Adaptive multivariable intermittent control: theory, development, and applications to real-time systems.

José Alberto Álvarez Martín

A thesis for the degree of *Doctor of Philosophy* (PhD)

Submitted to the College of Science and Engineering
University of Glasgow

January 2018

Abstract

Intermittent Control (IC), as a control scheme that switches between open and closed-loop configurations, has been suggested as an alternative model to describe human control and to explain the intermittency observed during sustained control tasks. Additionally, IC might be beneficial in the following scenarios: 1) in the field of robotics, where open-loop evolution could be used for computationally intensive tasks such as constrained optimisation routines, 2) in an adaptation context, helping to detect system and environmental variations. Based on these ideas, this thesis explored the application of real-time multivariable intermittent controllers in humanoid robotics as well as adaptive versions of IC implemented on inverted pendulum structures.

IC was evaluated on a humanoid robot during a balancing task, regulating the joint angles in order to maintain a standing position and to recover from perturbations exerted by a linear actuator. The experiment showed that IC is a viable alternative to traditional continuous control methods employed by roboticists, generating joint angles that are comparable in magnitude and rejecting the applied perturbations, while providing extra time resources in the form of open-loop intervals that could be allocated for other important goals. These results motivated the development of adaptive intermittent controllers (AIC) based on a self-tuning architecture and Kalman filtering, and were implemented for the first time on a real-time rotational pendulum using Extended and Unscented Kalman filters in combination with two versions of IC: the *system-matched hold* IC and the *tapping hold* IC. Simulations were performed assuming that some of the physical parameters of the pendulum were time-varying, and an experiment was carried out on the physical system considering a model that had parameters that were different in magnitude compared the nominal values. The results suggest that the IC inputs provide balance in terms of output and parameter estimation errors by constantly exciting the system, with the added benefit that controller redesign only happens at event times, reducing the computational burden. In conclusion, real-time controllers based on the IC framework were derived and potential benefits for generic engineering systems, that could help explain adaptive features of human motor control, were investigated.

Acknowledgements

I would like to thank my supervisor, Dr. Henrik Gollee. Without his advice, this would have never been possible. Thanks for understanding the highs and lows of being a student, for all the ideas and suggestions, for all the good times; but more importantly, for believing in me all these years. Thanks for introducing me to intermittent control, and for sharing your passion for control engineering with me. I will always remember the great discussions with the members of the intermittent control group, specially with Prof. Peter Gawthrop and Prof. Ian Loram, always insightful and deep. I would also like to thank Dr. Cornelis van de Kamp and Erik Vlasblom, who showed me the world of robotics and stayed late many nights, in Delft, working with me on our good old friend TUlip.

Thanks to all my friends in the Centre for Rehabilitation Engineering. Being a member of such a team was one of the greatest experiences during my time in Glasgow. Thanks to Dr. Aleksandra Vuckovic, Mohammed Jarjees, Salim Al-Wasity, Manaf Al-Taleb, Bethel Osuagwu, Lin Meng, Sarah Comincioli, Linda Binti, Anna Sosnowska, Anna Zulauf, Finda Putri, Gabriela Cruz and Jennifer Miller, for all the laughs and great memories. I would also like to thank Thomas O'Hara, Dr. Aldo Vargas, and Caesar Al-Ameri, you made my days in the University a real joy. To all the people in Mexico that supported and encouraged me from day one, thanks to Jonathan Sanata, Isidro Lázaro, Héctor Castañeda, Juan Luis Sandoval, Verónica Villaseñor, Omar Morán, Rodrigo Torres, Dr. Jesús Rico, and to the Guzmán-Fukumori familiy.

Thank you Margaret, you have been my guiding light on this path, thanks for caring the way you do and for always having the right word at the right time. This would have been impossible without you. Thank you Lucía and Diego, for all your love, patience, and for being a constant source of motivation. Finally, I would like to thank my parents, for all the sacrifices you have made to see my siblings and I succeed. Thanks for being who you are, for understanding me, and for being right next to me always.

Special thanks to CONACyT México, SCIDET Michoacán, and SEP México for providing the financial support to complete this project.

Author's declaration

I declare that, except where explicit reference is made to the contribution of others, this thesis is the result of my own work and has not been submitted for any other degree at the University of Glasgow or any other institution.

The experiment presented in chapter 4 was the result of a team effort between the University of Glasgow and Delft University of Technology. In particular, Dr. Cornelis van de Kamp allowed me work in the BioRobotics laboratory and lead the project as a visiting researcher, helping with the experimental design as well as being involved in the execution of the experiment. Erik Vlasblom helped me with the integration of the intermittent control routines to the exisiting code base implemented in the robot, and provided a solid robotics point of view and support to the overall project.

José Alberto Álvarez Martín

January 2018

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Our ancestors adopted a bipedal stance and left four-legged locomotion behind after a long process of adaptation and evolution. This gave them the possibility of using the upper limbs for complicated tasks like grasping with precision, allowing them to produce and manipulate tools for many other purposes. This behavioural advantage came with an associated cost, which is the problem of maintaining balance by keeping the centre of mass (CoM) within a small support area in order to avoid a fall. The way our biological controllers achieve this is by generating ankle torque based on the contraction of the soleus and gastrocnemius muscles (Basmajian and De Luca, 1985; Loram et al., 2005). Human balance is indeed a difficult problem due to the ratio between feet size and body height, and because of the constraints imposed by our joints, restricting our degrees of freedom (DoF), as well as the redundancy introduced by our muscles. Additionally, muscle fatigue comes into play rather quickly, affecting our movements and joint kinematics in general (Duarte and Zatsiorsky, 2001; Madigan et al., 2006). During standing, movement is normally dictated by a sagital sway pattern around the ankle joints (Gatev et al., 1999), which is commonly known as an *ankle strategy*. In the presence of external perturbations, the body can be pushed away from a stable sway motion, forcing the switch to recovery strategies which involve movement at the hip to maintain balance, hence a *hip strategy* (Nashner and McCollum, 1985), or even taking a step in the case of large perturbations. A standing posture also brings other common problems like lower back pain (Plomp et al., 2015) and blood circulation issues (Smith, 1990).

Nevertheless, balancing in human standing is natural to most of us, so much that we almost never pay attention to the actions that are required to stay on foot, even when an unexpected event throws us away from our equilibrium. In such cases, some of the reflex-like movements we use to compensate occur even against our will or without consciously deciding to execute them. The truth is that the underlying mechanisms

of quiet standing are still not fully understood by researchers; however, the study of this problem has allowed engineers to use ideas and concepts from the field of human physiology to get inspiration for new control methods (Bottaro et al., 2005, 2008; Gawthrop et al., 2011), it has allowed doctors and clinicians to better understand balance disorders (Bronstein et al., 1996; Mergner, 2010) and it provided insight from an evolutionary point of view to many biologists (Schmitt, 2003; Skoyles, 2006).

The study of standing in humans often implies the simplification of the problem, just as in many other areas of engineering. This involves resorting to models that summarise behavioural data, which helps to rationalise and understand human motion in order to visualise its fundamental properties. For instance, the body sway oscillations that are present in quiet standing fit the dynamics described by the equations of an inverted pendulum (Winter et al., 1998), which is convenient in the sense that the number of variables and parameters is greatly reduced, while still capturing the dynamic behaviour of the system. Many studies in movement science involve inverted pendulum studies as an analogue of the human body, ranging from single link pendulums (Loram and Lakie, 2002; Lakie et al., 2003; Loram et al., 2011), to multi-segmental structures (Gawthrop et al., 2015).

In the human standing context, inverted pendulum models are used to describe the part of the problem that it is supposed to be controlled, often represented by variables like the angle with respect to the vertical position or a CoM position with respect to the support base. In engineering terms, these are known as the *system* or the *plant*. Some systems might have a stable nature while others, like the inverted pendulum, exhibit unstable dynamics in the absence of a control command. Therefore, another kind of model is required to explain the way in which a stabilising action should be conceived, planned and executed, so that the system is regulated or achieves a predefined configuration. This broad idea of a *controller* is formed by three separate mechanisms: i) the intrinsic properties of a human joint such as damping and stiffness, contribute by generating torque responses to unknown perturbations. These are fast correcting actions with no delays. ii) there is a fast continuous loop of neural reflex-like responses running through the spinal cord, brain stem, motor cortex and the cerebellum (Deliagina et al., 2007) with delays up to 180 ms, that contributes to the overall torque commands. iii) A high-level control loop, in charge of planning and selecting appropriate motor commands (van de Kamp et al., 2013a). This high-level loop is capable of performing complex tasks such as adjusting its own commands to compensate for a change in the conditions or predicting the consequences of the applied commands to overcome considerable feedback and system delays.

Two important points of view about the functions and overall structure of these high-level processes have been around for some time now. Starting with a continuous control approach stating that the selection and planning stages are based on the continuous use of feedback to compute subsequent actions. Conversely, an intermittent approach suggests that the use of feedback is dictated by refractoriness, that task execution happens sequentially, and that feedback is used only when a previous action has been processed. This can be seen as a hybrid approach where the use of available information in the form of feedback is combined with open-loop evolution. This intermittency was observed initially in the early work of Craik (1947), leading to many experiments providing supporting evidence (Vince, 1948; Navas and Stark, 1968; Neilson et al., 1988; Loram et al., 2011). It also inspired control engineers to combine these findings with existing control theory to propose new controllers. As a result, intermittent control was presented initially by Ronco et al. (1999) as an attempt to extend the theory and applications of generalised predictive control; this initial version allowed Gawthrop et al. (2011) to formulate it in the context of human motor control.

A great effort has been made over the years to substantiate intermittent control as an explanation for human motor control, from both theoretical and experimental perspectives. Recent studies show that intermittent control is a robust alternative when trying to manually control a virtual inverted pendulum that changes its properties through time (van de Kamp et al., 2013b), suggesting that the switching between opening and closing the loop might contribute to identify the effects of our commands and to separate them from the natural dynamics of the system. In other words, the subjects were better at adapting to a new environment, where adaptation means the ability to change in order to improve according to some measure of performance. In that sense, an adaptive controller changes its performance in the presence of a new environment, while learning is associated with a controller that adjusts the performance when facing the same environment after several encounters.

The work of Gawthrop et al. (2015) on intermittent adaptive control is the first effort to extend the explanatory power of this framework to systems described by time-varying parameters or to situations where a plant/model mismatch requires some form of adaptation. In general, the field of adaptive control from an engineering perspective is well-established (Gawthrop, 1982; Åström and Wittenmark, 1995; Goodwin et al., 2001), and the body of work in this field is impressive, with applications in many different areas of engineering. This is motivated by the fact that in real life, no model is correct, there is always a certain amount of deviation from the truth. That characteristic makes the use of adaptive controllers necessary to reduce the effect of such differences. Gawthrop's contributions are based on continuous-time, non-minimal state-space for-

mulations and on state-variable filters, combined in a way that real-time system identification performed to update controller parameters. In his work, examples of adaptive intermittent control applied to human balance and manual control are presented in a simulation environment, showing the potential advantages of applying this in real-time scenarios. It is also stated that other alternatives should be considered to extend the adaptive intermittent controller to different formulations based on Extended and Unscented Kalman filters (Kalman, 1960; Julier et al., 1995).

Although continuous adaptive controllers can be effective in many situations, sometimes they generate control signals that do not excite the system enough to ensure that the parameter estimation process converges to the correct values. Many special techniques have been developed to counteract this problem, with most of them relying on the addition of auxiliary exciting signals (Landau et al., 2011), specially during the adaptation transients; however, adding these signals is not always feasible in practice. The *stability-plasticity dilemma* (Carpenter and Grossberg, 1988) helps to understand the trade-off that adaptive control in general should deal with, in the sense that exciting the system constantly means that the controller remains plastic (able to detect changes) while providing a response that ensures stability. The impulsive nature of the signal generated by an intermittent controller provides a balanced solution in terms of this trade-off.

While the idea of an intermittent controller being involved in high-level human control processes such as sensory analysis, task planning, and response selection, is still debatable and remains as an open question, the use of it as a control strategy for other purposes is quite appealing. Some possible applications in a biomedical context might lie in robotic-assisted rehabilitation (Loram et al., 2011) or in the modelling of cellular network systems (Gawthrop et al., 2015). Moreover, with the growing overlap between engineering and physiological research, and recent results from human balance, it is hypothesised that the field of humanoid autonomous robotics might benefit from the use of multi-input, multi-output intermittent controllers.

1.1 Aims and objectives

The overall aim of this research was to formulate multivariable adaptive intermittent controllers for real-time scenarios. The specific aims were (i) to investigate whether intermittent control could be applied to multi-segmental, autonomous, humanoid robots and to evaluate potential benefits of the IC approach for this field of engineering; with the expectation to achieve similar control performance levels when compared to tradi-

tional continuous control schemes. To do this, a balancing experiment was performed on TUlip, a robotic platform at the Technical University of Technology in Delft, the Netherlands (de Boer, 2012). (ii) to evaluate both in simulation and experimentally, the use of adaptive intermittent controllers based on state-space formulations, relying on nonlinear Kalman filters to perform real-time state and parameter estimation. In this case, the expectation was to obtain controllers that would perform similarly, in terms of output response, to continuous adaptive versions; with the added benefit of not having to redesign the controller continuously but only at the event times dictated by the IC triggering mechanism.

The first objective was to design multi-input, multi-output (MIMO) intermittent controllers to be used in real-time environments, tailor-made for a three-link inverted pendulum model of a humanoid robot.

The second objective was to implement the controllers on an advanced robotic platform (in terms of its design and hardware) such as TUlip, which is a 1.095 meters tall humanoid robot. The implementation involved the experimental evaluation of intermittent control in comparison with an analogous continuous predictive controller, by running a balancing task with perturbations during quiet standing.

The third objective was to design an adaptive intermittent controller capable of estimating the dynamic states of the system while tracking time-varying parameters. This real-time estimation routine would provide the necessary information to update the control law every intermittent interval in order to adjust the overall performance of the controller. The estimation was based on nonlinear versions of the Kalman filter, such as the Extended and Unscented formulations.

The final objective involved the validation of these adaptive intermittent controllers in simulation, to then evaluate their performance experimentally on a rotational inverted pendulum, which is an unstable, under-actuated system. The performance was compared to that of a continuous adaptive controller.

1.1.1 Contributions

The overall contributions of this thesis are as follows:

1. The experiment performed on TUlip is the first successful implementation of MIMO real-time intermittent controllers in an advanced engineering structure.

Moreover, the results suggest that intermittent control might be a robust alternative against un-modelled dynamics, as well as providing extra computational resources in the form of open-loop intervals.

2. The application of adaptive intermittent controllers on the rotational pendulum constitutes the first implementation of such architectures in real-time, showing that updating the control law using estimated parameters every intermittent interval produces similar results compared to a continuous update. The results also show that the control signals generated by the intermittent controllers provide a balanced solution in terms of output error and parameter estimation.

3. The implementation of the aforementioned controllers include the first use of 'tapping' (Gawthrop and Gollee, 2012) in an adaptation context, as an alternative to the classic intersample behaviour described by a system-matched hold (Gawthrop and Wang, 2011), showing that the use of a tapping hold results in a control signal that contributes positively to the parameter estimation process, while keeping similar output properties compared to the system-matched hold.

1.1.2 Publications

Conference proceedings

J.A. Álvarez-Martín, H. Gollee, I.D. Loram, P. Gawthrop. "A dual Kalman filter approach to adaptation in intermittent control", *In proceedings of the 21st ISEK Congress. Congress of the International Society of Electrophysiology and Kinesiology. Chicago, USA. 2016.* Selected for a student award.

In preparation

- J.A. Álvarez-Martín, E. Vlasblom, C. Van de Kamp, H. Gollee. "Event-driven intermittent control of a humanoid robot".
- J.A. Álvarez-Martín, H. Gollee, P. Gawthrop. "Adaptive intermittent control of a rotational pendulum, a system-matched hold approach.". Journal of Systems and Control Engineering. Institution of Mechanical Engineers.

1.2 Overview and structure of the thesis

This thesis is organised in six chapters as follows:

In chapter one, a brief introduction to the contents of this thesis is given, explaining the fundamental motivations behind using intermittent control to model human motion from the point of view of balance and quiet standing.

Chapter two presents a literature review of the physiological origins of intermittent control as well as its roots in control theory. The serial ballistic hypothesis is introduced and the most important contributions that led to the intermittent control paradigm are discussed. The connections between the fields of autonomous robotics and human balance are presented, as well as the links of intermittent control with classical adaptive control theory.

In chapter three, the theoretical background of intermittent control is introduced and a description of its different versions is given. A simulation example is presented to illustrate the most important features. Also, this chapter discusses the concept of adaptation in the IC framework, and it provides an explanation of the adaptive intermittent controller used in this thesis, with an emphasis on parameter and state estimation methods, and the self-tuning regulation architecture. The use of a tapping hold in an adaptation context is also presented. The chapter ends with an overall discussion.

Chapter four introduces the application of intermittent control to humanoid robotics, describing the experimental protocol for a balancing task as well as the considerations taken to design the implemented controllers. The experimental results are shown along with a discussion.

In chapter five, simulations of adaptive intermittent control applied to a rotational inverted pendulum are presented first, followed by the results of implementing these controllers in a real-time environment. The chapter ends with a general discussion.

The final chapter is devoted to the overall conclusions of the thesis. Additionally, a short discussion is presented, the limitations of the project are introduced, and some follow up ideas are mentioned in form of future work.

Chapter 2

Literature review

2.1 Introduction

Humans are capable of performing incredibly complex movements with dexterity, making everything look really simple and smooth in many cases. Playing an instrument, a dancing performance, or a high level athlete in competition, are all good examples of sustained motion control which is involved in our everyday life; yet there are still features of the underlying control processes related to human motion that we do not fully understand and that are actively discussed by the scientific community.

In particular, the idea of finding mathematical representations of our control mechanisms by means of computational algorithms has been an attractive notion to many people. For many years, a continuous negative feedback approach has been used to explain the task of maintaining balance during standing in humans and its intrinsic variability (van der Kooij et al., 1999; Peterka, 2002; Maurer and Peterka, 2005; Kooij and Vlugt, 2007). This implies that throughout the execution of the task, the sensorimotor information coming in the form of feedback is used all the time to generate appropriate control actions. Under this paradigm, the ideas behind optimal control theory and state prediction have been used to create models that describe physiological systems under the presence of motor and observation noise, with the model proposed by Kleinman (1969) as the leading paradigm.

However, the seminal work from Craik (1947, 1948) provided evidence of humans behaving in a ballistic, intermittent fashion for specific manual tasks. Moreover, this results led others (Vince, 1948; Navas and Stark, 1968) to test Craik's hypothesis experimentally, to find that when a subject is presented with a series of discrete stimuli, they respond at a rate of 2 to 3 actions per second, and that these actions only start

as soon as the previously executed action had elapsed. We know now that a property of the Central Nervous System (CNS) is the presence of the Psychological Refractory Period (PRP) originally discovered by Telford (1931), which is a well defined time interval where the control action acts without the influence of feedback information. In other words, if a response is already initiated, we can start planning and calculating a response for a second stimulus only after the PRP has elapsed. This means, in control engineering terms, that the controller in place operates in an open-loop regime for a minimum amount of time.

The work from Navas and Stark (1968) showed that humans use intermittent control in manual pursuit and tracking tasks, providing experimental evidence of human responses being modified approximately every 500 ms. Recently, in (Lakie et al., 2003), the same behaviour was observed when a group of subjects tried to manually control an inverted pendulum, confirming the similar action rates as in (Vince, 1948). In this same line, subsequent work from Lakie and Loram (2006) showed that even when different sensory information is provided, the rate of control actions stays the same when performing manual control tasks. These contributions resonated strongly within the movement science and physiology communities, leading to the consideration of intermittent control as a viable way to explain human motor control.

The theory of intermittent control has been advanced greatly since the work of Craik, building up a computational model that is capable of explaining sustained control and its intrinsic variability in different situations. This model has been defined from a linear perspective, in particular, treating the system at hand as a time-invariant model. This means that the system exhibits a linear behaviour and stays the same, in terms of the parameters that describe it, throughout its evaluation. It is also known that most systems in nature are not static, they are nonlinear and in many cases they are described by time-varying parameters. The work of van de Kamp et al. (2013a,b) and Gawthrop et al. (2015) suggest that human motor control uses adaptation capabilities to deal with uncertainties and environmental changes, which raises the question that adaptive intermittent control could possibly provide an explanation for these situations.

Additionally, in order to substantiate the theory of intermittent control, an important effort has been made to test these models in real-time systems (Gawthrop and Gollee, 2012) with the purpose of studying them from an engineering perspective, mostly for single input, single output systems. A natural consequence of these results, was to raise the question of the effects of a human inspired control algorithm such as intermittent control in the field of humanoid robotics.

In this chapter, the intermittent control framework will be described, considering both its physiology and engineering origins. The applications of intermittent control to real-time engineering scenarios and its possible benefits to the field of robotics will be introduced. Also, the connections of intermittent control with adaptation is reviewed in conjunction with the main adaptive control models in literature.

2.2 Serial ballistic hypothesis and intermittent control

It was the work of Craik (1947) that set the stage for new hypothesis on how humans control their movement. At the time, research on the interaction between man and machines was greatly influenced by the political climate and tension during the years after the second world war, where understanding how operators were capable of tracking moving targets was a fundamental motivation. When his seminal paper was published, the notion of a psychological refractory period was known, which states that there is a small amount of time where the control action is not affected by feedback that comes from the sensory system, discovered experimentally by Telford (1931). Craik postulated the fact that humans operate as intermittent servos, meaning that since the refractory period is present, the control output is generated as a sequence of serial ballistic control actions, where each individual control action lasted at least one refractory period. This evidence suggested that humans might modulate their control signals intermittently rather than continuously.

This intermittency motivated Vince (1948) to test this hypothesis experimentally, confirming Craik's evidence and suggesting that humans in fact operate intermittently under compensatory and pursuit tracking tasks. In one experiment, subjects were asked to keep a pointer aligned with a reference line that was drawn on a rotating smoked drum, the subject controlled the movement of the pointers using a handle. The pointer was perturbed by a motor causing a steady movement away from the line, forcing compensatory action by the subject. Similarly, in a second experiment the subjects were asked to keep the pointer aligned with a reference line that was changing in direction suddenly, based on a predefined course. One of the main findings was that the rate at which the subjects used corrective actions does not depend on the frequency of the disturbance signal and more importantly, it remained constant with a rate of 2 to 3 actions per second. Also, when the subjects were presented with quick series of changes in reference, it was noticed that the response to the second change in the series is delayed compared to that of the first one, providing more experimental evidence of the refractory period. Based on this, Vince suggested that the inability to respond on time to a second stimulus appears to be related to some central computing delay that

is used to select the appropriate response. If a second stimulus is presented during this delay, it will not evoke a response until an interval of half a second after the appearance of the first stimulus has elapsed.

