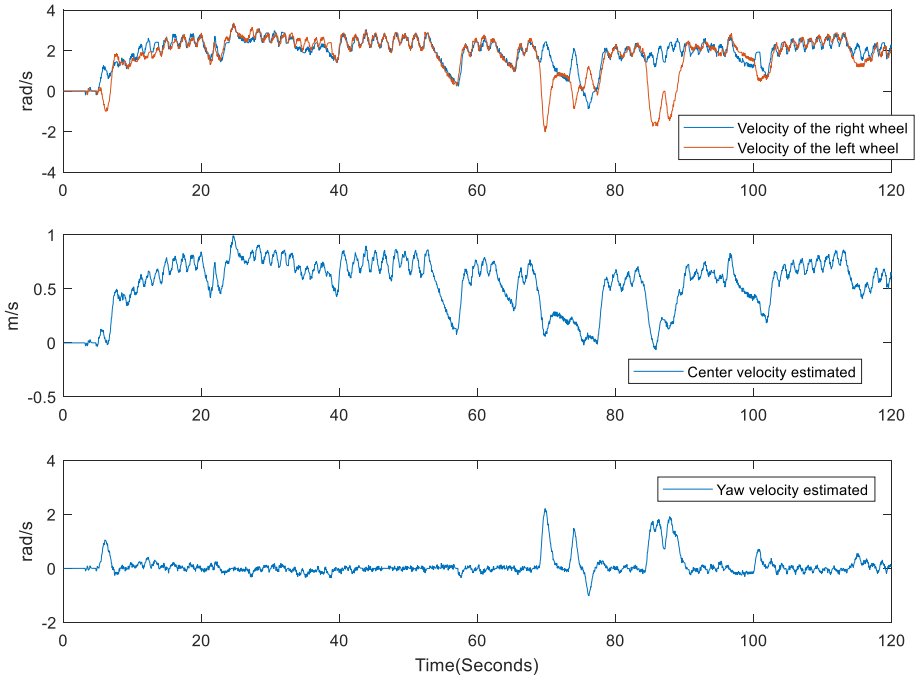

Figure 4.1.4. The measured angular velocity of each wheel, the estimated center velocity and the estimated yaw velocity (Second trial without assistive torques)

4.2 Trajectories tracking validation

Next, we report experimental results where the reference signals, the center velocity and the yaw velocity, are given by the reference generation algorithm presented in Section 3.3.3. The experience has been realized in the parking in front of Autonomad Mobility. Small stones are present on the ground. Moreover, the ground is not flat and some sections are rough (viscous friction coefficient changes).

4.2.1 Manual and assistance modes

To embed the full control, a detection mode must be incorporated. The switching conditions are depicted Figure 4.2.1.

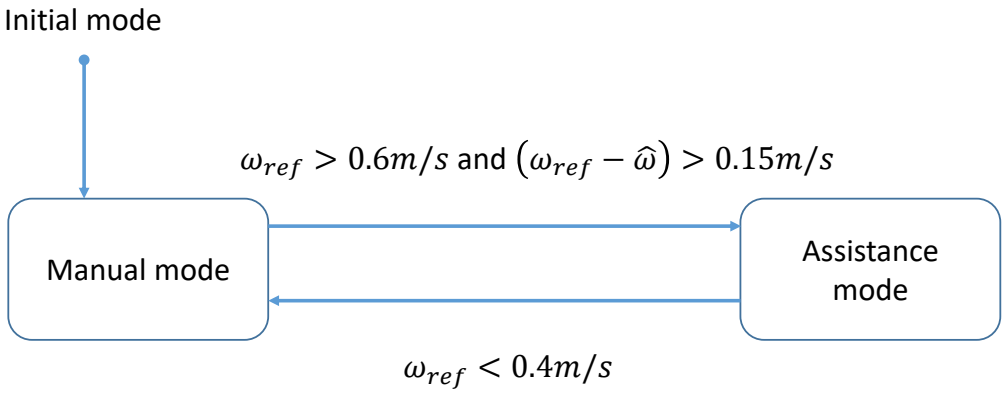

Figure 4.2.1. Manual mode and assistance mode

When the reference center velocity achieves a given threshold (0.6m/s), the wheelchair passes into the assistance mode. The condition $\omega_{ref} - \hat{\omega}(k) > 0.15\text{m/s}$ makes sure that the estimated center velocity is lower than the reference signal at the switching point from the manual mode to the assistance mode. Therefore, the motors do not brake abruptly and help users to accelerate. The wheelchair returns to the manual mode when the reference center velocity is below a 0.4m/s threshold. This value has to be different than the previous one (0.6m/s) in order to avoid unexpected switches (and a chattering effect) when the speed is close to the threshold.

4.2.2 Drivability and robustness tests

In order to test the feasibility of the proposed algorithms, especially robustness and performances, the trials presented in this section propose two different users with different mass and different ground conditions. The mass of user A is 63kg (the total mass including the wheelchair is 103kg) and the one of user B is 80 kg (total mass is 120kg). These values of mass are inside of the interval $[70,130]$, which is used for the robust observer-based tracking control design.

Figure 4.2.2 presents a $355s$ trial for user A, with his measured and estimated right and left torques. As already said, this estimation is crucial, since the reference signals are computed directly from two features of the signals e.g. frequency and direction.

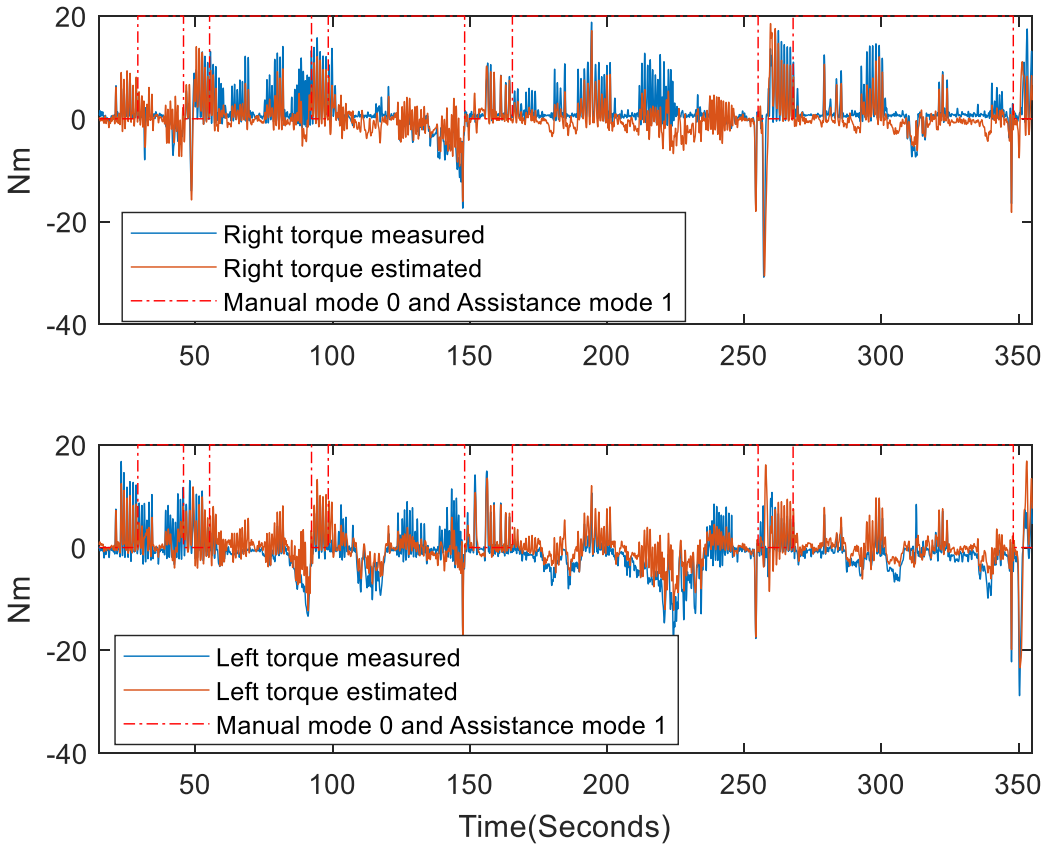

Figure 4.2.2. Human torque and estimated human torque of user A

The assistance algorithm switches automatically between the two modes depending on the need of the user. In the beginning, the reference center velocity is too low to activate the

assistance mode. After detecting that user A wishes to accelerate and the reference center velocity increases, the assistance mode is activated for helping him. When the user brakes to stop, the wheelchair switches from the assistance mode to the manual mode.

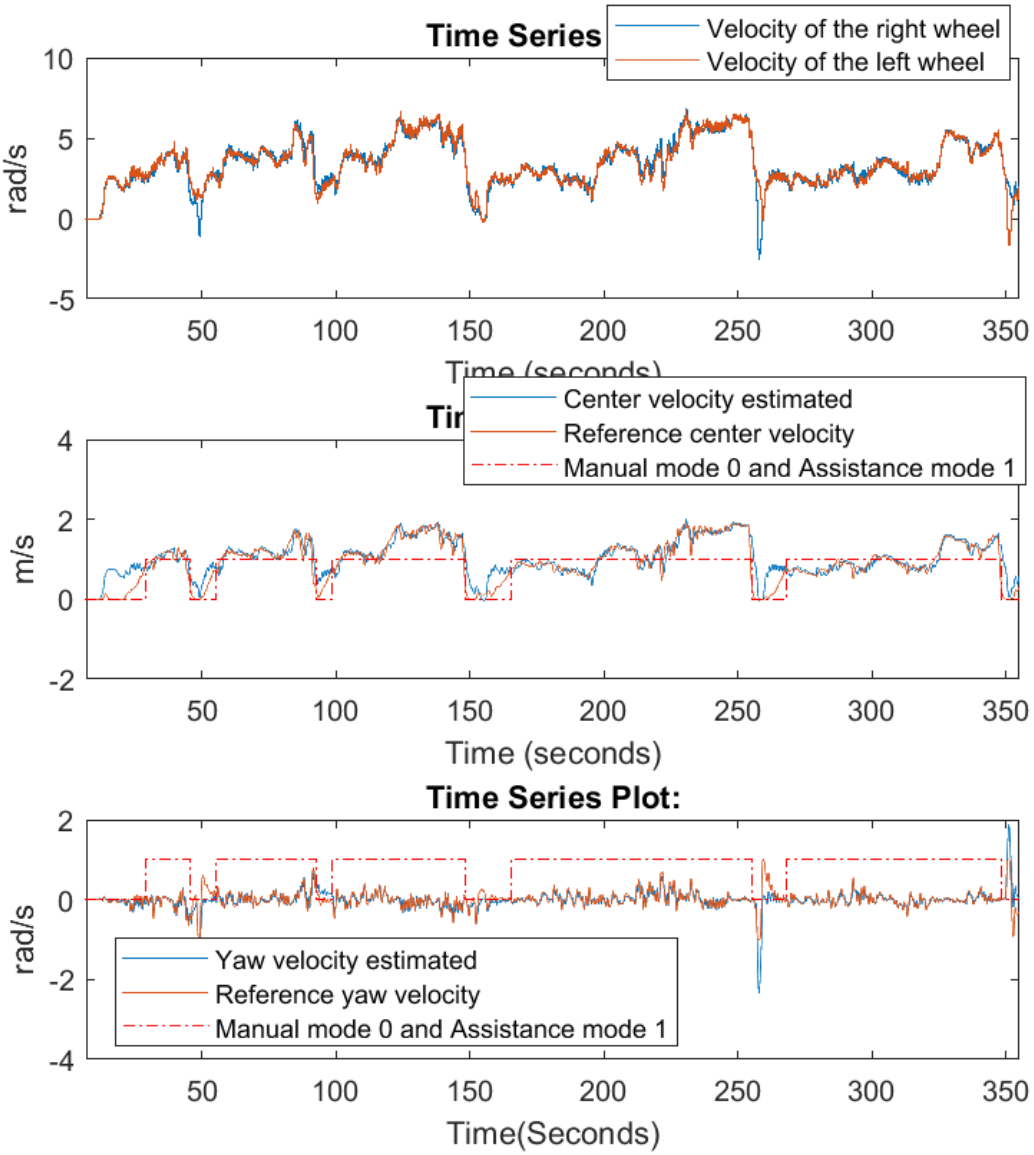

Figure 4.2.3. Velocity of each wheel, center velocity and yaw velocity of user A's trial

In the manual mode, the center velocity does not exceed 0.6m/s see Figure 4.2.3. One of the reasons could be that the user does not want to do too much physical exercise. When the system is in the assistance mode, the user easily achieves a higher center velocity. As shown in Figure 4.2.2, the user's torques are significantly smaller in the assistance mode than in the

manual mode. Thanks to the assistive system, the two electrical motors and the user track together the reference signals estimated from the user's pushing frequency. As shown in Figure 4.2.3, the proposed controller tracks well the reference signals produced by the algorithm, when the assistance mode is activated.

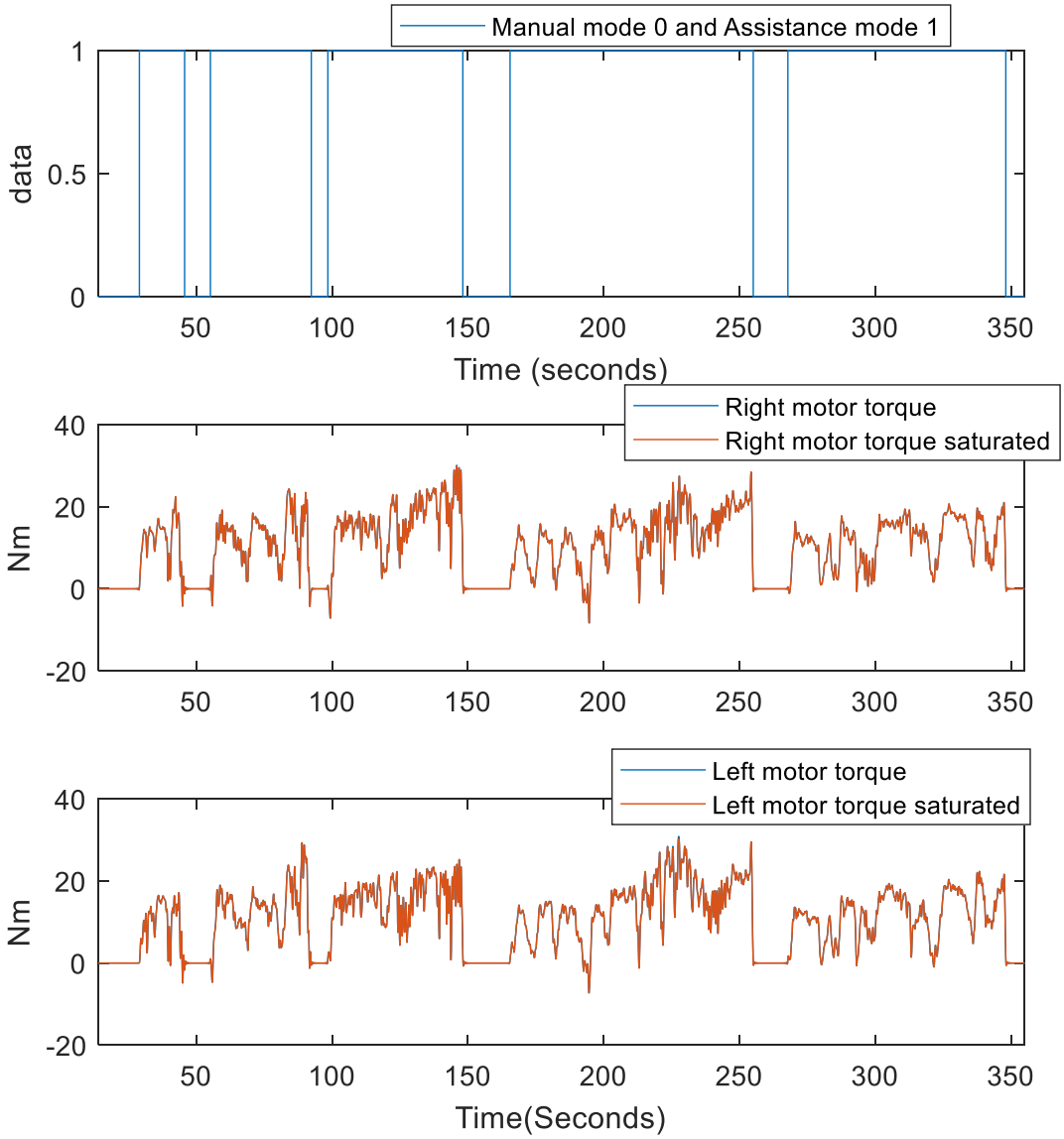

Figure 4.2.4. Mode of the wheelchair and assistive torque for user A

If the user pushes more, the assistive torques are reduced automatically for tracking the estimated reference signals. If the user feels tired and pushes less, the motors give naturally more assistive torque to accomplish the tracking task. As depicted in Figure 4.2.2, user A

reduces considerately his pushing at the end of the trial $t \in [300, 350] s$ and the motors generate the necessary assistive torques to ensure the reference tracking.

In addition, the proposed assistive algorithm is reactive enough for the user to manipulate easily the center velocity. At $200 s$, Figure 4.2.3, the $2 m/s$ center velocity is too high for the user to make a tight turn. Therefore, the user first slows down the wheelchair, second turns and lastly achieves quickly a desired velocity, around $1 m/s$ at $t = 260 s$, Figure 4.2.3.

Similar tests are done with user B, who is significantly heavier than user A. Figure 4.2.5 shows the results from the unknown input PI-observer estimation of the torques and almost similar behavior than user A.

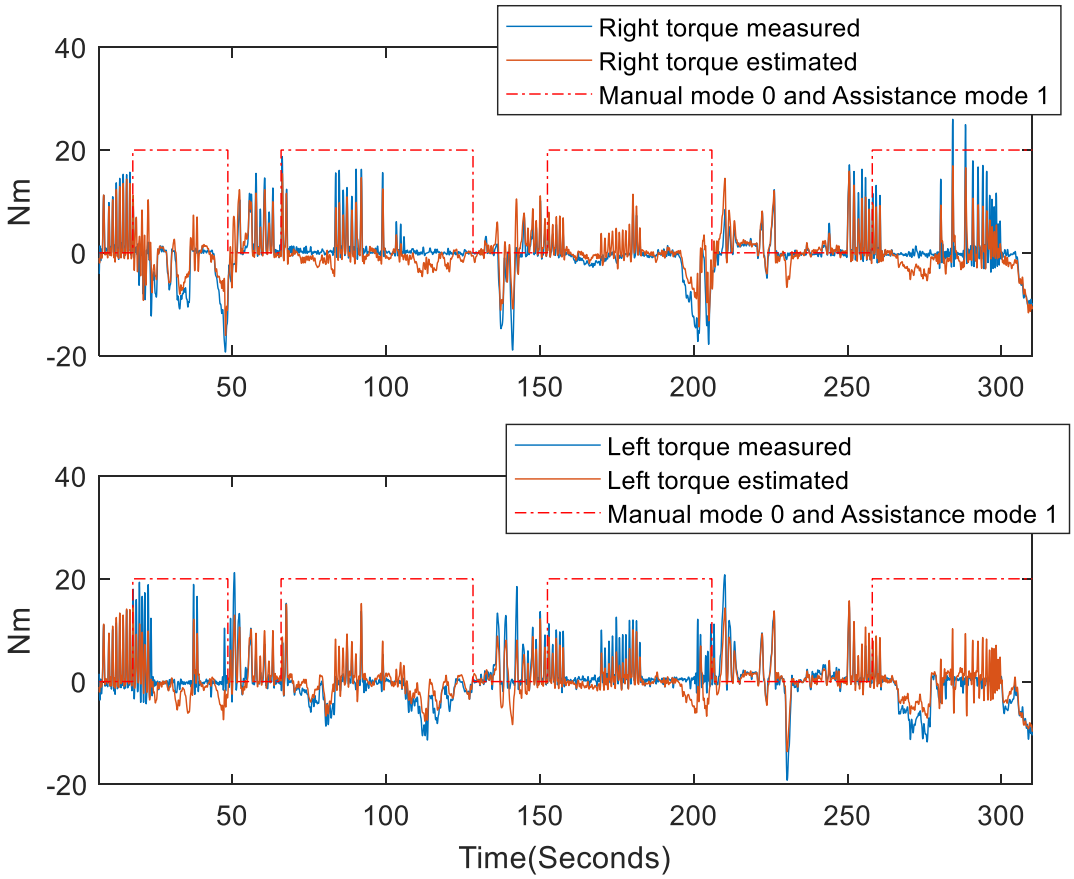

Figure 4.2.5. Human torque and estimated human torque of user B

Moreover, the user B being physically stronger than user A Figure 4.2.6 shows that the center velocity is already $1 m/s$ in the beginning of the trial in the manual mode. In this case, the reference center velocity is smaller than the center velocity see Figure 4.2.1 and the assistive

algorithm keeps the wheelchair in the manual mode. Once the user reduces his pushing effort, the center velocity decreases, when it goes below the reference center velocity, around $t = 20\,s$, Figure 4.2.6, the assistance is activated.

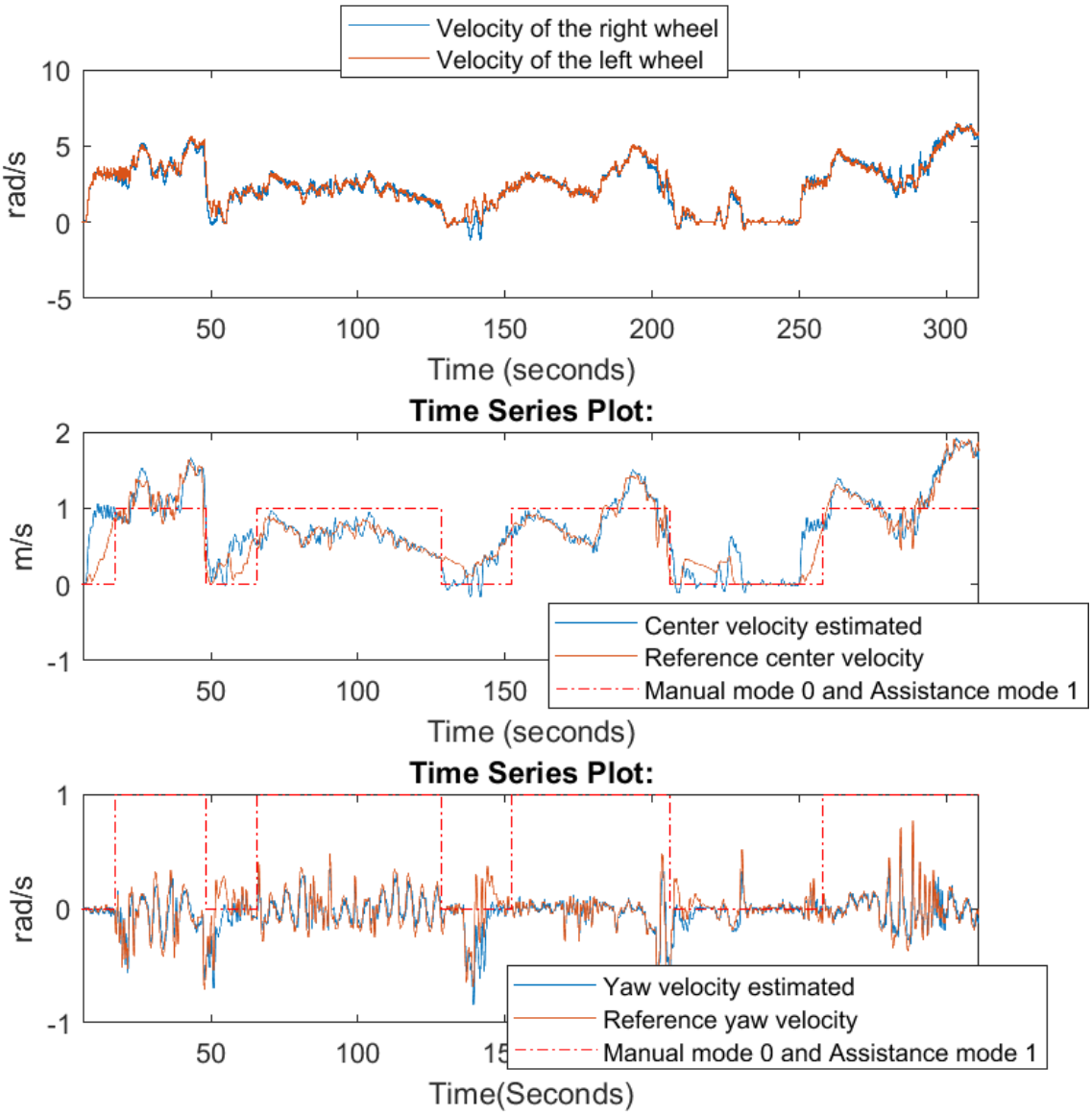

Figure 4.2.6. Velocity of each wheel, center velocity and yaw velocity of user B's trial

Between $180\,s$ and $200\,s$, the user almost stops pushing the wheelchair, Figure 4.2.5 and thanks to the assistance, the wheelchair still follows the reference signals, Figure 4.2.6 with the power coming principally from the electrical motors, Figure 4.2.7.