This inability to react to two different stimuli is still a source of debate. Welford (1952) proposed the fact that it is not possible for humans to select two responses at the same time, favouring the idea of processing happening sequentially. First, our sensorimotor system must provide information of the stimulus, then a feasible response must be selected, to finally execute the action. There is experimental evidence for this claim, showing also that this result is not affected by the kind of sensory information that is used to process the stimuli nor by the type of response (Pashler, 1992, 1994).

In support of this, Lakie et al. (2003) carried out experiments with the purpose of evaluating how balance control by means of a compliant element is performed. In this task the subjects were asked to manually balance a real single inverted pendulum with low intrinsic stiffness. In order to achieve balance, the subject had to pull a handle that was connected to the pendulum through an elastic element (a spring in this case). This is similar in many ways to the problem of human standing, where the calf muscle pulls through the Achilles tendon which is in turn attached to the ankle joint. In Lakie's experiment, the inverted pendulum resembles the human body as a single link structure. One of the important conclusions from this study is that the subjects applied discrete hand movements to maintain balance, and that these corrections came at the same rates reported by (Vince, 1948), moreover, human control in this particular setting seems to be subject to a refractory period and under the influence of a high level intermittent control process. The apparent limited bandwidth of these control adjustments (a maximum of 2 Hz) has been studied and demonstrated by Navas and Stark (1968); Miall et al. (1993a); Nielson (1999); Loram et al. (2009, 2011).

A natural consequence of the previous experiments was to evaluate the effect of the kind of feedback used to formulate control actions, as stated earlier. In an elegant experiment, Lakie and Loram (2006) showed that the correction rate was not affected by different forms of sensory information: visual, vestibular, and proprioceptive. Even when these forms of feedback were combined, the rate stayed the same. A similar attempt was made to evaluate if the nature of the load that should be balanced would have an effect on the aforementioned correction rates (Loram et al., 2006a). In this case, the time constant of the pendulum was doubled without observing any effect on the adjustment rates of the control signal. These two results suggest that balancing tasks might require a single computing neural process. This process might be consid-

ered a part of a more general delay known as *feedback time-delay*, which is comprised of the integration of sensory information, selection of the control action and planning its execution. However, this does not include the delay associated with the time elapsed between the control action and the response of the system as a consequence. Therefore, in this context, the effective time delay is formed by both a feedback time-delay and a system delay.

To further test the idea of intermittent control being behind human standing control, Loram et al. (2006b) showed that when evaluating the characteristic sway motion during standing, the calf muscle uses discrete corrections every 2 to 3 seconds on average, confirming results discussed earlier and suggesting that human standing is driven by intermittent control processes that are similar to those in manual tracking tasks.

Being unable to respond on time to sufficiently fast stimuli seems to be a limiting characteristic of human motor control, forcing us to disregard available information until we have processed previous actions. However, this can be seen as a built-in feature that allows us to predict the consequences of our actions in a better way. In control terms, opening and closing the loop can be used to clearly distinguish between the effect of new information (and its subsequent correcting action) from the unforced dynamics of the system. This idea led to a manual balancing experiment by Loram et al. (2011) that compared two strategies to control a virtual pendulum on a screen by means of a joystick. The subjects were asked to control the position of a pointer on a screen, that moved according to the second order, unstable dynamics of a single inverted pendulum, by holding the joystick throughout the entire trial or by *tapping* it. It is clear that the tapping resembles a pure intermittent approach where the each tap can be seen as a discrete control action.

One of the tasks in this experiment was to keep the position of the pointer in the centre of the screen (in terms of the pendulum that means to maintain the pendulum as close as possible to the upwards equilibrium point). During the trial, small random input disturbances were applied to ensure that the pointer would actually drift away from the centre. Additionally, the gain between the joystick signal and the virtual pendulum changed according to a piece-wise step function of three different amplitudes. These changes were unknown to the subject, resulting in a sudden adjustment on their strategy for most cases. For this task, tapping the joystick gave a better response compared to continuous contact. Also, tapping seems to be better way to deal with the changes in the joystick gain, showing less position error during gain transitions. An important conclusion based on this evidence is that intermittent control might be better at situations where there is not enough knowledge about the system, or

when this knowledge is not close to reality. Furthermore, the authors hypothesise that the constant switching between open and closed loop regimes imposed by the tapping strategy promotes a quicker identification of the system and the conditions affecting it.

Despite the growing interest in intermittent control and the experimental evidence described so far, there are continuous-time based theories that provide an opposing view while explaining some aspects of human motor control (Maurer and Peterka, 2005; Kooij and Vlugt, 2007). Overall, these theories have been taken by the community as the dominant paradigm for sustained control problems. However, the long standing debate between continuous and intermittent control has given great insight on the explanatory power of classical control theory applied to human physiology and movement science.

From this perspective, the use of engineering models to describe human motor control should not be considered as a direct analogue to the underlying physiological structures. Instead, they should be seen as sources of information about the performance of the human controller, providing solid grounds to asses behavioural data, to understand our control actions, and more importantly, to formulate predictions based on experimental data. An example of early work on engineering models for describing the manual interaction between pilots and machines comes from McRuer and Jex (1967). In his work, a detailed description of typical aircraft control models is presented, suggesting different transfer function representations for situations such as: pursuit, compensatory and periodic tasks. Primarily, the pilot is described as a transfer function formed by a gain and a delay element that changes according to the order of system to be controlled with the addition of a remnant signal that models the components of the output data that can not be described in linear terms.

A common approach that has been used extensively to model human stance is the proportional-integral-derivative (PID) controller (Johansson et al., 1988; Peterka, 2000; Maurer and Peterka, 2005). This model is based on the three important gain parameters that can be tuned using system identification or by optimisation procedures. Additionally, a time-delay is incorporated to the model with the purpose of reproducing the characteristic sway observed in quiet standing. Alexandrov et al. (2005), used the PID controller and a three segment, multi-joint model of a human to fit experimental data, which was collected from a balance perturbation task. As a conclusion the authors state that continuous feedback can describe human postural responses to stance perturbations and that these responses are also independently modulated via feed-forward corrections, and that the overall feedback loop can be modelled as a viscoelastic spring with a delay. However, a limitation of the PID model is that it has not

been able to explain the reduced control bandwidth and the long time delays observed in balance tasks. Evidence of this is provided by Gawthrop et al. (2009), where data from a simulated inverted pendulum controlled through a joystick was used to fit predictive and non predictive models. It is shown that the non-predictive PID controller is able to fit experimental data reasonably well, but it falls short in terms of identifying the time delay. On average, the PID model predicted smaller time delay values compared to those coming from a predictive state-feedback model. Also, the time delays generated by the PID model were less consistent with the ones estimated from the non-parametric impulse response obtained from the experiment.

The study by Gawthrop et al. (2009) contrasts the aforementioned PID model with an alternative state-space representation based on optimality and prediction. This is consistent with the ongoing trend of representing the human controller as an entity that uses prediction to counteract delayed information from the sensory system, while using internal states to form its own representation of the system dynamics. The information generated by this *internal* model is combined with known motor commands and measurements in the form of feedback. Some authors suggest that the brain relies on these internal representations, particularly in the cerebellum (Miall and Wolpert, 1996; Wolpert et al., 1998; Kawato, 1999; Schwartz, 2016), to achieve accurate motion patterns. With this in mind, Kleinman (1969) proposed a model based on three elements: an observer, a predictor, and state-based negative feedback. The purpose was to represent human control as a continuous controller that would find an *optimal* control action based on performance constraints, and using predicted states affected by observation noise. The internal models discussed earlier are implemented in Kleinman's architecture using state-observers, these operate based on a dynamical model of the system (thus an internal model) that receives the optimal control signals and the data being measured to update the overall state. In physiological terms, this observer receives an efference copy of the motor command and a reafference signal coming from the sensory system to calculate the new state; then a prediction based on the previously calculated state is made available to the motor controller, closing the loop. Neilson et al. (1988) proposed a detailed description of the relationship between internal models and the intermittent behaviour observed in tracking tasks, presenting a model comprising sensory analysis, response planning, and response execution stages. There, the human operator is represented using internal models that process sensory information continuously, whereas the response planning mechanism updates the remaining stages intermittently.

Kleinman's model is known as the *observer-predictor-feedback (OPF)* controller and it has been used in a wide variety of experiments, providing deep insight in terms

of the interaction between man and machine. At the time, research in this area was influenced greatly by military applications involving manual tracking of moving targets. This model has been used to represent the variability of the human operator by means of additive noise in visuo-manual tasks (Kleinman et al., 1970), to fit data from balance experiments (van der Kooij et al., 1999; Gawthrop et al., 2009) or to explain compensatory tracking with delays ranging from 150 to 250 ms (Baron et al., 1970). Despite its widespread adoption, the OPF model is not able to explain the intermittent behaviour mentioned previously, but it has provided a solid theoretical basis for Gawthrop et al. (2011) to formally introduce a computational-level theory for human control systems which is built around the intermittent control framework.

The intermittent control model, as presented in Gawthrop et al. (2011), has been used recently to evaluate why the biological control architecture might process parallel sensory information through a serial process that involves planning, selection, and inhibition of other plausible responses before generating a low dimensional motor output (van de Kamp et al., 2013a). The multi-segmental models of human standing with several degrees of freedom are redundant in the sense that the motor system generates parallel feasible solutions to achieve the same configuration or goal, yet it has to select instantaneously only one of them. From a continuous point of view, the solution for this problem is given by a planning stage that sets task goals and optimisation constraints, modulating low-level feedback controllers (Todorov, 2004; Lockhart and Ting, 2007; Karniel, 2011). This paradigm does not provide clear insight between the relationship of sensory input and motor output since the process of selecting an appropriate response from the pull of available solutions is often disregarded. Additionally, van de Kamp et al. (2013a) showed that refractoriness is relevant in whole-body control tasks. While visual information about the anterior-posterior position of the head was made available to the subject by showing it as a pointer on a screen, the subjects were asked to sway forward or backwards in order to make this pointer match a second target pointer that was moving according to a step-wise predefined sequence. The results showed that the delays were considerably different when the step of the target pointer was in close temporal proximity to a preceding step, providing evidence of refractoriness and of an underlying serial ballistic process.

Recently, Gollee et al. (2017) relied on intermittent control to provide an explanation for the nonlinear component of human motor output called the *remnant*. Traditionally, the remnant has been attributed to random sensorimotor noise coming from multiple sources and it has been explained from a computational point of view using continuous control with suitable added noise (Kleinman, 1969; Kleinman et al., 1970; Kooij and Vlugt, 2007; Van Der Kooij and Peterka, 2011; Kiemel et al., 2011).

However, the results shown by Gollee demonstrate that for the first and second order visuo-manual tracking tasks of the joystick experiments of Loram et al. (2011), the motor output comprised by a linear component and the nonlinear remnant can be explained by the triggering mechanism of intermittent control, which uses event-driven sampling based on prediction and thresholds. Moreover, the intermittent control model explained the total joystick power more accurately than a continuous model. The authors suggest that since executive decision making is related to aperiodic sampling processes within frontal, stratial, and parietal networks in the brain, then these structures are important for visuo-manual tracking tasks. The analysis methods presented in their work might contribute in the study of the links between executive functions and sensorimotor control, which is particularly relevant to neurological disorders like Parkinson's disease.

Evidence of movement intermittency has been observed also in complex motion tasks such as grasping and drawing. Doeringer and Hogan (1998) performed arm motion experiments to find out if a series of preplanned actions could appear in a drawing task even if visual feedback was not given to the subject. The results show that intermittency is not affected significantly if no visual aid is given and that it is not due to an additive noise process at the output of the system, suggesting that it is a direct result of how the movement is planned, being a fundamental feature of motion behaviour. Experiments in rapid pronation/supination movements in monkeys during a tracking task also showed that the observed movement irregularities could not be explained as a continuous control process, suggesting that they are a direct result of an intermittent control mechanism (Fishbach et al., 2005). Force field compensation has been a common paradigm to test movement under the influence of external perturbations. The results of Squeri et al. (2010) show that the overall control patterns in a visuo-manual tracking task are not affected by the influence of a large deviating force field and that they are characterised by intermittency.

Overall, the amount of evidence supporting intermittency in human motor control is significant and well documented. Nevertheless, different structures of the intermittent controller have been proposed over the years. The following section presents the most common representations of an intermittent controller from an engineering perspective and functional descriptions of the most important elements across them.

2.3 Models of intermittent control

The demanding control goals of most engineering problems have forced engineers to design algorithms that are flexible, capable of adapting to new conditions and to comply with operation conditions. Many applications are restricted in the sense that the outputs and inputs are only feasible if they are defined within a specific range, in other words, they are constrained. The development in computational power allowed engineers to implement optimisation strategies that would guarantee stable solutions while meeting constraints. A powerful scheme that is greatly influenced from this ideas is known as *model predictive control (MPC)*. This strategy has been used in many areas of engineering with interesting results (Morari and Lee, 1999; Wang, 2009) and it helped Ronco et al. (1999) to propose the first modern version of intermittent control. In discrete MPC, an optimisation procedure takes place every iteration or time step based on a time window, commonly known as moving horizon. This prediction calculates a control action for the entire moving horizon, yet it only applies the first value of the sequence, disregarding the rest. This requires that the optimisation procedure must finish within one time step to avoid timing problems. This problem was avoided by using a time window that remained static for some time, applying an open-loop control signal that was a result of the optimisation procedure; once the time in the moving horizon elapsed, a new optimisation is performed an a new control signal is calculated. This clever reformulation of the problem allows computationally expensive optimisation procedures to finish over an extended amount of time. This ideas were tested in a simulated inverted pendulum showing that both MPC and the intermittent approach give similar results when the optimisation windows are small; however, the intermittent controller gives better results when the sampling times are increased, also when un-modelled dynamics are considered in the simulation and in the presence of observation noise.

The work of Ronco et al. (1999) assumes that the moving horizon, or in other words, the open-loop interval is fixed, which is similar to the ideas proposed by Neilson et al. (1988) in their three stage model. Alternatively, the findings of Navas and Stark (1968); Loram et al. (2012) related to fixed sampling (or clock-driven behaviour) are not able to explain reaction times to discrete stimuli, leading to the suggestion that intermittent control adjusts the open-loop windows based on an error signal crossing a threshold, generating events. This resulted in a more general form known as event-driven intermittent control (Gawthrop and Wang, 2009a; Bottaro et al., 2005, 2008; Åström, 2008; Asai et al., 2009; Kowalczyk et al., 2012), which is capable of reproducing clock-driven control as a special case as well as continuous control. This property is powerful since it gives an explanation to cases where external perturbations or un-modelled dynamics

affect the system in such a way that events must be generated all the time to reject their effect. Very interesting results were then derived from a frequency point of view for clock-driven intermittent control, exploiting the regularity of a fixed open-loop interval to perform a frequency analysis (Gawthrop, 2009). These ideas were used by (Gollee et al., 2012) to propose a two-stage approach for the frequency domain identification of a closed-loop system, defined by the manual control of a virtual unstable load. The first stage involved the calculation of the frequency response of the system based on the experimental data. The second stage was the fitting of different parametric models, defined in this case by continuous and intermittent predictive controllers, as well as a non-predictive continuous version. The results show that both predictive controllers (continuous and intermittent) provide equally valid descriptions of the data, whereas the non-predictive controller resulted in time-delay estimates that were significantly smaller than the ones obtained with the predictive versions.

The alternation between closed and open-loop regimes, regardless if it is clock or event-driven, is what constitutes an intermittent controller and its most distinctive feature. In biological control systems this alternation might be helpful to understand the causality of actions and the properties and effects of external disturbances; some experiments show that during sustained movement, sensory perception is attenuated (Chapman et al., 1987; Collins et al., 1998), probably triggered by the expectation of voluntary movement (Voss et al., 2008). But perhaps what goes on during two different instances of feedback is another fundamental point of distinction between intermittent controllers.

Based on the concept of a switched system that turns on and off a stabilising controller, a possible way to describe an intermittent controller is by not applying any control signal during the open-loop intervals. This can be seen as a zero control approach that is alternated with a control signal from a feedback controller. This idea has been used by (Insperger, 2006; Asai et al., 2009; Kowalczyk et al., 2012) to propose intermittent models. Using a different approach, Gawthrop and Wang (2007) proposed an intermittent controller that would generate similar output properties compared to those of a continuous controller. They proposed that the open-loop evolution should be dictated by a dynamical system that matches the closed-loop response of the system. For this reason, this feature was called the *system-matched hold (SMH)*, referring to a hold mechanism that generates this signal, which is updated by feedback every time there is an event (Gawthrop and Wang, 2011). This resulted in an important property of intermittent control which is that its behaviour can be indistinguishable compared to a continuous controller, in the absence of external disturbances, and delayed only by a minimum intermittent interval (Gawthrop et al., 2011).

This *masquerading* property is a central idea in the interpretation of biological systems driven by intermittent control. The fact that an open-loop control model replicates the overall closed-loop behaviour of the system provides a solid ground to establish triggering mechanisms that would generate the events mentioned earlier. In event-driven control, an event is created if there is a significant discrepancy between the feedback information coming in the form of system states and the hold states. Normally, such a difference might be introduced by disturbances or even by noise. If this difference is greater than a predefined threshold, then the triggering mechanism creates an event. The events indicate that the controller should rely more heavily on the use of feedback to reduce the difference. This version of intermittent control was implemented in real-time engineering structures in (Gawthrop and Wang, 2006; Gawthrop and Gollee, 2012) opening the path to its implementation in other kind of systems.

Influenced by the results in (Loram et al., 2011), an alternative version of intermittent control was generated based on inter-sample behaviour generated by orthogonal functions and in particular, Laguerre polynomials (Gawthrop and Gollee, 2012), called *intermittent tapping control (ITC)*. This exploits the simplicity of Laguerre polynomials to generate control signals that approximate the notion of a *tap* or in other words, impulse-like control signals with a short time constant (i.e. as in the gentle tap approach to control a virtual inverted pendulum described in (Loram et al., 2011)). This version was proposed as a discontinuous approach to control mechanical systems, with possible applications to systems where nonlinear friction might be a problem, helping to overcome stiction in order to initiate movement.

Lastly, the SMH and ITC versions of intermittent control are both closely related to the OPF model by Kleinman (1969) since both implement predictors to compensate for feedback and processing delays. However, it is known that as time delays grow in a continuous control system, the stability margins get reduced (Goodwin et al., 2001; Bottaro et al., 2008) while intermittent control is more robust to this condition (Gawthrop et al., 2011). Prediction in this context is not only restricted to overcoming the effects of the different delays present in human control systems, it also considers the predictions that the brain makes about the consequences of applied motor commands based on internal-forward models operating optimally. These ideas have been studied and integrated in a framework relying on the concepts of optimal control and Bayesian integration of sensory information and predictions, which is considered the dominant paradigm to understand motor control from a computational point of view (Miall et al., 1993b; Miall and Wolpert, 1996; Wolpert et al., 1998; Bhushan and Shadmehr, 1999; Todorov and Jordan, 2002; Todorov, 2004; Shadmehr and Krakauer, 2008). Similarly, there are theories that introduce prediction as an intricate part of perception; where

the minimisation of prediction errors is what drives motor commands (Friston, 2008; Friston et al., 2009; Friston, 2010, 2011; Friston et al., 2011; Clark, 2013). This concept combines a forward model that outputs predictions of sensory inputs and state estimation into one *generative* model, which is expressed in terms of a Bayesian filtering problem that takes into account prior information and beliefs. It also makes a substantial difference in terms of replacing the traditional cost functions associated with optimal control by recasting them in terms of Bayesian inference.

Although explicit state-prediction might not be necessary to design a simple intermittent controller from an engineering perspective, it definitely needs to be included to model human control systems using control engineering concepts, where feedback and processing delays play a major role in the system dynamics.

2.4 Humanoid motion control

The rapidly evolving nature of technology and the widespread use of machines in everyday life has left us with an important question: Are machines going to be as intelligent as a human being? and if they do, would they look like like us? Since the beginning of cybernetics and automation, the race to build reliable and useful robotic machines started. Probably the field of industrial robotics got most of the attention because of the financial implications of having faster and more precise manufacturing, resulting in carefully optimised supply chains and reduced costs. However, the idea of developing a robot that is indistinguishable from a human is probably attractive and frightening at the same time, since it poses philosophical questions in many areas of man-machine interaction.

Building such a robot is certainly one of the main motivations behind many research teams and companies around the world, where not only the built-in intelligence must allow flexibility and computational power, but also the movements should be smooth and precise, being able to adapt to any possible scenario. These ideas resulted in the development of *soft robotics*, which aims to build flexible robots from compliant materials, mimicking the behaviour of living organisms.

Although the progress in terms of movement control for robots has been outstanding, it is still clear that most humanoid robots move in a mechanical way that could not be associated to that from a human (except from some notable exceptions), and that it is not easy to design effective control algorithms for this purpose. In this sense, appropriate bio-mechanically inspired motion controllers are needed, to leave behind

the conventional kinematic motion methods that have been inherited from industrial manipulators which are mainly based on the continuous use of feedback (Tomei, 1991; Qu and Dorsey, 1991; Kelly, 1997; Hsu and Fu, 2006; Park and Chung, 2000).

Computational power has been an important consideration in terms of determining the appropriate control method in humanoid robots. The earliest robots had very limited capabilities to compute motion trajectories online, therefore, roboticists relied on offline trajectory generation which implemented optimisation routines that took into account energy expenditure and planar models (Cabodevila et al., 1995; Chevallereau et al., 1998; Roussel et al., 1998). The start of the 21st century brought the first optimal motion patterns based on complete nonlinear dynamics, including enough computational power to consider open-loop trajectory stability and mass distributions (Buschmann et al., 2005; Mombaur, 2009). Still, the computations associated with these schemes were too complex to be performed online in order to meet the real-time constraints of the system, this resulted in control approaches that used banks of offline generated trajectories that were retrieved online depending on the state of the robot, to then be stabilised by a feedback controller (Denk and Schmidt, 2001; Wieber and Chevallereau, 2006; Liu and Atkeson, 2009; Tedrake et al., 2010).