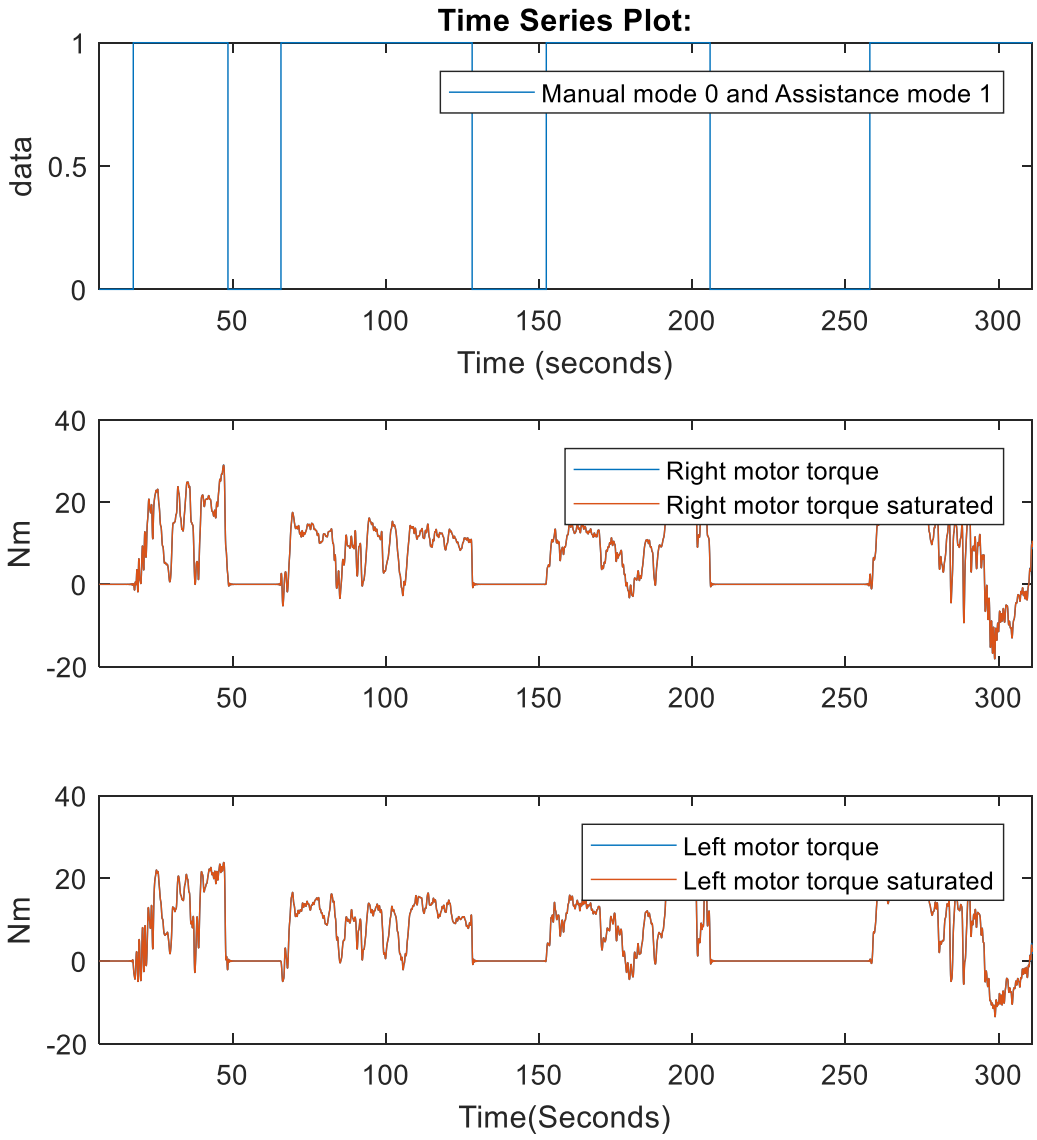

Figure 4.2.7. Mode of the wheelchair and assistive torque for user B

4.2.3 Predefined tasks

Some more tests were performed, among which some predefined tasks proposed to the users, in order to see the difficulties in maneuvering to perform them. For the first trial, user A has been asked to perform eight-shaped and oval trajectories on the parking of the Autonomad Mobility company. This trial includes different ground adhesion, obstacle avoidance and different (reasonable) slopes, Figure 4.2.8.

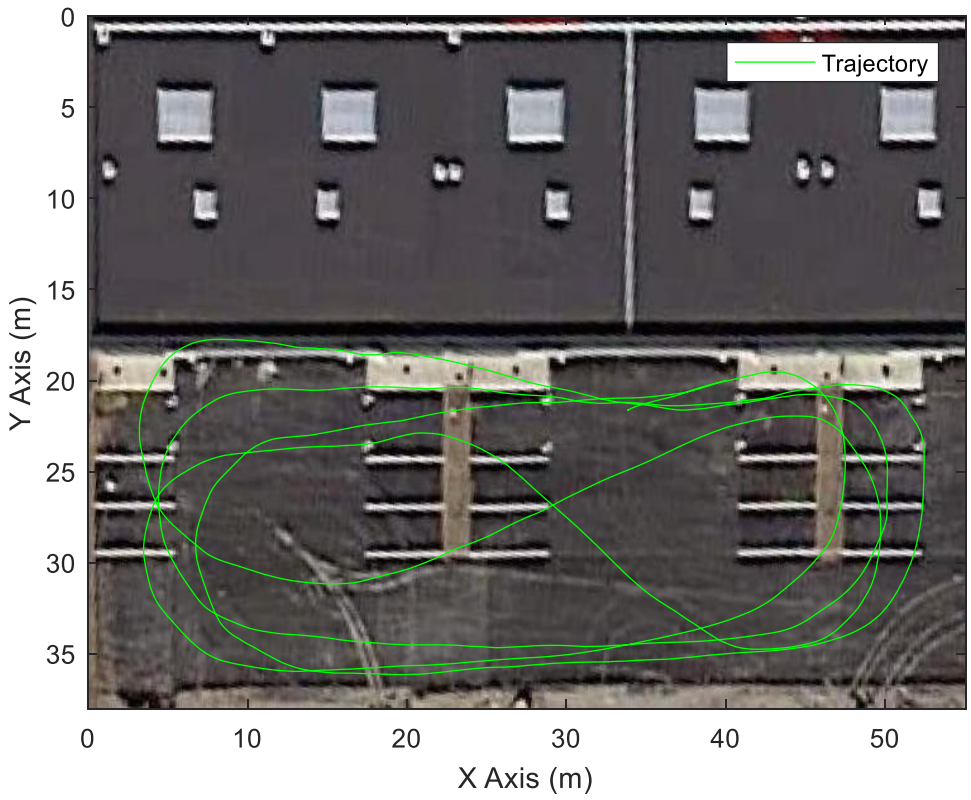

Figure 4.2.8. Two oval-shaped trajectories and one eight-shaped trajectory performed by user A under the assistive control

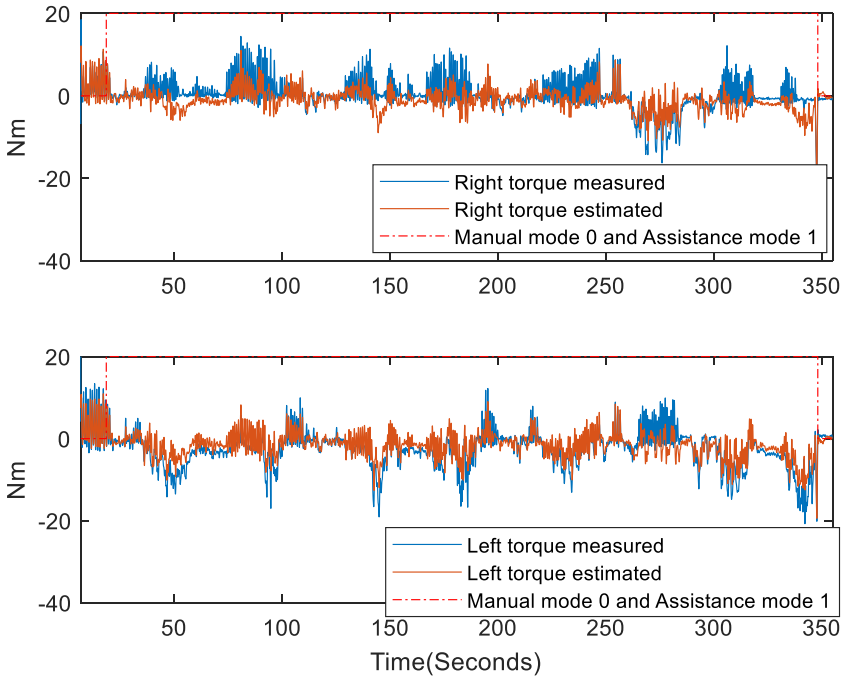

Figure 4.2.9. Human torque and estimated human torque of the trajectory tracking (User A)

The obtained trajectory is shown Figure 4.2.8 and user A was perfectly able to accomplish this complicated task with the help of the proposed assistive algorithm. The human torque measured by the sensors and the estimated human torques are given Figure 4.2.9, the assistance torquesFigure 4.2.10. On this figure, it can be seen that between $225s$ and $275s$, an actuator saturation occurs. Thanks to the anti-windup design in the controller (3.5.14) the saturated action is perfectly taken into account and we notice that the controller gives priority to the yaw velocity tracking, Figure 4.2.11. However, the electrical motors do not have enough power to ensure center velocity tracking during these periods.

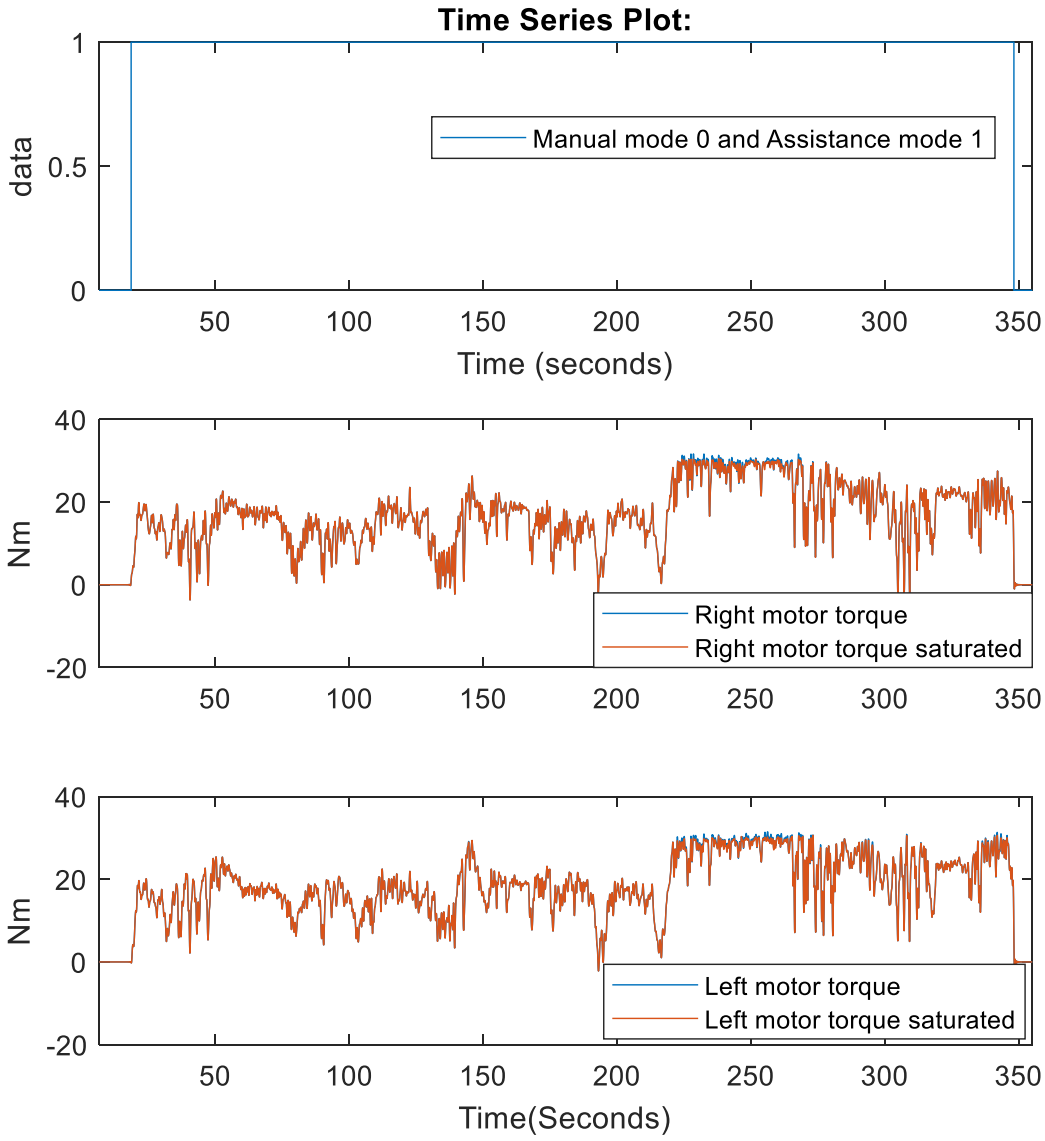

Figure 4.2.10. Mode of the wheelchair and assistive torque of the trajectory tracking (User A)

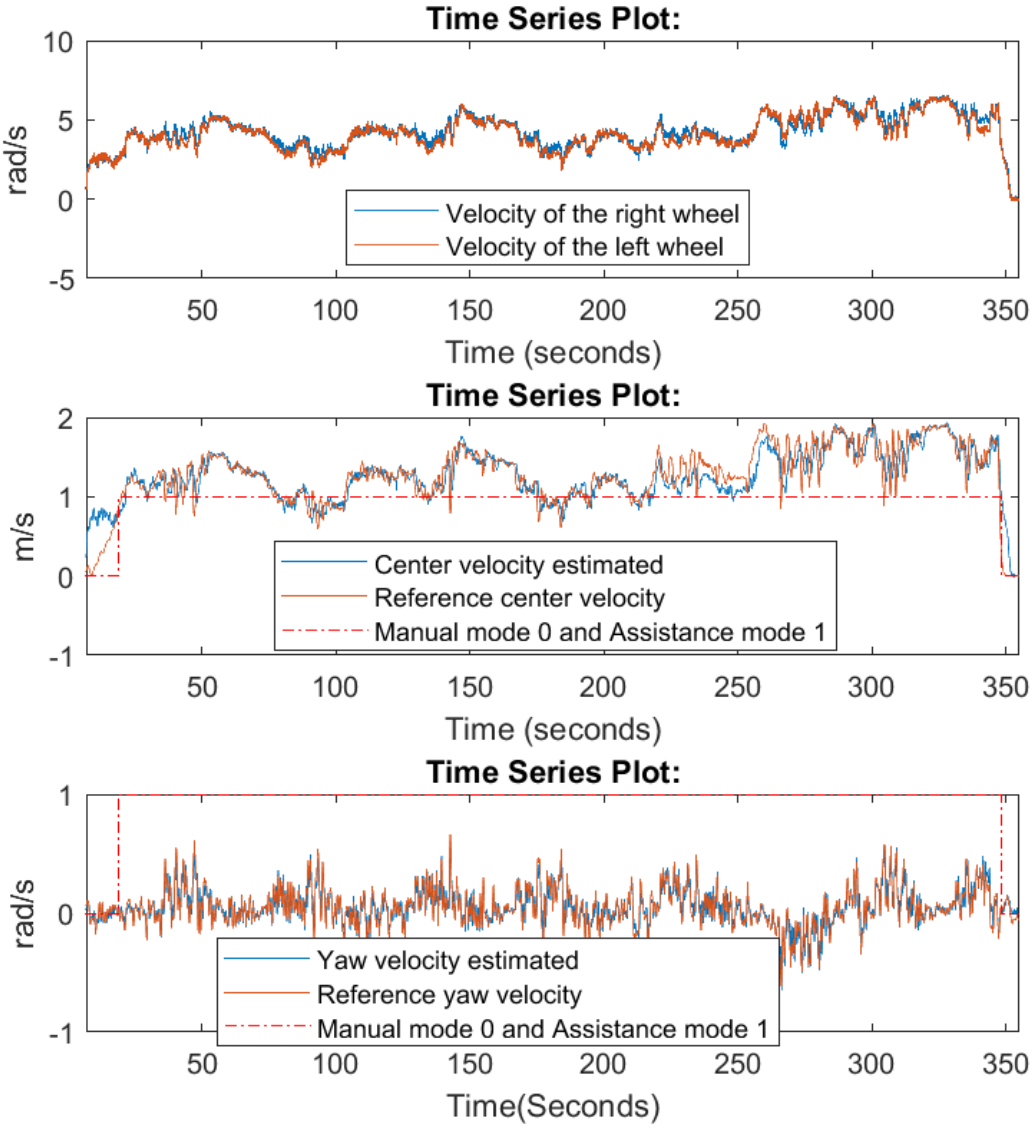

Figure 4.2.11. Velocity of each wheel, center and yaw velocities of the first user's trajectory tracking

For the second trial, user B has been asked to perform a round-trip between the points, indicated in red in Figure 4.2.12. The figure also shows in green the trajectory of the wheelchair during the driving task. Both user B and estimated torques are presented Figure 4.2.13.

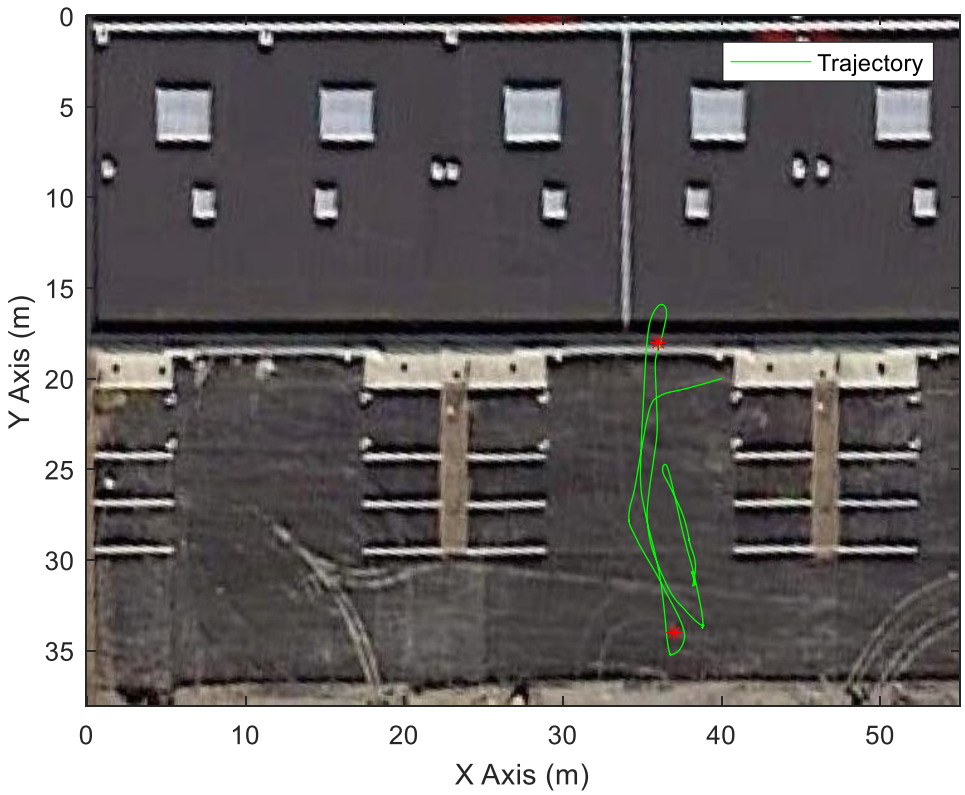

Figure 4.2.12. Trajectory tracking by the user B

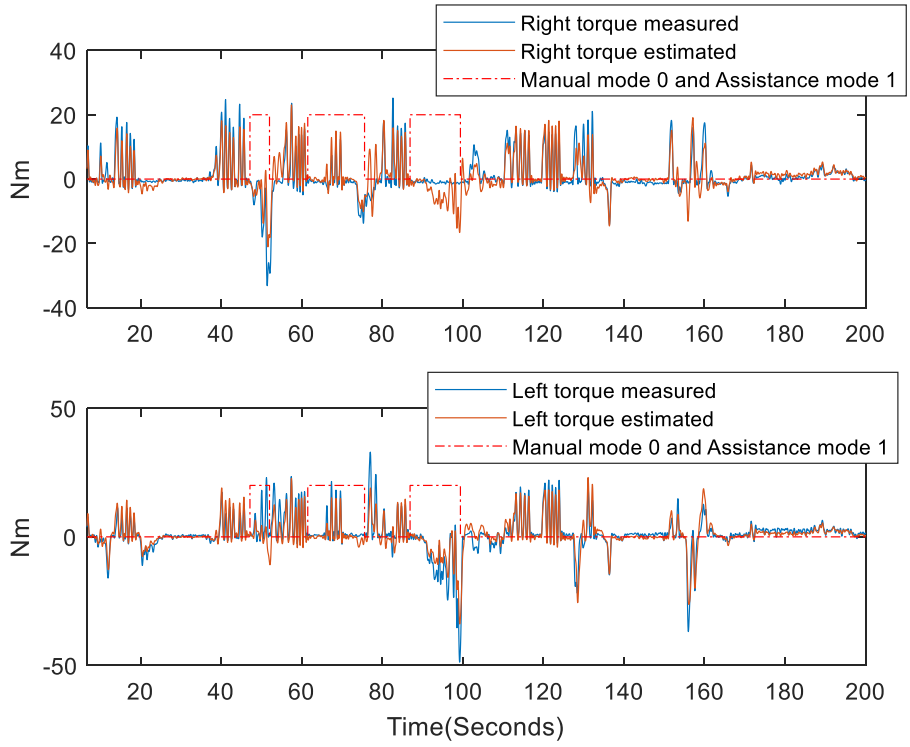

Figure 4.2.13. Human torque and estimated human torque of the trajectory tracking (User B)

Figure 4.2.14 presents the modes as well as the assistance torques. We can note that the manual mode has been preferentially used by user B. One of the reasons is that he was willing to accomplish the task by himself. Between $90s$ and $100s$, the control actions are saturated. As shown in Figure 4.2.15, the yaw velocity is ensured in the presence of actuator saturations as expected from the theoretical part.