In recent years, numerical optimisation routines developed considerably and processing power stopped being a such a strong limitation, this led to more online based control algorithms and planning strategies in humanoid control. Most of the work has been based on simplified inverted pendulum models (Kajita et al., 2001; Loffler et al., 2004; de Boer, 2012; Englsberger et al., 2011), with solutions that applied finite difference approximations (Kagami et al., 2002), optimal control theory (Urata et al., 2011; Hu and Mombaur, 2017; Hawley and Suleiman, 2017), and MPC (Mayne et al., 2000; Erez et al., 2013; Koenemann et al., 2015; Castano et al., 2016). In this context, MPC has gained popularity due to relatively simple implementation and the flexibility to deal with hard constraints (Richalet, 1993; Naveau et al., 2017).

The inverted pendulum model has been used significantly to derive controllers for humanoid robots, allowing descriptions that capture the essential features of the whole-body dynamics. The control approach taken most of the time implies the careful tracking of the CoM with respect to the contact points and the reaction forces they produce. The Zero Moment Point (ZMP) is normally defined as the distribution of forces around the robot's foot and the ground (Vukobratović and Stepanenko, 1972). The relationship between the CoM and the ZMP has been captured in models that have been used to provide motion trajectories and stable waking patterns in humanoids. Wieber (2006) used the ZMP in combination with linear MPC to generate stable walking mo-

tion in the presence of strong lateral perturbations, including a stability analysis of the control scheme and the inclusion of constraints on the ZMP. Another example is the balance and walking patters achieved in Honda's Asimo robot, which used ZMP as a measure of dynamic balance and as a reference, to implement MPC based on a three mass model. After the introduction of the concept of the Capture Point (CP) by Pratt et al. (2006), as the point on the ground where the control mechanism should place the ZMP in order to stop the motion of the CoM, Krause et al. (2012) provided a solution in terms of MPC to stabilise the unstable dynamics related to the CP by relying on feedback instead of an online pattern generator. They achieved this by separating the problem into a an outer CP control loop which generated an input to an inner loop ZMP controller.

The work by Koenemann et al. (2015) also uses MPC in combination with an optimal control strategy for the problem of whole-body motion on the HRP-2 robot. This was the first implementation of its kind on a humanoid robot directed to balance control including constraints such as collisions and joint limits. Similarly, the work by Castano et al. (2016) involves the use of Robust MPC, on the COMAN humanoid robot, to compensate for the deviations of the CoM using a primary ZMP controller. By incorporating a second stabilising controller in charge of body orientation, they were able to reject external disturbances.

MPC is based on an receding horizon optimisation procedure that calculates a control signal within a defined time-step. The fact that MPC can be applied to a wide variety of systems including actuator limitations and allowing safe operation closer to actual system constraints, imposes a trade-off between generality and efficiency, that leads in some cases to computationally demanding routines. A way of dealing with this timing requirement is to design efficient and fast optimisation routines (Slotine and Yang, 1989; Diehl et al., 2006; Todorov, 2004; Todorov and Li, 2005). The work of Alamir and Boyer (2003); Alamir and Marchand (2003); Alamir (2004); Alamir and Boyer (2006) shows that MPC implementations might be obtained for problems where constrained stabilisation is required; in fact, they are based on the efficient calculation of open-loop steering trajectories. These trajectories are system dependent, therefore in order to improve the efficiency of the overall algorithm, a low dimensional parameterisation of them can be obtained resulting in a low dimensional MPC problem. Similarly, Bobrow et al. (2006) developed efficient optimal solvers for inverse and forward dynamics through the use of Riccati differential equations, applying them to under-actuated robots and one-dimensional hopping structures. Along the same line, the work of Featherstone (2010a,b) uses spatial vectors to reduce the number of equations and algebraic operations involved in solving dynamic equations and

kinematic chains using recursive Newton-Euler algorithms. This has been exploited by roboticists to simplify optimisation routines in robotic structures. Featherstone's methods are particularly suitable for robots having branched connectivity, which include robot arms, multifingered grippers and humanoid robots.

Neunert et al. (2014) proposed a control scheme that used full-state feedback MPC on a real robot called Rezero. To solve the optimisation an iterative linear quadratic Gaussian method was implemented in real-time and in an efficient way since it only requires first order derivatives of the system dynamics. This scheme uses an outer control loop that modulates the MPC algorithm influencing specific features like the time horizon and the cost function used in the optimisation procedure or even change the assumed system dynamics. This outer loop effectively introduces changes in the behaviour of the robot in a similar way to the proposed architecture of (Loram et al., 2009; van de Kamp et al., 2013a), the similarity relies on having a high-level controller that provides goals and references to low-level fast continuous controllers. These ideas allow us to hypothesise that intermittent control might be able to not only perform as high-level controller in humanoid robotics but also as a mechanism that reduces the computational load on the system by introducing open-loop intervals.

2.5 Adaptation and intermittent control

Adaptation can be defined broadly as the processing of sensory information, in a changing environment, with the purpose of adjusting the control law in order to improve the overall performance for a particular task. This is a very simple definition, yet it illustrates a powerful capability of biological control. Human motor control not only has to be adaptive, but it also needs to learn new skills, implement new functions, and find optimal performance (Karniel, 2011). This point of view by Karniel is helpful to understand what adaptation means in this context, and with the use of concepts from control engineering it is possible to establish a more specific definition. A visual representation about the elements that constitute adaptation is given in Fig. 2.1, where a distinction is made in terms of the amount of knowledge about the system to be controlled (or the task to be performed), and in terms of what elements of the controller are being adjusted.

From a control engineering perspective, the overall structure of the system must be known in order to formulate an appropriate controller, this includes the parameters (coefficients) that are affecting the main variables of the system. If this is the case, then a fixed control law is suitable, meaning that the gains or constants describing

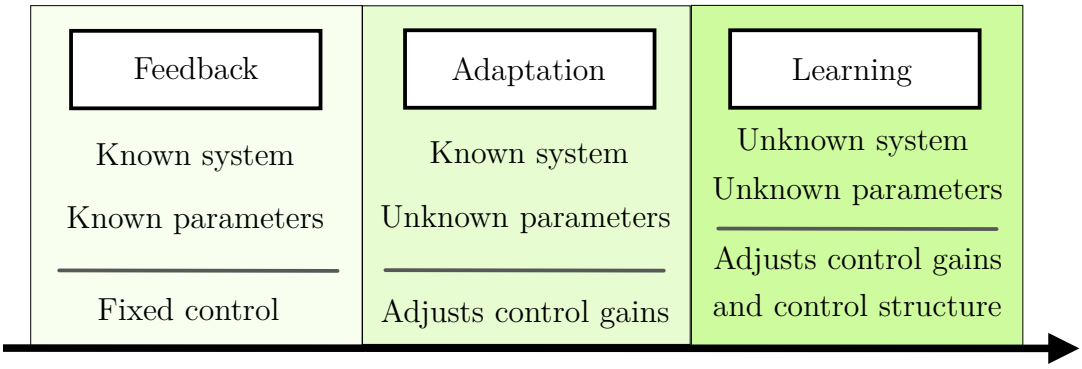


Figure 2.1: Feedback - Adaptation - Learning diagram. These mechanisms are shown in terms of the available amount of knowledge about the system structure and its parameters, with a distinction based on what control features are adjusted.

the control law do not change throughout the task or during the time the controller is operating. This scheme implies that the control signal is modulated only by the changes in the feedback signals and that there are no variations in both the structure and parameters of the system; the control signals are generated in real-time delayed only by the inherent transmission delays. This situation is represented in Fig. 2.1 by the box labelled as *Feedback*.

There are also situations where the system parameters are not known precisely or they even have a time-varying nature, while the system structure remains the same. This is what we would consider throughout this thesis as *Adaptation* (represented by the middle box in Fig. 2.1), since in order to reduce the effects of the uncertainties, the parameters must be estimated and used to adjust the control law accordingly. An important distinction is that adaptation forgets about a particular configuration as new information comes in, meaning that if the system parameters return to a set that the controller has *seen* before (due to variations), it would still try to obtain a new set of gains for the control law based on estimates, instead of recalling it from a memory mechanism. In contrast with a feedback only approach, an adaptive controller not only modulates the control signal by using the sensory signals provided as feedback, but it also does it by adjusting the gains of the control law based on the available information.

Learning is a process that would use both feedback and adaptation against parametric uncertainties, but it would also be capable of adjusting the structure of the control law in order to produce a new behaviour or function. This can be seen as using a different control strategy depending on the task or the structure of the controlled system. While it is possible to argue that adaptation might also modify the structure of the control law specially when a specific parameter takes a value of zero, we shall consider this situations as special cases of parametric adaptation. A learning system

then, uses these changes to gradually improve the performance for a known task, while being flexible to adjust itself if there is a new task to learn. This means that learning also identifies structural changes in the system that is being controlled, in contrast with adaptation, which is used for the case when parametric changes are present. Some forms of adaptation might be able to also modify the structure of the control law, as in learning. For instance, a switching control system might achieve this by using an array of control laws, which might be structurally different between them, and switch according to a specific criterion. The difference between this adaptation approach and a learning system relies in the fact that learning finds out the correct control law (including its structure) on the go, without or with a little amount previous knowledge, whereas a switching system uses previously tested controllers to deal with different operating conditions and changes in the environment.

In human motor control, adaptation and learning are processes that are tightly related, and in some cases it is not easy to establish a clear line as to when adaptation ends and when learning starts taking place. Nevertheless, Fig. 2.1 provides a simple classification that relies heavily on the concept of a *System*, viewed as a process that can be described by a set of differential equations that are weighted by parameters. If the parameters change or if there is uncertainty about their values, then adaptation is the process that tracks the varying parameters or obtains the unknown values using an estimation procedure, to then adjust the control parameters or gains.

Based on the previous ideas, it can be said that the main features of the biological controller are flexibility and generality. With the emergence of computational models of the human controller, ideas coming from adaptive control and learning theory have been used to give a functional explanation of these features. Normally these models are based on mathematical descriptions of the system to be controlled, which are improved by fitting the parameters of the model to experimental data. This is not a simple procedure since the model would have to explain reliably a highly nonlinear system, that is redundant in many levels. Still, there are many examples of adaptation models that have been able to replicate some properties in motor control.

Donchin et al. (2003) studied adaptation occurring trial to trial of a task that involved reaching movements in a velocity-dependent force field. The results showed that under the assumption of a fixed desired trajectory, error in a specific movement direction could be generalised in a bimodal pattern to directions that are in close proximity, suggesting that adaptation can be described as a multidimensional hidden state that changes according to the errors experienced in previous trials and depending on the basis functions used to represent the task. In similar experiments, Izawa et al. (2008)

evaluated reaching movements in velocity dependent force fields from a stochastic point of view, where the variance of the force field changed every trial and the force was perpendicular to the direction of motion. In these experiments the results showed that overcompensation in the force produced by the subjects disappeared and the peak movement speed increased. In a deterministic environment, overcompensation does not occur. These results imply that motor adaptation is a process of adjusting our internal models through time when facing a new environment rather that cancelling the effects of it to return to a baseline state.

Motor adaptation has been tested in bimanual tasks involving rhythmic movements of the index finger (Klaiman and Karniel, 2006), showing evidence of the existence of internal representations of coordinative motor tasks. In these experiments, the subjects were trained using altered visual feedback of the required tapping frequency, nevertheless a gradual decrease in the task performance error was observed throughout the experiment which was visible from the beginning of the training trials; this suggests a learning process that starts a soon as the subject goes into training. Also, they observed aftereffects and washout when the altered visual feedback returned back to normal suggesting an underlying adaptive process. Levy et al. (2010) performed interesting experiments linking motor adaptation and delayed force perturbations. Two groups performed reaching movements under the influence of force fields, one did it without any delay, the other had a delay of 50 ms acting on the applied force. They observed deviations from the reference trajectory that were shifted in time between test and control groups during catch trials (trials where the forced field was turned off), which indicates that the subjects were expecting a force field. This shows that subjects successfully adapted to the 50 ms delay. An important conclusion supported by this study is that the internal representation that the brain uses to perform this task is capable of employing time representations, suggesting that the adaptation mechanisms might actively compensate for the varying delays in the sensorimotor system.

Adaptation in control engineering has been studied extensively with applications in different areas (Åström and Wittenmark, 1995; Feng and Lozano, 1999; Landau et al., 2011); however, many opinions and feelings about what constitutes an adaptive controller have been exposed, without getting to an actual agreement for a long time. Eventually, the field converged to the idea that if a controller is capable of changing its parameters to obtain a better performance, then this would be some kind of adaptive controller. Often, an adaptive controller is seen as controller that learns how to operate in a new environment to produce acceptable performance, this point of view puts adaptive control as some sort of very basic learning machine.

Adaptive controllers have been proposed mostly for cases where the plant can be represented as a linear time-varying system, where the parameters associated to the dynamical equations change through time. In that sense, and adaptive controller has two main purposes: i) to detect this variations in the parameters or in the output values, ii) and adjust a predefined control law according to this changes. This can be done in different ways, for instance, a model of the desired output behaviour can be built in order to compare it against the output of the real system. The difference between the two outputs is then used to drive an adjustment mechanism that attempts to make the outputs match. This strategy is known as *model reference adaptive control (MRAC)* (Chalam, 1987). Also, the concept of *gain scheduling* as explained by Narendra and Annaswamy (1989), has been used as a basic form of adaptive control to deal with systems over a wide range of operating conditions. The basic idea is that for some systems the amplitude of the control input might change substantially as the system enters a different operating condition and in order to keep the output at the desired average value, a gain is adjusted. The key concept is that before the execution, an array of gains can be computed for each of the operating conditions that the system might be working on, and assuming that measurements are available, the controller selects between them online.

Perhaps the most common form of an adaptive controller is to use a self-tuning architecture (Åström et al., 1977; Clarke and Gawthrop, 1981; Gawthrop, 1982), with direct and indirect representations. The indirect self-tuning architecture uses an online parameter estimator to identify the varying parameters of the plant, it also uses a control design algorithm that takes the newly estimated parameters to adjust the control law accordingly. It can be seen as a two stage sequential approach. On the contrary, a direct approach is parameterised in such a way that parameter estimation routine would yield the controller parameters directly, without the need of a control design stage. An advantage of indirect self-tuning controllers is that the parameter estimation algorithm is completely independent from the design stage, allowing for easy testing and clear differentiation of the two stages.

Most of the aforementioned adaptive controllers are based on *continuous* monitoring of inputs and outputs in order to estimate parameters, and a subsequent *continuous* redesign of the controller gains. If the parameter estimation procedure is operating, then the control law is adjusted every sampling interval. Some authors suggest that once the parameters have been estimated accurately, the parameter estimation procedure can be disabled in order to avoid parameter drift (Giri et al., 1991), which is a consequence of the low excitation levels of the control input, specially once the system reaches the desired steady-state. Ensuring that the controller will produce a persis-

tently exciting signal is one of the main concerns of adaptive control in general, since the estimation procedure must remain aware of any possible changes in the parameters at all times while maintaining its stability properties, this trade-off has been formulated in other contexts as the *stability-plasticity dilemma* (Carpenter and Grossberg, 1988). The introduction of auxiliary signals that provide extra levels excitation, allowing the parameter estimation procedure to converge to the correct values even when not much is happening in the system, has been proposed as a counteract measure (Landau et al., 2011); however, this is not always possible, specially in some restricted real-time environments, with the potential drawback of increasing the steady-state error of the system outputs.

In the previous sections, the virtual inverted pendulum experiments performed by Loram et al. (2011) were discussed. One of the important conclusions from this work is that opening the loop for a small amount of time using a tapping strategy had interesting benefits, arguing that between taps it is easier to establish the *causality* of the signals in the system. Also, they suggest that the impulse-like control signal that is generated by the tapping strategy provides a natural mechanism to probe the system constantly. It is possible to relate this idea to the persistent excitation problem of continuous adaptive control exposed in the previous paragraph, suggesting that a hybrid strategy such as intermittent control might provide a control signal that excites the system in a way that parameter estimation can be performed while having output errors that are comparable to those from a continuous approach. These ideas led to the first formulation of adaptive intermittent control.

Gawthrop et al. (2015) explored the application of adaptive intermittent control in simulation examples covering human standing and reaching movements under a force-field. In both cases, the system was expressed in a non-minimal state-space (NMSS) form that is linear in the parameters (Young et al., 1987, 1991; Taylor et al., 2000) under an indirect self-tuning configuration. The NMSS approach allows to measure the states directly, therefore an observer is not needed. In this case, continuous-time parameter estimation was implemented using recursive least squares methods with a forgetting factor (Åström and Wittenmark, 1995; Åström et al., 1977) in combination with state-variable filters (Young, 1981). Gawthrop's results showed that it is possible to evaluate adaptation in a human motor context with parameter estimation; additionally, a suggestion is made in terms of exploiting the structure of self-tuners in order to test other state-space estimation methods such as Kalman filters, which can be extended easily to a multivariable scenario.

Kalman filters have been used in many areas of engineering since their introduction

(Kalman, 1960), specially after many improvements that have allowed the original structure of the filter to deal with nonlinear problems Julier et al. (1995); Simon (2006). These filters are unbiased, minimum error variance recursive algorithms that can be used to optimally estimate the unknown states of a dynamical system as well as its parameters, from noisy data measured in real-time. Because of its properties, the Kalman filter is the best possible linear estimator (Haykin, 2001). It has a successful history of applications where both parameters and states of the controlled plant had to be estimated (Van Der Merwe and Wan, 2001; Simon, 2006; Manganiello et al., 2015; Morrison and Cebon, 2016).

Many interesting versions of the filter have been proposed recently, however, the ones that are probably more common in literature are known as Extended and Unscented Kalman filters. The Extended Kalman filter (EKF) is a powerful approach in the sense that it uses a linearisation procedure to obtain linear representations of a nonlinear plant every time step, reducing the complexity of the problem considerably (Mcgee et al., 1985), whereas the Unscented Kalman filter (UKF) is based on a statistical unscented transformation that projects mean and covariance estimates of the original nonlinear system. These two filters are interesting from the perspective of adaptive control since they provide a way of solving the state and parameter estimation problem in one single algorithm.

Wolpert (1997) proposed an elegant formulation of the Kalman filter as a fundamental entity in a model of sensorimotor integration. In this paper, the Kalman filter is seen as an internal model that combines predicted and sensory feedback, to generate an optimal state estimate which is used by a forward model that describes the dynamics of a human arm. From this perspective, the notion of the intermittent controller being an *intermittent action, continuous observation* scheme (Gawthrop et al., 2011), allows us to hypothesise that the Kalman filter is an algorithm that fits in the adaptation context accordingly, capable of solving the continuous monitoring of the system states, while tracking time-varying parameters.

The following chapter explains the theoretical aspects of the existing framework of intermittent control, introducing it from a control engineering perspective and providing mathematical details of each of its components. This is followed by the introduction of adaptive intermittent control based on Kalman filters and a self-tuning achitecture to estimate both parameters and states of a given system; this control scheme constitutes an important contribution of this thesis.

Chapter 3

Intermittent control theory and adaptation

3.1 Introduction

The theory of continuous control has been used in the past to describe human motor control (Kleinman, 1969; Baron et al., 1970), where the available sensory information is used as feedback; this fundamental concept of a feedback mechanism commonly implies that the information that the sensors in place are gathering, is used every single instant in time to compute a suitable control action. This is in fact how the vast majority of the control problems in engineering have been solved, backed up by a great amount of research devoted to fully understand the use of different control architectures, where all of them share the central feature of continuous feedback.

Alternatively, the characteristics of some control problems forced the engineers to look for other solutions. For instance, the sensors used to convey the information back to the controller might have a finite rate at which the information can actually be transmitted (Nair and Evans, 2003), leaving the controller with less information available, which in some cases, can result in unstable responses.

Similarly, the system to be controlled might have to comply with input and output restrictions, also known as constraints in the control literature; for example, the output of a system might have to stay bounded to ensure safe operation, or the input should be restricted as well in order to avoid wear or even failure of the devices generating the control signal. These conditions are commonly solved by optimisation approaches (Gawthrop and Wang, 2009b), where an online routine based on the available information should be executed to ensure that the constraints are met. A limitation of such

controllers is that since they have to be implemented digitally, the sampling frequency of the system might not allow the optimisation procedure to finish before the next sample is taken, making the implementation difficult in some cases (Morari and Lee, 1999). Some authors have developed efficient methods to perform faster optimisations in order to circumvent this limitation (Diehl et al., 2006) as well as simplifying the computational effort by converting a nonlinear problem into a linear one (Lee and Ricker, 1993).

The previous ideas gave rise to hybrid approaches that alternated feedback control with open-loop trajectories in a variety of ways, generating a range of controllers that were able to deal systems over low-bandwidth channels and using the open-loop interval to finish complex optimisation procedures (Ronco et al., 1999; Gawthrop et al., 2012, 2013).

In the human motor control context, there is evidence showing that some human control systems are driven by intermittent ballistic actions or events (Craik, 1947; Vince, 1948; Navas and Stark, 1968) and several authors have used the concept of Intermittent Control (IC) as a computational model to support these ideas (Miall et al., 1993a; Bhushan and Shadmehr, 1999; Loram and Lakie, 2002; Loram et al., 2011). These schemes are based on the *psychological refractory period (PRP)* observed by Telford (1931) in the context of double-response time experiments. This result suggests that humans, when asked to respond to a series of rapid stimuli in a manual tracking task, have to wait until the PRP has elapsed in order to initiate a response to the new stimuli in the sequence, becoming unable to respond to newly available information. This concept matches the alternate nature of the hybrid controllers described in previous paragraphs, with the caveat of having a minimum open-loop interval condition before a new set of feedback values could be used.

For these reasons IC has been proposed as computational paradigm to explain human motor control under different experimental conditions (Gawthrop et al., 2011; van de Kamp et al., 2013a,b) and as a suitable alternative to control engineering systems with low computational power, low bandwidth channels and/or with system uncertainty.

As stated by (Gawthrop et al., 2014), there are different architectures of IC. The main difference between most of them is in terms of the inter-sample behaviour, where some approaches reset the control signal to zero according to a predefined criterion (Insperger, 2006; Estrada and Antsaklis, 2008; bin Mohd Taib et al., 2013), and others use a *generalised hold* (Montestruque and Antsaklis, 2003; Gawthrop and Wang,

2007). Another distinction of IC is that it can be set use feedback in fixed intervals, giving a clock-driven (periodic) behaviour, or based on events generated by a threshold mechanism. For the purpose of this thesis, special emphasis is made on event-based IC based on a generalised hold.

This chapter is organised into two main parts, first an overall explanation of existing theory about intermittent control is given, including a simulation example that illustrates its use by means of a simulation case study that involves the inverted pendulum model for human standing. The second part discusses the concept of adaptation and its integration into the IC framework, it introduces a state-space formulation of an adaptive intermittent controller that relies on Kalman filters to perform state and parameter estimation. The formulation of this controller, as well as its validation both in simulation and experimentally, constitutes one of the main contributions of this thesis.

3.2 Intermittent control

The intermittent controller discussed in (Gawthrop et al., 2011), is based on the *observer-predictor-feedback (OPF)* model of Kleinman (1969), which is continuous in nature. This controller was used to model human motor control tasks in the presence of time delays. Fig. 3.1 shows a block-diagram representation of this scheme.

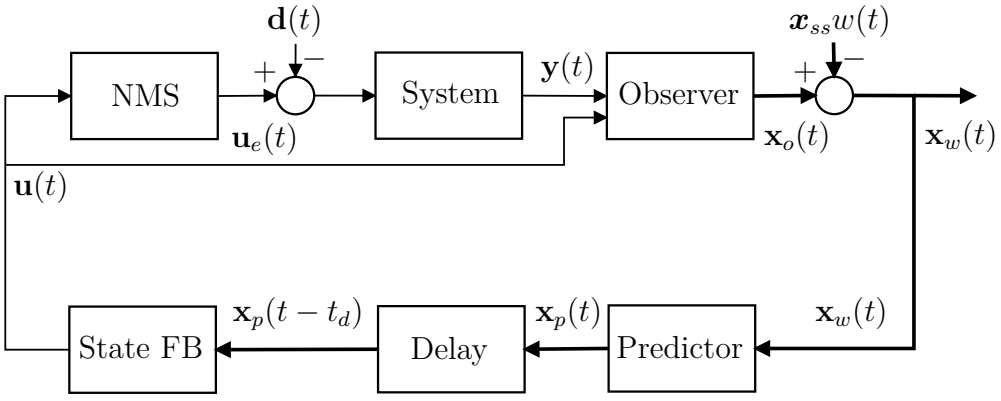


Figure 3.1: The quantities $\mathbf{d}$, $\mathbf{y}$, $\mathbf{u}$, w represent disturbances, outputs, inputs, and setpoints respectively. The product $\boldsymbol{x}_{ss}w$ is the vector version of the setpoint w. The observed states are defined by $\mathbf{x}_o$. The time delay is denoted by t_d which is compensated in the predicted states represented by $\mathbf{x}_p$. $\mathbf{u}_e$ is the control signal after being affected by the dynamics of the neuro-muscular system. The thick lines/arrows of the diagram represent vector signals, whereas the thin ones represent scalar versions, for the single-input single-output case. This figure is based on the representation given in (Gawthrop et al., 2011).

It can be seen from Fig. 3.1 that the model uses an *Observer*, which provides estimates of the relevant states in the system, resembling the integration of sensory information. Similarly, there is a *Predictor* in series with a *Delay*. Their purpose is to model various delays that are present in the human controller, such as the central nervous system delay. They can be seen as a mechanism that predicts the consequences of our motor commands to overcome these delays.

The *State FB* block stands for state-feedback, it has the purpose of computing the appropriate control action based on the predicted states. Finally, the *NMS* block represents the *neuro-muscular system* (a model that captures the dynamics of our muscles/actuators) that would eventually transmit the requested commands to the *System* that is being controlled.

Designing a stable continuous controller as the one shown in Fig. 3.1 is fundamental to implement an intermittent controller. This procedure is called *the underlying continuous control design* and it is based on linear control theory. It is comprised of the following steps: steady-state design, feedback gain design, observer design, and state-prediction implementation. The most important concepts of this procedure are discussed in section 3.2.1.

The IC framework discussed in Gawthrop et al. (2011), extends the underlying continuous controller by adding three key features: a generalised hold, an intermittent predictor, and a triggering mechanism. These features are explained in detail later in this chapter, specifically in 3.2.3, 3.2.4, 3.2.5 respectively. Fig. 3.2 shows a block diagram of the intermittent controller.

3.2.1 Underlying (continuous) control design

In order to ensure acceptable performance levels, this *underlying* continuous controller must be designed carefully. This implies using modern control techniques to define appropriate steady-state inputs and to calculate feedback gains that would comply with specific system requirements. The following sections provide an explanation of each step involved in the aforementioned continuous control design.

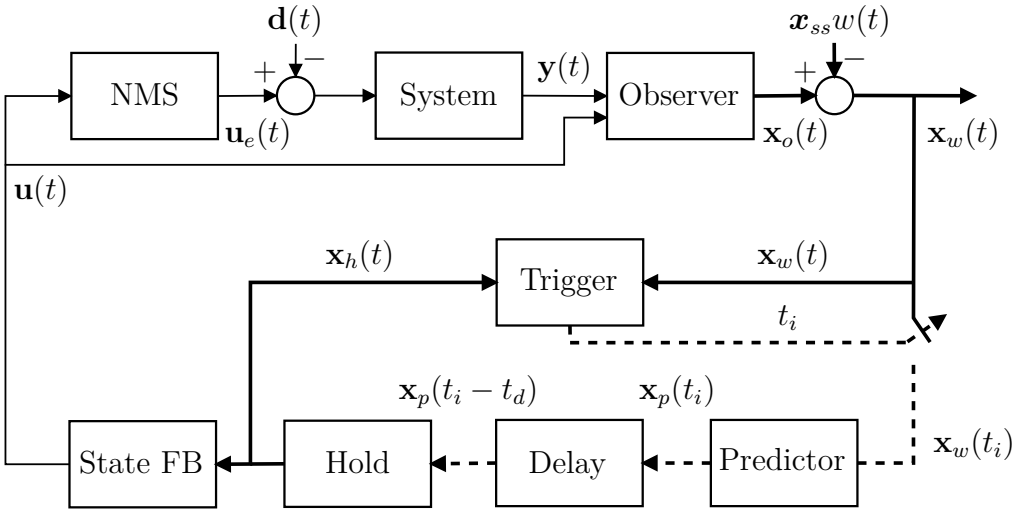


Figure 3.2: Diagram of the intermittent controller (Gawthrop et al., 2011). The hold states $\mathbf{x}_h$ are compared with the state-estimates $\mathbf{x}_w$ in the Trigger block. If the difference exceeds a predefined threshold, then the block create events at times denoted by t_i. The hold states $\mathbf{x}_h$ are used to generate the control signal $\mathbf{u}$ during the open-loop period, and it is reset only at times t_i by the predictor block. The dashed lines represent signals that are defined only at t_i. The thick lines/arrows of the diagram represent vector signals, whereas the thin ones represent scalar versions, for the single-input single-output case. This figure is based on the representation given in (Gawthrop et al., 2011).

Closed-loop system

The plant to be controlled can be described by the following linear dynamical system of order n

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{B}_d\mathbf{d}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) \\ \mathbf{x}(0) &= \mathbf{x}_0 \,, \end{aligned} \tag{3.1}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^{n_y}$, $\mathbf{u} \in \mathbb{R}^{n_u}$ and $\mathbf{d} \in \mathbb{R}^{n_u}$ correspond to the system state, output, input and disturbance respectively, and t represents continuous time. $\mathbf{A}$ is an $n \times n$ matrix, $\mathbf{B}$ and $\mathbf{B}_d$ are $n \times n_u$, and $\mathbf{C}$ is $n_y \times n$.

A state-feedback controller with gain $\mathbf{k}$, of the form

$$\mathbf{u}(t) = -\mathbf{k}\mathbf{x}(t) \,, \tag{3.2}$$

would stabilise the system given that it is *controllable* (Goodwin et al., 2001). The resulting closed-loop system, defined by the state vector $\mathbf{x}_c$, after substituting (3.2) in

(3.1) is

$$\begin{aligned} \dot{\mathbf{x}}_c(t) &= \mathbf{A}_c\mathbf{x}_c(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}_c(t) \\ \mathbf{x}_c(0) &= \mathbf{x}_0 \,, \end{aligned} \tag{3.3}$$

where $\mathbf{A}_c$ is defined as follows

$$\mathbf{A}_c = \mathbf{A} - \mathbf{B}\mathbf{k} \,. \tag{3.4}$$

The selection of the control gain $\mathbf{k}$ determines the stability of the closed-loop matrix $\mathbf{A}_c$. In particular, if $\mathbf{k}$ is chosen in such a way that the real part of all the eigenvalues of $\mathbf{A}_c$ is negative, then the response of the system to finite initial conditions will converge exponentially to zero as $t \to \infty$, and is said to be *asymptotically stable* (Kwakernaak and Sivan, 1972).

Steady-state design

Generally, a control-law would serve two main purposes, one is known as a *regulation* problem, where the controller aims at steering an initial state x_0 to the origin in a finite amount of time by means of an input $\mathbf{u}$ while showing good disturbance rejection properties. On the other hand, if the controller must also force the output $\mathbf{y}$ to follow a reference signal, then this is known as a *tracking* problem.

To solve both problems successfully, the reference signal must be properly introduced to the system equations. There are different ways to achieve this, as stated by Franklin et al. (1994), although for the purpose of this thesis only one method is used, which is described in the following paragraphs.

The control-law in (3.2) is commonly used to solve the regulation problem discussed earlier. If this equation is to be used to control a system that must follow some sort of reference, it will almost surely exhibit an error once the system reaches the steady-state. To reduce this error to zero, the steady-state values of all states and of the control input must be obtained.

Consider the system in (3.1) where $\mathbf{d} = 0$. The system can be written as follows for a constant steady-state regime

$$\begin{aligned} \mathbf{0}_{n\times 1} &= \mathbf{A}\boldsymbol{x}_{ss}(t) + \mathbf{B}\boldsymbol{u}_{ss}(t) \\ \boldsymbol{y}_{ss}(t) &= \mathbf{C}\boldsymbol{x}_{ss}(t) \,, \end{aligned} \tag{3.5}$$

where $\boldsymbol{x}_{ss}$, $\boldsymbol{u}_{ss}$, and $\boldsymbol{y}_{ss}$ correspond respectively to the steady-state versions of the states, inputs, and outputs. Equation (3.5) can be solved for the case when the steady-state output is equal to the reference trajectory or setpoint $\boldsymbol{y}_{ss} = w$, for any value of w. For the case where the output is $\boldsymbol{y}_{ss} = 1$, it is possible to rewrite the steady-state system (3.5) as follows

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{ss} \\ \boldsymbol{u}_{ss} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{n\times 1} \\ 1 \end{bmatrix} , \tag{3.6}$$

and then solve for $\boldsymbol{x}_{ss}$ and $\boldsymbol{u}_{ss}$. Thus, the control input in (3.2) can be redefined as

$$\begin{aligned} \mathbf{u}(t) &= -\mathbf{k}\left(\mathbf{x}(t) - \boldsymbol{x}_{ss}w(t)\right) + \boldsymbol{u}_{ss}w(t) \\ &= -\mathbf{k}\mathbf{x}(t) + \left(\boldsymbol{u}_{ss} + \mathbf{k}\boldsymbol{x}_{ss}\right) w(t) . \end{aligned} \tag{3.7}$$

By defining $\mathbf{r} = \boldsymbol{u}_{ss} + \mathbf{k}\boldsymbol{x}_{ss}$, a simplified expression is obtained.

$$\mathbf{u}(t) = -\mathbf{k}\mathbf{x}(t) + \mathbf{r}w(t) . \tag{3.8}$$

Expression (3.8) can be used for both *regulation* and *tracking* problems, allowing the designer to compute matrices $\mathbf{r}$ and $\mathbf{k}$ offline. It is clear now that the only quantity left to define in (3.8) is the feedback gain $\mathbf{k}$, which multiplies the system states.

One common procedure to obtain $\mathbf{k}$ is to use a method known as *pole-placement*, where the main advantage is that the poles of the closed-loop system can be arbitrarily fixed at desired locations. Another possibility is to use an optimisation approach, commonly referred to as the linear quadratic regulator (LQR) (Kwakernaak and Sivan, 1972; Franklin et al., 1994; Goodwin et al., 2001), which is a part of optimal control theory. This method is briefly explained in the following section.

Linear quadratic regulator

The LQR method is based on the optimisation of a cost function (also known as Performance Index) that depends on both the states and the inputs. Consider the following cost function:

$$J_{LQR} = \int_0^\infty \left[\mathbf{x}(t)^T\mathbf{Q}_c\mathbf{x}(t) + \mathbf{u}(t)^T\mathbf{R}_c\mathbf{u}(t)\right] dt . \tag{3.9}$$

Equation (3.9) can be interpreted as function that describes the energy of the closed-loop system described by (3.3). If J_{LQR} is kept small, then the energy of the closed-loop system would be small as well. Both $\mathbf{x}$ and $\mathbf{u}$ in (3.9) are weighted by design matrices

$\mathbf{Q}_c$ (an $n \times n$ matrix) and $\mathbf{R}_c$ (an $n_u \times n_u$ matrix). A large $\mathbf{Q}_c$ means that in order to keep J_{LQR} small, the state $\mathbf{x}$ should be smaller. Similarly, selecting a large $\mathbf{R}_c$ means that the control input $\mathbf{u}$ must be smaller to keep J_{LQR} small. In terms of the closed-loop system, a large $\mathbf{Q}_c$ would place the poles of (3.4) further to the left of the s-plane resulting in a faster decay to zero. A large $\mathbf{R}_c$ would result in in less control effort used to drive the system and poles that are slower (closer to the s-plane origin) and as a consequence, larger values of state $\mathbf{x}$.

The selection of the design matrices should ensure that $\mathbf{Q}_c$ is selected to be positive semi-definite and $\mathbf{R}_c$ to be positive definite. This guarantees that the cost function J_{LQR} is well defined by forcing the scalar $\mathbf{x}(t)^T\mathbf{Q}_c\mathbf{x}(t)$ to be always greater or equal to zero for all t and the term $\mathbf{u}(t)^T\mathbf{R}_c\mathbf{u}(t)$ to be positive for all values of $\mathbf{u}$ and t.

To obtain the closed-loop gain $\mathbf{k}$, the following equation must be considered

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} + \mathbf{Q}_c - \mathbf{P}\mathbf{B}\mathbf{R}_c^{-1}\mathbf{B}^T\mathbf{P} = 0\,, \tag{3.10}$$

where (3.10) is known as the algebraic Riccati equation (ARE), which can be solved for the positive-definite matrix $\mathbf{P}$. Once the solution is known, the control gain $\mathbf{k}$ can be obtained as follows

$$\mathbf{k} = \mathbf{R}_c^{-1}\mathbf{B}^T\mathbf{P}\,. \tag{3.11}$$

Solving equation (3.10) is not a simple task and a great amount of work has been done to obtain reliable, numerically stable solutions. Most modern solvers use a combination of matrix factorisation/manipulation routines and iterative methods to solve the problem as described by Arnold and Laub (1984). A good introduction to some of these methods and how to implement them in real-time can be found in Bini et al. (2011).

State observers

In the previous sections, it has been assumed that all the states that describe the system behaviour are accessible for measurement. The reality is that in many situations it is not possible to physically measure all variables, this poses a problem to all state-feedback approaches since the control input relies heavily on past information of all states. To solve this, a virtual sensor or *observer* can be implemented, that would have the main goal of estimating the states that can not be measured based on the available measurements. In other words, by recording the output of the system to a known input signal, for a finite amount of time, it is possible to reconstruct the full state vector, given that the dynamical equations of the system are known and if the system is *observable* (Goodwin et al., 2001).

In order derive a traditional observer, also known as a *Luenberger* observer (Luenberger, 1971), the error between our state estimates $\mathbf{x}_o$ and the true state $\mathbf{x}$ must be defined as follows

$$\tilde{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{x}_o(t)\,, \tag{3.12}$$

and the dynamics of the error are given by

$$\dot{\tilde{\mathbf{x}}}(t) = \dot{\mathbf{x}}(t) - \dot{\mathbf{x}}_o(t) = \mathbf{A}\tilde{\mathbf{x}}\,. \tag{3.13}$$

The purpose is to force the error described by (3.12) to be as small as possible, this can be achieved by feeding back a weighted difference of the measured outputs and their estimates to the system described in (3.1)

$$\dot{\mathbf{x}}_o(t) = \mathbf{A}\mathbf{x}_o(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{L}(\mathbf{y}(t) - \mathbf{C}\mathbf{x}_o(t))\,. \tag{3.14}$$

The matrix $\mathbf{L} \in \mathbb{R}^{n_y \times n}$ is known as the observer gain and the term $\mathbf{L}(\mathbf{y}(t) - \mathbf{C}\mathbf{x}_o(t))$ is called the *innovation* or correction term. It is important to see that if $\mathbf{L} = 0$, then expression (3.14) becomes the open-loop model described in (3.1).

Substituting both (3.14) and (3.1) in (3.13) yields

$$\dot{\tilde{\mathbf{x}}}(t) = \mathbf{A}_o\tilde{\mathbf{x}}(t)\,, \tag{3.15}$$

where $\mathbf{A}_o = \mathbf{A} - \mathbf{L}\mathbf{C}$. It can be seen that if $\mathbf{L}$ is designed in such a way that $\mathbf{A}_o$ is stable, then the error between the estimates and the true states would eventually converge to zero. It is normally a good idea to design $\mathbf{L}$ with the purpose of generating faster error dynamics in comparison to the open-loop evolution of the system, ensuring that the error would decay to zero in a short period of time. The observer gain $\mathbf{L}$ can be designed using a pole-placement approach or by using the LQR method (Kwakernaak and Sivan, 1972; Goodwin et al., 2001), discussed in the previous section.

There is a trade-off between the speed of convergence and the transient response of the observer since both depend on the eigenvalues of $\mathbf{A}_o$. Depending on the design considerations, the resulting observer might not be able to suppress high frequencies entirely in the noise. In addition, model uncertainties can lead to biased estimates if the operation range is beyond the linearity assumptions.

The Kalman filter

The use of state observers has proven to be an effective way to estimate variables in many areas of engineering and research, this led to the development of different versions

and extensions to nonlinear systems. One particular formulation of these extensions is the Kalman filter (Kalman, 1960), which is a linear, unbiased, and minimum error variance recursive algorithm that optimally estimate the states of a stochastic dynamical system.

The very first use of these filters was by engineers at NASA's space program in the 1960s (Mcgee et al., 1985), and soon after that it became one of the most popular and widely accepted estimation algorithms available, with applications in satellite navigation, ballistic missile control, laser systems, radars, mapping and localisation in robotics, etc.

The Kalman filter can be interpreted as a least squares approach where the error is minimised and most of its derivations are made based on this idea (Gelb, 1974; Haykin, 2001; Simon, 2006). Although there are derivations of these filters for continuous systems (Bucy and Joseph, 1968; Lewis et al., 2007), the most common way of presenting the algorithm is using a discrete framework, since it was conceived to be implemented recursively in a digital computer. The next paragraphs give an introduction to the basic formulation of the linear Kalman filter used as a state observer.

Consider the following linear system with Gaussian noise

$$\begin{aligned} \mathbf{x}_k &= \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{v}_k \,, \end{aligned} \tag{3.16}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^{n_y}$, and $\mathbf{u} \in \mathbb{R}^{n_u}$ correspond to the system state, output and input respectively, and k represents the discrete time index. Also, $\mathbf{w} \in \mathbb{R}^n$ is the process noise vector, $\mathbf{v} \in \mathbb{R}^{n_y}$ is the measurement noise vector, $\mathbf{A}$ is an $n \times n$ matrix, $\mathbf{B}$ is $n \times n_u$, and $\mathbf{C}$ is $n_y \times n$.

Both process and measurement noise covariance terms are assumed to be uncorrelated, additive, white, and Gaussian with zero mean and known covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ respectively, as in

$$\begin{aligned} \mathbf{w}_k &\sim N(0, \mathbf{Q}) \\ \mathbf{v}_k &\sim N(0, \mathbf{R}) \,, \end{aligned} \tag{3.17}$$

and the initial state of the system $\hat{x}_0$ is known with a corresponding uncertainty expressed by the initial error covariance matrix $\mathbf{P}_0$.

The Kalman filter works in a predictor-corrector sequence, this implies that the first step of the filter is to predict the states $\hat{\mathbf{x}}_k$ and the error covariance $\mathbf{P}_k$ in order to obtain

a-priori estimates (denoted by superscript $-$) for the next iteration. In this context, the term *prediction* refers to the use of the available information and the system model to evaluate the state vector one iteration ahead. The second step is to combine the measurements $\mathbf{y}_k$ coming in the form of feedback with *a-priori* estimates to improve the result, obtaining *a-posteriori* estimates (denoted by superscript $+$).

The algorithm is defined by the following set of equations

- **Prediction step**

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1}^+ + \mathbf{B}\mathbf{u}_{k-1} \tag{3.18}$$

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}^+\mathbf{A}^T + \mathbf{Q} \tag{3.19}$$

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{C}^T\left[\mathbf{C}\mathbf{P}_k^-\mathbf{C}^T + \mathbf{R}\right]^{-1} \tag{3.20}$$

- **Correction step**

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left[\mathbf{y}_k - \mathbf{C}\hat{\mathbf{x}}_k^-\right] \tag{3.21}$$

$$\mathbf{P}_k^+ = \left[\mathbf{I} - \mathbf{K}_k\mathbf{C}\right]\mathbf{P}_k^- \,, \tag{3.22}$$

where $\mathbf{I}$ is an identity matrix of appropriate dimensions and $\mathbf{K}_k$ is known as the Kalman gain. The algorithm can be initialised at $k = 0$ as follows

$$\begin{aligned} \hat{\mathbf{x}}_0 &= E\left[\mathbf{x}_0\right] \\ \mathbf{P}_0 &= E\left[\left(\mathbf{x}_0 - E\left[\mathbf{x}_0\right]\right)\left(\mathbf{x}_0 - E\left[\mathbf{x}_0\right]\right)^T\right] \,, \end{aligned} \tag{3.23}$$

and in this case, $\mathbf{P}_0$ represents the uncertainty in our initial state estimate $\hat{\mathbf{x}}_0$ and $E\left[\cdot\right]$ stands for the expectation operator. For implementation purposes, it is generally possible to measure the noise covariance $\mathbf{R}$ by taking offline measurements to determine the variance of the measurement noise. However, the situation is not the same for the process noise covariance $\mathbf{Q}$ since normally it is not possible to directly observe the process. Most of the time, $\mathbf{Q}$ and $\mathbf{R}$ are considered design parameters of the filter, tuning both of them until the desired performance is obtained.