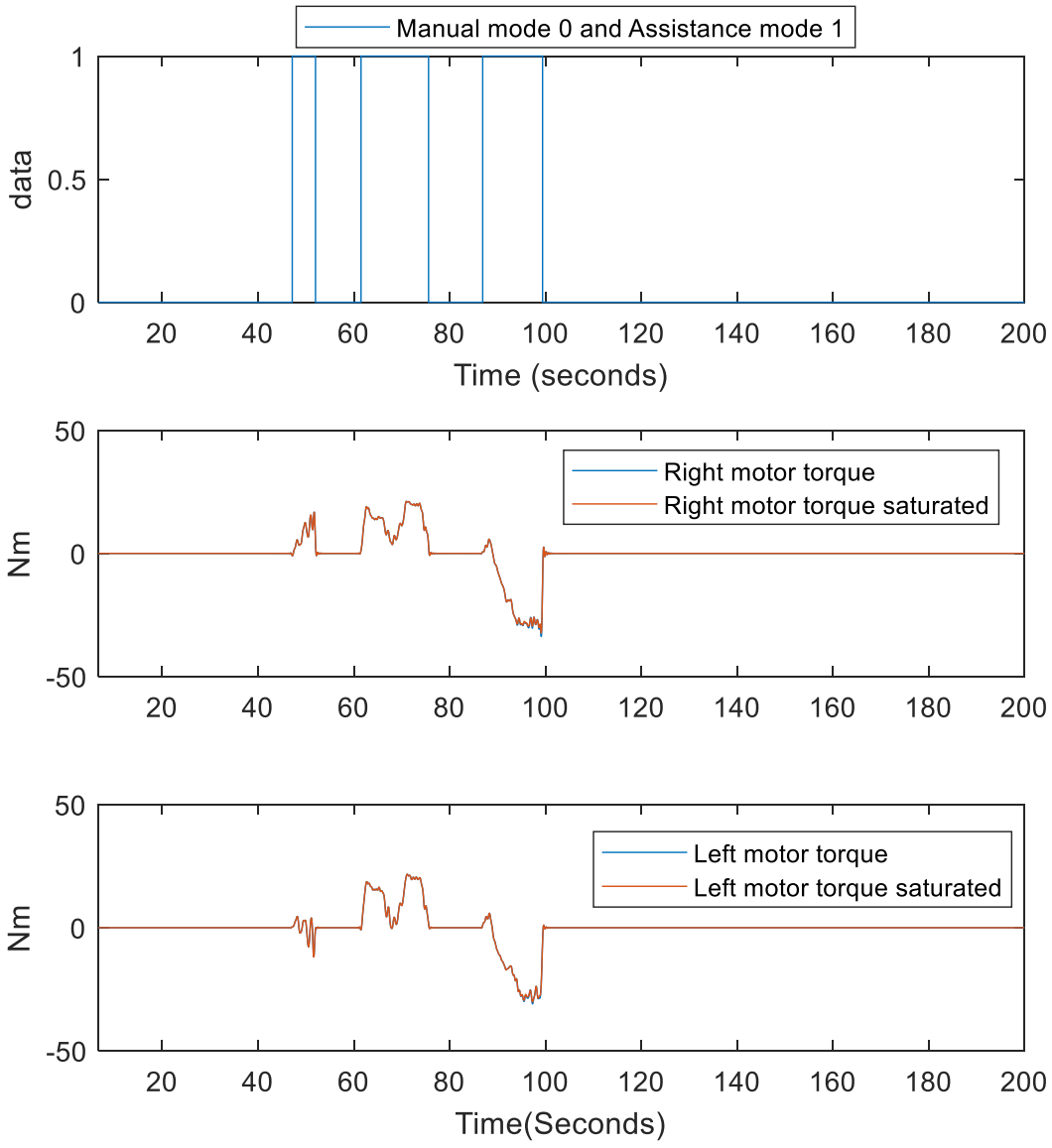

Figure 4.2.14. Mode of the wheelchair and assistive torque of the trajectory tracking (User B)

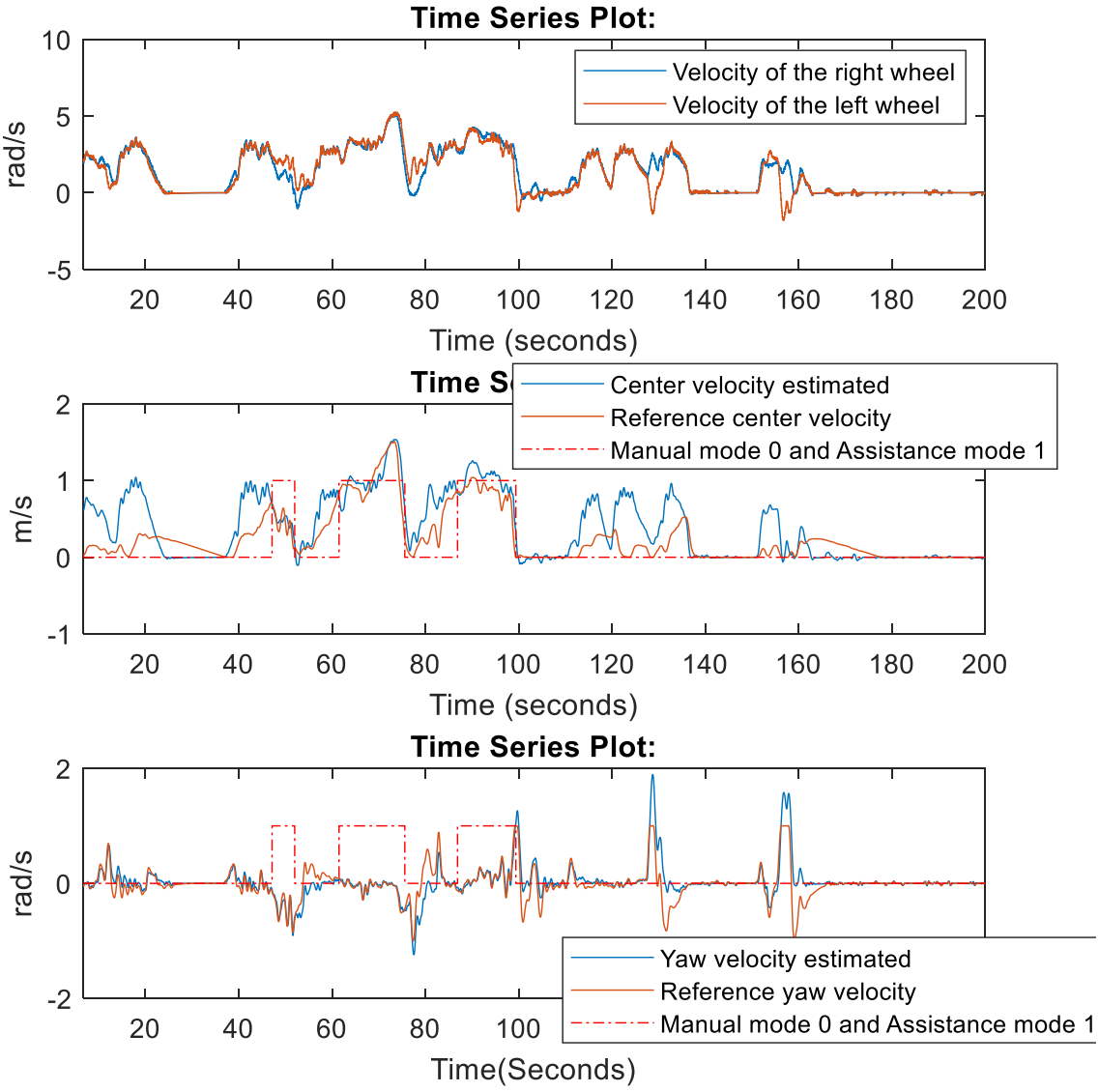

Figure 4.2.15. Velocity of each wheel, center and yaw velocities of user B trajectory tracking

4.3 Conclusion

The experimental results have been made step-by-step to show the capabilities of the unknown input PI-observer first, then to validate both the reference algorithm proposition and the robust observer-based tracking controller. Robustness tests according to the mass (2 different users) and to the ground adhesion has also been performed to show the effectiveness of the approach.

However, the observer design of Section 3.3 under time varying sampling has not been validated by experimental tests, due to the hardware constraints. One of the main future work directions focuses on validating experimentally this observer design as it should exhibit better performances especially for low speeds.

Moreover, the parameters of the reference generation algorithm may not be optimally calibrated for a particular user. For example, to achieve a same desired center velocity, different users may perform different pushing frequency. Therefore, an adaptability to these heterogeneous human behaviours seems necessary. Before proposing an idea to integrate such adaptability to the presented assistive control, the next chapter introduces a proof-of-concept study using the model-free reinforcement learning framework.

Chapter 5. Model-free optimal control design subject to PAWs

5.1 Introduction

In the two previous chapters, a robust observer-based tracking controller and a stability analysis have been provided for the mechanical part of the human-wheelchair system. However, the human fatigue dynamics, which are virtually always unknown, have not been taken into account. To deal with unknown human fatigue dynamics, we propose model-free approaches for our PAW application in this chapter.

With PAWs, depending on different human fatigue dynamics, users can perform a tuneable and suitable level of physical activities which could not be achieved with traditional manual wheelchairs or fully electric wheelchairs. Moreover, PAWs are driven by a hybrid energy source consisting of human metabolic power and electrical power from a battery. Thanks to this hybrid energy storage structure of PAWs, more degrees of freedom are available to design an optimal energy management strategy.

In this context, the major novelty we propose is a reinforcement learning control strategy for PAWs that optimizes electrical energy while also taking into account human fatigue. We formulate the assistive task as a constrained optimal control problem: the assistive algorithm is expected to produce a desired fatigue variation of users while using minimal electrical energy for a given driving task. With the initial-to-final fatigue constraint, a (near-) optimal assistance is found so that users contribute efficiently their metabolic energy.

In contrast to hybrid electrical bicycles (Corno et al. 2016; Guanetti et al. 2017; Wan et al. 2014) little work is done in the PAW literature to address energy optimization with human fatigue considerations. In (Seki et al. 2009), a regenerative braking control is applied to PAWs for safe downhill driving and electrical energy savings. In (Tanohata et al. 2010), the control system is based on a fuzzy algorithm and the fuzzy rules are designed by an expert, aiming to increase the energy efficiency. However, human fatigue has not been taken into account to design PAWs in the literature. The adaptability of optimal solutions with respect to different human fatigue dynamics is not analysed. An adaptable solution would be vital for

PAW designs, since different users may have different fatigue dynamics. Consequently, the existing model-based approaches would not be appropriate for our PAW energy management problem.

The present study relies on Patent WO2015173094 ("US20170151109A1 - Method and device assisting with the electric propulsion of a rolling system, wheelchair kit comprising such a device and wheelchair equipped with such a device") and designs assistive strategies for paraplegic wheelchair users. Specifically, we propose to use model-free reinforcement learning methods to calculate the optimal assistance while respecting a desired fatigue variation over a prescribed driving task. The optimal control method of choice is the direct Policy Search Policy Gradient (PG) (Sutton et al. 2000; Williams 1992). Compared with policy iteration (Buşoniu et al. 2010) and temporal difference learning (Boyan 2002), PG directly provides continuous actions without computing the value function (Bellman 1966), which renders it more practical in robotics (Kober et al. 2009). Another crucial advantage of PG is its online model-free nature: it treats the wheelchair dynamics, human fatigue dynamics, and human controller as a "black box", and the algorithm only needs state measurements and rewards (negative costs) in order to learn the solution. This is important in practice, since the true human dynamics will almost never be available.

Before moving on to practical implementation, the learning methodology must be evaluated in simulation with mathematical models which can roughly represent the human-wheelchair behaviors. We select the human state of fatigue (S_{of}) model from (Fayazi et al. 2013), the human controller model (Ronchi et al. 2016) and the wheelchair model simplified from the dynamics (2.2.5) and use these models to verify numerically the optimality of the solution found by PG. A baseline solution is given by a finite-horizon extension of the fuzzy Q-iteration in (Buşoniu et al., 2010). The two PG methods used in this paper are Gradient of a Partially Observable Markov Decision Processes (GPOMDP) (Peters et al. 2006), (Baxter et al. 2000) and Policy learning by Weighting Exploration with the Returns (PoWER) (Kober et al. 2009).

First of all, we apply the policy-gradient approach GPOMDP for our PAW application (Feng et al. 2018). To verify if this approach is able to provide a sub-optimal solution, we compare the solution provided by GPOMDP with the baseline solution provided by fuzzy Q-iteration. Next, we aim to improve considerably the data efficiency of the approach, by employing a different learning algorithm, PoWER, and by simplifying the parametrization of the

controller. The idea is to find a near-optimal policy in much fewer trials, so as make the method better in practice. We also derive a new near-optimality analysis of fuzzy Q-iteration, which is not included in (Buşoniu et al., 2010). Moreover, the learning method is expected to be adaptable to either different users or changes in the same user. To verify this possibility, a novel investigation is performed in this chapter. We modify the human fatigue dynamics to represent three categories of users: physically strong, normal and weak. Simulations are conducted to confirm if the proposed learning method is able to provide a solution that adapts to these cases. We also study the different convergence speeds to the baseline solution when using the parameters learned with the nominal fatigue model versus resetting them to zero defaults.

Our objective with the simulations described above is to evaluate, as a proof of concept, the effectiveness of the learning methodology in the PAW domain. To this end, we select the coarse models (Fayazi et al. 2013), (Ronchi et al. 2016), (Tashiro et al. 2008). While these models do generate qualitatively and physically meaningful interconnected human-wheelchair behaviours (Feng et al. 2018), and thus are useful as an initial validation step, they are not required to be very accurate. Indeed, the main strength of the learning algorithm is that it does not depend on the details of the particular model or notion of fatigue used, instead working for a wide range of unknown dynamics. Having performed these simulations, our next step is to conduct an experiment with the real PAW, where the fatigue model is replaced by a joystick, using which users return a discrete subjective evaluation of their S_{of} to the learning algorithm (too fatigued, OK, and insufficiently fatigued/desiring more exercise). This experiment serves to verify whether the learning methodology works in the real application, which by necessity is quite different from the simulation model.

This chapter is organized as follows: In Section 5.2, we present the human-wheelchair model and the problem formulation. In Section 5.3, we formulate the optimal problem for our PAW application. Section 5.4 introduces the baseline solution derived from the approximate dynamic programming and its optimality analysis. Section 5.5 provides the first reinforcement learning algorithm and its comparison with the baseline solution. In section 5.6, we improve the data-efficiency by applying the second reinforcement learning method i.e. PoWER and presents the experimental results. Section 5.7 gives our conclusion and discusses direction for future work.

5.2 Models for simulation validations

Next, we introduce a human-wheelchair model which is used to validate in simulation the proposed model-free PG approaches. The proposed human model represents only coarsely human behaviours in practice, since human muscle fatigue would be difficult to precisely model or quantitatively measure (Fayazi et al. 2013). However, the model is sufficiently representative to validate numerically the learning approach.

5.2.1 Human fatigue dynamics

Owing to the repetitive nature of wheelchair pushing and the absence of a dynamical human fatigue model dedicated to PAWs in the literature, we apply the muscle fatigue model from (Fayazi et al. 2013) used for a cycling application. The chosen single-state human fatigue model takes into account the fatigue effect and the recovery effect which usually happen for long-term sports such as wheelchair pushing (Rodgers et al. 1994). Considering these two effects, an "intelligent" assistance can be devised to save electrical energy. Although significant differences exist between the bicycle problems and PAW problems, this model is still qualitatively meaningful and therefore useful for numerical validation.

The dynamic of the maximum available force F_m provided by human is:

$$\dot{F}_m(t) = -\left(\mathrm{R} + \frac{\mathrm{F}\,F_h(t)}{M_{vc}}\right)F_m(t) + \mathrm{R}\,M_{vc} \tag{5.2.1}$$

where $0 \le F_h(t) \le F_m(t) \le M_{vc}$, and M_{vc} is the Maximum Voluntary Contraction force at rest, and $F_h(t)$ is the actual human applied force. Moreover, $\mathcal{F}$ and $\mathcal{R}$ represent the fatigue coefficient and the recovery coefficients respectively.

When $F_h = F_m$, F_m decreases at its maximum rate. This leads (5.2.1) to an equilibrium point where the fatigue rate is identical to the recovery rate, $\dot{F}_m = 0$, and the positive solution is:

$$F_{eq} = \frac{\mathrm{R}\,M_{vc}}{2\mathrm{F}}\left(-1 + \sqrt{1 + \frac{4\mathrm{F}}{\mathrm{R}}}\right) \tag{5.2.2}$$

This positive solution F_{eq} is also the minimum threshold that $F_m(t)$ can achieve. Thus $F_{eq} \le F_m \le M_{vc}$. Using the first-order Euler's method, a discrete-time version of (5.2.1) is:

$$F_{m_{k+1}} = \left[1 - T_e\left(\mathrm{R} + \frac{\mathrm{F}F_{h_k}}{M_{vc}}\right)\right]F_{m_k} + T_e \mathrm{R} M_{vc} \tag{5.2.3}$$

with the sampling time T_e. Then, the state of fatigue S_{of} in discrete time is defined as:

$$S_{of\,k} = \frac{M_{vc} - F_{mk}}{M_{vc} - F_{eq}} \tag{5.2.4}$$

The S_{of} is therefore the normalized value of F_m and is used as an indicator to quantify the human fatigue.

5.2.2 Simplified wheelchair dynamics and Human controller

The wheelchair is simplified from the original model (2.2.1) and described by the following:

$$\begin{bmatrix} d_{k+1} \\ v_{k+1} \end{bmatrix} = \mathbf{A}\begin{bmatrix} d_k \\ v_k \end{bmatrix} + \mathbf{B}\left(U_k + F_{hk}\varsigma\right) \tag{5.2.5}$$

where the system matrix $\mathbf{A} \in \mathrm{R}^{2\times 2}$ and the input matrix $\mathbf{B} \in \mathrm{R}^{2\times 1}$. With the nominal prameters in Table I, we have:

$$\mathbf{A} = \begin{bmatrix} 1 & 0.05 \\ 0 & 0.9406 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0.0059 \end{bmatrix}.$$

The control input is the motor torque U and ς is the wheel radius of hand-rims. The variables d and v are the wheelchair position and velocity, respectively. Note that the human torque satisfies $U_{hk} = F_{hk}\varsigma$.

We assume that the human force F_h depends on the fatigue state S_{of}, the electrical motor torque U, and the wheelchair velocity v (all perceived by the user):

$$F_{hk} = y\left(U_k, S_{of_k}, v_k\right) \tag{5.2.6}$$

Here, we extend the fatigue-motivation model (Ronchi et al. 2016) to describe roughly how the fatigue and the assistance affect human motivation. An accurate model of the motivation would require significant further studies that are outside the scope of this work, since it would not contribute significantly to our initial objective of validating the learning methodology

with a coarse model. Human fatigue decreases the motivation and the perceived help increases it. The normalized help is:

$$H_k = U_k / U_{max} \in [0,1] \tag{5.2.7}$$

where U_{max} is the maximum motor torque. The equilibrium point between the perceived fatigue and the perceived help is:

$$f_k = \frac{H_k - S_{of_k}}{H_k + S_{of_k}} \in [-1,1] \tag{5.2.8}$$

The motivation $\mathcal{M}$ is:

$$\mathrm{M}_k = \begin{cases} f(1+f_k) & \text{if } f_k < 0 \\ f + (1-f) f_k & \text{if } f_k \geq 0 \end{cases} \tag{5.2.9}$$

where $\mathcal{M} \in [0,1]$ and the parameter $f \in [0,1]$. The user motivation in (5.2.9) affects proportionally the desired wheelchair velocity V_r of the user, so that a higher motivation leads to a higher desired velocity, i.e. $V_r = V_{max}\mathcal{M}$ (where V_{max} is the maximum velocity of the wheelchair). Finally, the human force is modelled as a proportional velocity-tracking controller:

$$F_{h_k} = K_p \left(V_{max} \mathrm{M}_k - v_k \right) \tag{5.2.10}$$

Moreover, the human force should be saturated by F_m, and only positive human force is taken into account:

$$F_{h_k} = \mathrm{sat}\left(0, F_{m_k}, F_{h_k}\right) \tag{5.2.11}$$

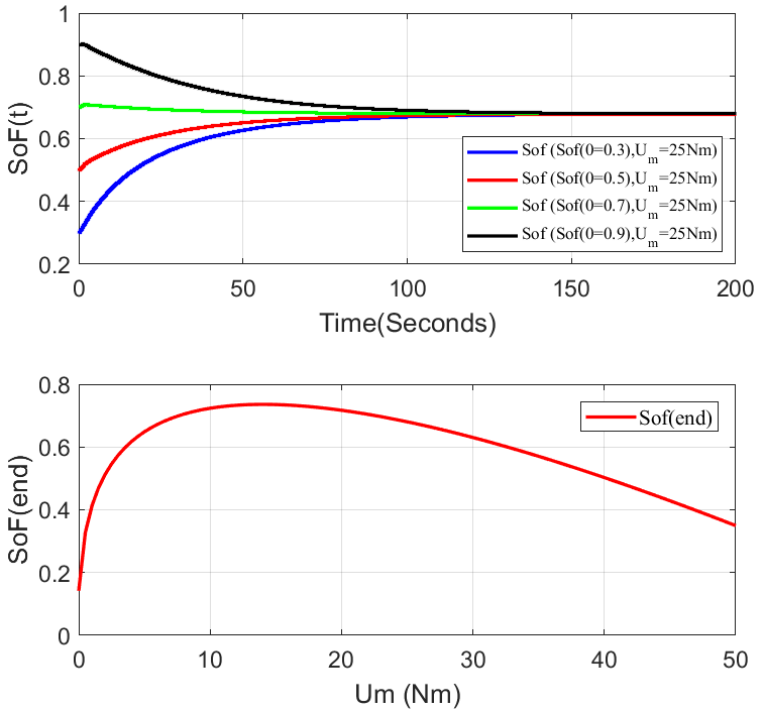

Figure 5.2.1. $S_{of}(k)$ evolution with a constant $U_m = 25Nm$ (above) and $S_{of}(end)$ evolution respect to U_m (below)

Remark 13. The human controller represented by (5.2.6) is an implicit $S_{of}-$tracking controller for the interconnected wheelchair/human dynamics. Simulation results (with $f_n = 0.5$) in Figure 5.2.1 show that the S_{of} converges to a specific value for a constant U_m; the proposed model manages the human fatigue depending on the perceived environment. Interestingly, the left part of the second curve, Figure 5.2.1 illustrates the fact that increasing perceived help motivates the user to do more physical exercise. The right part of this curve shows that when the assistive torque is increased, the motor assists the user to decrease his/her physical workload.

5.3 Optimal control problem formulation

For simplicity, we consider the electric energy consumption E_{elec} to be a quadratic function of U via the finite horizon criterion:

$$E_{elec} = \frac{1}{2}\sum_{k=0}^{K-1} U_k^2 \tag{5.3.1}$$

Over a predefined time horizon, the optimal solution minimizing (5.3.1) without considering any constraint corresponds to a manual propulsion strategy in which all the kinetic energy comes from the human. To avoid this trivial solution, we impose the following fatigue constraint. Knowing the initial S_{of_0}, the final S_{of_K} should reach a desired level S_{of-ref}:

$$S_{of\,K} = S_{of-ref} \tag{5.3.2}$$

while minimizing (5.3.1) over the considered driving profile. The wheelchair should also travel a required distance. Knowing the initial d_0, we impose the following distance constraint:

$$d_K = d_{ref} \tag{5.3.3}$$

including the terminal distance d_K and the desired terminal d_{ref}. Rather than solving explicitly a constrained problem, we represent the constraints (5.3.2)-(5.3.3) with a terminal reward, leading to the following optimal control problem:

$$\max_U R = -\left[w_{e1}\, w_{e2}\right]\begin{bmatrix} \left(d_K - d_{ref}\right)^2 \\ \left(S_{of\,K} - S_{of-ref}\right)^2 \end{bmatrix} - \frac{1}{2}\sum_{k=0}^{K-1} U_k^2 \tag{5.3.4}$$

with w_{e1} , w_{e2} the reward weights and K the finite time horizon. Note that in classical control theory the return R in (5.3.4) is often replaced by a positive cost function and must be minimized. Here, we use Artificial Intelligence techniques, so we follow the maximization convention in this field.

The system considered is described in general by the deterministic state transition function:

$$x_{k+1} = \xi\left(x_k, u_k\right) \tag{5.3.5}$$

where x and u are state vector and control input respectively. The general return R to optimize over a finite-horizon is:

$$R(\tau) = \gamma^K T\left(x_K\right) + \sum_{k=0}^{K-1} \gamma^k r\left(x_k, u_k\right) \tag{5.3.6}$$

where $\tau = (x_0, u_0, x_1, u_1, \ldots x_{K-1}, u_{K-1}, x_K)$ is a trajectory of the system, $T(x_K)$ is the terminal reward, and $r(x_k, u_k)$ is the stage reward. A discount factor $\gamma \in (0, 1]$ may be used; in the finite-horizon case, γ is often taken equal to 1. The optimization problem (5.3.4) is a

specific case of the general form (5.3.6). The following algorithms are presented for the general case defined in (5.3.5)-(5.3.6).

5.4 Baseline solution: Approximated dynamic programming

5.4.1 Finite-horizon fuzzy Q-iteration

Fuzzy Q-iteration (Buşoniu et al., 2010) is originally given in the infinite-horizon case, and the horizon-K solution can be obtained simply by iterating the algorithm K times. However, the entire time-varying solution must be maintained, and special care must be taken to properly handle the terminal reward. So for clarity we restate the entire algorithm, adapting it to the finite-horizon case.

The idea is to approximate the optimal time-varying solution, which can be expressed using Q-functions of the state in the state-space X and action in the action space U. These Q-functions are generated backwards in time:

$$\begin{aligned} &Q_{K-1}^{*}\left(x_{K-1},u_{K-1}\right)=r\left(x_{K-1},u_{K-1}\right)+\gamma T\left(\xi\left(x_{K-1},u_{K-1}\right)\right)\\ &Q_{k}^{*}\left(x_{k},u_{k}\right)=r\left(x_{k},u_{k}\right)+\gamma\max_{u_{k+1}}Q_{k+1}^{*}\left(\xi\left(x_{k},u_{k}\right),u_{k+1}\right),\\ &\text{for } k=K-2,\ldots,0 \text{ and } \forall x\in X,\ \forall u\in U \end{aligned} \tag{5.4.1}$$

The advantage of using Q-functions is that the optimal control can then be computed relatively easily, using the following time-varying state-feedback:

$$\pi^{*}\left(x_{k},k\right)=\arg\max_{u_{k}}Q_{k}^{*}\left(x_{k},u_{k}\right) \tag{5.4.2}$$

Since the system is nonlinear and the states and actions are continuous, in general it is impossible to compute the exact solution above. We will therefore represent Q^* with an approximator that relies on an interpolation over the state space, and on a discretization of the action space. First, to handle the action, the approximate Q-value of the pair (x,u) is replaced by that of the pair (x,u_d), where u_d has the closest Euclidean distance to u in a discrete subset of actions $U_d=\{\bar{u}_j|\bar{u}_j\in U, j=1,\ldots,N_u\}$. To handle the state, a grid of discrete values $X_d=\{\bar{x}_i|\bar{x}_i\in X, i=1,\ldots,N_x\}$ in the state space is chosen for the centers of triangular membership functions $\phi(x)=[\phi_1(x),\ldots,\phi_{N_x}(x)]$ (Buşoniu et al. 2010). A parameter vector

$\theta \subset \mathbb{R}^{N_x \times N_u \times K}$ is defined, and the approximate Q-function is linearly interpolated by overlapping the membership functions ϕ on the grid of the centers X_d as follows:

$$\hat{Q}_k(x,u) = \sum_{i=1}^{N_x} \phi_i(x)\theta_{i,j,k} \tag{5.4.3}$$

with $j \epsilon \arg\min_{j'} \|u - \bar{u}_{j'}\|^2$. Thus, each individual parameter corresponds to a combination between a point i on the state interpolation grids, a discrete action j, and a time stage k. The approximated optimal solution $\hat{\pi}$ can be obtained as follows:

$$\hat{\pi}(x,k) = \bar{u}_j \quad \text{with} \quad j = \arg\max_{j'} \sum_{i=1}^{N_x} \phi_i(x)\theta_{i,j',k}. \tag{5.4.4}$$

Algorithm 1. Finite-horizon fuzzy Q-iteration

1 **for** $i = 1, \ldots, N_x, j = 1, \ldots, N_u$ **do**
2 $\quad \theta_{i,j,K-1} = r(\bar{x}_i, \bar{u}_j) + \gamma T(\xi(\bar{x}_i, \bar{u}_j))$
3 **end for**
4 **for** $k = K-2, \ldots, 0$ **do**
5 $\quad$ **for** $i = 1, \ldots, N_x, j = 1, \ldots, N_u$ **do**
6 $\quad\quad \theta_{i,j,k} = r(\bar{x}_i, \bar{u}_j) + \gamma \max_{j'} \sum_{i'=1}^{N_x} \phi_{i'}(\xi(\bar{x}_{i'}, \bar{u}_{j'}))\theta_{i,j,k+1}$
7 $\quad$ **end for**
8 $\quad \hat{\pi}(x,k) = \bar{u}_j, \quad j = \arg\max_{j'} \sum_{i=1}^{N_x} \phi_i(x)\theta_{i,j',k} \quad \forall x,k$
9 **end for**

Algorithm 1 gives the complete version of Fuzzy Q-iteration. To understand it, note that the main update in line 6 is equivalent to the following approximate variant of the iterative update in (5.4.1):

$$\hat{Q}_k(\bar{x}_i, \bar{u}_j) = r(\bar{x}_i, \bar{u}_j) + \gamma \max_{\bar{u}_{j,k+1}} \hat{Q}_{k+1}(\xi(\bar{x}_i, \bar{u}_j), \bar{u}_{j,k+1}) \tag{5.4.5}$$

This is because, firstly, due to the properties of triangular basis functions the parameter $\theta_{i,j,k}$ is equal to the approximate Q-value $\hat{Q}_k(\bar{x}_i, \bar{u}_j)$. Secondly, the maximization over the discretized actions is done by enumeration over j; and thirdly, the summation is just the approximate Q-value at the next step, via (5.4.3). Line 2 simply sets the parameters at step K-1 via the initialization in (5.4.1).

For clarity, the algorithm shows in line 8 how the near-optimal control is computed via maximization over the discrete actions. In practice, this maximization is done on-demand, only for the states encountered while controlling the system, so an explicit function $\hat{\pi}$ of the continuous state does not have to be stored. Instead, only the parameters are stored.

5.4.2 Optimality analysis

In contrast to the algorithm itself, the infinite-horizon analysis does not easily extend to the finite-horizon case, e.g. we need to account for the possibility that $\gamma = 1$. Thus, the upcoming study, which has been presented in (Feng et al. 2019), provides a complete analysis.

The error ε_k between $\hat{Q}_k$ and Q_k^* for sample k is defined as:

$$\varepsilon_k = \left\| \hat{Q}_k(x,u) - Q_k^*(x,u) \right\| \tag{5.4.6}$$

The state resolution step δ_x is defined as the largest distance between any two neighbouring triangular MF cores, i.e.

$$\delta_x = \max_{i \in \{1,\dots N_x\}} \min_{i' \in \{1,\dots N_x\}, i' \neq i} \left\| \bar{x}_i - \bar{x}_{i'} \right\|_2$$

The action resolution step δ_u is defined similarly for the discrete actions. Moreover, for every x, only $2^{N_{state}}$ (where N_{state} is the number of states) triangular membership functions are activated. Let the infinite norm $\|\theta_k\|_\infty = \max_{i \in \{1,\dots N_x\}, j \in \{1,\dots N_u\}} |\theta_{i,j,k}|$ denotes the largest parameter magnitude at sample k. Note that triangular membership functions are Lipschitz-continuous, so there exists a Lipschitz constant $L_\phi > 0$ such that $\|\phi_i(x) - \phi_i(x')\|_2 \leq L_{\phi_i}(\|x - x'\|_2)\ \forall x, x' \in X, \forall i$. Moreover, we say that a function of the state and action, such as the deterministic state transition function ξ, is Lipschitz continuous with constant $L_\xi > 0$ if $\|\xi(x,u) - \xi(x',u')\|_2 \leq L_\xi(\|x - x'\|_2 + \|u - u'\|_2)\ \forall x, x' \in X, u, u' \in U$.

Assumption 1: The reward function r, the terminal function T, and the deterministic state transition function ξ are Lipschitz-continuous with the Lipschitz constants L_r, L_T, and L_ξ respectively.

We present an explicit bound on the near-optimality of the Q-function as a function of the grid resolutions. This bound has the nice feature that it converges to zero when the grid becomes infinitely dense, which is a consistency property of the algorithm.

Proposition 1: Under Assumption 1, there exists an error bound $\bar{\varepsilon}_k$ so that $\hat{Q}$, i.e. the approximate Q-function obtained by (5.4.6) satisfies $\varepsilon_k \leq \bar{\varepsilon}_k$ and $\lim_{\delta_x,\delta_u\to 0} \bar{\varepsilon}_k = 0$ for $k = K-1, \ldots, 0$. Depending on the discount factor γ and the Lipschitz constant L_ξ, the bound is given as follows:

$$\gamma L_\xi < 1, \qquad \bar{\varepsilon}_k = \left(\delta_x + \delta_u\right)\left(\left(K-k\right)L_r + \sum_{z=1}^{K-k}\left(L_T\left(\gamma L_\xi\right)^z + L_r\frac{\left(\gamma L_\xi\right)^z - \gamma L_\xi}{\gamma L_\xi - 1}\right)\right) \tag{5.4.7}$$

$$\gamma L_\xi = 1, \qquad \bar{\varepsilon}_k = \left(\delta_x + \delta_u\right)\left(K-k\right)\left(L_r + L_T + \frac{\left(K-k-1\right)}{2}L_r\right) \tag{5.4.8}$$

$$\gamma L_\xi > 1, \qquad \bar{\varepsilon}_k = \left(\delta_x + \delta_u\right)\left(\left(K-k\right)L_r + 2^{N_{state}}\gamma L_\xi L_\phi \sum_{z=1}^{K-k}\left\|\theta_{K-z+1}\right\|_\infty\right) \tag{5.4.9}$$

5.4.2.1 Lipschitz property of Q^*

Before giving the proof of proposition 1, we explore the Lipschitz property of Q^*. Hereafter, we prove that the Q_k^* function is Lipschitz for $k = 0,1, \ldots K$. Knowing that T function is Lipschitz and the exact optimal Q function Q_K^* is equal to the terminal return as $\forall x \in X, u \in U$:

$$Q_K^*\left(x,u\right) = T\left(x\right)$$

Consequently, Q_K^* is Lipschitz. Considering an arbitrary time stage k ($k = 0,1, \ldots K-1$), we obtain the following inequality $\forall x, x' \in X, u, u' \in U$:

$$\begin{aligned}
&\left|Q_k^*\left(x,u\right) - Q_k^*\left(x',u'\right)\right| \\
&= \left|r\left(x,u\right) + \gamma \max_{\tilde{u}} Q_{k+1}^*\left(\xi\left(x,u\right),\tilde{u}\right) - r\left(x',u'\right) - \gamma \max_{\tilde{u}'} Q_{k+1}^*\left(\xi\left(x',u'\right),\tilde{u}'\right)\right| \\
&\leq \left|r\left(x,u\right) - r\left(x',u'\right)\right| + \gamma \max_{\tilde{u}'}\left|Q_{k+1}^*\left(\xi\left(x,u\right),\tilde{u}'\right) - Q_{k+1}^*\left(\xi\left(x',u'\right),\tilde{u}'\right)\right| \\
&\leq \left|r\left(x,u\right) - r\left(x',u'\right) + \gamma \max_{\tilde{u}'}\left[Q_{k+1}^*\left(\xi\left(x,u\right),\tilde{u}'\right) - Q_{k+1}^*\left(\xi\left(x',u'\right),\tilde{u}'\right)\right]\right|
\end{aligned} \tag{5.4.10}$$

Note that the stage reward function r is Lipschitz. Therefore, we can bound (5.4.10) using the triangular inequality property as follows:

$$\begin{aligned}&\left|Q_k^*(x,u)-Q_k^*(x',u')\right|\\&\le L_r\left(\|x-x'\|_2+\|u-u'\|_2\right)+\gamma\max_{\tilde{u}'}\left|Q_{k+1}^*\left(\xi(x,u),\tilde{u}'\right)-Q_{k+1}^*\left(\xi(x',u'),\tilde{u}'\right)\right|\end{aligned}\tag{5.4.11}$$

Supposing that for an arbitrary time stage k ($k=0,1,\dots K-1$), Q_{k+1}^* is Lipschitz and its Lipschitz constant is L_{k+1}. The inequality (5.4.11) can be expressed as:

$$\begin{aligned}&\left|Q_k^*(x,u)-Q_k^*(x',u')\right|\\&\le L_r\left(\|x-x'\|_2+\|u-u'\|_2\right)+\gamma\max_{\tilde{u}'}\left|L_{k+1}\left\|\xi(x,u)-\xi(x',u')\right\|_2\right|\\&\le L_r\left(\|x-x'\|_2+\|u-u'\|_2\right)+L_{k+1}\gamma L_\xi\left(\|x-x'\|_2+\|u-u'\|_2\right)\\&\le\left(L_r+L_{k+1}\gamma L_\xi\right)\left(\|x-x'\|_2+\|u-u'\|_2\right)\end{aligned}$$

Then, Q_k^* is also Lipschitz and its Lipschitz constant is $L_k=L_r+L_{k+1}\gamma L_\xi$. As a result, the Q_k^* function is Lipschitz for $k=0,1,\dots K-1$. Now we write the general form for the Lipschitz constant L_k as follows:

When $\gamma L_\xi=1$, $L_k=L_r+L_{k+1}$. The Lipschitz constant L_k is:

$$L_k=(K-k)L_r+L_T$$

When $\gamma L_f\neq 1$, $L_k=L_\rho+L_{k+1}\gamma L_f$. The Lipschitz constant L_k is:

$$L_k=\left(L_T+\frac{L_r}{\gamma L_\xi-1}\right)\left(\gamma L_\xi\right)^{K-k}-\frac{L_r}{\gamma L_\xi-1}=L_T\left(\gamma L_\xi\right)^{K-k}+L_r\frac{\left(\gamma L_\xi\right)^{K-k}-1}{\gamma L_\xi-1}$$

At last, before giving the proof of the proposition let us give also a property shared by the membership functions $\phi_i(x)$. They hold a convex property, i.e. for any state x:

$$\sum_{i=1}^{M}\phi_i(x)=1\tag{5.4.12}$$

Therefore, trivially we can decompose (5.4.12) as:

$$\sum_{i=1}^{M}\phi_i(x)=\sum_{i\in\{i|\phi_i(x)\neq 0\}}\phi_i(x)+\sum_{i\in\{i|\phi_i(x)=0\}}\phi_i(x)$$

With $\sum_{i\in\{i|\phi_i(x)=0\}}\phi_i(x)=0$ and $\sum_{i\in\{i|\phi_i(x)\neq 0\}}\phi_i(x)=1$. Moreover, for the terms of the second sum, denoting $I_k=\{i\,|\,\phi_i(x)\neq 0\}$, $i\in I_k$ defining δ_x as the state resolution step we can write:

$$\|x-x_i\|_2\leq\delta_x$$

5.4.2.2 Proof of proposition 1

The exact optimal time-varying Q-function can be expressed as ($k=K-1,K-2,\ldots,1$ and $\forall x\in X,u\in U$):

$$Q_k^*=r_k+\gamma\max_{u_{k+1}}Q_{k+1}^*(\xi_k,u_{k+1})$$

when $k=K$,

$$Q_K^*=Q^*(x_K,u_K)=T(x_K)$$

Or

$$Q^*(\xi_{K-1},u_K)=T(\xi_{K-1})$$

The approximate Q-function is $(k=K-1,K-2,\ldots,1$ and $\forall x\in X,u\in U)$:

$$\hat{Q}_k=\hat{Q}(x_k,u_k)=\sum_{i=1}^{M}\phi_i(x_k)\left[r(\bar{x}_i,\bar{u}_k)+\gamma\max_{u_{k+1}}\hat{Q}_{k+1}(\xi(\bar{x}_i,\bar{u}_k),u_{k+1})\right]\qquad(5.4.13)$$

with $\bar{u}_k=\underset{u_k^j}{\operatorname{argmin}}\|u_k-u_k^j\|$ and $u_k^j\in U_d$. With the set $I_k=\{i\,|\,\phi_i(x)\neq 0\}$, the approximation (5.4.13) becomes:

$$\hat{Q}_k=\sum_{i\in I_k}\phi_i(x_k)\left[r(\bar{x}_i,\bar{u}_k)+\gamma\max_{u_{k+1}}\hat{Q}_{k+1}(\xi(\bar{x}_i,\bar{u}_k),u_{k+1})\right]$$

When $k=K$,

$$\hat{Q}_k=\hat{Q}_K=Q^*(x_K,u_K)=T(x_K)$$

Or

$$\hat{Q}(\xi_{i,K-1},u_K)=T(\xi_{i,K-1})$$

The error between the approximate Q-function and the optimal one for arbitrary k:

$$\begin{aligned}
\varepsilon_k &= \left|\hat{Q}_k - Q_k^*\right| \\
&= \left|\sum_{i\in I_k}\phi_i\left(x_k\right)\left[r\left(\bar{x}_i,\bar{u}_k\right)+\gamma\max_{u_{k+1}}\hat{Q}_{k+1}\left(\xi_{i,k},u_{k+1}\right)\right]-r_k-\gamma\max_{u_{k+1}}Q_{k+1}^*\left(\xi_k,u_{k+1}\right)\right| \\
&\le \sum_{i\in I_k}\phi_i\left(x_k\right)\left|r\left(\bar{x}_i,\bar{u}_k\right)+\gamma\max_{u_{k+1}}\hat{Q}_{k+1}\left(\xi_{i,k},u_{k+1}\right)-r_k-\gamma\max_{u_{k+1}}Q_{k+1}^*\left(\xi_k,u_{k+1}\right)\right| \\
&\le \sum_{i\in I_k}\phi_i\left(x_k\right)\left|L_r\left(\left\|\bar{x}_i-x_k\right\|_2+\left\|\bar{u}_k-u_k\right\|_2\right)+\gamma\max_{u_{k+1}}\hat{Q}_{k+1}\left(\xi_{i,k},u_{k+1}\right)-\gamma\max_{u_{k+1}}Q_{k+1}^*\left(\xi_k,u_{k+1}\right)\right|
\end{aligned} \tag{5.4.14}$$

Using the triangular inequality property and introducing $Q_{k+1}^*\left(\xi_{i,k},u_{k+1}\right)-Q_{k+1}^*\left(\xi_{i,k},u_{k+1}\right)$ the error can be bounded as:

$$\begin{aligned}
\varepsilon_k &= \left|\hat{Q}_k - Q_k^*\right| \\
&\le L_r\left(\delta_x+\delta_u\right)+\sum_{i\in I_k}\phi_i\left(x_k\right)\gamma\max_{u_{k+1}}\left|\hat{Q}_{k+1}\left(\xi_{i,k},u_{k+1}\right)-Q_{k+1}^*\left(\xi_{i,k},u_{k+1}\right)+Q_{k+1}^*\left(\xi_{i,k},u_{k+1}\right)-Q_{k+1}^*\left(\xi_k,u_{k+1}\right)\right| \\
&\le L_r\left(\delta_x+\delta_u\right)+\varepsilon_{k+1}+\sum_{i\in I_k}\phi_i\left(x_k\right)\gamma\max_{u_{k+1}}\left|Q_{k+1}^*\left(\xi_{i,k},u_{k+1}\right)-Q_{k+1}^*\left(\xi_k,u_{k+1}\right)\right|
\end{aligned}$$

Thus:

$$\varepsilon_k-\varepsilon_{k+1}\le L_r\left(\delta_x+\delta_u\right)+\sum_{i\in I_k}\phi_i\left(x_k\right)\gamma\max_{u_{k+1}}\left|Q_{k+1}^*\left(\xi_{i,k},u_{k+1}\right)-Q_{k+1}^*\left(\xi_k,u_{k+1}\right)\right| \tag{5.4.15}$$

Since the optimal Q-function Q_{k+1}^* is proved previously to be a Lipschitz function with the corresponding Lipschitz constant L_{k+1}, the inequality (5.4.15) can be expressed as:

$$\varepsilon_k-\varepsilon_{k+1}\le L_r\left(\delta_x+\delta_u\right)+\sum_{i\in I_k}\phi_i\left(x_k\right)\gamma\max_{u_{k+1}}L_{k+1}\left\|\xi_{i,k}-\xi_k\right\|_2$$

With the Lipschitz property of ξ and the convex sum property $\sum_{i\in I_k}\phi_i\left(x\right)=1$, we have:

$$\varepsilon_k-\varepsilon_{k+1}\le\left(L_r+L_{k+1}\gamma L_\xi\right)\left(\delta_x+\delta_u\right)$$

With the same reasoning, the error between the approximate Q-function and the optimal one for $k=K-2,K-1,\dots K-m$:

$$\begin{aligned}
\varepsilon_{K-1}-\varepsilon_K &\le\left(L_r+L_K\gamma L_\xi\right)\left(\delta_x+\delta_u\right) \\
\varepsilon_{K-2}-\varepsilon_{K-1} &\le\left(L_r+L_{K-1}\gamma L_\xi\right)\left(\delta_x+\delta_u\right) \\
\varepsilon_{K-3}-\varepsilon_{K-2} &\le\left(L_r+L_{K-2}\gamma L_\xi\right)\left(\delta_x+\delta_u\right)
\end{aligned}$$

$$\vdots$$

$$\varepsilon_{K-m} - \varepsilon_{K-m+1} \leq \left(L_r + L_{K-m+1}\gamma L_\xi\right)\left(\delta_x + \delta_u\right)$$

Summing up the right and the left sides of the inequalities above, we obtain the error ε_{K-m} as:

$$\varepsilon_{K-m} - \varepsilon_K \leq \sum_{z=1}^{m}\left(L_r + L_{K-z+1}\gamma L_\xi\right)\left(\delta_x + \delta_u\right) \tag{5.4.16}$$

Since we can compute the exact Q-function of the final state, the error $\varepsilon_K = 0$. With $\varepsilon_K = 0$ and (5.4.16), we have:

$$\varepsilon_{K-m} \leq \sum_{z=1}^{m}\left(L_r + L_{K-z+1}\gamma L_\xi\right)\left(\delta_x + \delta_u\right) \tag{5.4.17}$$

For the special case $\gamma L_\xi = 1$, the bound of (5.4.17) can be expressed as:

$$\varepsilon_{K-m} \leq m\left(\delta_x + \delta_u\right)\left(L_r + L_T + \frac{m-1}{2}L_r\right)$$

And with $k = K - m$ it corresponds to (5.4.8).