An important feature to notice is the similarity between (3.14) and (3.21), essentially both equations share the same structure with the difference that in the Kalman filter case, the gain $\mathbf{K}$ is adjusted every iteration, whereas in the observer formulation, the gain $\mathbf{L}$ is constant throughout the operation.

State prediction

A common situation that arises in many control problems is that a system might evolve under the influence of time delays, which might introduce negative effects in terms of performance and stability, specially if a strategy to deal with them is not in place (Niculescu, 2001). In human control systems for instance, time delays are present within the feedback loop as stated by Kleinman (1969). This characteristic led him to use the OPF model to accurately account for a time delay t_d and to eliminate the effect of it by using a state predictor.

Different versions of state predictors have been used successfully in a wide range of applications (Fuller, 1968; Gawthrop, 1976; Witrant et al., 2007; Kharitonov, 2017), and many of them took the work of Smith (1959) as a starting point to develop predictors that would perform well for both open and closed-loop systems.

The main idea behind these methods is to solve the system in (3.1) from time t to time $t + t_d$ using the estimated states $\mathbf{x}_w$ as the initial condition. With this in mind, the state predictor can be described by the following expression

$$\mathbf{x}_p(t) = e^{\mathbf{A}t_d}\mathbf{x}_w(t) + \int_0^{t_d} e^{\mathbf{A}t'}\mathbf{B}\mathbf{u}(t - t')dt' \,, \tag{3.24}$$

where $\mathbf{x}_p$ are the predicted states of the system, at time $t+t_d$, computed using available information at time t. Notice that knowledge about the size of the time-delay t_d is needed to implement the predictor.

The second term in (3.24) is known as a convolution integral and it is normally implemented by approximating its solution using numerical methods, which is a procedure that can impose a computational burden in real-time. Therefore, there is always a trade-off between solution accuracy and execution speed when a continuous predictor is to be implemented.

3.2.2 Time frames of intermittent control

An important distinction in IC is the fact that it uses three different time frames during its execution. Understanding these time frames is fundamental if a reliable and stable controller is to be designed. Fig. 3.3 displays a graphic representation of the evolution of IC, that showcases the role of each time frame.

The time frames can be defined formally as follows:

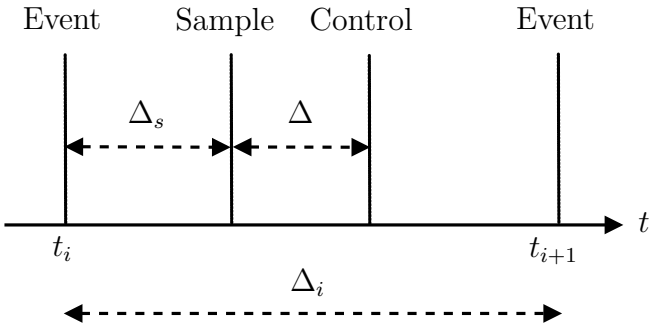


Figure 3.3: Timing diagram of IC, where Δ is the time delay of the system, Δ_s is the sampling delay, and Δ_i is the ith intermittent interval. This diagram is based on the one presented in (Gawthrop et al., 2015).

1. **Continuous-time (t)**, represents the time in which the system evolves.

2. **Discrete-time (t_i)**, time instants at which an event is detected, indexed by i. The time between event instants is known as the intermittent interval Δ_i and it is described by the following expression

$$\Delta_i = t_{i+1} - t_i \, . \tag{3.25}$$

   Once an event occurs, the observer states are sampled in order to be used as feedback. This sampling procedure can happen at a fixed time Δ_s after an event time t_i

$$t_i^s = t_i - \Delta_s \, , \tag{3.26}$$

   where Δ_s is known as the sampling delay.

3. **Intermittent-time (τ)**, a continuous variable that is restarted every intermittent interval according to

$$\tau = t - t_i \, . \tag{3.27}$$

   It is also possible to define the intermittent time τ^s between an event and the end of the sampling delay as follows

$$\tau^s = t - t_i^s \, . \tag{3.28}$$

   Alternatively, a lower limit Δ_{min} can be specified within a given intermittent interval

$$\Delta_i > \Delta_{min} > 0 \, . \tag{3.29}$$

   The lower limit Δ_{min}, also known as *minimum open-loop interval*, can be interpreted following Ronco's ideas (Ronco et al., 1999) as the time it takes to compute and select the ensuing control action and it is particularly helpful when

a state predictor is used since the equations are simple and easy to implement for the case where $\Delta \leq \Delta_{min}$.

Another interpretation of Δ_{min} is the one given in Gawthrop et al. (2011), where it is used to model the *psychological refractory period* of human motor control, that was observed originally by Telford (1931).

3.2.3 The generalised hold

In addition to these time frames, the IC framework uses a generalised hold to generate the open-loop behaviour of the system. The control signal of the intermittent controller is defined in terms of the states produced by the hold

$$\mathbf{u}(t) = \mathbf{u}(t_i + \tau) = -\mathbf{k}\mathbf{x}_h(\tau) \text{ for } t_i \leq t < t_{i+1} \,, \tag{3.30}$$

where the hold states $\mathbf{x}_h$ evolve in the intermittent time τ according to the following autonomous system

$$\frac{d}{d\tau}\mathbf{x}_h(\tau) = \mathbf{A}_h\mathbf{x}_h(\tau) \,. \tag{3.31}$$

The dynamics of (3.31) are defined by the hold matrix $\mathbf{A}_h$, this expression is known as a *generalised hold* and it is one of most important features of the IC framework.

System-matched hold

One possible approach to design $\mathbf{A}_h$ is to set it equal to the closed-loop system matrix $\mathbf{A}_c$, producing a *system-matched hold (SMH)* (Gawthrop and Wang, 2011).

$$\mathbf{A}_h = \mathbf{A}_c \,. \tag{3.32}$$

An interesting consequence from this choice is that hold states would differ from the estimated system states $\mathbf{x}_w$ only in the presence of a disturbance $\mathbf{d}$. At each feedback instant $t = t_i$, the intermittent control signal is defined by the vector $\mathbf{U}_i$ as follows

$$\mathbf{U}_i = \mathbf{K}_h\mathbf{x}_p(t_i - t_d) \,, \tag{3.33}$$

where the square matrix $\mathbf{K}_h = \mathbf{I}_{n\times n}$ is the intermittent control gain. Expression (3.33) is used to reset the hold state $\mathbf{x}_h$ at the start of each intermittent interval

$$\mathbf{x}_h(t_i) = \mathbf{U}_i \,. \tag{3.34}$$

The system-matched hold is then a natural way to implement an intermittent controller, giving rise to a system behaviour that can easily be mistaken for that of the underlying continuous-time controller (Gawthrop et al., 2011). On the other hand, this is not the only possibility for the selection of $\mathbf{A}_h$ as stated by Gawthrop et al. (2014). In the next section, the concept of using orthogonal functions to design the hold is explained.

Tapping hold

The system-matched hold is not the only possibility when it comes to select the open-loop behaviour of the system. One interesting option discussed by Wang (2009), is to implement a generalised hold based on Laguerre functions.

Definition 3.2.1. *Laguerre functions* are a set of orthonormal functions defined by

$$\begin{aligned} l_1(t) &= \sqrt{2p}e^{-pt} \\ l_2(t) &= \sqrt{2p}(-2pt+1)e^{-pt} \\ &\vdots \\ l_j(t) &= \sqrt{2p}\frac{e^{pt}}{(j-1)!}\frac{d^{j-1}}{dt^{j-1}}\left[t^{j-1}e^{-2pt}\right] . \end{aligned} \tag{3.35}$$

For any positive value of p, which is a design parameter that determines the shape of the function (i.e., how fast the exponentials of each function decay to zero).

The use of a state-space description to represent Laguerre functions simplifies their implementation. This can be done by defining the following state vector

$$L(t) = \left[\begin{array}{cccc} l_1(t) & l_2(t) & \dots & l_N(t) \end{array}\right]^T . \tag{3.36}$$

Using (3.36), the state-space equations are

$$\left[\begin{array}{c} \dot{l}_1(t) \\ \dot{l}_2(t) \\ \vdots \\ \dot{l}_N(t) \end{array}\right] = \left[\begin{array}{cccc} -p & 0 & \dots & 0 \\ -2p & -p & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ -2p & \dots & -2p & -p \end{array}\right] \left[\begin{array}{c} l_1(t) \\ l_2(t) \\ \vdots \\ l_N(t) \end{array}\right] . \tag{3.37}$$

Solving (3.37) yields Laguerre functions for $i = 1, 2, ..., N$. The compact form of the solution is

$$\begin{aligned} L(t) &= e^{\mathbf{A}_p t}L(0) \\ L(0) &= \sqrt{2p}\left[\begin{array}{cccc} 1 & 1 & \dots & 1 \end{array}\right]^T , \end{aligned} \tag{3.38}$$

where the matrix $\mathbf{A}_p$ is

$$\mathbf{A}_p = \begin{bmatrix} -p & 0 & \dots & 0 \\ -2p & -p & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ -2p & \dots & -2p & -p \end{bmatrix} . \tag{3.39}$$

Laguerre functions have been used in the past to approximate impulse responses (Lee, 1962; Wahlberg and Mäkilä, 1996). This feature is important to the notion of Intermittent Tapping Control (ITC) (Gawthrop and Gollee, 2012), where the concept of a *tap* can be described as the portions of the control signal where peaks of large amplitude are present for a brief period of time, thus resembling Dirac's delta function.

In ITC, the *taps* in the control signal can be approximated if the generalised hold takes the following form

$$\mathbf{A}_h = \mathbf{A}_p . \tag{3.40}$$

This particular choice of the hold is known as the *tapping hold*. In contrast with the SMH, the dimensions of $\mathbf{A}_h$ are $N \times n$, and by adjusting p and N, the shape of the tap can be controlled. Commonly, $\mathbf{A}_h$ is designed in such a way that the time constant of the tap is short, resulting in an impulse-like control input. The $N \times n$ gain $\mathbf{K}_h$, in combination with the state-feedback gain $\mathbf{k}$, define the size of each of the taps.

An important feature of ITC is that $\mathbf{A}_h$ can be designed to generate a response that is similar to that of the SMH but with a control signal that is different in shape. If the state-feedback gain $\mathbf{k}$ is held constant, then the design task is simplified to calculate the IC gain $\mathbf{K}_h$ that would generate the desired response.

$\mathbf{K}_h$ gain design

In order to obtain the corresponding $\mathbf{K}_h$, a modified version of the optimisation approach in (3.9) should be used. Consider the following cost function

$$J_{IC} = \int_0^{\tau_1} \left[\mathbf{x}(\tau)\mathbf{Q}_c\mathbf{x}(\tau) + \mathbf{u}(\tau)^T\mathbf{R}_c\mathbf{u}(\tau) \right] d\tau + \mathbf{x}(\tau_1)^T\mathbf{P}\mathbf{x}(\tau_1) . \tag{3.41}$$

Equation (3.41) is a receding-horizon optimisation in the time frame of τ with an added terminal cost, where $\mathbf{Q}_c$ and $\mathbf{R}_c$ are the same design matrices of the LQR in (3.9), and $\mathbf{P}$ is the solution to (3.10). Since the purpose of ITC is to approximate the effect of the underlying continuous controller in the system using a different control signal,

the integral term in (3.41) can be eliminated resulting in an updated cost function as follows

$$J_{ITC} = \mathbf{x}(\tau_1)^T \mathbf{P} \mathbf{x}(\tau_1) \,. \tag{3.42}$$

Notice that this expression is still influenced by (3.9) since it uses the same solution matrix $\mathbf{P}$. To find the value of (3.42), a state-space system combining the dynamics of the state $\mathbf{x}$ and the hold state $\mathbf{x}_h$ is created

$$\begin{aligned} \dot{\boldsymbol{X}}(\tau) &= \mathbf{A}_{xu} \boldsymbol{X}(\tau) \\ \boldsymbol{X}(0) &= \boldsymbol{X}_i \,, \end{aligned} \tag{3.43}$$

where the augmented state vector is $\boldsymbol{X} = \begin{bmatrix} \mathbf{x} & \mathbf{x}_h \end{bmatrix}^T$, the initial conditions are $\boldsymbol{X}_i = \begin{bmatrix} \mathbf{x}(t_i) & \mathbf{U}_i \end{bmatrix}^T$, and

$$\mathbf{A}_{xu} = \begin{bmatrix} \mathbf{A} & \mathbf{Bk} \\ 0_{n\times n} & \mathbf{A}_h \end{bmatrix} . \tag{3.44}$$

The solution of (3.43) has an explicit form and can be written as

$$\boldsymbol{X}(\tau) = e^{\mathbf{A}_{xu}\tau} \boldsymbol{X}_i \,. \tag{3.45}$$

Using the augmented state $\boldsymbol{X}$, the ITC cost function in (3.42) becomes

$$J_{ITC} = \boldsymbol{X}(\tau_1)^T \mathbf{P}_{xu} \boldsymbol{X}(\tau_1) \tag{3.46}$$

where

$$\mathbf{P}_{xu} = \begin{bmatrix} \mathbf{P} & 0_{n\times n} \\ 0_{n\times n} & 0_{n\times n} \end{bmatrix} . \tag{3.47}$$

This allows the use of $\boldsymbol{X}$ in (3.43) to reformulate the cost function (3.46) as follows

$$J_{ITC} = \boldsymbol{X}_i^T J_{XX} \boldsymbol{X}_i \,, \tag{3.48}$$

where J_{XX} is a $2n \times 2n$ matrix defined as

$$J_{XX} = e^{\mathbf{A}_{xu}^T \tau_1} \mathbf{P}_{xu} e^{\mathbf{A}_{xu}\tau_1} \,, \tag{3.49}$$

which can be partitioned to have a compact form

$$J_{XX} = \begin{bmatrix} J_{xx} & J_{xU} \\ J_{Ux} & J_{UU} \end{bmatrix} . \tag{3.50}$$

The IC gain $\mathbf{K}_h$ is computed from (3.50) as follows

$$\mathbf{K}_h = J_{UU}^{-1} J_{Ux} \, . \tag{3.51}$$

This choice of $\mathbf{K}_h$ ensures the intermittent control vector $\mathbf{U}_i$ minimises the value of the cost function J_{ITC}. An advantage of this approach is that $\mathbf{K}_h$ can be computed prior to the execution. As mentioned earlier, an important distinction is that by design $\mathbf{K}_h$ is not a square matrix, whereas in the SMH case it is defined as an identity matrix.

The selection of a tapping hold might be beneficial in systems where the effect of nonlinear friction is considerable (Gawthrop and Gollee, 2012), due to the nature of the control signals that it generates. It is hypothesised that this same feature could be used in adaptive control to excite the system regularly, helping the parameter estimation procedure to converge to the true values, specially when the outputs have reached the steady-state.

3.2.4 Intermittent state prediction

One of the main benefits of using IC in terms of state prediction, is the fact that (3.24) can be replaced by an expression that avoids the approximation of the convolution integral mentioned in section 3.2.1 (Gawthrop and Wang, 2007). Equation (3.24) is the solution to the following dynamical system during the intermittent time frame τ

$$\frac{d}{d\tau}\mathbf{x}_p(\tau) = \mathbf{A}\mathbf{x}_p(\tau) + \mathbf{B}\mathbf{u}(\tau) \, , \tag{3.52}$$

with $\mathbf{x}_p(0) = \mathbf{x}_w(t_i)$ as initial condition and evaluated at $\tau = t_d$. Combining (3.52) and (3.31) yields the following extended system

$$\frac{d}{d\tau}\bar{X}(\tau) = \mathbf{A}_{ph}\bar{X}(\tau) \tag{3.53}$$

$$\bar{X}(0) = \bar{X}_i \, . \tag{3.54}$$

The combined state vector $\bar{X}$ is defined as follows during the open-loop interval

$$\bar{X}(\tau) = \begin{bmatrix} \mathbf{x}_p(\tau) \\ \mathbf{x}_h(\tau) \end{bmatrix} \, . \tag{3.55}$$

For the instances where feedback is used (t_i), $\bar{X}$ takes the following form

$$\bar{X}_i = \begin{bmatrix} \mathbf{x}_w(t_i) \\ \mathbf{x}_p(t_i - t_d) \end{bmatrix} \, . \tag{3.56}$$

The matrix $\mathbf{A}_{ph}$ is composed by the hold $\mathbf{A}_h$, the state-feedback gains $\mathbf{k}$, and the system matrices $\mathbf{A}$ and $\mathbf{B}$ giving

$$\mathbf{A}_{ph} = \begin{bmatrix} \mathbf{A} & -\mathbf{Bk} \\ 0_{n\times n} & \mathbf{A}_h \end{bmatrix} . \tag{3.57}$$

Solving (3.52) at $\tau = t_d$ yields

$$\bar{X}(t_d) = e^{\mathbf{A}_{ph}t_d}\bar{X}_i . \tag{3.58}$$

From (3.58), the predicted states $\mathbf{x}_p$ can be obtained every intermittent interval and are given by

$$\mathbf{x}_p(t_i) = \mathbf{E}_{pp}\mathbf{x}_w(t_i) + \mathbf{E}_{ph}\mathbf{x}_h(t_i) , \tag{3.59}$$

where the matrices $\mathbf{E}_{pp}$ and $\mathbf{E}_{ph}$ of dimension $n \times n$, are partitions of the $2n \times 2n$ matrix $\mathbf{E}$ defined as

$$\mathbf{E} = \begin{bmatrix} \mathbf{E}_{pp} & \mathbf{E}_{ph} \\ \mathbf{E}_{hp} & \mathbf{E}_{hh} \end{bmatrix} , \tag{3.60}$$

and $\mathbf{E}$ comes from

$$\mathbf{E} = e^{\mathbf{A}_{ph}t_d} . \tag{3.61}$$

The matrices $\mathbf{E}_{pp}$ and $\mathbf{E}_{ph}$ can be obtained offline, when the controller is being designed. This is a convenient feature from the computational point of view, since only matrix products are involved to obtain the predicted states $\mathbf{x}_p$ once the controller is operating.

3.2.5 Event detection and thresholds

The intermittent controller uses a triggering mechanism to generate sample times t_i and force the use of feedback. There are two operation modes for this purpose (Gawthrop and Wang, 2009a):

- **Clock-based**: in this mode, Δ_i on (3.25) is constrained to have a minimum value of at least Δ_{min}. This generates events at fixed intervals regardless of the state of the system. A disadvantage of this mode is that the controller would still use feedback periodically even if there is no need for it; for instance, once the system reaches a desired steady-state value and there are no disturbances acting on it.

- **Event-based**: this mode generates an aperiodic sequence of events, determined by the error between either predicted or hold states and the estimated states. If the errors are greater than a threshold q, then an event is generated. The definition of both of these errors is as follows:

1. *Prediction error*: the estimated states $\mathbf{x}_w$ are compared to the predicted states $\mathbf{x}_p$ according to the following expression

$$e_x = \mathbf{x}_p(t) - \mathbf{x}_w(t) \,. \tag{3.62}$$

   This error is used when a system-matched hold is driving the open-loop behaviour.

2. *Closed-loop error*: introducing the autonomous dynamical system driven by the closed-loop dynamics as

$$\frac{d}{d\tau}\mathbf{x}_c(\tau) = \mathbf{A}_c\mathbf{x}_c(\tau) \,, \tag{3.63}$$

   where $\mathbf{x}_c(0) = \mathbf{x}_p(t_i - t_d)$, this formulation allows the states $\mathbf{x}_c$ to be used in the closed-loop error equation as follows

$$e_x = \mathbf{x}_c(t) - \mathbf{x}_w(t) \,. \tag{3.64}$$

   The closed-loop error is normally used in conjunction with the tapping hold. The reason for this is based on the fact that the tapping hold does not include information in its states to detect events, therefore equation (3.63) is used instead. Notice that this formulation is equivalent to that in (3.31), which is the definition of a system-matched hold.

   Regardless of which error equation is used, an event in generated when e_x is greater than q, this can be formally defined by using the following quadratic criterion

$$e_x^T(t)\mathbf{Q}_t e_x(t) - q^2 \geq 0 \,, \tag{3.65}$$

   where $\mathbf{Q}_t$ is a positive semi-definite matrix that can be used to define which states are used to generate events. For example, choosing an identity matrix would force the controller to trigger on all system states.

For the particular case where $q = 0$, the event-based mode becomes clock-based since the condition in (3.65) is always true, thus triggering at every minimum open-loop interval Δ_{min}. Therefore, clock-based IC can be seen as a special case of the event-based mode.

3.3 The inverted pendulum model of human standing

The single inverted pendulum model is a traditional example in control systems and it is widely used to illustrate the results and the implementation details of many control strategies. In addition, some authors have used this model to describe the human balance control problem, contrasting simulation and experimental data (Loram et al., 2009; Nomura et al., 2013). For the purpose of this simulation, the dynamic bias model of human standing described in (Lakie et al., 2003; Loram et al., 2005, 2009; Gawthrop et al., 2011) is used. This model considers that the control signal that is applied to maintain a human inverted pendulum balanced is generated by a tendon that is connected in series with a contractile element (in this case the calf muscle) which is in charge of generating a torque. The equation of motion of the pendulum is as follows

$$J\ddot{\theta} = mgh\sin(\theta) + T\,, \tag{3.66}$$

where J is the moment of inertia, θ is the angular position with respect to the vertical, m is the mass of the pendulum, g is the gravitational acceleration, and h is the distance from the joint to the centre of mass. The small angle approximation $\theta = \sin(\theta)$ is used to simplify the equation to a linear model. The ankle torque T is defined as

$$T = -cmgh(\theta - \theta_0) - V\dot{\theta}\,, \tag{3.67}$$

with c being the ratio between the tendon stiffness k_p, the load stiffness defined by the product $k_e = mgh$, and V is the ankle viscosity. The input is provided by θ_0 (known as the bias) which represents the active muscle shortening in angular terms. Considering the pendulum angle θ (in radians) and the bias θ_0 as output and input respectively, the model can be written as a transfer function, resulting in

$$\theta = \frac{\frac{cmgh}{J}}{s^2 + \frac{V}{J}s + (c-1)\frac{mgh}{J}}\theta_0\,. \tag{3.68}$$

The constants in (3.68) are: $c = 0.85$, $J = 77$ Kg m^2, $V = 2.9$ Nm rad^{-1}, $m = 70$ Kg, $h = 0.92$ m, $g = 9.81$ m s^{-2} as reported in (Loram et al., 2009). The fact that c is smaller than 1 implies that the tendon stiffness is not enough to stabilise the pendulum on its own, requiring additional control effort provided by the muscle. In (Loram et al., 2009), this model was implemented in simulation, where the input θ_0 was provided by a subject holding a joystick, thus θ_0 was proportional to the motion of the joystick.