Otherwise, when $\gamma L_\xi < 1$ and $\gamma L_\xi > 1$, with $k = K - m$, (5.4.17) can be bounded as:

$$\varepsilon_k \leq \left(K-k\right)L_r\left(\delta_x + \delta_u\right) + \sum_{z=1}^{K-k}\left(L_T\left(\gamma L_\xi\right)^z + L_r\frac{\left(\gamma L_\xi\right)^z - \gamma L_\xi}{\gamma L_\xi - 1}\right)\left(\delta_x + \delta_u\right) \tag{5.4.18}$$

Which corresponds to (5.4.7). Note that if $\gamma L_\xi < 1$ or $\gamma L_\xi = 1$, the error bound ε_k converges to zero when the resolution steps δ_x and δ_u tend to zero. For $\gamma L_\xi > 1$, due to $\left(\gamma L_\xi\right)^z$ the proposed error bound ε_k above increases exponentially, when the horizon increases. Since the horizon is finite, the error bound converges still to zero when the resolution steps δ_x and δ_u tend to zero. In what follows, we search a new error bound which provides a better feature in terms of convergence.

When $\gamma L_f > 1$, another error bound can be considered as follows. Consider again the error ε_k between the approximate Q-function and the optimal one, in (5.4.14) and introducing the null quantity $\hat{Q}_{k+1}(\xi_k, u_{k+1}) - \hat{Q}_{k+1}(\xi_k, u_{k+1})$ a new bound can be obtained as:

$$\begin{aligned}
&\varepsilon_k = \left|\hat{Q}_k - Q_k^*\right| \\
&\le L_r(\delta_x + \delta_u) + \sum_{i \in I_k} \phi_i(x_k)\gamma \max_{u_{k+1}} \left|\hat{Q}_{k+1}(\xi_{i,k}, u_{k+1}) - \hat{Q}_{k+1}(\xi_k, u_{k+1}) + \hat{Q}_{k+1}(\xi_k, u_{k+1}) - Q_{k+1}^*(\xi_k, u_{k+1})\right| \\
&\le L_r(\delta_x + \delta_u) + \varepsilon_{k+1} + \sum_{i \in I_k} \phi_i(x_k)\gamma \max_{u_{k+1}} \left|\hat{Q}_{k+1}(\xi_{i,k}, u_{k+1}) - \hat{Q}_{k+1}(\xi_k, u_{k+1})\right|
\end{aligned}$$

Define the new set of indexes: $I'_{k+1} = \{i' | \phi_{i'}(\xi_{i,k}) \neq 0 \; or \; \phi_{i'}(\xi_k) \neq 0\}$, we have:

$$\varepsilon_k \le L_r(\delta_x + \delta_u) + \varepsilon_{k+1} + \sum_{i \in I_k} \phi_i(x_k)\gamma \max_{u_{k+1}} \left| \sum_{i' \in I'_{k+1}} \left[\left(\phi_{i'}(\xi_{i,k}) - \phi_{i'}(\xi_k)\right)\theta_{i',j,k+1} \right] \right| \quad (5.4.19)$$

With $j = \underset{j'}{\operatorname{argmin}} \left\| u_{k+1} - u_k^{j'} \right\|$ and $u_k^{j'} \in U_d$. Using the Lipschitz property of ξ and the convex um property of the triangular membership function ϕ with the Lipschitz constant L_ξ and L_ϕ respectively,

$$\left\| \phi_{i'}(\xi_{i,k}) - \phi_{i'}(\xi_k) \right\|_2 \le L_\xi L_\phi (\delta_x + \delta_u) \quad (5.4.20)$$

With the inequality (5.4.19), the inequality (5.4.20) can be relaxed:

$$\varepsilon_k \le L_r(\delta_x + \delta_u) + \varepsilon_{k+1} + \sum_{i \in I_k} \phi_i(x_k)\gamma L_\xi L_\phi (\delta_x + \delta_u) \left| \sum_{i' \in I'_{k+1}} \theta_{i',j,k+1} \right|$$

For every x, only a finite number $2^{N_{state}}$ of MFs are non-zero and the cardinality of I'_{k+1} $|I'_{k+1}| \le 2^{N_{state}}$. Then,

$$\begin{aligned}
&\varepsilon_k \le L_r(\delta_x + \delta_u) + \varepsilon_{k+1} + \sum_{i \in I_k} \phi_i(x_k) * 2^{N_{state}} \gamma L_\xi L_\phi (\delta_x + \delta_u) \left\|\theta_{k+1}\right\|_\infty \\
&\le L_r(\delta_x + \delta_u) + \varepsilon_{k+1} + 2^{N_{state}} \gamma L_\xi L_\phi (\delta_x + \delta_u) \left\|\theta_{k+1}\right\|_\infty
\end{aligned}$$

And

$$\varepsilon_k - \varepsilon_{k+1} \le L_r(\delta_x + \delta_u) + 2^{N_{state}} \gamma L_\xi L_\phi (\delta_x + \delta_u) \left\|\theta_{k+1}\right\|_\infty$$

With the same reasoning, the error between the approximate Q function and the optimal one for $k = K-2, K-1, \ldots K-m$:

$$\varepsilon_{K-1} - \varepsilon_K \le L_r\left(\delta_x + \delta_u\right) + 2^{N_{state}} \gamma L_\xi L_\phi \left(\delta_x + \delta_u\right) \left\|\theta_K\right\|_\infty$$
$$\varepsilon_{K-2} - \varepsilon_{K-1} \le L_r\left(\delta_x + \delta_u\right) + 2^{N_{state}} \gamma L_\xi L_\phi \left(\delta_x + \delta_u\right) \left\|\theta_{K-1}\right\|_\infty$$

$$\vdots$$

$$\varepsilon_{K-m} - \varepsilon_{K-m+1} \le L_r\left(\delta_x + \delta_u\right) + 2^{N_{state}} \gamma L_\xi L_\phi \left(\delta_x + \delta_u\right) \left\|\theta_{K-m+1}\right\|_\infty$$

Summing up the right and the left sides of the inequalities above, we obtain the error ε_{K-m} as:

$$\varepsilon_{K-m} \le mL_r\left(\delta_x + \delta_u\right) + 2^{N_{state}} \gamma L_\xi L_\phi \left(\delta_x + \delta_u\right) \sum_{z=1}^{m} \left\|\theta_{K-z+1}\right\|_\infty$$

And we get with $k = K - m$:

$$\varepsilon_k \le \left(K-k\right) L_r\left(\delta_x + \delta_u\right) + 2^{N_{state}} \gamma L_\xi L_\phi \left(\delta_x + \delta_u\right) \sum_{z=1}^{K-k} \left\|\theta_{K-z+1}\right\|_\infty \tag{5.4.21}$$

That corresponds to (5.4.9), the last expression of proposition 1. At last notice that we have also for the last case (5.4.21) $\lim_{\delta_x \to 0, \delta_u \to 0} \bar{\varepsilon}_k = 0$, $k = K-1, \ldots, 0$ that ends the proof.■

5.5 Reinforcement Learning for Energy Optimization of PAWs

To represent the optimal control problem (5.3.4), where the objective is to minimize the electric energy consumption for a given driving task while producing a desired initial-to-final constraint of users, the terminal reward and the stage reward of (5.3.6) are defined as follows:

$$T\left(x_N\right) = -\left[w_{e1}\, w_{e2}\right] \begin{bmatrix} \left(d_K - d_{ref}\right)^2 \\ \left(S_{of\,K} - S_{of-ref}\right)^2 \end{bmatrix} \tag{5.5.1}$$
$$r\left(x_k, u_k\right) = -\frac{1}{2} U_k^2$$

where the state vector is $x_k = \left[d_k, v_k, S_{of_k}\right]^T$ and the control input is the motor torque $u_k = U_k$.

Since the driving task is to travel a predefined distance, negative human torque and negative motor torque are inefficient in terms of metabolic-electrical energy consumption over the driving task. Moreover, due to the actuator limitations, the maximum torque that the motor can provide is U_{max}. Therefore, the control is bounded: $0 \leq U \leq U_{max}$. Since the distance is monotonic, it acts as a proxy for time, which can be implicitly used by the algorithm instead of an explicit time variable. Therefore, we can use a time-invariant solution $\bar{\pi}_\lambda(x_k)$ to approximate the optimal time-varying solution in (5.4.2). We approximate the deterministic part $\bar{\pi}$ of the motor torque by the following RBF expansion:

$$\bar{\pi}_\lambda\left(x_k\right) = \lambda_l^T \varphi\left(x_k\right) \tag{5.5.2}$$

where the RBF $\varphi_i = \exp(-\beta\|x_k - c_i\|^2)$, $c_{i=1,\ldots,M}$ is the center vector of the RBFs, M is the total number of RBFs and β is the radial parameter. Since the radial parameter β is the same for each RBF, all the RBFs have the same shape.

Hereinafter, for each variable, a subscript or index P (resp. G) stands for PoWER (resp. GPOMDP).

In model-free Policy Search, exploration is indispensable to learn the unknown dynamics. Stochastic policies are needed to explore. To this end, we use a parameterized policy with the parameters λ. Then, the stochastic policy distribution is $\tilde{\pi}_\lambda(u_k|x_k, k)$. Under this stochastic policy, the probability distribution $p_\lambda(\tau)$ over trajectories τ can be expressed in the following way:

$$p_\lambda\left(\tau\right) = p\left(x_0\right)\prod_{k=0}^{K-1}\tilde{\pi}_\lambda\left(u_k|x_k, k\right) \tag{5.5.3}$$

where $p(x_0)$ is the initial state distribution. Under trajectories τ generated by $\tilde{\pi}_\lambda$, the expected return is:

$$\bar{R}_\lambda = \int p_\lambda\left(\tau\right)R\left(\tau\right)d\tau \tag{5.5.4}$$

5.5.1 GPOMDP

The GPOMDP (Gradient of a Partially Observable Markov Decision Processes) algorithm (Peters et al. 2006) updates the control parameters λ in the steepest ascent direction so that the expected return (5.5.4) is maximized. We apply this algorithm to estimate the gradient $\nabla_\lambda \bar{R}_\lambda$, which can be obtained from the stage rewards r_j and the distribution $\tilde{\pi}_\lambda$. The entire procedure is given in Algorithm 2, where Γ is the total trials.

Algorithm 2. GPOMDP

1 Initialize λ_0

2 **for** $l = 0, 1, 2, \ldots \Gamma$

3 Generate N_τ trajectories τ of length K using λ_l

4 $\nabla_\lambda \bar{R}_\lambda = \frac{1}{N_\tau} \sum_{\varsigma=1}^{N_\tau} \left[\sum_{k=0}^{K-1} \sum_{h=0}^{k} \left[\nabla_\lambda \log \tilde{\pi}_\lambda \left(u_h^\varsigma | x_h^\varsigma, k \right) \right] r_k^\varsigma \right]$

5 $\lambda_{l+1} = \lambda_l + \alpha \cdot \nabla_\lambda \bar{R}_\lambda$ with the learning rate $\alpha > 0$

6 **end for**

In line 3 of Algorithm 2, for each iteration l we generate N_τ trajectories using the stochastic policy with λ_l. Applying the Likelihood Ratio Estimator, calculating the gradient $\nabla_\lambda \bar{R}_\lambda$ is transformed to calculating $\nabla_\lambda \log \tilde{\pi}_\lambda (u_k | x_k, k)$. To this end, zero mean Gaussian noise z_G is added to the executed action and renders the policy (5.5.2) stochastic. In order to prevent the executed action from violating the action saturation limits, the stochastic motor torque is selected with:

$$q_{\text{sat}} \left[\lambda_l^{GT} \varphi \left(x_k \right) + z_G \right] \tag{5.5.5}$$

where q_{sat} is a smooth saturation (the Gaussian error function (5.5.5) shown at the top of Figure 5.5.1) between $[0, U_{max}]$ such that the stochastic action is differentiable with respect to λ_l^G. When the optimal action is close to the borders of the interval $[0, U_{max}]$, using the original return (5.5.1) without input saturation can lead to the divergence of the parameters. To address this problem, a penalty function P is added to the stage reward (5.5.1) as follows:

$$r \left(x_k, u_k \right) = -\left[\frac{1}{2} U_k^2 + w_{e3} P \left(U_k \right) \right] \tag{5.5.6}$$

where w_3 is the constraint penalty weight. The function P, shown in Figure 5.5.1 bottom, is defined as follows:

$$P=\begin{cases}\sin\left(\dfrac{\pi\left(U-U_{max}\right)}{0.04U_{max}}\right)+1 & 0.98U_{max}\le U\le U_{max}\\ 0 & 0.02U_{max}\le U\le 0.98U_{max}\\ \sin\left(\dfrac{\pi\left(-U\right)}{0.04U_{max}}\right)+1 & 0\le U\le 0.02U_{max}\end{cases} \tag{5.5.7}$$

which penalizes the (stochastic) action when it is close to the saturation value. The objective of P is to keep the mean value of the stochastic actions inside the interval $[0, U_{max}]$.

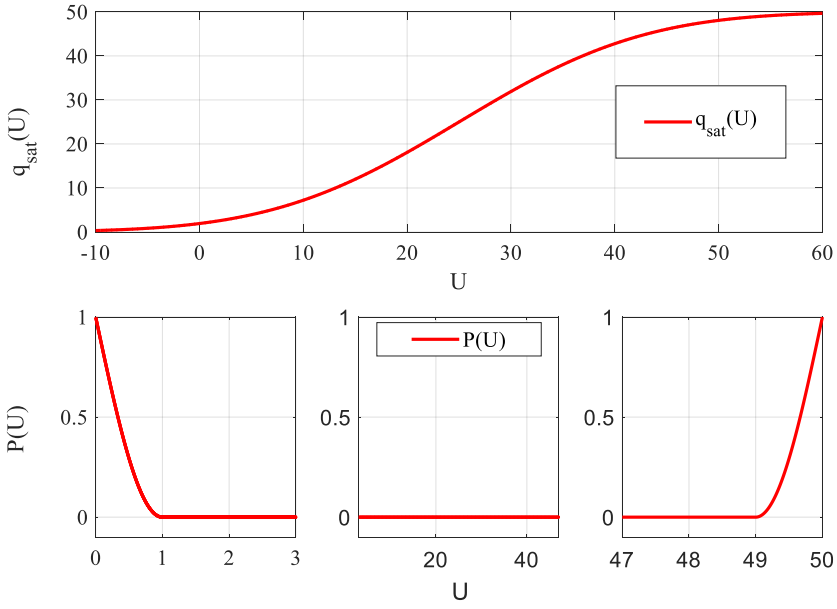

Figure 5.5.1. Smooth saturation function q_{sat} (above) and penalty function P_s for $U_{\max}=50N$ (below)

Recall that we use a time-invariant policy. Consequently, the stochastic action distribution does not depend on the time stage k, but on the state x_k. According to (5.5.5), the distribution $\tilde{\pi}_{\lambda}^{G}(U_k|x_k)$ of the stochastic motor torque U is:

$$\tilde{\pi}_{\lambda}^{G}\left(U_k|x_k\right)=\frac{1}{\sqrt{2\pi\sigma_G^2}}\exp\left(-\frac{\left[q_{\text{sat}}^{-1}\left(U_k\right)-\lambda_l^{GT}\varphi\left(x_k\right)\right]^2}{2\sigma_G^2}\right) \tag{5.5.8}$$

The derivative of (5.5.8) with respect to λ_l^G is used to estimate the gradient $\nabla_{\lambda}\bar{R}_{\lambda}$ in Algorithm 2 and to update the parameter vector λ_l^G. By tuning the parameters (β, c, M) of the basis

functions (5.5.2), the standard deviation σ_P and σ_G, the reward weights (w_1, w_2), the learning rate α and the penalty weight w_3, we have all the conditions to update the parameters λ^G.

The stochastic policy distribution $\tilde{\pi}_\lambda$ is available, so that the gradient $\nabla_\lambda \bar{R}_\lambda$ can be computed. The expected value is approximated by Monte Carlo techniques using the N_τ trajectories. The learning rate α has to be tuned manually in order for the control parameters λ to converge efficiently.

5.5.2 Simulation validation with baseline solution

To solve the finite-horizon problem, we use first the model-based approach i.e. the algorithm 1 to derive a baseline solution. The whole set of parameters is shown in Table II. We choose the discount factor γ as 1. For a horizon of 10s with a sampling time 0.05s, the number of the backward iteration is 200. To represent the finite-horizon return, the terminal cost is used firstly to compute the Q-function of the last time step, and then each stage is gradually added via the backward dynamic programming iterations. In total, 200 Q-functions are generated to represent a time-varying Q-function for a horizon of 10s. Moreover, we derive the policy from the obtained time-varying Q-function in the forward direction, by choosing the action which maximizes the Q-function of that step and apply it to the system.

Table II. PARAMETERS OF THE CONSIDERED HUMAN-WHEELCHAIR DYNAMICS

Meaning	Notation [units]	Value or domain
Sampling time	T_e [s]	0.05
Human parameters		
Recovery coefficient	$\mathcal{R}[\mathrm{s}^{-1}]$	0.0063
Fatigue coefficient	$\mathcal{F}$ $[\mathrm{s}^{-1}]$	0.153
MVC	M_{vc} [N]	100
Fraction of V_{max}	f	0.5
Human control gain	K_p	30
Wheelchair parameters		
Wheel radius	ς [m]	0.33
Maximum velocity	V_{max} [rad/s]	7
System matrix	$\mathbf{A}$	$\begin{bmatrix} 1 & 0.05 \\ 0 & 0.9406 \end{bmatrix}$
Input matrix	$\mathbf{B}$	$\begin{bmatrix} 0 \\ 0.0059 \end{bmatrix}$
Driving schedule configuration		

Finite horizon	K	200
Initial state of fatigue	S_{of_0}	0.5
Desired final human fatigue	S_{of-ref}	0.5
Distance-to-go	d_{ref} [rad]	20
State-space and action-space region		
Distance	d [rad]	[0,20]
Velocity	v [rad/s]	[0,7]
State of fatigue	S_{of}	[0.35,0.7]
Motor torque	U [Nm]	[0,50] ($U_{max} = 50$)

For the PG approach, an equidistant three dimensional $5 \times 5 \times 8$ grid is selected as the centers of the RBFs. In total, 200 RBFs ($M = 200$ and $\beta = 0.5$), together with a parameter vector $\theta \epsilon \mathcal{R}^{200}$ are used to approximate the controller (5.5.2). The learning rate α is chosen as 10^{-5}.

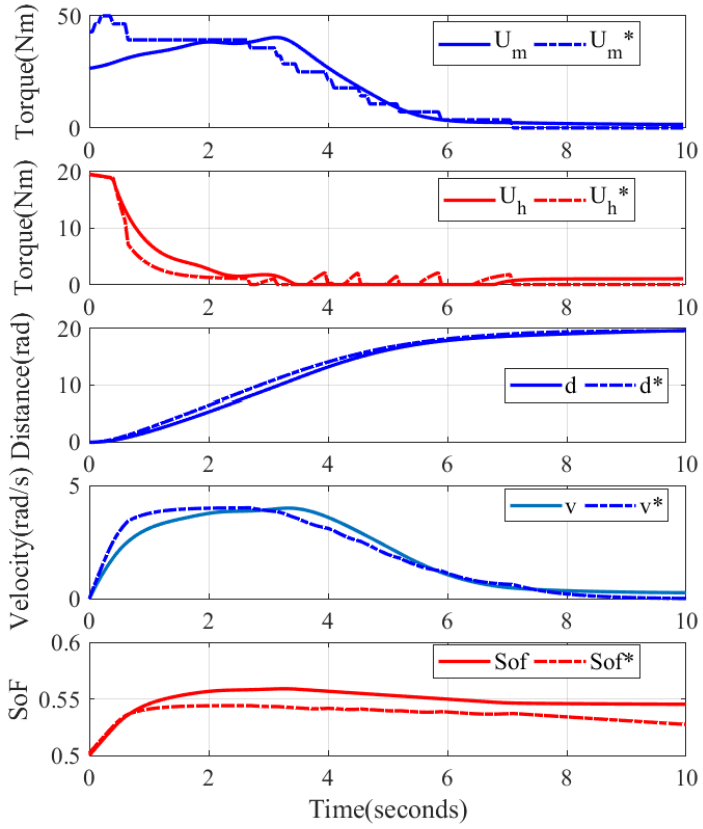

Figure 5.5.2. Simulation results provided by GPOMDP algorithm and ADP algorithm

A number of 8000 trajectories of 10s are performed to learn the control parameters θ. We compare the solution of PG with the solution obtained by the ADP. As shown in Figure 5.5.2,

PG approach (solid line) has a similar performance with ADP approach (dotted line). The simulation results are: the final return is $-8.20e4$ for PG (the energy consumption: $-5.39e4$ and the terminal penalty: $-2.81e4$). The final return is $-6.98e4$ for ADP (the energy consumption: $-6.4726e4$ and the terminal penalty: $-5.1108e3$). The PG approach provides 12.7% less return than ADP. However, the PG approach eliminates the need for a model by accepting this loss in return. It is important to mention this 12.7% difference includes both an electrical energy component and a difference in the final *Sof* reached by both methods.

From a practical point of view, first the "user" (of course it is an abuse of language as this is only simulation) cooperates with the motor to push the wheelchair. After reaching a suitable velocity between $1s$ and $3.5s$, the "user" reduces his applied force to reduce his fatigue. During this time, the electrical motor provides the main input to maintain this velocity. In the reminder of the driving, the motor assistance is reduced gradually to minimize the energy consumption. Moreover, the "user" tries to attain the desired final fatigue level by reducing his force. The system uses the kinetic energy given previously by the user and the motor to end the mission. During the driving task, the provided assistive algorithm tries to provide an energy-efficient assistance to the user so that his final fatigue level reaches the desired one.

For the model-free PG approach, we have a terminal error of 0.05 between the final $S_{of\,K}$ and the desired final value S_{of-ref} (0.02 for the ADP approach). This error can be reduced by increasing the weighting factor w_{e2}. However, the energy consumption should have a significant weight in the return function (34) to fulfill the optimization objective. The weight parameters w_{e1}, w_{e2} and w_{e3} must be tuned to have a tradeoff between reaching the terminal conditions and minimizing the energy consumption. The learning rate α tuning also depends on the weighting factors and parameters (β, c, M). Since no prior knowledge about the optimal policy is available, an equidistant grid on the given intervals is chosen for the centers of the RBFs. If we increase the number of RBFs, the approximate controller may tend to the optimal solution after receiving enough training. Roughly speaking, around 20-30 preliminary experiments are required to fix the 4 parameters and the RBFs used in this work.

As a considerable amount of data is needed to obtain a high performance controller, more PG learning techniques will be investigated to reduce the learning time in the next section. The ultimate objective is to develop an efficient real-time learning control of PAWs.

5.6 Applying PoWER to improve Data-Efficient

The energy optimization problem in the previous section requires a considerable amount of data to get a solution, which in practice would be impossible to obtain. Therefore, the main purpose of this section is to increase the data efficiency. To achieve this goal, we propose two ideas. The first one is to use a more efficient PG algorithm, namely PoWER. Secondly, as observed in (Feng et al. 2018) the operating region in the state space is concentrated on a few radial basis functions (RBFs); therefore, for the remaining RBFs the parameters remain constant or have a very small gradient. Reducing the parameters to the significant ones will accelerate the learning speed. Using Fuzzy Q-iteration as the baseline solution, we compare the performance of the two PG algorithms (PoWER and GPOMDP) with the new controller parameterizations and the one in previous section.

5.6.1 PoWER

To obtain a higher expected return, we may consider a new distribution $p_{\lambda'}(\tau)$ over trajectories that might provide a better expected return than the previous one i.e. $\int p_{\lambda'}(\tau)R(\tau)d\tau \geq \int p_{\lambda}(\tau)R(\tau)d\tau$. The new expected return $\int p_{\lambda'}(\tau)R(\tau)d\tau$ with parameters λ' is lower-bounded by a quantity $L_{\lambda}(\lambda')$ that depends on λ. The analytical expression of $L_{\lambda}(\lambda')$ (Kober and Peters 2009) is expressed as follows:

$$L_{\lambda}(\lambda') = -D\left(p_{\lambda}(\tau)R(\tau) \| p_{\lambda'}(\tau)\right) \tag{5.6.1}$$

The selection of λ' can be done by maximizing the lower bound $L_{\lambda}(\lambda')$ to implicitly maximize(5.5.4). In (Dayan and Hinton 1997), the authors show that maximizing $L_{\lambda}(\lambda')$ guarantees the improvement of the expected return. The intuition is that if $R(\tau_1) > R(\tau_2)$, the new λ' will put more probability mass on τ_1 than λ does.

PoWER (Policy learning by Weighting Exploration with the Returns) is one of gradient-free optimization criterion which works by maximizing the lower bound $L_{\lambda}(\lambda')$. Moreover, a deterministic policy is approximated by general basis functions ψ i.e. $\bar{\pi}_{\lambda}(x_k) = \lambda^T \psi(x_k, k)$. For exploration, Gaussian noise is added directly to the parameter vector λ. Using importance sampling (Neal 2001), the parameters λ are updated with the N_s trials which have the highest return among the performed trials. The formula to update the parameters is (Kober and Peters 2009):

$$\lambda_{l+1} = \lambda_l + \frac{\sum_{s=1}^{N_s} (\lambda_s - \lambda_l) R(\tau_s)}{\sum_{s=1}^{N_s} R(\tau_s)} \tag{5.6.2}$$

The whole method is given in Algorithm 3.

Algorithm 3. PoWER

1 Initialize λ_0

2 **for** $l = 0, 1, 2, \ldots \Gamma$

3 Generate a new trajectory τ of length K using λ_l

4 Sort the performed trials decreasingly by return

5 Select the N_s trials with the highest return

6 Update parameters $\lambda_{l+1} = \lambda_l + \frac{\sum_{s=1}^{N_s} (\lambda_s - \lambda_l) R(\tau_s)}{\sum_{s=1}^{N_s} R(\tau_s)}$

7 **end for**

The exploration is carried out in the parameter space as previously explained. The zero mean Gaussian noise vector z_P with the standard deviation σ_P is added to the parameters and renders the action stochastic as follows:

$$\text{sat}\left(0, U_{max}, \left(\lambda_l^P + z_P\right)^T \varphi(x_k)\right) \tag{5.6.3}$$

where the stochastic motor torque is saturated between 0 and U_{max} and the parameter vector λ_l^P is updated by (5.6.2). By tuning the parameters (β, c, M) of the basis functions (5.5.2), the standard deviation σ_P and we have all the conditions to update the parameters λ^P.

5.6.2 Learning time comparison between GPOMDP and PoWER

In this section, simulations are carried out to compare the proposed methods. The whole set of parameters is shown in Table II. The human model parameters are adapted from (Fayazi et al. 2013) to have a reasonable fatigue and recovery rate to avoid a trivial optimal solution. The control strategy is approximated over the state-space and action-space region given in Table III. The configurations and learning parameters of the return function, penalty function, model-based policy, and model-free policies are shown in the following Table III.

Table III. RETURN FUNCTION, PENALTY FUNCTION, MODEL-BASED POLICY, MODEL-FREE POLICIES CONFIGURATIONS, AND LEARNING PARAMETERS

Return function and penalty function configuration	
Reward weight matrix $[w_{e1} \quad w_{e2}]$	$[4000 \quad 10^7]$
Penalty weight w_{e3}	800
Q-function approximation	
Centers of triangular functions ϕ distributed on an equidistant grid	$10 \times 10 \times 41$ over the state-space $\left(x = [d, v, S_{of}]^T\right)$
Number of equidistant discrete actions	15
Radial basis functions (5.5.2) configuration 1	
Radial parameter β	0.5
Centers of RBFs distributed on an equidistant grid	$5 \times 5 \times 8$
Total number of RBFs M	200
Radial basis functions (5.5.2) configuration 2	
Radial parameter β	0.5
Centers of RBFs distributed on an equidistant grid	$5 \times 5 \times 1$
Total number of RBFs M	25
GPOMDP parameters	
Learning rate α	10^{-5}
Standard deviation σ_G	5
PoWER parameters	
Importance sampling N_s	10
Standard deviation σ_P	1

The number in the legend gives the total parameters of the controller approximation (5.5.2) for each simulation. A mean value along with a 95% confidence interval calculated for 10 independent simulations is given (each simulation with 400 trials). Figure 5.6.1 shows that with the same policy parametrization, PoWER has a considerably higher data efficiency than GPOMDP. GPOMDP-25 and GPOMDP-200 give a similar final performance. Considering the mean, 90% of the baseline return is provided in around 100 trials by PoWER-25. The same performance is given in around 200 trials by PoWER-200. Overall, PoWER-25 is the best choice among the considered configurations.

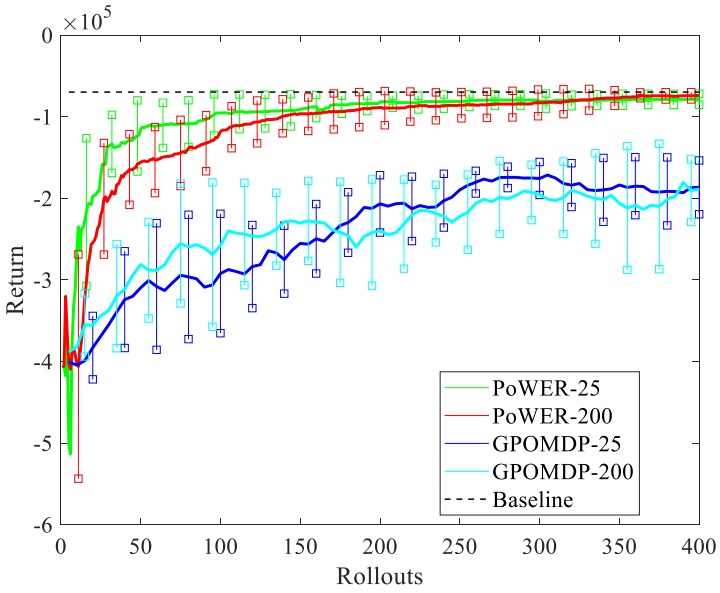

Figure 5.6.1. The mean performance and 95% confidence interval on the mean value of PoWER with 25 control parameters (PoWER-25), PoWER with 200 (PoWER-200), GPOMDP with 25 control parameters (GPOMDP-25) and GPOMDP with 200 (GPOMDP-200)

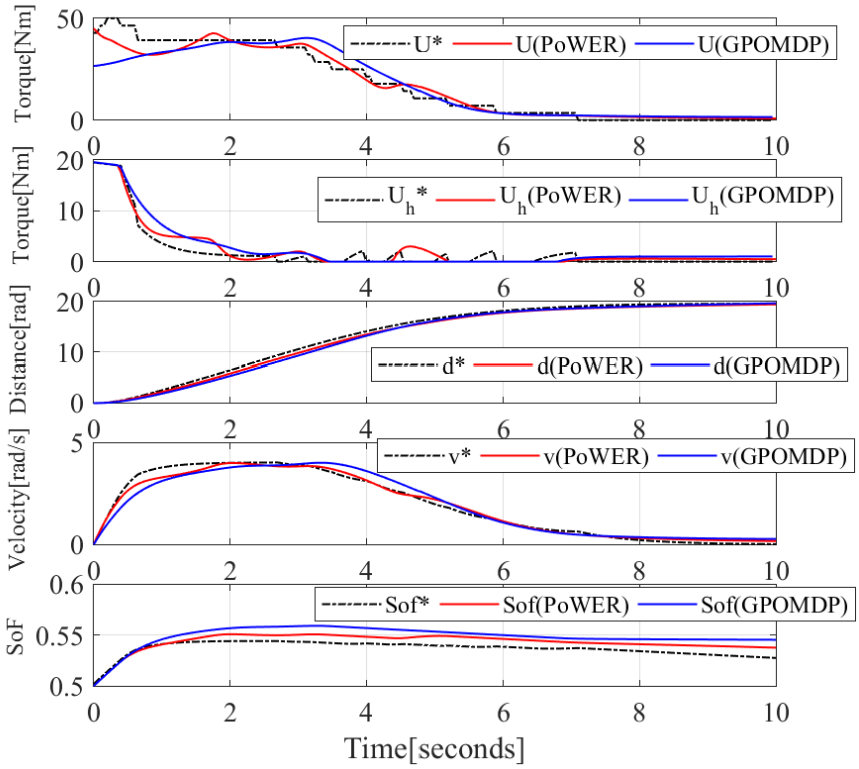

Figure 5.6.2. Controlled trajectories provided by GPOMDP and PoWER and fuzzy Q-iteration algorithm

For the next simulations, we focus on the final near-optimal behaviours provided by PoWER-25 and GPOMDP-25. To this end, 400 trials and 8000 trials are performed to learn the parameter vectors λ^P and λ^G, respectively. The slow learning speed of GPOMDP is mainly due to the exploration noise added directly to actions at every step. This type of exploration strategy can cause a high variance for learning algorithm (Kober et al. 2009) and leads to a poor performance in terms of data-efficiency. As shown in Figure 5.6.2, the first solution of the model-free methods PoWER (red solid line) and GPOMDP (blue solid line) are comparable to the model-based fuzzy Q-iteration (black dotted line baseline solution). PoWER-25 has the fastest convergence among other approach and other configuration. The final return is -82017, -70242, and -69837 for GPOMDP, PoWER and fuzzy Q-iteration respectively. Here again, PoWER delivers a better solution than GPOMDP in terms of final return.

The simulation was done on an Intel Core i7-6500 CPU @ 2.50GHz. The average elapsed CPU time to compute a control action is $1.0 * 10^{-4}$s, $7.6 * 10^{-4}$s, $1.5 * 10^{-4}$s, and $7.6 * 10^{-4}$s respectively for PoWER-25, PoWER-200 GPOMDP-25 and GPOMDP-200. As their elapsed CPU time is significantly less than the sampling time of 0.05s, it is possible to embed them into a real PAW.

5.6.3 Adaptability to different human fatigue dynamics

In this section, we turn our focus towards adaptation to human fatigue variability, which is crucial for a personalized PAW. In what follows, we investigate only the adaptability of PoWER-25 to these changes, since it provided the best results in the previous section. The objective of this investigation is to confirm the possibility of having a generic solution for different human fatigue dynamics. To represent various human fatigue dynamics, we change the parameters of the human fatigue (5.2.1) as follows:

$$\mathrm{F}' = \frac{1}{\boldsymbol{\eta}}\mathrm{F}\,; \qquad \mathrm{R}' = \boldsymbol{\eta}\mathrm{R}\,; \qquad M_{vc}^{'} = \boldsymbol{\eta} M_{vc}$$

where $\mathcal{F}$, $\mathcal{R}$, M_{vc} are the nominal parameters used in Table II. A value $\boldsymbol{\eta} > 1$ corresponds to a user physically stronger than the nominal one, because they get exhausted slower, recover faster and have more Maximum Voluntary Contraction force. On the contrary, $\boldsymbol{\eta} < 1$ corresponds to a physically weaker user. Adaptation starts from the parameters found using the nominal model. As a baseline, we compare this adaptation procedure with simply

resetting the parameters to zero values when the model changes. The same variance σ_P of Section 5.6.2 is applied for exploration. Both stronger $(\boldsymbol{\eta}=2)$ and weaker $(\boldsymbol{\eta}=1/2)$ users are studied. Figure 5.6.3 shows that PoWER is clearly much more efficient, when initialized with the nominal model, being able to provide a good return directly and to find a new near-optimal solution for the new fatigue dynamics in less than 50 trials.

In order to confirm that the assistive control can adapt to a bigger range of parameter changes, we carry out the same comparison for $\eta = 8, 4, 3,\ 1/3,\ 1/4,\ 1/8$. Table III gives the baseline return for each η, the minimal return for each case and the number of trials to converge to 90% of the corresponding baseline return for both initializations. The asterisk * represents situations where the learning algorithm fails to converge to the 90% of the baseline return within 400 trials.

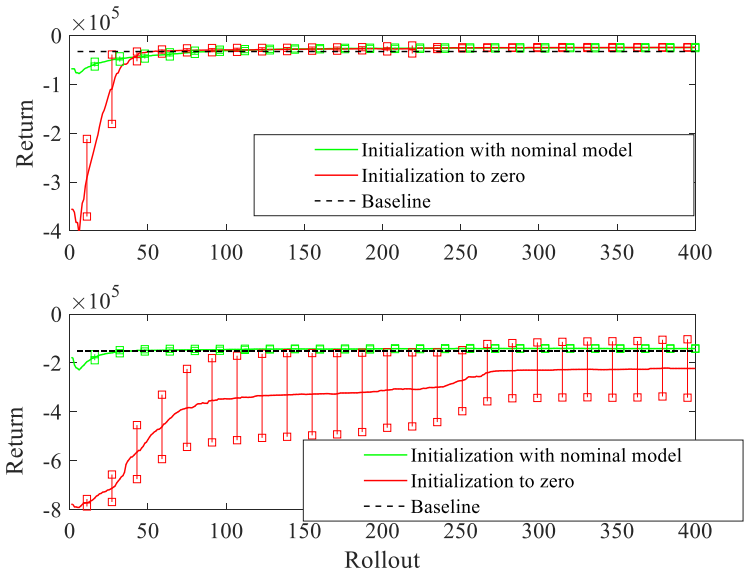

Figure 5.6.3. The mean performance of PoWER for both initialization (Top: $\boldsymbol{\eta}=2$ and bottom $\boldsymbol{\eta}=1/2$)

Table IV shows that both initializations have similar convergence for $\boldsymbol{\eta}=8$. For $\boldsymbol{\eta}=3$, the initialization to zero has a faster convergence. This result may be because that the initialization to zero is closer to the optimal solution. Nevertheless, for all the other $\boldsymbol{\eta}$, the initialization with the nominal model converges faster. Overall, starting learning with the nominal solution can guarantee a higher minimum return. Moreover, PoWER with prior knowledge adapts reasonably well to human fatigue dynamic changes without tuning again

the learning parameter σ. This study therefore confirms the possibility of providing an adaptive solution for different human fatigue dynamics.

Table IV. PoWER WITH VARYING FATIGUE MODEL (ZERO: INITIALIZATION TO ZERO, NOMINAL: INITIALIZATION WITH THE NOMINAL MODEL. THE MINIMAL RETURN IS NORMALIZED BY THE CORRESPONDING BASELINE RETURN)

η	Baseline return (fuzzy Q-iteration)	PoWER			
		Minimal return		Number of trials	
		Nominal	Zero	Nominal	Zero
8	-361950	1.25	2.30	39	37
4	-96723	1.76	4.96	33	56
3	-54018	2.14	7.58	47	34
2	-32744	2.38	12.43	48	65
1/2	-150920	1.51	5.25	10	*
1/3	-207400	1.86	5.52	198	*
1/4	-299540	1.76	4.50	37	*
1/8	-657620	1.56	2.73	30	*

5.6.4 Experimental Validation

To demonstrate the effectiveness of the proposed learning algorithm, proof-of-concept experiments have been conducted on the PAW prototype. Via a joystick, the user can return a subjective evaluation of his/her S_{of} to the control algorithm. When the user pushes the joystick to the negative or positive Y-direction, the joystick returns to the algorithm a discrete value -1 or 1, respectively. The neutral position of the joystick returns a discrete value 0. These three discrete values $-1, 0$ and 1 mean respectively that the user feels too tired, is comfortable, and feels insufficiently tired (is willing to exercise more). The discrete signal is filtered so that when it changes between two levels (among -1, 0 and 1), its filtered version I provides a gradual transition between these levels. Furthermore, to avoid the need for too many pushes of the joystick, after such a transition the filtered signal is kept nearly constant for a certain duration.

The driving scenario consists in riding on a straight flat road with a given reference velocity v_{ref} set by the user. The velocity estimated from the position encoders is available via the computer connected to the data acquisition system. The control objective is to minimize both the electrical energy and the use of the joystick, while tracking the reference velocity. Therefore, the stage reward function is:

$$r_k = -w_{e1}\left(v_k - v_{ref\,k}\right)^2 - w_{e2}I_k^2 - w_{e3}U_k^2 \tag{5.6.4}$$

where v_{ref_k} is the given reference velocity at sample k. The reward weights are $w_{e1} = 10$, $w_{e2} = 0.25$ and $w_{e3} = 0.05$. Note that any joystick signal $I \neq 0$ is penalized. The controller is configured as a PI-type law:

$$U_k = \lambda_1\left(v_k - v_{ref\,k}\right) + \lambda_2\sum_{\mathrm{i}=0}^{k}\left(v_{\mathrm{i}} - v_{ref\,\mathrm{i}}\right) + \lambda_3 I_k + \lambda_4\sum_{\mathrm{i}=0}^{k} I_{\mathrm{i}} - \lambda_5 F_{hk}\varsigma \tag{5.6.5}$$

The first four terms of the controller (5.6.5) are used to track the reference v_{ref} while keeping the filtered joystick signal I to 0. The term $\lambda_5 F_{hk}\varsigma$ is for compensating the human input.

One healthy male volunteer (29-year-old) performed the proof-of-concept experiments. There are 5-minute rest periods between consecutive trials. In total, 24 trials with the same driving condition have been carried out on the same day to learn the parameter vector λ in (5.6.4). Figure 5.6.4 shows the total return of each trial. Among the 24 trials, 3 trials went unstable at the beginning of learning. For these trials, the velocity is oscillating around the set-point and the amplitude of oscillation is increasing. Therefore, the user stopped immediately the wheelchair and a very low return was given to the learning algorithm to avoid such situations in the future. The return tends to increase gradually after performing these trials. We notice that the obtained curve of return is noisy. Due to the time-consuming nature of the experiment, it is not feasible to perform many trials to obtain a smooth mean return.

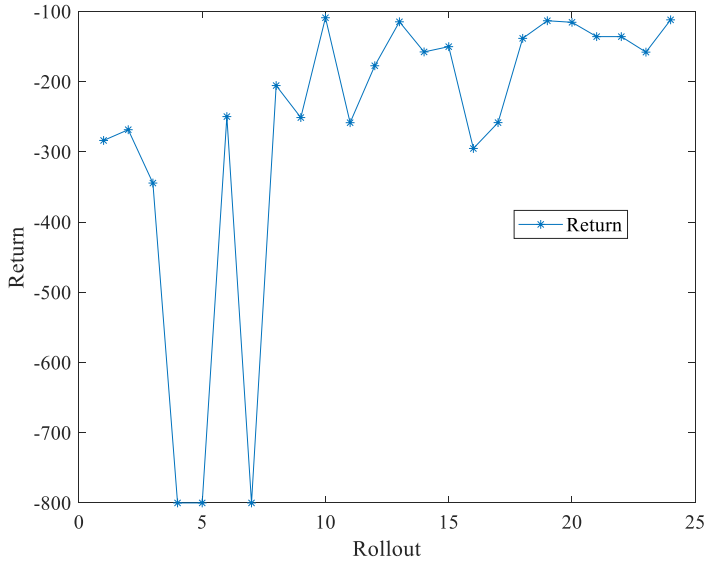

Figure 5.6.4. The total return of each trial

Figure 5.6.5 shows the trajectories of the first four stable trials and the last four trials. The motor torques are normalized between -1 and 1. The values 1 and -1 represent respectively the maximum torque in the position direction and the maximum torque in the negative direction. We remark that the user does not push the joystick anymore in the last four trials. The joystick signal *I* sums up the influence of main physiological and psychological factors to tell the learning algorithm what assistive torque is suitable to users. The fact that the user does not use the joystick at the end means that after training, the provided assistive torques are acceptable in terms of the sensation of fatigue. Another consequence of training is that the user and the controller track together the given velocity more smoothly.

Through these proof-of-concept experiments, we conclude that the proposed learning algorithm PoWER is able to improve the performance of the controller (5.6.4). For a final commercial product, there will be a certain accommodation time to obtain a satisfactory performance, during which a health professional would help the user interact with the PAW.