Defining the state vector $x(t) = \begin{bmatrix} \dot{\theta} & \theta \end{bmatrix}^T$, and using the constants described above,

the transfer function in (3.68) can be written as state-space model as follows

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} -0.037 & 1.231 \\ 1 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 6.98 \\ 0 \end{bmatrix} \mathbf{u}(t) \tag{3.69}$$

$$\mathbf{y} = \begin{bmatrix} 0 & 1 \end{bmatrix} \mathbf{x}(t) \, . \tag{3.70}$$

In terms of the block diagram in Fig. 3.2, the block labelled as *System* represents the inverted pendulum model in 3.68. In Fig. 3.4 a diagram of the inverted pendulum is presented. The following second order, neuro-muscular system (*NMS* block in Fig. 3.2), that describes the dynamics of both the hand and joystick is considered (Navas and Stark, 1968; Gawthrop et al., 2011)

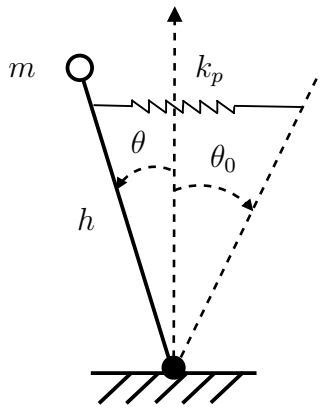


Figure 3.4: Inverted pendulum model of human standing. The input θ_0 represents the contraction of the muscle as an angle that influences the pendulum via a spring of stiffness k_p, which represents the ankle joint tendon. The output θ is the angle of the pendulum with respect of the vertical line, m is the mass, and h is the distance from the joint to the centre of mass.

$$G_s(s) = \frac{1}{s(0.1s+1)} \, , \tag{3.71}$$

which corresponds to the following state-space realisation

$$A_s = \begin{bmatrix} -10 & 0 \\ 1 & 0 \end{bmatrix}, \; B_s = \begin{bmatrix} 10 \\ 0 \end{bmatrix}, \; C_s = \begin{bmatrix} 0 \\ 1 \end{bmatrix}^T \, . \tag{3.72}$$

The combination of these two systems yields a fourth order model. Therefore the input that is applied to the system defined by (3.69) is the output of the neuro-muscular model in (3.72), and corresponds to the angle θ_0. However, the input $\mathbf{u}(t)$ that the intermittent controller generates serves as an input to (3.72) as shown in the IC diagram (Fig. 3.2), and is the one presented in the simulation as $\mathbf{u}$. The purpose of this example

is to illustrate the design of intermittent controllers using system-matched and tapping holds, while combining them with Luenberger and Kalman observers.

The inverted pendulum model was used to run a simple simulation where the main goal is to apply a control signal to keep the pendulum as close as possible to the vertical position, using the two versions of intermittent control presented in this chapter (SMH and ITC). Also, two types of state estimators were used: a Luenberger observer and a Kalman filter. Throughout the simulation, a random disturbance signal $\mathbf{d}(t)$ with an amplitude of 0.01 rad was applied to the control input as shown in 3.2.

3.3.1 Controller and estimator design

The design parameters used to implement the controllers and the state estimators is presented in this section.

Controller gains and system-matched hold: with the LQR approach from section 3.2.1 and the system matrices in (3.69), it is possible to define the square design matrices $\mathbf{R}_c = \mathbf{I}_{4\times4}$ and $\mathbf{Q}_c$, where the diagonal of $\mathbf{Q}_c$ is defined by the vector $\begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}$ and the rest of the entries being zero, to obtain the following controller gains

$$\mathbf{k} = \begin{bmatrix} 1.8845 & 2.0625 & 1.2118 & 15.0545 \end{bmatrix} . \tag{3.73}$$

In this case, matrix $\mathbf{Q}_c$ only contains elements on the first two elements of its diagonal. This two values have a direct effect on both velocity and position states of the pendulum. With the values of the state-feedback gain $\mathbf{k}$, it is possible to build the closed-loop matrix $\mathbf{A}_c$ which is used to define the system-matched hold as in (3.4)

One possible way to compute these gains is to use packages such as Matlab (Mathworks, Inc.) and GNU Octave. Both of them have specific routines to solve the ARE in (3.10) and to obtain expression (3.11).

Luenberger observer and Kalman filter: similarly, to design a Luenberger state observer using the LQR approach, the matrix $\mathbf{Q}_o$ is defined as $\mathbf{Q}_o = q_o\mathbf{B}\mathbf{B}^T$ where $q_o = 10$. This yields the following observer gain

$$\mathbf{L} = \begin{bmatrix} 17.0218 & 5.8347 & 0.0789 & 1.2570 \end{bmatrix}^T . \tag{3.74}$$

The process and measurement noise covariance matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$ of the Kalman filter were defined as identity matrices of appropriate dimensions.

Tapping hold: values of $p = 10$ and $N = 1$ were selected to define the Laguerre functions defined by (3.39), to then build the hold matrix $\mathbf{A}_h$. This leads to the following IC gain matrix

$$\mathbf{K}_h = \begin{bmatrix} 2.8099 & 3.0969 & 2.0912 & 25.3240 \end{bmatrix} . \tag{3.75}$$

Timing and triggering: Table 3.1 shows the timing parameters considered for all controllers

Table 3.1: Intermittent control timing parameters

Δ_{min}	Δ_s	t_d	Threshold q
0.25 sec	0 sec	0.01 sec	0.01 rad

Finally, the intermittent controller was designed to trigger only on the position state θ by setting matrix $\mathbf{Q}_t$ in (3.65) to have the vector $\begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}$ as its diagonal. The following initial condition vector $\mathbf{x}(0) = \begin{bmatrix} 0 & 0.05 & 0 & 0 \end{bmatrix}^T$ was used to start the simulation.

3.3.2 Simulation results

The simulation shows the output and input for the SMH and ITC controllers, when using both Luenberger and Kalman filters (Fig. 3.5), a comparison between the estimated angular velocity $\hat{\dot{\theta}}$ (generated by the state estimators) and the simulated value of θ (Fig. 3.6), and the corresponding open-loop distributions (Fig. 3.7).

In Fig. 3.5, the output for both controllers is shown in sub-figures (a) and (b), and the inputs in (c) and (d). The left column corresponds to the data generated with the Luenberger observer and the one in the right for the Kalman filter.

The outputs from both controllers oscillate around zero. It is clear that for the considered design parameters, the SMH output error is smaller compared to ITC (for both state estimators); however, both responses are comparable in terms of amplitude. The situation is different for the control inputs, ITC generates an impulsive-like control signal that is considerably higher in amplitude compared to SMH.

Fig. 3.6 shows the behaviour of the estimated angular velocity in comparison with the *real* value that is extracted from the simulated system. The role of the state observers in this case is to generate the estimates only from measurements of the system output θ.

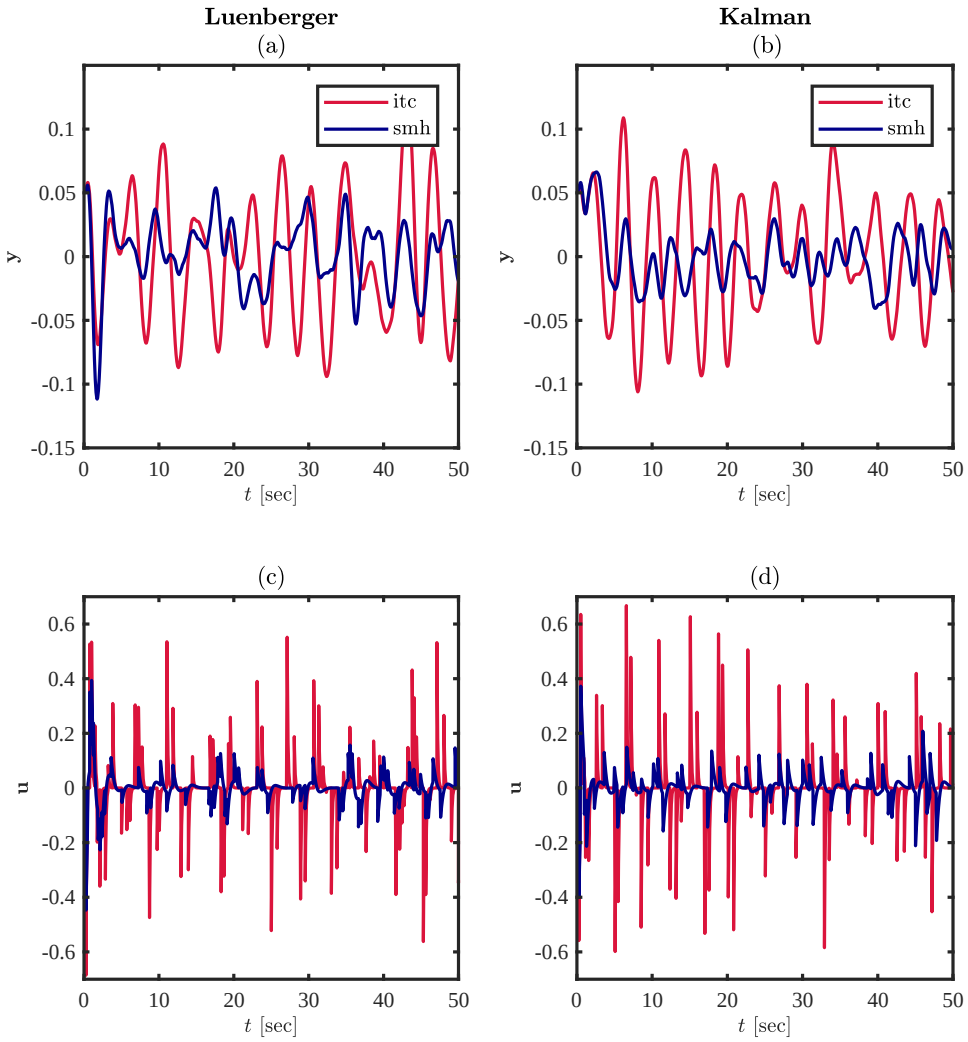


Figure 3.5: Input and output. (a) and (b) show the output of the inverted pendulum for both controllers (SMH in blue, ITC in red). (c) and (d) correspond to the control signals. The data shown in the left column ((a) and (c)) was generated using a Luenberger observer, whereas the right column ((b) and (d)) corresponds to the Kalman filter.

In (a) and (b), the estimates obtained when the SMH controller is used are presented, (c) and (d) correspond to ITC. For both controllers, the state estimators generate angular velocities that match closely the real values of $\dot{\theta}$, being different between them only in terms of amplitude since the ITC shows higher velocity values on average compared to SMH.

In Fig. 3.7, the open-loop distributions for SMH and ITC (in combination with both estimators) are shown. In general, the open-loop intervals Δ_{ol} corresponding to ITC are smaller on average compared to SMH. The minimum open-loop interval was set

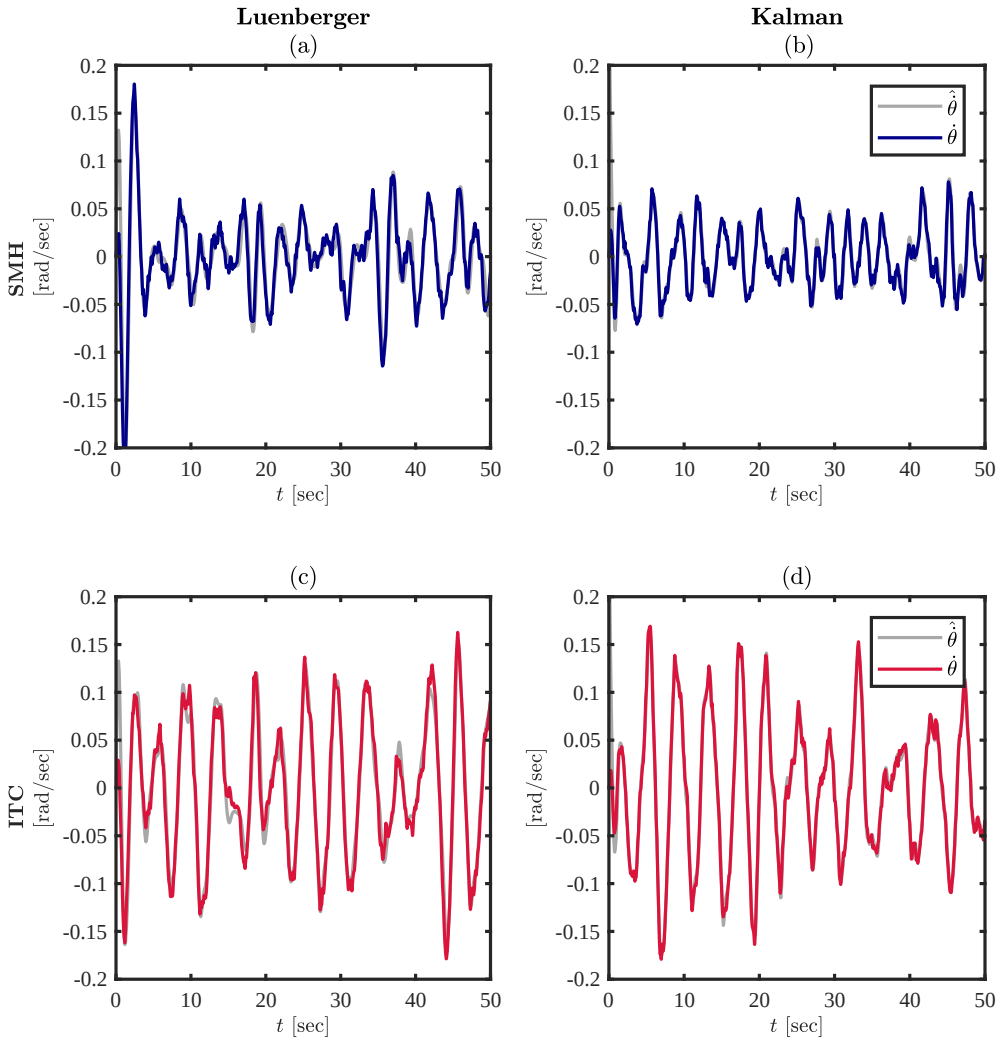


Figure 3.6: Angular velocity. Comparison between the estimated angular velocity $\hat{\dot{\theta}}$ and its emphreal value $\dot{\theta}$. The data shown in the left column ((a) and (c)) was generated using a Luenberger observer, whereas the right column ((b) and (d)) corresponds to the Kalman filter. (a) and (b) correspond to the SMH, (c) and (d) to ITC.

to 0.25 sec for this simulation, it is possible to see that the controllers generate events at this minimum rate only during the first seconds of the simulation, to then grow to values between 0.5 sec and 2.5 sec for ITC, and between 0.5 sec and almost 3.5 sec for SMH. This distributions indicate for how long the intermittent controllers were evolving in an open-loop configuration.

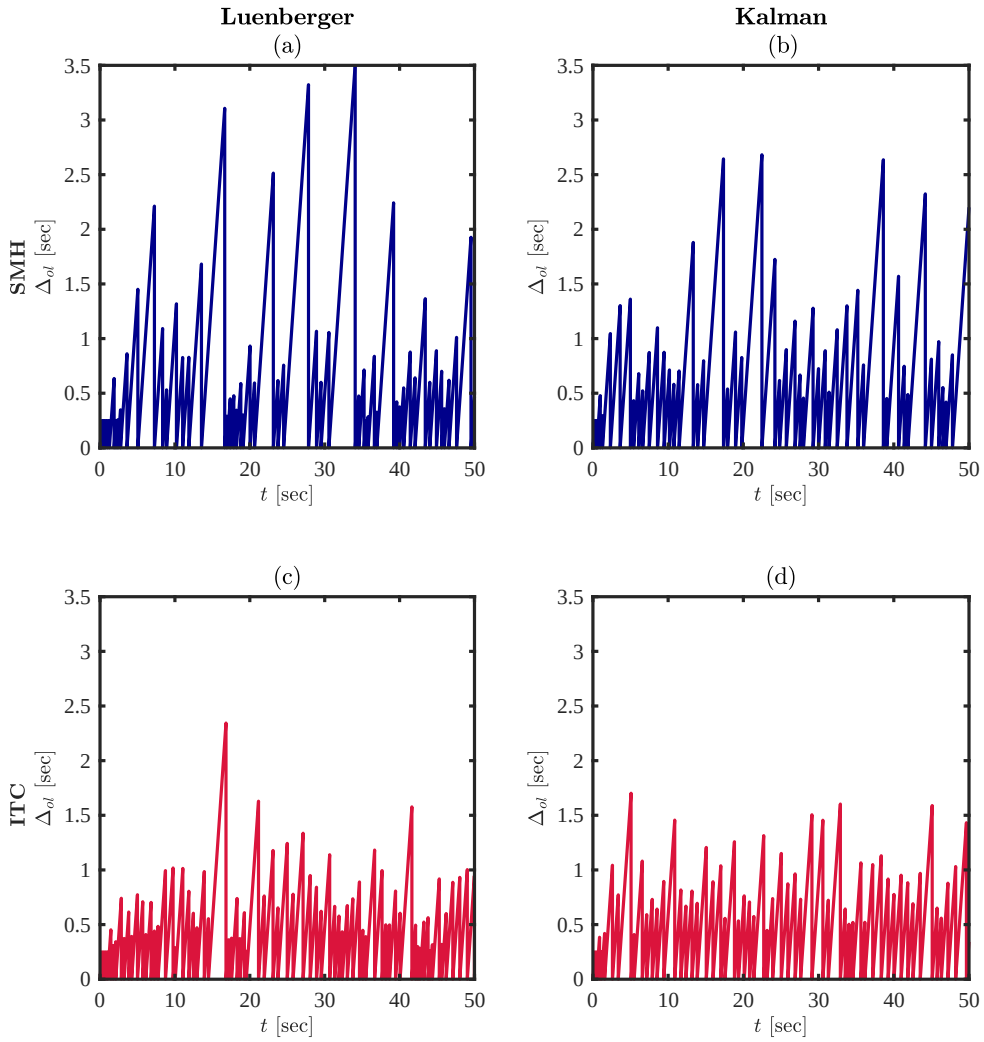


Figure 3.7: Open-loop intervals. The time the controllers evolved in an open-loop configuration is shown in this figure. The data shown in the left column ((a) and (c)) was generated using a Luenberger observer, whereas the right column ((b) and (d)) corresponds to the Kalman filter. (a) and (b) correspond to the SMH, (c) and (d) to ITC.

3.4 Adaptation in the context of intermittent control

Apparent simple tasks like walking, grasping or even quiet balancing are dominated by two important processes: i) planning and generating commands to mechanically achieve a certain goal, ii) and predicting sensory consequences of these commands (Shadmehr and Mussa-Ivaldi, 2012). Predicting allows us to overcome the delays that the central nervous system imposes in our sensory information as well as the delays associated with the generation of a motor command via muscle activation. In fact, this is critical since it is known that delays can affect the overall stability of a control

system considerably (Niculescu, 2001). Since the moment we are born, humans start perfecting their ability to predict by tuning their internal models and by becoming more accurate at formulating beliefs based on the information that the models provide; this tuning is greatly related to our motion performance and it is probably one of reason why high-level athletes are capable of controlling their movements more accurately than average people. However, these predictions are greatly improved by the delayed information that our sensory system gathers continuously, even if it is delayed and noisy. Combining these two streams of information in an optimal way is a task that our brain performs automatically, giving us more complete description of the state of our body in the environment.

We constantly make wrong predictions in daily life, and when this happens, we must rely on available information to update our prediction and obtain an updated belief of our state. For instance, predicting the mass of an empty box often leads to over estimate the amount of force that is needed in order to move it, resulting in erroneous initial motor commands (Gordon et al., 1991). Soon after, these commands get updated based on an improved estimate of the state of the arm (i.e., positions and velocities), allowing us to apply the correct amount of force to move the box. Interestingly, the unknown weight of the box is also estimated and combined with our previous beliefs, allowing us to apply the correct force the next time we need to move the box. In this sense, it is possible to say that this mechanism is flexible enough to update itself, adapting to prediction errors and wrong assumptions about the environment, an it is capable of identifying the *real* system that is trying manipulate or control. This two step process of (1) combining state and parameter information with our predictions to then (2) correct our control actions can be formulated in terms of adaptive control theory, where having a reliable real-time procedure to optimally combine information while estimating parameters determines the performance of the controller.

Computational level theories of human control incorporate adaptation and learning to fully explain the features of our internal control mechanisms. In particular, state estimation in the adaptation context has been formulated within the framework of Bayesian estimation and Kalman filtering, which produces minimum variance estimates of stochastic processes under the influence of Gaussian noise. Moreover, it has been shown experimentally that in some scenarios, humans use forward models in a Bayesian framework to estimate their state and its relationship with the environment (Wolpert et al., 1995; Körding and Wolpert, 2004). The flexible nature of the Kalman algorithm can estimate not only the state vector that describes the system dynamics but also some of its parameters. This can be used to track time-varying changes in these parameters in order to update a previously designed controller. This concept is

known in adaptive control as a self-tuning algorithm, which provides the advantage of clearly separating the estimation and control update steps.

Based on recent theoretical and experimental results (Gawthrop and Gollee, 2012; Loram et al., 2011; van de Kamp et al., 2013b; Gawthrop et al., 2015), it has been argued that the natural switching between open and closed loop configurations provided by intermittent control contributes to the clear distinction of the effects of our motor commands. This hypothesis comes from the results of a group of subjects that tried to control a simulated inverted pendulum using a joystick. The signal provided by the joystick was multiplied by a gain that changed throughout the trial. In this experiments, the subjects that applied continuous hand contact with the joystick performed worse that those who used a tapping strategy. The time between taps can be seen as an open-loop period, where the natural dynamics of the system are clearly separated from the inputs provided by the joystick, which helped the participants to realise that some conditions had changed, leading to quicker adaptation. These ideas motivated the results presented in this chapter, which include the formulation of an adaptive intermittent controller that is based on the Kalman estimation framework and a self-tuning architecture, with the purpose of testing if indeed the hybrid switching strategy of intermittent control provides an advantage in terms of parameter estimation and adaptation compared to a continuous adaptation scheme, while evaluating the effects of using nonlinear Kalman filters for the first time in this context.