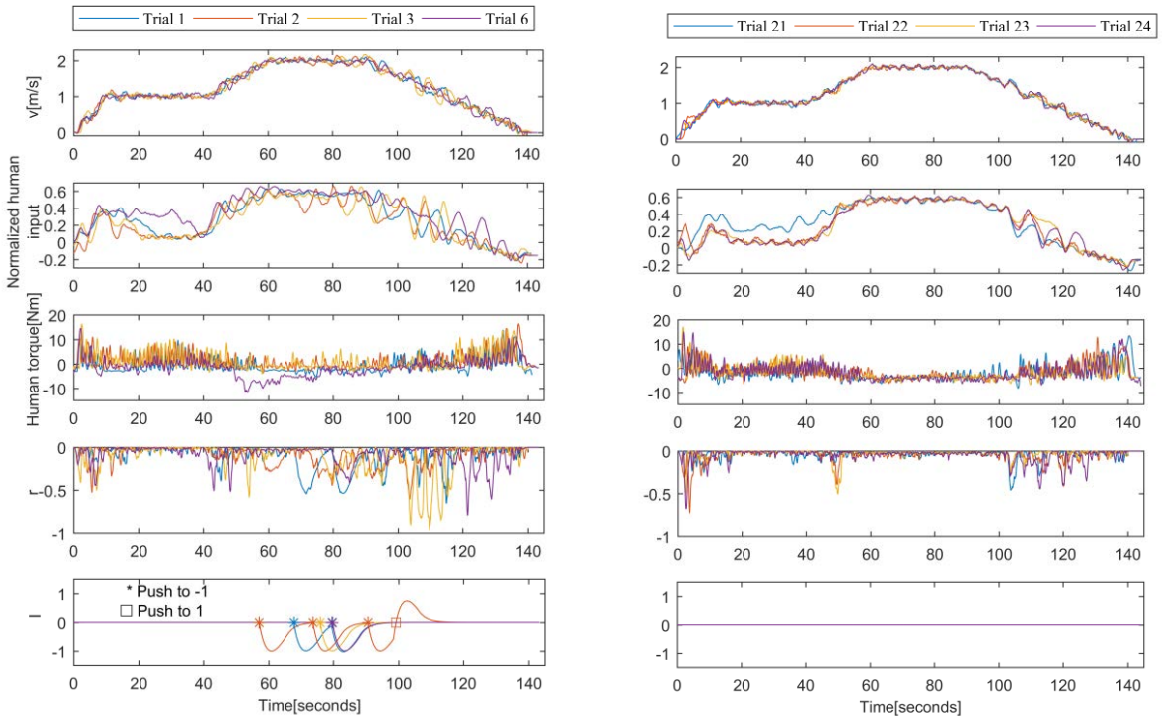

Figure 5.6.5. The trajectories of the first four stable trials and the last four trial. (The instant where the joystick is pushed is indicated on the *I* signal)

5.7 Summary

In this chapter, a novel PAW control design has been proposed for paraplegic wheelchair users. The assistive strategy is based on energy optimization, while maintaining a suitable fatigue level for users and using minimal electrical energy over a distance-to-go. This optimal control problem was solved by the online model-free reinforcement learning methods PoWER and GPOMDP. Their near-optimality was confirmed by the model-based approach finite-horizon Q-fuzzy iteration. An important contribution is that the near-optimality of finite-horizon Q- uzzy iteration was proven. In addition, simulation results confirmed that PoWER with a simplified controller parameterization provides a considerably higher data efficiency, which renders the model-free framework better applicable in practice. Moreover, an investigation has been done to illustrate that PoWER is also able to adapt to human fatigue dynamic changes. Finally, a proof-of-concept experiment has been carried out to demonstrate the feasibility of the approach in practice.

Based on the proof-of-concept study of this chapter, next chapter gives the conclusion of the thesis and proposes an idea to integrate the model-free approach into the model-based assistive control.

Chapter 6. Conclusion and future works

6.1 Conclusion

The work presented in this thesis tried to propose solutions for the assistance of Power-Assisted Wheelchair (PAW) with a minimum of sensors for the larger possible population of disabled persons. The goals were twofold: reducing the hardware cost to render the assistance kit as affordable as possible and rendering the assistance adaptable (as transparent as possible) to this highly heterogeneous population.

A “pure” classical approach of automatic control via an exhaustive modelling of both the wheelchair and the human was therefore prohibited. Effectively, not only heterogeneity would have been an important issue, but also feeding the model parameters would have been impossible case-by-case. Thus, two completely, not opposite, but different ways were explored during this work. The first one was to take profit of the best possibilities of performance and robust control techniques based on a mechanical model of the wheelchair. Results obtained were beyond our initial expectations, especially because we were able to do the assistance design without needing the main corrupted variable which is the user’s amplitude of propelling. The second way was to explore the possibilities of learning techniques applied without model for the assistive control. It was done keeping in mind, as people from automatic control community, that issues about proofs of convergence were important.

In order to highlight the current disability issues, Chapter 1 presented first the economic and social context and their corresponding challenges, such as high cost of assistive devices and heterogeneous disabled population. In such context, the chapter 2 provided a literature review of mechanical models of the wheelchair, model-based control and model-free control free techniques. This chapter were useful for the model-based PAW assistance design presented in Chapter 3 and Chapter 4 and for the model-free PAW assistance design of Chapter 5. The main contributions of this work are resumed as follows:

An innovative model-based assistive control has been proposed for a Power-Assisted Wheelchair. Using an unknown input observer (“software” or “virtual” sensor), human torques sensors are not required anymore and the cost of the assistive device is reduced

(Mohamad et al. 2015). Thanks to the reconstructed human torques, an algorithm is provided that allows defining reference trajectories for the center and yaw velocities. In addition, actuator saturation and uncertainties (user's mass and road conditions) were taken into account to design a robust observer-based tracking controller. The stability analysis of the complete closed-loop system was possible using an LMI constraints formalism under a two-steps algorithm. The effectiveness of the whole assistive control was confirmed via both simulations and experimental real-time tests.

A model-free assistance was designed for the PAW application. A case study illustrates the possibility to adapt the heterogonous disabled population (such as different human fatigue dynamics) using learning algorithms. To verify numerically the optimality of the model-free design, we used a model-based approach, such as finite-horizon fuzzy Q iteration, to derive a based-line solution. The (near-)optimality and the consistency of the finite-horizon fuzzy Q iteration were proved analytically. Moreover, a proof-of-concept experience was performed for validating the model-free design.

Based on the above design experiences, we finally proposed an idea to combine the model-based control and the model-free control to design a kind of personalized assistance of a PAW. This future direction for research seems promising as it will combine the advantages of both fields. References parameters would be adapted from learning techniques with guarantees of convergence and a high level of confidence, whereas tracking of the references would be ensured by the robust observer-based tracking controller. we give more perspectives in the next section.

6.2 Control-learning framework proposal and future works

In the previous three chapters, model-based control and model-free control solve separately two main problems of the PAW application. Based on a model-based design, an assistive control for PAW applications has been presented in Chapter 3 and validated with experimental results in. Adaptability to unknown human fatigue dynamic has been achieved by the model-free approach in Chapter 5. With the design experiences obtained in this thesis, we propose an innovative idea to combine control and learning for constructing an intelligent PAW. Furthermore, some theoretical perspectives are given in this chapter.

Results obtained in Chapter 5 show the proposed model-free approach is able to improve the assistive control after training. However, the wheelchair is modelled as one dimensional and goes only straight. For a practical PAW application, the rotation of the wheelchair has to be considered. For such a consideration, the state vector consists of two states e.g. center velocity and yaw velocity. The control inputs are the right and the left motor torques. Since both the number of states and the number of control inputs increase, the number of the control parameters becomes important. Thus, the time for learning a satisfying control may also increase considerably. In addition, torque sensors are needed to compute the control action.

From the results of Chapter 5, we can conclude that modelling the human-wheelchair system as a black-box may not be the best solution. Instead going from black-box to a grey-box seems a promising way. The prior knowledge of the human-wheelchair system has to be exploited. The simplified mechanical dynamic of the wheelchair is known in general. In order to remove torque sensors, a sufficiently precise model is first used to estimate human torques. To this end, an unknown input observer is designed. The simulation results in Section 3.2 and the experimental results in Section 4.1 confirm that a satisfying estimation performance is obtained.

Despite of a satisfying performance provided by the model-based assistive control, the reference generation may not be optimal respect to a particular user. To give an obvious example, we analysis the braking performances according to the two different weight users of Chapter 3. During braking, the decreasing rate of the center velocity for a heavy user should be smaller than the one for a light user. The reason is that the wheelchair with a heavier user needs more braking distance to disperse an important kinetic energy. Such a longer braking distance can be obtained by a smaller decreasing rate of the center velocity.

We show braking scenarios of both users in Figure 6.2.1 and Figure 6.2.2. These sequences are extracted from the experimental validations of Chapter 4. With these two example, we explain that a same constant decreasing rate of the center velocity for different users may not be optimal to obtain a personalized braking.

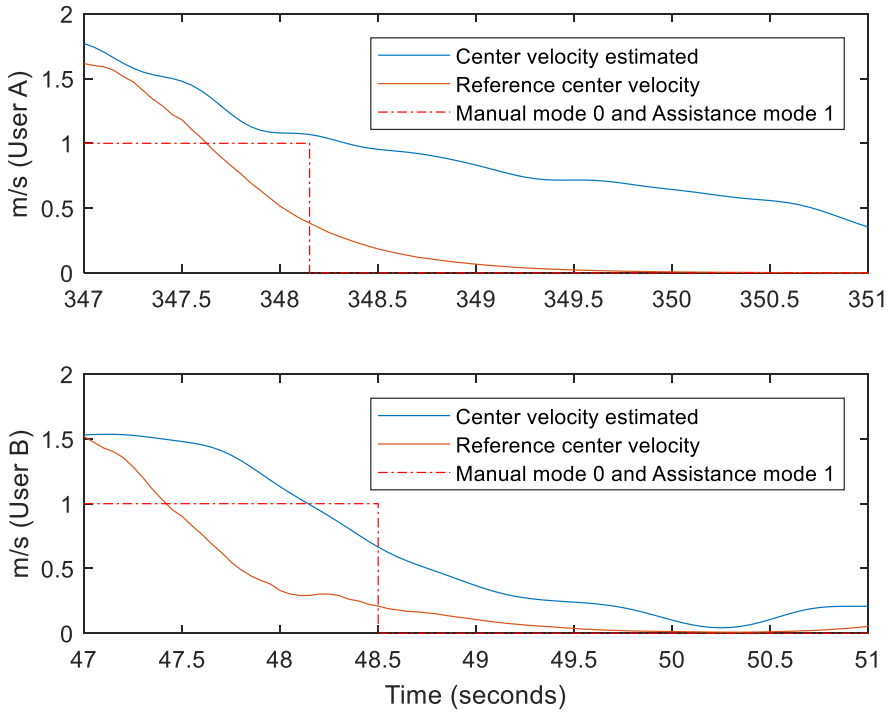

Figure 6.2.1. Center velocity and center velocity reference during braking for user A and user B (experimental results)

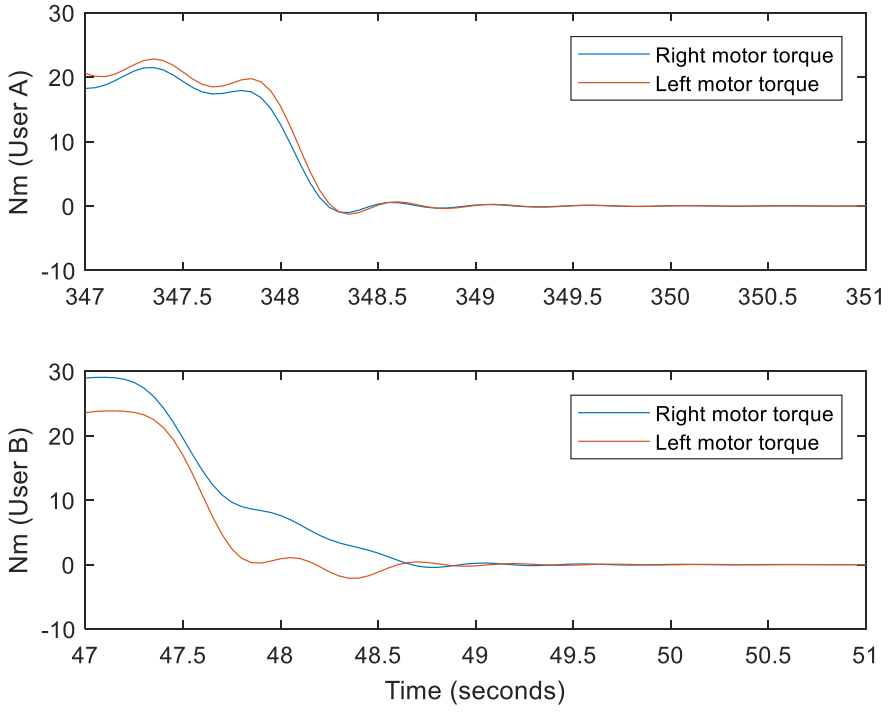

Figure 6.2.2. Motor torques during braking for user A and user B (experimental results)

Figure 6.2.1 provides the center velocity, the reference center velocity and the operating mode. Both trials have a similar initial velocity before braking. Moreover, for both trials the reference center velocity computed by the algorithm for the braking are similar, since the assistive control has the same parameters for both users. To follow such a reference, the assistive system slows down the wheelchair by reducing the assistance in assistance mode, as shown in Figure 6.2.2. Then, the wheelchair is braked by the friction in manual mode. Since user A is light and user B is heavy, more braking torque is needed for user B to stop the wheelchair than for user A. To this end, the assistive system reduces more importantly and more quickly the assistive torque for user B than for user A, see in Figure 6.2.2.

However, this quick change of assistive torque could make the user feeling uncomfortable and unsafe. From the feedback of user B, the assistive system brakes too strongly and he feels uncomfortable with it. Nonetheless, the assistive system provides a braking such that the lighter user feels comfortable and safe. Therefore, the decreasing rate η of the center velocity should to be adapted according to the user.

In addition, different users may perform different pushing frequency to achieve a same desired center velocity. For example, since user B is physically strong, his propelling is more high amplitude and low frequency, whereas for user A it is medium amplitude and frequency. Of course, some level of pathology and/or weak disabled person will end with low amplitude and high frequency. As shown in Figure 6.2.3, the center velocity under the propulsion of user B is higher than the estimated reference. Therefore, the assistive system brakes constantly the wheelchair see Figure 6.2.4. To assist better the user, the reference generation function should provide a higher reference center velocity with a same pushing frequency for user B. Thus, having an adaptive δ according to the user would be profitable.

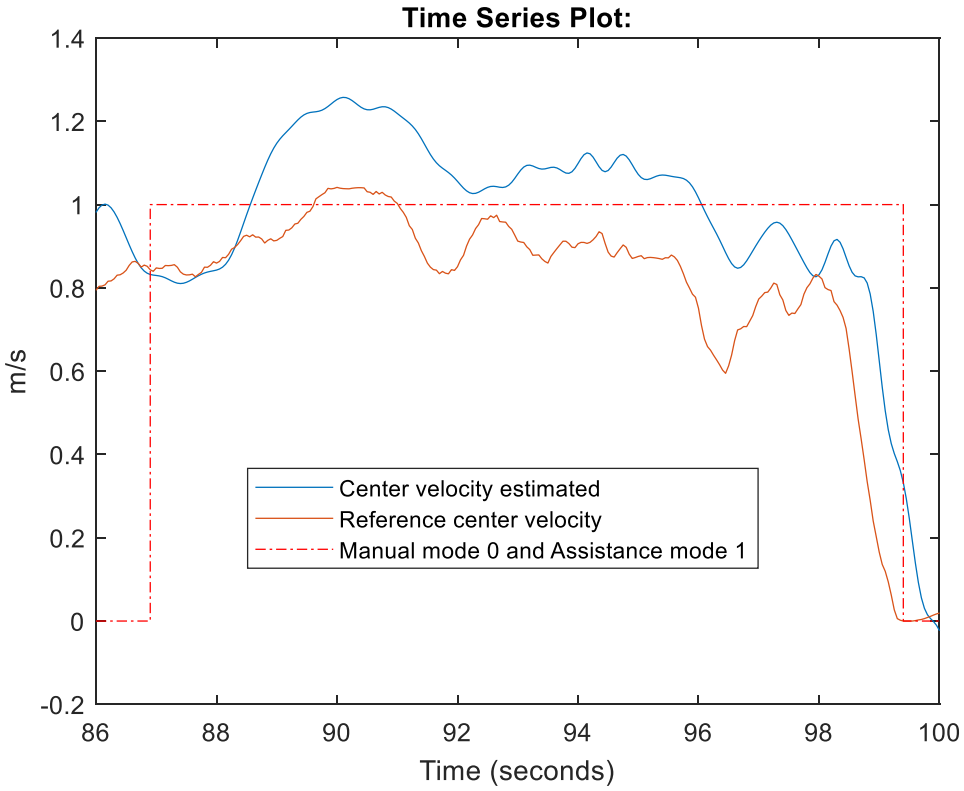

Figure 6.2.3. Reference estimation with an inappropriate parameter δ

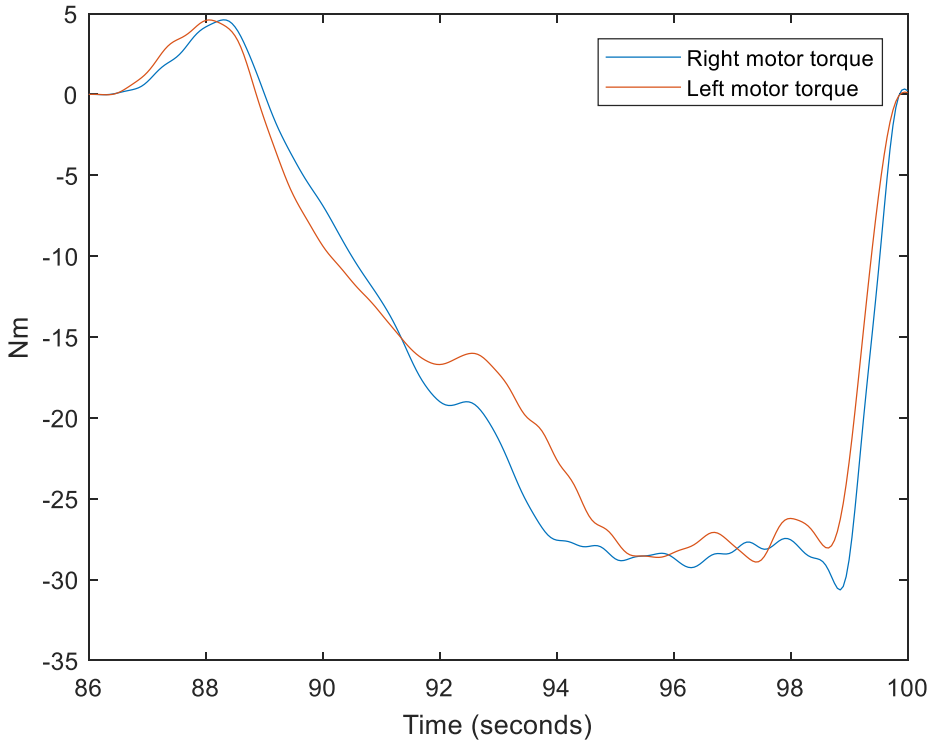

Figure 6.2.4. Assistive torque with an inappropriate parameter δ

Of course, this illustrative example could cover more important issues, such has, braking acceptability for some disabled person pathologies. The same kind issues can appear for acceleration, turning and so on. Therefore, a more "personalized" assistance, especially through trajectory generation has to be thought for the future. This personalized assistance could be the right place for learning.

In this context, we propose the idea shown in Figure 6.2.5 to integrate an adaptability to the proposed model-based design. In such a framework, the functionalities of the model-based control are to ensure the reference tracking and the stability of the system wheelchair + human. The functionalities of the reference generation are to produce the references based on an estimate of the human intention from the measurements. The quality of the estimated reference signals depend partly on the parameter vector $\begin{bmatrix}\zeta & \delta & \eta\end{bmatrix}^T$. With the help of a feedback from the human (for example via a button), the learning algorithm could produce and adapt a (near-)optimal parameter vector $\begin{bmatrix}\zeta* & \delta* & \eta*\end{bmatrix}^T$ and generate a (near-)optimal reference signal for a particular user.

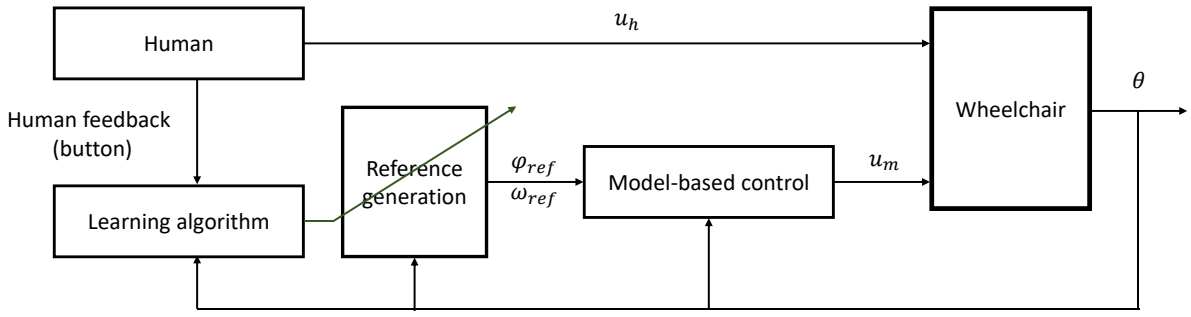

Figure 6.2.5. Control-Learning framework proposal for PAW designs

If the learning function is removed in the framework of Figure 6.2.5, the assistive control can still provide the performance obtained in Chapter 3 and in Chapter 4. Based on this performance of the model-based design, this framework is expected to improve the assistive control and provide a satisfying performance at the very beginning of learning.

Furthermore, the proposed idea exploits the prior knowledge of the human-wheelchair system. Based on the model-based design and the structure of the reference generation, the learning algorithm has only the parameter vector $\begin{bmatrix}\zeta & \delta & \eta\end{bmatrix}^T$ to learn. The objective is to finding a (near-)optimal solution with few data.

To extract meaningful feature of human behaviours from the limited measurement, the learning algorithm may need long-term trials for an adaptive strategy. Therefore, in the proposed framework, the learning control is modelled as a high level control which collects enough information to update the parameter vector $\begin{bmatrix} \zeta & \delta & \eta \end{bmatrix}^T$. The frequency of the parameter updates would be an important issue. Considering usual trips, long trips, user's state of fatigue, the frequency of update should be in the range of some hours or every day or every few days.

These three parameters are just given as an illustration of the global idea. Of course, a deeper research has to be done to determine the principal variables to adapt in order to gain a high level of drivability and fulfil the comfort requirements of final users. The way they have to return their feedback is also an important issue. The assistive control has to be as natural as possible, in order neither to increase their workload, nor to make them feel this task uncomfortable. At last, a critical issue would be to ensure that the low level model-based control can cope safely and robustly to the changes of reference. Moreover, the levels of safety and security have to be kept at a very high level. Therefore, some theoretical aspects have to be considered during the switching sequences of modification of the parameters. It is certainly challenging to combine proofs of robustness and convergence issues of the learning in a global framework.

Bibliography

Algood, S. David et al. 2004. "Impact of a Pushrim-Activated Power-Assisted Wheelchair on the Metabolic Demands, Stroke Frequency, and Range of Motion among Subjects with Tetraplegia." *Archives of physical medicine and rehabilitation* 85(11): 1865–1871.

Aula, A., S. Ahmad, and R. Akmeliawati. 2015. "PSO-Based State Feedback Regulator for Stabilizing a Two-Wheeled Wheelchair in Balancing Mode." In *2015 10th Asian Control Conference (ASCC)*, , 1–5.

BA, Brian Woods et al. 2003. "A Short History of Powered Wheelchairs." *Assistive Technology* 15(2): 164–80.

Baxter, J., and P. L. Bartlett. 2000. "Direct Gradient-Based Reinforcement Learning." In *2000 IEEE International Symposium on Circuits and Systems. Emerging Technologies for the 21st Century. Proceedings (IEEE Cat No.00CH36353)*, , 271–74 vol.3.

Bellman, Richard. 1966. "Dynamic Programming." *Science* 153(3731): 34–37.

Bennani, C. et al. 2017. "A Modified Two-Step LMI Method to Design Observer-Based Controller for Linear Discrete-Time Systems with Parameter Uncertainties." In *2017 6th International Conference on Systems and Control (ICSC)*, , 279–84.

Bertsekas, Dimitri P., Dimitri P. Bertsekas, Dimitri P. Bertsekas, and Dimitri P. Bertsekas. 1995. 1 *Dynamic Programming and Optimal Control*. Athena scientific Belmont, MA.

Blandeau, M. et al. 2018. "Fuzzy Unknown Input Observer for Understanding Sitting Control of Persons Living with Spinal Cord Injury." *Engineering Applications of Artificial Intelligence* 67: 381–89.

Boninger, Michael L. et al. 2000. "Manual Wheelchair Pushrim Biomechanics and Axle Position." *Archives of Physical Medicine and Rehabilitation* 81(5): 608–13.

Boyan, Justin A. 2002. "Technical Update: Least-Squares Temporal Difference Learning." *Machine Learning* 49(2): 233–46.

BOYD, S. 1994. "Linear Matrix Inequalities in System and Control Theory." *SIAM*. https://ci.nii.ac.jp/naid/10000022326/ (April 30, 2019).

Bradtke, Steven J., and Andrew G. Barto. 1996. "Linear Least-Squares Algorithms for Temporal Difference Learning." *Machine learning* 22(1–3): 33–57.

Buşoniu, L., D. Ernst, B. De Schutter, and R. Babuška. 2010. "Online Least-Squares Policy Iteration for Reinforcement Learning Control." In *Proceedings of the 2010 American Control Conference*, , 486–91.

Buşoniu, Lucian et al. 2018. “Reinforcement Learning for Control: Performance, Stability, and Deep Approximators.” *Annual Reviews in Control* 46: 8–28.

Busoniu, Lucian, Robert Babuska, Bart De Schutter, and Damien Ernst. 2010. *Reinforcement Learning and Dynamic Programming Using Function Approximators*. 0 ed. CRC Press. https://www.taylorfrancis.com/books/9781439821091 (May 3, 2019).

Buşoniu, Lucian, Damien Ernst, Bart De Schutter, and Robert Babuška. 2010. “Approximate Dynamic Programming with a Fuzzy Parameterization.” *Automatica* 46(5): 804–14.

Chadli, M., and T. M. Guerra. 2012. “LMI Solution for Robust Static Output Feedback Control of Discrete Takagi–Sugeno Fuzzy Models.” *IEEE Transactions on Fuzzy Systems* 20(6): 1160–65.

Chadli, M., and H. R. Karimi. 2013. “Robust Observer Design for Unknown Inputs Takagi–Sugeno Models.” *IEEE Transactions on Fuzzy Systems* 21(1): 158–64.

Chen, W., J. Yang, L. Guo, and S. Li. 2016. “Disturbance-Observer-Based Control and Related Methods—An Overview.” *IEEE Transactions on Industrial Electronics* 63(2): 1083–95.

Chibani, Ali, Mohammed Chadli, and Naceur Benhadj Braiek. 2016. “A Sum of Squares Approach for Polynomial Fuzzy Observer Design for Polynomial Fuzzy Systems with Unknown Inputs.” *International Journal of Control, Automation and Systems* 14(1): 323–30.

Choi, Jong-Woo, and Sang-Cheol Lee. 2009. “Antiwindup Strategy for PI-Type Speed Controller.” *IEEE Transactions on Industrial Electronics* 56(6): 2039–2046.

Cooper, R. A. et al. 2002. “Performance Assessment of a Pushrim-Activated Power-Assisted Wheelchair Control System.” *IEEE Transactions on Control Systems Technology* 10(1): 121–26.

Cooper, Rory A. et al. 2001. “Evaluation of a Pushrim-Activated, Power-Assisted Wheelchair.” *Archives of Physical Medicine and Rehabilitation* 82(5): 702–708.

Corno, M., D. Berretta, P. Spagnol, and S. M. Savaresi. 2016. “Design, Control, and Validation of a Charge-Sustaining Parallel Hybrid Bicycle.” *IEEE Transactions on Control Systems Technology* 24(3): 817–29.

Darouach, M., M. Zasadzinski, and S. J. Xu. 1994. “Full-Order Observers for Linear Systems with Unknown Inputs.” *IEEE Transactions on Automatic Control* 39(3): 606–9.

Dayan, Peter, and Geoffrey E. Hinton. 1997. “Using Expectation-Maximization for Reinforcement Learning.” *Neural Computation* 9(2): 271–78.

De La Cruz, Celso, Teodiano Freire Bastos, and Ricardo Carelli. 2011. “Adaptive Motion Control Law of a Robotic Wheelchair.” *Control Engineering Practice* 19(2): 113–25.

Delrot, Sabrina, Thierry Marie Guerra, Michel Dambrine, and François Delmotte. 2012. “Fouling Detection in a Heat Exchanger by Observer of Takagi–Sugeno Type for

Systems with Unknown Polynomial Inputs." *Engineering Applications of Artificial Intelligence* 25(8): 1558–1566.

Ding, B. 2010. "Homogeneous Polynomially Nonquadratic Stabilization of Discrete-Time Takagi–Sugeno Systems via Nonparallel Distributed Compensation Law." *IEEE Transactions on Fuzzy Systems* 18(5): 994–1000.

Ding, D., and R. A. Cooper. 2005. "Electric Powered Wheelchairs." *IEEE Control Systems Magazine* 25(2): 22–34.

Duan, Yan et al. 2016. "Benchmarking Deep Reinforcement Learning for Continuous Control." In *International Conference on Machine Learning*, , 1329–1338.

Edwards, Richard, and Tara Fenwick. 2016. "Digital Analytics in Professional Work and Learning." *Studies in Continuing Education* 38(2): 213–227.

Estrada-Manzo, V., Z. Lendek, and T. M. Guerra. 2015. "Unknown Input Estimation for Nonlinear Descriptor Systems via LMIs and Takagi-Sugeno Models." In *2015 54th IEEE Conference on Decision and Control (CDC)*, , 6349–54.

Estrada-Manzo, Victor, Zsófia Lendek, and Thierry Marie Guerra. 2016. "Generalized LMI Observer Design for Discrete-Time Nonlinear Descriptor Models." *Neurocomputing* 182: 210–20.

Fantuzzi, Cesare, and R. Rovatti. 1996. "On the Approximation Capabilities of the Homogeneous Takagi-Sugeno Model." In *Proceedings of IEEE 5th International Fuzzy Systems*, IEEE, 1067–1072.

Faure, J. L. 2009. *Le Rapport de l'Observatoire National Sur La Formation, La Recherche et l'innovation Sur Le Handicap, 2008*. Paris: Observatoire national sur la formation, la recherche et l'innovation ….

Fay, Brain T., and Michael L. Boninger. 2002. "The Science behind Mobility Devices for Individuals with Multiple Sclerosis." *Medical engineering & physics* 24(6): 375–383.

Fayazi, S. A. et al. 2013. "Optimal Pacing in a Cycling Time-Trial Considering Cyclist's Fatigue Dynamics." In *2013 American Control Conference*, , 6442–47.

Feng, G., L. Buşoniu, T. M. Guerra, and S. Mohammad. 2018. "Reinforcement Learning for Energy Optimization Under Human Fatigue Constraints of Power-Assisted Wheelchairs." In *2018 Annual American Control Conference (ACC)*, , 4117–22.

Feng, G., L. Busoniu, T. Guerra, and S. Mohammad. 2019. "Data-Efficient Reinforcement Learning for Energy Optimization of Power-Assisted Wheelchairs." *IEEE Transactions on Industrial Electronics*: 1–1.

Feng, G., T. M. Guerra, L. Busoniu, and S. Mohammad. 2017. "Unknown Input Observer in Descriptor Form via LMIs for Power-Assisted Wheelchairs." In *2017 36th Chinese Control Conference (CCC)*, , 6299–6304.

Feng, Guoxi, Thierry Marie Guerra, Sami Mohammad, and Lucian Busoniu. 2018. "Observer-Based Assistive Control Design Under Time-Varying Sampling for Power-Assisted Wheelchairs." *IFAC-PapersOnLine* 51(10): 151–56.

Floquet, T., C. Edwards, and S. K. Spurgeon. 2007. "On Sliding Mode Observers for Systems with Unknown Inputs." *International Journal of Adaptive Control and Signal Processing* 21(8–9): 638–56.

Giesbrecht, Edward M., Jacqueline D. Ripat, Arthur O. Quanbury, and Juliette E. Cooper. 2009. "Participation in Community-Based Activities of Daily Living: Comparison of a Pushrim-Activated, Power-Assisted Wheelchair and a Power Wheelchair." *Disability and Rehabilitation: Assistive Technology* 4(3): 198–207.

Grondman, Ivo, Lucian Busoniu, Gabriel AD Lopes, and Robert Babuska. 2012. "A Survey of Actor-Critic Reinforcement Learning: Standard and Natural Policy Gradients." *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42(6): 1291–1307.

Guan, Y., and M. Saif. 1991. "A Novel Approach to the Design of Unknown Input Observers." *IEEE Transactions on Automatic Control* 36(5): 632–35.

Guanetti, Jacopo, Simone Formentin, Matteo Corno, and Sergio M. Savaresi. 2017. "Optimal Energy Management in Series Hybrid Electric Bicycles." *Automatica* 81: 96–106.

Guerra, T. M., H. Kerkeni, J. Lauber, and L. Vermeiren. 2012. "An Efficient Lyapunov Function for Discrete T–S Models: Observer Design." *IEEE Transactions on Fuzzy Systems* 20(1): 187–92.

Guerra, Thierry M., Antonio Sala, and Kazuo Tanaka. 2015. "Fuzzy Control Turns 50: 10 Years Later." *Fuzzy sets and systems* 281: 168–182.

Guerra, Thierry Marie, and Laurent Vermeiren. 2004. "LMI-Based Relaxed Nonquadratic Stabilization Conditions for Nonlinear Systems in the Takagi–Sugeno's Form." *Automatica* 40(5): 823–29.

Han, Hugang, Jiaying Chen, and Hamid Reza Karimi. 2017. "State and Disturbance Observers-Based Polynomial Fuzzy Controller." *Information Sciences* 382: 38–59.

Heydt, G. T. et al. 1999. "Applications of the Windowed FFT to Electric Power Quality Assessment." *IEEE Transactions on Power Delivery* 14(4): 1411–1416.

Howard, Ronald A. 1960. *Dynamic Programming and Markov Processes*. Oxford, England: John Wiley.

Ichalal, D., B. Marx, J. Ragot, and D. Maquin. 2009. "Simultaneous State and Unknown Inputs Estimation with PI and PMI Observers for Takagi Sugeno Model with Unmeasurable Premise Variables." In *2009 17th Mediterranean Conference on Control and Automation*, , 353–58.

Kalsi, Karanjit, Jianming Lian, Stefen Hui, and Stanislaw H. Żak. 2010. "Sliding-Mode Observers for Systems with Unknown Inputs: A High-Gain Approach." *Automatica* 46(2): 347–53.

Kerkeni, H., J. Lauber, and T. M. Guerra. 2010. “Estimation of Individual In-Cynlinder Air Mass Flow via Periodic Observer in Takagi-Sugeno Form.” In *2010 IEEE Vehicle Power and Propulsion Conference*, , 1–6.

Kober, Jens, J. Andrew Bagnell, and Jan Peters. 2013. “Reinforcement Learning in Robotics: A Survey.” *The International Journal of Robotics Research* 32(11): 1238–1274.

Kober, Jens, and Jan R. Peters. 2009. “Policy Search for Motor Primitives in Robotics.” In *Advances in Neural Information Processing Systems 21*, eds. D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou. Curran Associates, Inc., 849–856. http://papers.nips.cc/paper/3545-policy-search-for-motor-primitives-in-robotics.pdf (April 30, 2019).

Lagoudakis, Michail G., and Ronald Parr. 2003. “Least-Squares Policy Iteration.” *Journal of machine learning research* 4(Dec): 1107–1149.

Lampert, Christoph H., and Jan Peters. 2012. “Real-Time Detection of Colored Objects in Multiple Camera Streams with off-the-Shelf Hardware Components.” *Journal of Real-Time Image Processing* 7(1): 31–41.

Leaman, Jesse, and Hung Manh La. 2017. “A Comprehensive Review of Smart Wheelchairs: Past, Present, and Future.” *IEEE Transactions on Human-Machine Systems* 47(4): 486–499.

Lendek, Z., T. Guerra, and J. Lauber. 2015. “Controller Design for TS Models Using Delayed Nonquadratic Lyapunov Functions.” *IEEE Transactions on Cybernetics* 45(3): 439–50.

LEVANT, ARIE. 1993. “Sliding Order and Sliding Accuracy in Sliding Mode Control.” *International Journal of Control* 58(6): 1247–63.

Lewis, F. L., and D. Vrabie. 2009. “Reinforcement Learning and Adaptive Dynamic Programming for Feedback Control.” *IEEE Circuits and Systems Magazine* 9(3): 32–50.

Löfberg, Johan. 2004. “YALMIP: A Toolbox for Modeling and Optimization in MATLAB.” In *Proceedings of the CACSD Conference*, Taipei, Taiwan.

Losero, R., J. Lauber, and T. Guerra. 2015. “Discrete Angular Torque Observer Applicated to the Engine Torque and Clutch Torque Estimation via a Dual-Mass Flywheel.” In *2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)*, , 1020–25.

Losero, R., J. Lauber, and T. -M. Guerra. 2018. “Virtual Strain Gauge Based on a Fuzzy Discrete Angular Domain Observer: Application to Engine and Clutch Torque Estimation Issues.” *Fuzzy Sets and Systems* 343: 76–96.

Losero, R., J. Lauber, T. Guerra, and P. Maurel. 2016. “Dual Clutch Torque Estimation Based on an Angular Discrete Domain Takagi-Sugeno Switched Observer.” In *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, , 2357–63.

Luenberger, D. 1971. “An Introduction to Observers.” *IEEE Transactions on Automatic Control* 16(6): 596–602.

Maeda, G., M. Ewerton, D. Koert, and J. Peters. 2016. “Acquiring and Generalizing the Embodiment Mapping From Human Observations to Robot Skills.” *IEEE Robotics and Automation Letters* 1(2): 784–91.

Marx, B., D. Koenig, and J. Ragot. 2007. “Design of Observers for Takagi–Sugeno Descriptor Systems with Unknown Inputs and Application to Fault Diagnosis.” *IET Control Theory & Applications* 1(5): 1487–95.

Mayne, D. Q., J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. 2000. “Constrained Model Predictive Control: Stability and Optimality.” *Automatica* 36(6): 789–814.

Mnih, Volodymyr et al. 2015. “Human-Level Control through Deep Reinforcement Learning.” *Nature* 518(7540): 529.

Mulder, Eric F., Pradeep Y. Tiwari, and Mayuresh V. Kothare. 2009. “Simultaneous Linear and Anti-Windup Controller Synthesis Using Multiobjective Convex Optimization.” *Automatica* 45(3): 805–11.

Nair, Ashvin et al. 2018. “Overcoming Exploration in Reinforcement Learning with Demonstrations.” In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 6292–6299.

Neal, Radford M. 2001. “Annealed Importance Sampling.” *Statistics and computing* 11(2): 125–139.

Nguyen, Anh-Tu, Thierry-Marie Guerra, and Chouki Sentouh. 2018. “Simultaneous Estimation of Vehicle Lateral Dynamics and Driver Torque Using LPV Unknown Input Observer.” *IFAC-PapersOnLine* 51(26): 13–18.

Oh, Sehoon, Kyoungchul Kong, and Yoichi Hori. 2014. “Operation State Observation and Condition Recognition for the Control of Power-Assisted Wheelchair.” *Mechatronics* 24(8): 1101–11.

Ohishi, K., M. Nakao, K. Ohnishi, and K. Miyachi. 1987. “Microprocessor-Controlled DC Motor for Load-Insensitive Position Servo System.” *IEEE Transactions on Industrial Electronics* IE-34(1): 44–49.

Oonishi, Y., S. Oh, and Y. Hori. 2010. “A New Control Method for Power-Assisted Wheelchair Based on the Surface Myoelectric Signal.” *IEEE Transactions on Industrial Electronics* 57(9): 3191–96.

Organization, World Health. 2011. “World Report on Disability 2011.”

Pai, M. A. 1981. 3 *Power System Stability: Analysis by the Direct Method of Lyapunov*. North-Holland Amsterdam.

Peters, J., and S. Schaal. 2006. “Policy Gradient Methods for Robotics.” In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, , 2219–25.

Peters, Jan, Katharina Mulling, and Yasemin Altun. 2010. “Relative Entropy Policy Search.” In *Twenty-Fourth AAAI Conference on Artificial Intelligence*,.

Peters, Jan, and Stefan Schaal. 2008. “Natural Actor-Critic.” *Neurocomputing* 71(7–9): 1180–1190.

Phillips, A. M., and M. Tomizuka. 1995. “Multirate Estimation and Control under Time-Varying Data Sampling with Applications to Information Storage Devices.” In *Proceedings of 1995 American Control Conference - ACC’95*, , 4151–55 vol.6.

Pogorzelski, Kamil, and Franz Hillenbrand. “The Incremental Encoder – Operation Principals & Fundamental Signal Evaluation Possibilities.” *produktiv messen*. https://www.imc-tm.com/download-center/white-papers/the-incremental-encoder-part-1/ (June 5, 2019).

Precup, Radu-Emil, and Hans Hellendoorn. 2011. “A Survey on Industrial Applications of Fuzzy Control.” *Computers in Industry* 62(3): 213–26.

Rodgers, Mary M. et al. 1994. “Biomechanics of Wheelchair Propulsion during Fatigue.” *Archives of physical medicine and rehabilitation* 75(1): 85–93.

Ronchi, Enrico, Paul A. Reneke, and Richard D. Peacock. 2016. “A Conceptual Fatigue-Motivation Model to Represent Pedestrian Movement during Stair Evacuation.” *Applied Mathematical Modelling* 40(7): 4380–96.

Rummery, Gavin A., and Mahesan Niranjan. 1994. 37 *On-Line Q-Learning Using Connectionist Systems*. University of Cambridge, Department of Engineering Cambridge, England.

Scherer, C W. 2001. “LPV Control and Full Block Multipliers&.” : 15.

Scherer, Carsten, and Siep Weiland. 2015. “Linear Matrix Inequalities in Control.” : 293.

Schulman, John et al. 2015. “Trust Region Policy Optimization.” In *International Conference on Machine Learning*, , 1889–1897.

Seki, H., K. Ishihara, and S. Tadakuma. 2009. “Novel Regenerative Braking Control of Electric Power-Assisted Wheelchair for Safety Downhill Road Driving.” *IEEE Transactions on Industrial Electronics* 56(5): 1393–1400.

Seki, H., and A. Kiso. 2011. “Disturbance Road Adaptive Driving Control of Power-Assisted Wheelchair Using Fuzzy Inference.” In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, , 1594–99.

Seki, H., and S. Tadakuma. 2004. “Minimum Jerk Control of Power Assisting Robot on Human Arm Behavior Characteristic.” In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, , 722–27 vol.1.

———. 2006. “Straight and Circular Road Driving Control for Power Assisted Wheelchair Based on Fuzzy Algorithm.” In *IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics*, , 3898–3903.

Seki, Hirokazu, Takeaki Sugimoto, and Susumu Tadakuma. 2005. “Novel Straight Road Driving Control of Power Assisted Wheelchair Based on Disturbance Estimation and

Minimum Jerk Control." In *Fourtieth IAS Annual Meeting. Conference Record of the 2005 Industry Applications Conference, 2005.*, IEEE, 1711–1717.

Shung, J. B., G. Stout, M. Tomizuka, and D. M. Auslander. 1983. "Dynamic Modeling of a Wheelchair on a Slope." *Journal of Dynamic Systems, Measurement, and Control* 105(2): 101–106.

Silver, David et al. 2016. "Mastering the Game of Go with Deep Neural Networks and Tree Search." *Nature* 529(7587): 484–89.

———. 2017. "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm." *arXiv preprint arXiv:1712.01815*.

Simpson, Richard C. 2005. "Smart Wheelchairs: A Literature Review." *Journal of rehabilitation research and development* 42(4): 423–36.

Sutton, Richard S., and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. MIT press.

Sutton, Richard S, David A. McAllester, Satinder P. Singh, and Yishay Mansour. 2000. "Policy Gradient Methods for Reinforcement Learning with Function Approximation." In *Advances in Neural Information Processing Systems 12*, eds. S. A. Solla, T. K. Leen, and K. Müller. MIT Press, 1057–1063. http://papers.nips.cc/paper/1713-policy-gradient-methods-for-reinforcement-learning-with-function-approximation.pdf (April 30, 2019).

Szepesvári, Csaba. 2010. "Algorithms for Reinforcement Learning." *Synthesis lectures on artificial intelligence and machine learning* 4(1): 1–103.

Takagi, TOMOHIRO, and MICHIO Sugeno. 1993. "Fuzzy Identification of Systems and Its Applications to Modeling and Control." In *Readings in Fuzzy Sets for Intelligent Systems*, eds. Didier Dubois, Henri Prade, and Ronald R. Yager. Morgan Kaufmann, 387–403. http://www.sciencedirect.com/science/article/pii/B9781483214504500456 (April 30, 2019).

Taniguchi, T., K. Tanaka, H. Ohtake, and H. O. Wang. 2001. "Model Construction, Rule Reduction, and Robust Compensation for Generalized Form of Takagi-Sugeno Fuzzy Systems." *IEEE Transactions on Fuzzy Systems* 9(4): 525–38.

Tanohata, N., H. Murakami, and H. Seki. 2010. "Battery Friendly Driving Control of Electric Power-Assisted Wheelchair Based on Fuzzy Algorithm." In *Proceedings of SICE Annual Conference 2010*, , 1595–98.

Tashiro, S., and T. Murakami. 2008. "Step Passage Control of a Power-Assisted Wheelchair for a Caregiver." *IEEE Transactions on Industrial Electronics* 55(4): 1715–21.

Tesauro, Gerald. 1995. "Temporal Difference Learning and TD-Gammon." *Communications of the ACM* 38(3): 58–68.

Thieffry, M., A. Kruszewski, C. Duriez, and T. Guerra. 2019. "Control Design for Soft Robots Based on Reduced-Order Model." *IEEE Robotics and Automation Letters* 4(1): 25–32.

Tsai, M., and P. Hsueh. 2012. “Synchronized Motion Control for 2D Joystick-Based Electric Wheelchair Driven by Two Wheel Motors.” In *2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, , 702–7.

Tsai, Mi-Ching, and Po-Wen Hsueh. 2013. “Force Sensorless Control of Power-Assisted Wheelchair Based on Motion Coordinate Transformation.” *Mechatronics* 23(8): 1014–24.

Tsuyoshi Shibata, and Toshiyuki Murakami. 2008. “Power Assist Control by Repulsive Compliance Control of Electric Wheelchair.” In *2008 10th IEEE International Workshop on Advanced Motion Control*, Trento, Italy: IEEE, 504–9. http://ieeexplore.ieee.org/document/4516118/ (April 30, 2019).

Umeno, T., T. Kaneko, and Y. Hori. 1993. “Robust Servosystem Design with Two Degrees of Freedom and Its Application to Novel Motion Control of Robot Manipulators.” *IEEE Transactions on Industrial Electronics* 40(5): 473–85.

“US20170151109A1 - Method and Device Assisting with the Electric Propulsion of a Rolling System, Wheelchair Kit Comprising Such a Device and Wheelchair Equipped with Such a Device - Google Patents.” https://patents.google.com/patent/US20170151109A1/en (April 30, 2019).

Vecerik, Mel et al. 2019. “A Practical Approach to Insertion with Variable Socket Position Using Deep Reinforcement Learning.” In *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 754–760.

Wan, N., S. A. Fayazi, H. Saeidi, and A. Vahidi. 2014. “Optimal Power Management of an Electric Bicycle Based on Terrain Preview and Considering Human Fatigue Dynamics.” In *2014 American Control Conference*, , 3462–67.

Williams, Ronald J. 1992. “Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning.” *Machine Learning* 8(3): 229–56.