3.5 Adaptive control

Adaptive control consists of two feedback loops. The first loop is an inner loop which is formed by a parameter dependent controller in series with the system/plant that is being controlled, commonly known as the *feedback loop*. The outer loop has a supervisory role and is normally referred as the *adaptation loop*. This adaptation loop updates its own states, which correspond to the parameters (i.e., coefficients of the dynamical system describing the plant) to then pass them to the controller that is part of the inner loop. Fig. 3.8 shows the general scheme of an adaptive controller, including the adaptation and feedback loops.

In the scheme shown in Fig. 3.8, the *System* is normally considered as a linear time-variant system, where the parameters are changing through time. It can be said then that the adaptation loop must gather the maximum amount of information about the plant constrained to a defined parameter set and then update a linear controller. This idea assumes that there is a design procedure that would generate a stabilising controller that complies with performance requirements if there is enough information

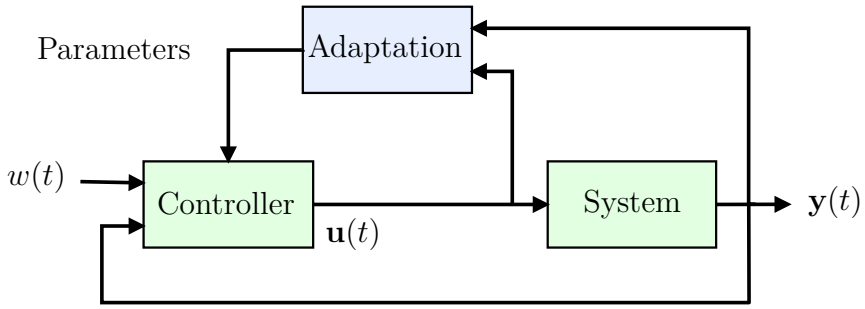


Figure 3.8: General adaptive control. An *adaptation loop* is in charge of estimating parameters for an adjustable controller. The control signal $\mathbf{u}(t)$ is computed based on the new controller parameters, the setpoint $w(t)$, and the output $\mathbf{y}(t)$, which comes in the form of feedback.

about the plant and the environment in which operates. The *adaptive control problem* then can be understood as a method that adjusts a controller when the parameters of the system are unknown or changing. These previous ideas can be summarised in three broad steps that are necessary in adaptive control:

- Definition of the desired performance of the closed-loop system using a fixed controller and find a suitable control law.
- Implementation of an adjustable version of the control law, with coefficients that can be modified.
- Implementation of a method that estimates the changing or unknown system parameters.

Different adaptive control schemes have been proposed to deal with uncertainties, unknown constant parameters or time-varying parameters, such as *gain scheduling*, *model reference adaptive control*, and *multiple model networks*, with successful applications in real-time situations (Åström and Wittenmark, 1995; Feng and Lozano, 1999; Landau et al., 2011). The controllers implemented in this chapter are based on an idea originally introduced by Kalman (1958), known as *self-tuning adaptive control*, which is described in the following section.

3.5.1 Self-tuning adaptive control

A self-tuning controller can be described as having two very distinctive elements included in the *adaptation loop*, as indicated in Fig. 3.9:

- A *recursive estimator* that computes a set of parameters based only on information of the applied input $\mathbf{u}$, the measured output $\mathbf{y}$ of the system, and an

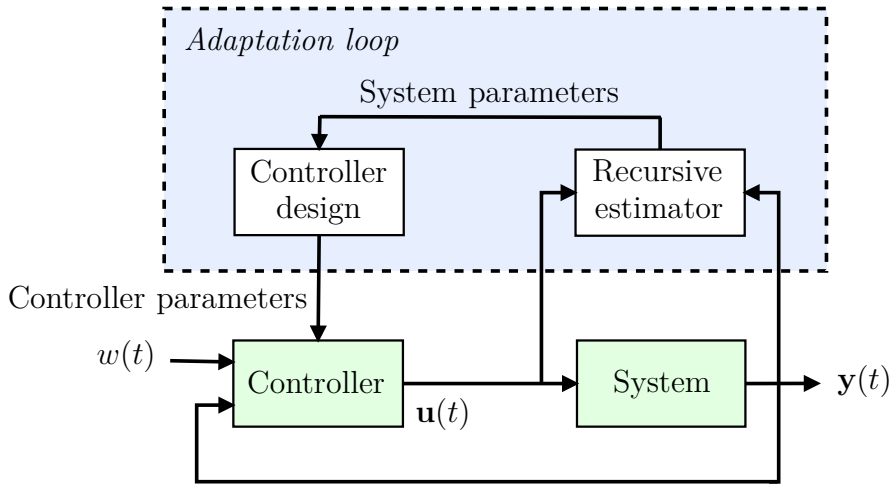


Figure 3.9: Adaptive self-tuning control diagram. The *adaptation loop* is comprised of a recursive estimator in charge of providing system parameters to the controller design stage, which computes the parameters of an adjustable control law $\mathbf{u}(t)$, based on the setpoint $w(t)$ and the measured output $\mathbf{y}(t)$.

assumed model structure. These parameters can be the coefficients of the differential equations describing the process, denoted by φ_m, from which the controller parameters φ_c could be obtained, or they can be generated in a form that allows their direct application to the controller, as in $\varphi_m = \varphi_c$.

- For the case where $\varphi_m \neq \varphi_c$, there is the need of a *controller design stage*, that would yield the parameters that the adjustable control law is supposed to use.

When the adaptation loop in self-tuning control is comprised of both a recursive estimator and a controller design stage, the overall scheme is referred as indirect (or explicit) self-tuning adaptive control. If there is no controller design stage it means that the recursive estimator is directly producing the controller parameters, this is known as direct (or implicit) self-tuning.

There are advantages for using both schemes, for instance, direct self-tuning control often leads to simpler controller structures that have been related to reinforcement learning algorithms previously (Sutton et al., 1992). Indirect self-tuners are flexible in the sense that the recursive estimation algorithm is independent from the control design stage, allowing more options when it comes to testing different estimators and controllers, while giving insight on how they influence each other. These ideas are based on linear control theory, specifically on the *separation principle* (Luenberger, 1979).

The stability of a self-tuning controller has been studied extensively in the control community (Åström et al., 1977; Gawthrop and Lim, 1982; Narendra and Annaswamy, 1989;

Feng and Lozano, 1999; Landau et al., 2011) and proofs for simple schemes in ideal conditions have been derived (Egardt, 1980; Goodwin et al., 1980; Morse, 1980). The stability of the system that is being controlled must be guaranteed by the online parameter estimation routine in combination with the plant and the controller itself. A simple notion is that the feedback loop will be stable as long as the estimated parameters are sufficiently close to the *true* values. However, the convergence of the estimation routine depends on control signals that provide persistent excitation to the system (Sastry and Bodson, 2011). In this sense, the concepts of stability and parameter convergence rely on the idea that the state of a parameter error system converges to zero exponentially if and only if the control signal, which is related to the plant parameterisation, is persistently exciting (Tao, 2014).

The nature of the model that should be estimated has an important effect on stability. In order to obtain an accurate model, the controller should generate input signals that contain sufficient energy around the cross-over frequency of the system and it has been suggested that the estimation routine should be active only if the absolute value of the energy associated with the control input is above a certain limit (Egardt, 1979; Peterson and Narendra, 1982). If the control signals do not excite the system enough, two common strategies can be used to avoid divergence and improve adaptation: the first one involves breaking the adaptation-loop, stopping the controller redesign, and the second one is about the inclusion of artificial perturbation signals (Wittenmark and Åström, 1984).

3.6 Adaptive intermittent controller

The adaptive intermittent controller (AIC) that is presented in this section is one of the main contributions of the thesis; in particular, it is an extension to the existing theory of intermittent control and its links with adaptation. This controller is based on nonlinear Kalman filters to perform joint state and parameter estimation, and it can be designed to use both a *system-matched hold* or a tapping-hold. Its validation in simulation and experimentally is discussed in the following chapters.

AIC can be implemented by exploiting the benefits from having separate design and estimation stages in the self-tuning architecture. The diagram on Fig. 3.9 assumes that part of the adaptation loop uses a recursive estimator to identify the unknown system parameters φ_m, based on measurements of the input and the output. On the other hand, state observers also use the same information to estimate the portion of the state vector $\mathbf{x}_o(t)$ that is not available for direct measurement using sensors, which

is a common situation in most applications. Therefore, algorithms that can solve both problems concurrently are ideal in this context. Before introducing the AIC in the form of a diagram, it is convenient to look at the elements of a continuous adaptive controller in order to visualise the differences between them. In Fig. 3.10, the block diagram of an adaptive continuous controller is shown.

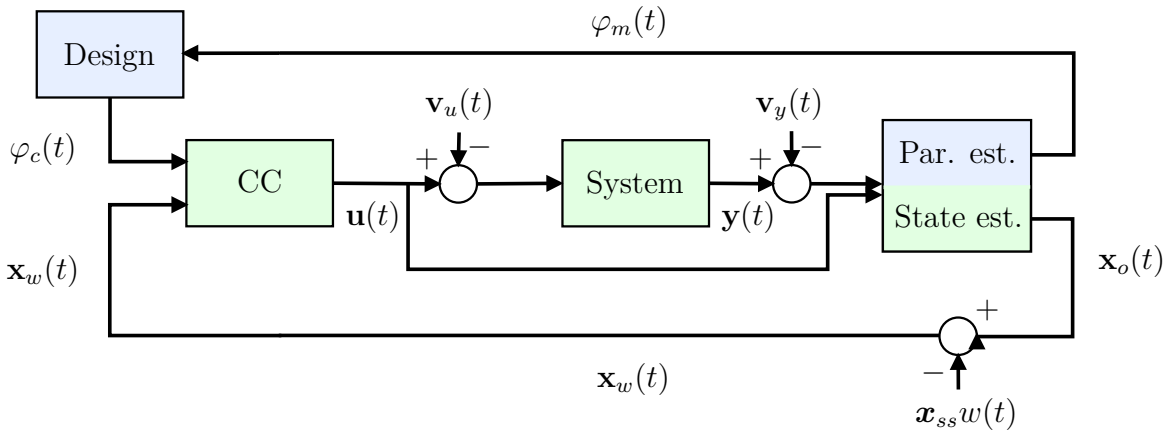


Figure 3.10: Adaptive continuous control diagram. The block labelled as *Par est./State est.* represents a recursive algorithm that continuously estimates the system states $\mathbf{x}_o(t)$ and the model parameters $\varphi_m(t)$. Both the input $\mathbf{u}(t)$ and the output $\mathbf{y}(t)$ can be affected by input noise $\mathbf{v}_u(t)$ and measurement noise $\mathbf{v}_y(t)$ respectively. The *Design* block represents the controller redesign stage and it provides all the parameters needed by the continuous controller *CC*. The blue colour represents the components of the *adaptation loop*, while green is used to identify the components in the *feedback loop*.

The scheme presented in Fig. 3.10 uses a state and parameter estimator (Par est./State est. block) that is continuously receiving the input $\mathbf{u}(t)$ provided by a standard CC as well as the output of the process $\mathbf{y}(t)$. Both of them might be corrupted by input and observation noise vectors $\mathbf{v}_u$ and $\mathbf{v}_y$ respectively. With this information, the estimator generates observed states $\mathbf{x}_o(t)$ and the model parameters $\varphi_m(t)$ throughout the entire time the controller is operational. The design stage (Design block) receives *continuously* the model parameters $\varphi_m(t)$ in order to generate the new controller parameters $\varphi_c(t)$. The CC controller also uses the estimated states $\mathbf{x}_w(t)$ in the form of continuous feedback in order to modulate the control input $\mathbf{u}(t)$.

Based on the scheme shown in Fig. 3.10, one possible structure for AIC is presented in Fig. 3.11. In the AIC case, the model parameters $\varphi_m(t)$ are also used by the design stage to update the internal components of the IC block; however, this happens only when an event is detected at discrete times t_i. In other words, the design stage uses sampled model parameters $\varphi_m(t_i)$ to generate a new set of controller parameters $\varphi_c(t_i)$, which are passed to the standard IC.

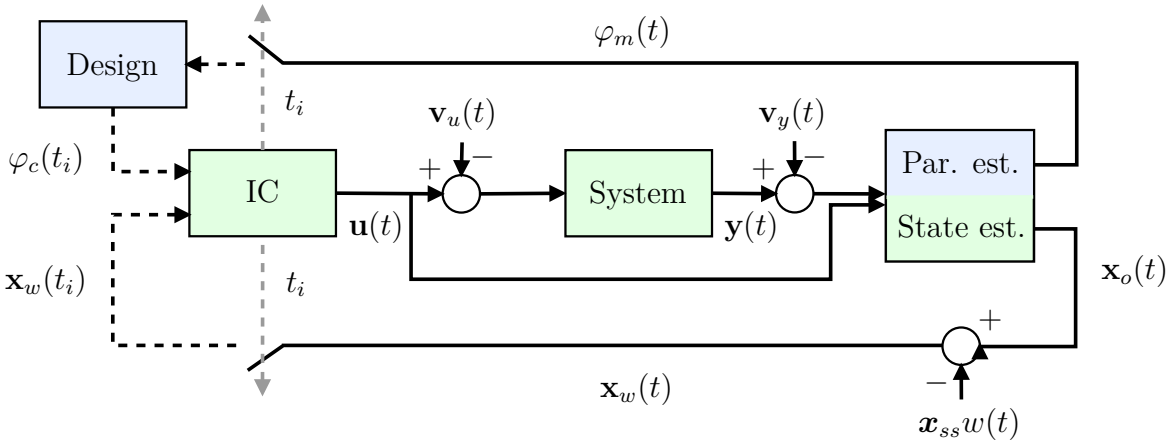


Figure 3.11: Adaptive intermittent control diagram. The block labelled as *Par est./State est.* represents a recursive algorithm that continuously estimates the system states $\mathbf{x}_o(t)$ and the model parameters $\varphi_m(t)$. Both the input $\mathbf{u}(t)$ and the output $\mathbf{y}(t)$ can be affected by input noise $\mathbf{v}_u(t)$ and measurement noise $\mathbf{v}_y(t)$ respectively. The intermittent controller block *IC* samples $\varphi_m(t)$ and the setpoint version of the states $\mathbf{x}_w(t)$ at discrete points in time t_i determined by the event generated by its triggering mechanism. The *Design* block represents the controller redesign stage and it provides all the parameters needed by the standard intermittent controller *IC*. The grey dashed lines represent event times, that determine when the *IC* the loop is opened or closed. The blue colour represents the components of the *adaptation loop*, while green is used to identify the components in the *feedback loop*.

The setpoint $w(t)$ is introduced to the system by subtracting it from the state estimates $\mathbf{x}_o(t)$, resulting in $\mathbf{x}_w(t)$, to then be used as feedback by the IC. In contrast with the continuous version in Fig. 3.10, the vector $\mathbf{x}_w(t)$ is sampled every time the loop is closed by the triggering mechanism. The Design block in the AIC case is in charge of three main tasks, which can be summarised as follows:

1. Compute a new set of closed-loop controller gains based on available estimated parameters.
2. Update the elements that are part of the hold mechanism, including the intermittent control gains.
3. Update the predictor matrices.

The tasks mentioned above are also part of the design of standard (non-adaptive) intermittent controllers, with the difference that in the adaptation context, these tasks are executed when there is an event. For standard non-adaptive IC, the computations can be performed offline. The details and equations of the design stage are introduced in section 3.6.3. The following section describes the algorithms that are used to perform the state and parameter estimation procedure shown in the aforementioned controllers.

3.6.1 State and parameter estimation

A common way to implement state estimation in situations where one or more elements of the state vector are not directly measurable is using state observers. An observer is also referred to as a soft sensor since it can generate the trajectories of unknown states only using past input and output data, and a dynamical model of the system. For cases where the dynamics can be approximated by a linear system, a Luenberger observer or a linear Kalman filter can be used to estimate the system states.

The adaptation layer of AIC assumes that there is a way to estimate the time-varying parameters of the system that is being controlled. This can be achieved by a recursive algorithm that performs online system identification. The flexible structure of the Kalman filter has been exploited in the past to formulate the filter either as a parameter estimator, a state estimator, or as *joint estimator* which estimates both at the same time (Haykin, 2001). To formally introduce this idea, consider the following nonlinear dynamical system:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \boldsymbol{f}(\mathbf{x}(t), \mathbf{u}(t), \varphi_m(t)) + \mathbf{w}(t) \\ \mathbf{y}(t) &= \boldsymbol{h}(\mathbf{x}(t), \mathbf{u}(t)) + \mathbf{v}(t) \,, \end{aligned} \tag{3.76}$$

where $\mathbf{w} \in \mathbb{R}^n$ and $\mathbf{v} \in \mathbb{R}^{n_y}$ are the process and measurement noise vectors, respectively. Also, they are assumed to be additive, Gaussian, with zero mean and known covariance matrices $\mathbf{Q}$ and $\mathbf{R}$. The dimension of the output vector is represented by n_y, and n is the number of states.

The *joint estimation* problem consists of augmenting the state vector $\mathbf{x}(t)$ to include the model parameters as extra states, implicitly assuming that they stay constant. Defining the augmented state-vector $\mathbf{x}_{aug}(t) = [\mathbf{x}(t)\ \varphi_m(t)]^T$, the new dynamical model is

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\varphi}_m(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}\left(\mathbf{x}(t), \mathbf{u}(t), \varphi_m(t)\right) \\ 0 \end{bmatrix} . \tag{3.77}$$

The equations of the Kalman filter are then applied to the system defined in (3.77).

Another formulation that allows the estimation of both states and parameters is known as *dual estimation*, where two separate filters are implemented concurrently, one estimates states and the second one does the same for the parameters. Dual estimation introduces extra computations since a second filter is needed, however it is possible to disable the parameter estimation if the filter converged to the real parameters. In (Haykin, 2001), it is argued that the dual schemes have better convergence properties over the joint schemes, although in (Hegyi et al., 2006) a detailed comparison between

them revealed that the errors of Kalman based dual estimation are larger compared to joint estimators for cases where the covariance matrices differ by several orders of magnitude. Additionally, the joint estimators explicitly allow statistical dependence between states and parameters, whereas in dual estimation the cross covariances are not estimated, implying total independence. Some situations might assume that there is some degree of correlation between states and parameters, therefore the joint estimator might be better suited for these scenarios (Van Der Merwe, 2004).

3.6.2 The extended and unscented Kalman filters

The Kalman filter is considered as an optimal Bayesian estimator for linear, Gaussian systems. This implies that if the system is truly linear, the filter would provide unbiased estimates. However, most of the problems in engineering are better described by nonlinear dynamics under a stochastic environment. For this reason, the linear Kalman filter has been extended in several ways in order to be applied to nonlinear systems (Lee and Ricker, 1993; Julier et al., 1995; Van Der Merwe and Wan, 2001). This section discusses two of these extensions: the Extended Kalman filter (EKF) and the Unscented Kalman filter (UKF).

The extended Kalman filter

An intuitive way to work with nonlinear systems is to assume that they would behave linearly for a very short period of time. In other words, if the nonlinear equations are approximated at each time step by linear system, then a standard Kalman filter could be applied iteratively. This is the working principle of the EKF. The most common way to implement this is to approximate the system using a Taylor series expansion around the estimate, resulting in linearised Jacobian matrices that are then used by the Kalman filter algorithm.

The main benefit of the EKF is that its formulation is relatively simple compared to other nonlinear versions of the filter, making it easier to understand and implement. Probably this is the reason why it has been adopted as the de facto alternative for estimation problems in many areas of engineering. It must be said that the EKF would perform poorly if the assumptions of local linearity are violated, in this case, the linearisation would introduce errors that might end in estimates that do not converge to the real values. Also, in order to avoid the online calculation of the derivatives involved in the linearisation process, explicit versions of the Jacobian matrices are normally computed offline and then implemented in the algorithm. This is a simple

task for systems with a small number of states, however, large systems might require long expressions that might introduce errors when translated into code.

Normally, the equations for Kalman filtering algorithms are presented from a discrete point of view since it was conceived to be implemented in a computer, although continuous formulations are also possible (Kalman and Bucy, 1961; Ruymgaart and Soong, 1988). The steps involved in the EKF algorithm for a discretised version of the system described in (3.76) are as follows:

1. **Initialisation step**: define initial values for the estimated states $\hat{\mathbf{x}}_0$, the initial error covariance matrix $\mathbf{P}_0$, and the process and measurement noise covariance matrices $\mathbf{Q}$ and $\mathbf{R}$.

2. **Prediction step**: compute system Jacobian $\mathbf{A}_k$, the a-priori state estimate $\hat{\mathbf{x}}_k^-$, and the a-priori error covariance matrix $\mathbf{P}_k^-$.

$$\mathbf{A}_k = \left.\frac{\partial \boldsymbol{f}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_{k-1}} \tag{3.78}$$

$$\hat{\mathbf{x}}_k^- = \boldsymbol{f}(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_{k-1}) \tag{3.79}$$

$$\mathbf{P}_k^- = \mathbf{A}_k \mathbf{P}_{k-1}^+ \mathbf{A}_k^T + \mathbf{Q}\,. \tag{3.80}$$

3. **Correction step**: compute the observation Jacobian $\mathbf{C}_k$ and the Kalman gain $\mathbf{K}_k$, correct the a-priori state estimates using measurements $\mathbf{y}_k$ to produce a-posteriori estimates $\hat{\mathbf{x}}_k^+$, and obtain the a-posterirori error covariance matrix $\mathbf{P}_k^+$ for the next iteration.

$$\mathbf{C}_k = \left.\frac{\partial \boldsymbol{h}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_k^-, \mathbf{u}_{k-1}} \tag{3.81}$$

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{C}_k^T \left[\mathbf{C}_k \mathbf{P}_k^- \mathbf{C}_k^T + \mathbf{R}\right]^{-1} \tag{3.82}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left[\mathbf{y}_k - \boldsymbol{h}(\hat{\mathbf{x}}_k^-, \mathbf{u}_{k-1})\right] \tag{3.83}$$

$$\mathbf{P}_k^+ = \left[\mathbf{I} - \mathbf{K}_k \mathbf{C}_k\right] \mathbf{P}_k^-\,. \tag{3.84}$$

Equations (3.78) and (3.81) should be implemented carefully, in some situations it is possible to obtain analytical expressions of the derivatives which reduce the computational burden of the filter. This is preferable compared to the online approximation of the derivatives. This motivated the extension of the filter based on fully nonlinear methods such as the Unscented Kalman filter.

The unscented Kalman filter

The truncation of the higher order terms made in the linearisation step of the EKF algorithm (3.78) is the reason why the EKF is considered a suboptimal filter (Cox, 1964; Athans et al., 1968). In order to capture the behaviour of the neglected terms, it has been proposed that using the full nonlinear equations might result in more accurate estimates. This idea motivated nonlinear extensions of the filter that essentially eliminated the explicit derivatives involved in the Jacobians of the EKF while increasing the convergence properties. One particular formulation known as the Unscented Kalman filter has gained attention since it has produced higher accuracy in different situations compared to the EKF (Julier et al., 1995, 2000; Arulampalam, 2004; Chowdhary and Jategaonkar, 2010; Meskin et al., 2013; Biswas et al., 2017).

The UKF implements a statistical alternative to the Jacobian matrices of the EKF known as the *unscented transformation*, which consists of using a set of deterministically selected points, collected from the a-priori mean and covariance of the states (also known as sigma points) and passing them through a nonlinear transformation. The a-posteriori mean and covariance of the states are obtained from the transformed sigma points. The spread of the sigma points is determined based on design parameters that represent the scaling of the *unscented transformation*.

The steps involved in the UKF algorithm are summarised as follows:

1. **Initialisation step**: define initial values for the estimated states $\hat{\mathbf{x}}_0$, the initial error covariance matrix $\mathbf{P}_0$, and the process and measurement noise covariance matrices $\mathbf{Q}$ and $\mathbf{R}$.

2. **Define design parameters** α, β, κ, and calculate the scaling parameter λ. Then compute the weight vectors η_0^m, η_0^c, and η_i^m, considering that n represents the number of states.

$$\lambda = c - n \quad \text{where} \quad c = \alpha^2(n + \kappa) \tag{3.85}$$

$$\eta_0^m = \lambda/(n + \lambda) \tag{3.86}$$

$$\eta_0^c = \lambda/(n + \lambda) + 1 - \alpha^2 + \beta \tag{3.87}$$

$$\eta_i^m = \eta_i^c = 1/\left[2(n + \lambda)\right], \quad \text{for} \quad i = 1, ..., 2n\,. \tag{3.88}$$

3. **Compute the sigma points**

$$\boldsymbol{\chi} = \begin{bmatrix} \hat{\mathbf{x}}_{k-1}^{+} & \hat{\mathbf{x}}_{k-1}^{+} + \sqrt{c\,\mathbf{P}_{k-1}^{+}} & \hat{\mathbf{x}}_{k-1}^{+} - \sqrt{c\,\mathbf{P}_{k-1}^{+}} \end{bmatrix} . \tag{3.89}$$

4. **Prediction transformation**: propagate the sigma points $\boldsymbol{\chi}$ through the process model, compute the mean of a-priori predicted states $\hat{\mathbf{x}}_k^-$, finally compute the a-priori state covariance matrix $\mathbf{P}_k^-$.

$$\boldsymbol{\chi}_k^- = \boldsymbol{f}(\boldsymbol{\chi}, \mathbf{u}_{k-1}) \tag{3.90}$$

$$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2n} \eta_i^m \boldsymbol{\chi}_{k,i}^- \tag{3.91}$$

$$\mathbf{P}_k^- = \mathbf{Q} + \sum_{i=0}^{2n} \eta_i^c \, (\boldsymbol{\chi}_{k,i}^- - \hat{\mathbf{x}}_k^-)(\boldsymbol{\chi}_{k,i}^- - \hat{\mathbf{x}}_k^-)^T . \tag{3.92}$$

5. **Observation transformation**: propagate the a-priori sigma points $\boldsymbol{\chi}_k^-$ through the observation model, compute the mean of the predicted output $\hat{\mathbf{y}}_k^-$ and its covariance matrix $\mathbf{P}_k^{yy}$, finally obtain the cross-covariance matrix $\mathbf{P}_k^{xy}$.

$$\boldsymbol{\psi}_k^- = \boldsymbol{h}(\boldsymbol{\chi}_k^-, \mathbf{u}_{k-1}) \tag{3.93}$$

$$\hat{\mathbf{y}}_k^- = \sum_{i=0}^{2n} \eta_i^m \boldsymbol{\psi}_{k,i}^- \tag{3.94}$$

$$\mathbf{P}_k^{yy} = \mathbf{R} + \sum_{i=0}^{2n} \eta_i^c \, (\boldsymbol{\psi}_{k,i}^- - \hat{\mathbf{y}}_k^-)(\boldsymbol{\psi}_{k,i}^- - \hat{\mathbf{y}}_k^-)^T \tag{3.95}$$

$$\mathbf{P}_k^{xy} = \sum_{i=0}^{2n} \eta_i^c \, (\boldsymbol{\chi}_{k,i}^- - \hat{\mathbf{x}}_k^-)(\boldsymbol{\psi}_{k,i}^- - \hat{\mathbf{y}}_k^-)^T . \tag{3.96}$$

6. **Correct estimates using measurements**: compute the Kalman gain $\mathbf{K}_k$, correct the a-priori state estimates using measurements $\mathbf{y}_k$ to produce a-posteriori estimates $\hat{\mathbf{x}}_k^+$, and obtain the a-posterirori error covariance matrix $\mathbf{P}_k^+$ for the next iteration.

$$\mathbf{K}_k = \mathbf{P}_k^{xy} \, (\mathbf{P}_k^{yy})^{-1} \tag{3.97}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left[\mathbf{y}_k - \hat{\mathbf{y}}_k^- \right] \tag{3.98}$$

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}_k^{yy} \mathbf{K}_k^T . \tag{3.99}$$

The *unscented transformation* scaling parameter α determines the spread of the sigma points. A small value of leads to tighter selection of the sigma-points. A range of commonly used values is from 0.0001 to 1. The β parameter includes information of the

a-priori distribution, with the consideration that for Gaussian distributions the value of $\beta = 2$ is optimal. Lastly, the κ parameter is normally set to 0. The calculation of the sigma points in (3.89) involves the implementation of the square root of the state covariance matrix $\mathbf{P}_{k-1}^{+}$, and in order to do this efficiently, a Cholesky decomposition method can be used, which yields a lower triangular matrix that is used as a representation of the square root (Press et al., 1996).

Both the EKF and UKF can be used to provide states $\mathbf{x}_o$ and parameters φ to the design step of the AIC. The following section explains the steps needed to redesign the standard intermittent controller based on available estimates.

3.6.3 Adaptive IC design

The general intermittent control law can be formulated as

$$\mathbf{u}(t) = \mathbf{u}(t_i + \tau) = -\mathbf{k}\mathbf{x}_h(\tau) + \boldsymbol{u}_{ss} w(t) \quad \text{for} \quad t_i \leq t < t_{i+1} \,, \tag{3.100}$$

which considers the time between events determined by τ. including t_i, and that uses the states generated by the hold mechanism $\mathbf{x}_h$ to drive the system. When there is an event, however, the hold states are reset according to the following expression

$$\mathbf{x}_h(t_i) = \mathbf{U}_i \,, \tag{3.101}$$

where $\mathbf{U}_i = \mathbf{K}_h \mathbf{x}_p(t_i - t_d)$, and $\mathbf{x}_p$ are the states that predictor system outputs after considering the time-delay t_d, which are based on the states provided by the estimation process in place, referred as $\mathbf{x}_w$. This allows us to write the control law at t_i as

$$\mathbf{u}(t_i) = -\mathbf{k}\mathbf{K}_h \mathbf{x}_p(t_i - t_d) + \boldsymbol{u}_{ss} w(t_i) \,. \tag{3.102}$$

During $t_i < t < t_{i+1}$, the hold states are determined by the following system

$$\dot{\mathbf{x}}_h(\tau) = \mathbf{A}_h \mathbf{x}_h(\tau) \,, \tag{3.103}$$

where $\mathbf{A}_h$ is the generalised hold, which dictates the dynamics of $\mathbf{x}_h$. Equations (3.100),(3.101), and (3.103), depend directly on the values of the system matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, that are part of a linear state-space representation that approximates the dynamics of the system that is being controlled, as in (3.1). Normally, these matrices contain entries that are linear expressions based on parameters of the system, in other words, they are parameterised. Therefore, it is possible to use a parameter estimation routine to update these matrices in real-time. Once the updated matrices are obtained,

a controller redesign can take place.

Based on a recursive state-parameter estimation algorithm such as the EKF or the UKF, it is possible to establish the previously mentioned control design stage that obtains an updated model and a new control signal according to variations in the system parameters. In this sense, two controllers based on the IC framework can be proposed: 1) Adaptive system-matched hold IC, and 2) Adaptive tapping IC. The design of both controllers starts with an *underlying continuous design* (UCD), then a *generalised hold* is established, and finally if there is the need of implementing state prediction, the *intermittent prediction* matrices must be obtained. In a non-adaptive case, the entire design procedure can be done offline, whereas in the adaptive version the aforementioned design procedure must be done when an event is generated.

To illustrate the steps involved in the design stage, let us assume that an estimation algorithm is capable of tracking time-varying parameters in the plant, feeding them back to the design stage *continuously*. The effect of these parameter variations can affect the system dynamics considerably, and since the initial controller was designed for a "different" plant, events would be generated in consequence by the triggering mechanism of the IC. Therefore, when there is an event, a new set of system matrices is generated using the estimated parameters as follows

$$\mathbf{A}_i = \mathbf{A}(\varphi_m(t_i)) \ , \ \mathbf{B}_i = \mathbf{B}(\varphi_m(t_i)) \, , \tag{3.104}$$

where $\varphi_m(t_i)$ represents the set of parameters that are being tracked by the recursive estimator at t_i, which are part of the parameterisation of matrices $\mathbf{A}$ and $\mathbf{B}$. This implies that the value of $\mathbf{A}_i$ and $\mathbf{B}_i$ remains constant until the next event is generated at t_{i+1}, this can be defined as

$$\mathbf{A}_i = \mathbf{A}(t_i + \tau) \ , \ \mathbf{B}_i = \mathbf{B}(t_i + \tau) \quad \text{for} \quad t_i \leq t < t_{i+1} \, . \tag{3.105}$$

The computation of $\mathbf{A}_i$ and $\mathbf{B}_i$ is considered the initial step of the design stage. A fast and accurate estimation of the parameter would reduce the effect of the variations considerably. For this reason, it is important to have robust algorithms that can provide the parameters reliably since the entire controller redesign depends on them.

Underlying continuous design

A carefully designed control law should be obtained before establishing the components of an intermittent controller. The fact that it is called *continuous* reflects that on its own it can be considered as a standard state-feedback controller, that if implemented,

would use feedback continuously. This step should meet all design specifications and requirements such as rise time, overshoot and steady-state error. In essence, the underlying continuous design is comprised of two independent steps: i) steady-state design, and ii) the computation of state-feedback gains, as shown in section 3.2.1. In the adaptation context, these steps should be performed at every event time t_i and can be summarised as follows

1. **Steady-state design**: compute $\boldsymbol{x}_{ss}(t_i)$ and $\boldsymbol{u}_{ss}(t_i)$ by solving the following system
$$\begin{bmatrix} \boldsymbol{x}_{ss}(t_i) \\ \boldsymbol{u}_{ss}(t_i) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_i & \mathbf{B}_i \\ \mathbf{C} & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0}_{n\times 1} \\ 1 \end{bmatrix} . \tag{3.106}$$
2. **State-feedback gains**: using the LQR approach, compute the gains
$$\mathbf{k}_i = \mathbf{R}_c^{-1}\mathbf{B}_i^T\mathbf{P}_i \, , \tag{3.107}$$
where $\mathbf{P}_i = \mathbf{P}(t_i + \tau)$ is the solution of the ARE defined in (3.10) at time t_i.

The steps of the UCD serve as the basis of IC regardless of the nature of the hold mechanism.

Generalised hold

The generalised hold $\mathbf{A}_h$ determines the dynamics of the system during the open-loop interval Δ_{ol}. The two adaptive controllers proposed in this chapter use different hold mechanisms. One uses a system-matched hold which effectively approximates the dynamics of the closed-loop system defined by the UCD, whereas the tapping hold is based on an impulse-like control signal defined by Laguerre functions. Both schemes are explained in section 3.2.3. The equations involved in defining $\mathbf{A}_h$ for an adaptive context are explained in the following paragraphs.

1. **System-matched hold**: this hold requires the computation of the closed-loop matrix $\mathbf{A}_c$ as follows
$$\mathbf{A}_h(t_i) = \mathbf{A}_c(t_i) = \mathbf{A}_i - \mathbf{B}_i\mathbf{k}_i \, . \tag{3.108}$$
This definition converts the generalised hold $\mathbf{A}_h$ into a time-varying matrix that is redefined every t_i and stays constant throughout the open-loop interval as in $\mathbf{A}_h(t_i) = \mathbf{A}_c(t_i + \tau)$. The hold mechanism in general requires the computation of an intermittent control gain $\mathbf{K}_h$, which for the system-matched hold case is defined as a $n \times n$ identity matrix where n is the number of states in (3.1).

2. **Tapping hold**: this hold requires the design of a square matrix $\mathbf{A}_p$ whose elements are defined by Laguerre functions, as in (3.39). This matrix defines the shape of the tap and its value is assigned to the generalised hold ($\mathbf{A}_h = \mathbf{A}_p$). Notice that since $\mathbf{A}_p$ can be designed offline, $\mathbf{A}_h$ stays constant during the entire time the controller is operational, contrary to the system-matched hold case. However, the intermittent control vector $\mathbf{K}_h$ is recalculated once an event is detected. First, an autonomous system formed by $\boldsymbol{X} = \begin{bmatrix} \mathbf{x} & \mathbf{x}_h \end{bmatrix}^T$ is defined as

$$\dot{\boldsymbol{X}}(\tau) = \mathbf{A}_{xu}(t_i)\boldsymbol{X}(\tau) \ , \tag{3.109}$$

where τ is the intermittent time, the initial conditions are $\boldsymbol{X}_i = \begin{bmatrix} \mathbf{x}(t_i) & \mathbf{U}_i \end{bmatrix}^T$, and the matrix $\mathbf{A}_{xu}(t_i)$ is

$$\mathbf{A}_{xu}(t_i) = \begin{bmatrix} \mathbf{A}_i & \mathbf{B}_i\mathbf{k} \\ 0_{n\times n} & \mathbf{A}_h \end{bmatrix} . \tag{3.110}$$

Notice the appearance of the state feedback gain $\mathbf{k}$ in (3.110). Following the procedure in (Gawthrop and Gollee, 2012), the approach is to fix $\mathbf{k}$ as a unity vector of $n \times n_u$, where n_u corresponds to the number of inputs in the system, and only determine $\mathbf{K}_h$.

The solution of (3.109) is then used to formulate a cost function that depends on $\mathbf{P}_i$, which is the solution to the ARE obtained in the UCD step

$$J_{ITC} = \boldsymbol{X}(\tau_1)^T\mathbf{P}_{xu}(t_i)\boldsymbol{X}(\tau_1) \, , \tag{3.111}$$

where τ_1 is the optimisation horizon used in (3.41) and $\mathbf{P}_{xu}(t_i)$ is

$$\mathbf{P}_{xu}(t_i) = \begin{bmatrix} \mathbf{P}_i & 0_{n\times n} \\ 0_{n\times n} & 0_{n\times n} \end{bmatrix} . \tag{3.112}$$

By calculating the matrix $J_{XX}(t_i) = e^{\mathbf{A}_{xu}(t_i)^T\tau_1}\left(\mathbf{P}_{xu}(t_i)\right)e^{\mathbf{A}_{xu}(t_i)\tau_1}$, it is possible to rewrite the cost function as $J_{ITC} = \boldsymbol{X}_i^T J_{XX}(t_i)\boldsymbol{X}_i$. Then, the intermittent control gain at t_i can be computed using

$$\mathbf{K}_h(t_i) = J_{UU}^{-1}J_{Ux} \, , \tag{3.113}$$

with J_{UU} and J_{Ux} being entries of (3.50).

Intermittent prediction

If state prediction is needed to compensate against system delays, then the matrices $\mathbf{E}_{pp}$ and $\mathbf{E}_{ph}$ in (3.59) should be evaluated every t_i, resulting in the following expression

$$\mathbf{x}_p(t_i) = \mathbf{E}_{pp}\mathbf{x}_w(t_i) + \mathbf{E}_{ph}\mathbf{x}_h(t_i)\,. \tag{3.114}$$

These matrices come from an autonomous dynamical system formed by a state-vector defined by the predicted and the hold states $\bar{X}(\tau) = [\,\mathbf{x}_p(\tau) \quad \mathbf{x}_h(\tau)\,]$

$$\frac{d}{d\tau}\bar{X}(\tau) = \mathbf{A}_{ph}(t_i)\bar{X}(\tau)\,, \tag{3.115}$$

which is driven by the dynamics imposed by matrix $\mathbf{A}_{ph}(t_i)$, defined as

$$\mathbf{A}_{ph}(t_i) = \begin{bmatrix} \mathbf{A}_i & -\mathbf{B}_i\mathbf{k}_i \\ 0_{n\times n} & \mathbf{A}_h \end{bmatrix}\,. \tag{3.116}$$

Notice that $\mathbf{A}_{ph}(t_i)$ contains the generalised hold matrix $\mathbf{A}_h$ defined in the previous section. The solution of (3.115) involves the computation of matrix $\mathbf{E}(t_i)$ (defined in (3.60)), which contains the necessary matrices to obtain (3.114).

3.6.4 Summary

The previous section defined the equations for system-matched hold and tapping based adaptive intermittent controllers. Both evolve in an open-loop configuration between events, based on dynamics dictated by a generalised hold. The occurrence of an event forces the use of feedback, sampling the estimated states and parameters and then redesigning the controller for the next open-loop interval. A summary of the main steps of AIC is presented in the following list:

Adaptive Intermittent Control

Initialisation (offline): design an initial IC based on the state-space model defined in (3.1), and the initial parameters for the EKF/UKF.

Open-loop control $(t \neq t_i)$

Apply $\mathbf{u}(t) = -\mathbf{k}\mathbf{x}_h(\tau) + \boldsymbol{u}_{ss}w(t)$ for $t_i < t < t_{i+1}$.

Closed-loop control $(t = t_i)$

Apply $\mathbf{u}(t_i) = -\mathbf{k}\mathbf{K}_h\mathbf{x}_p(t_i - t_d) + \boldsymbol{u}_{ss}w(t_i)$ and update control parameters for the next Δ_{ol} as follows:

1. Update state-space model

Compute the new system matrices based on the set of estimated parameters φ_m, $\mathbf{A}_i = \mathbf{A}(\varphi_m(t_i)),\ \mathbf{B}_i = \mathbf{B}(\varphi_m(t_i))$.

2. Underlying continuous design

- Steady-state: compute $\boldsymbol{x}_{ss}(t_i)$ and $\boldsymbol{u}_{ss}(t_i)$ by solving (3.106)
- State-feedback (LQR): compute $\mathbf{k}_i = \mathbf{R}_c^{-1}\mathbf{B}_i^T\mathbf{P}_i$, by solving the ARE in (3.10)

3. Generalised hold

System-matched

- Obtain $\mathbf{A}_c(t_i)$

$$\mathbf{A}_h(t_i) = \mathbf{A}_c(t_i) = \mathbf{A}_i - \mathbf{B}_i\mathbf{k}_i$$

Tapping

- Find $\mathbf{A}_{xu}(t_i)$ based on (3.110), with $\mathbf{A}_h = \mathbf{A}_p$ defined in (3.39)
- Find $J_{XX}(t_i)$ based on (3.112)
- Obtain $\mathbf{K}_h(t_i) = J_{UU}^{-1}J_{Ux}$ using (3.50)

4. Intermittent prediction

- Find $\mathbf{E}(t_i)$ based on (3.60) and the value of $\mathbf{A}_{ph}$ defined in (3.116)
- Compute the predicted states using $\mathbf{x}_p(t_i) = \mathbf{E}_{pp}\mathbf{x}_w(t_i) + \mathbf{E}_{ph}\mathbf{x}_h(t_i)$

3.7 Discussion

In this section, we will discuss some aspects that should be considered when implementing AIC. For this purpose, the ideas about AIC as a whole are considered first, followed by parameter and state estimation related remarks to then end with the ones that are control specific.

IC is a model based control strategy, which implies that dynamical model of the system is needed to design a controller. When the model matches reality closely, the designed controller would have the expected performance. However, modelling is not an easy problem in higher dimension systems with complex nonlinear dynamics and interactions. A first source of discrepancy is introduced when the values of the parameters used to design the controller are not close to the *true* system parameters. Although it is not possible to know the *true* parameters, a strategy like adaptive control is useful

when the discrepancy affects the dynamics considerably, and reducing it is critical to achieve the desired performance. Nevertheless, there are situations where neglected dynamics might introduced an extra source of error that the adaptive control might not be able to counteract unless the model used to design the controller includes them explicitly. Therefore, having a model that is as close as possible to reality greatly improves the performance of any adaptive controller.

The aforementioned adaptive controllers are all based on the assumption that the dynamical model of the system can be written in a state-variable form, where the state vector includes system states $\mathbf{x}$ and also the parameters of interest $\varphi(t)$, which is fundamental to formulate a joint Kalman filter as a recursive estimator by means of an augmented state vector. A linearisation process on the nonlinear system yields linear system matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, which contain entries that depend on the parameters of interest. For implementation purposes, only the entries dependent on $\varphi(t)$ must be recomputed every t_i.

The indirect self-tuning architecture of the proposed AIC allows the formulation controllers that might rely on different estimation techniques such as a more traditional recursive least squares algorithm. This provides flexibility in the design of AIC and makes testing a simpler process since the controller is unaffected by the chosen estimation algorithm in terms of its architecture. However, the accuracy of the parameter estimation does play an important role in order to improve performance.

State and parameter estimation

The initial values of the EKF and UKF have an important role in the performance of both filters as stated by (Crassidis, 2006), where large initialisation errors might have a greater impact in terms of the linearisation errors associated with the EKF. There are some authors that state that the reported under-performing EKF results in literature are mostly related to a trial and error selection of its design parameters such as the process covariance matrix $\mathbf{Q}$ and measurement noise covariance matrix $\mathbf{R}$, mentioning that a well tuned EKF might perform as well as more complex filters (Daum, 2005; Schneider and Georgakis, 2013). The EKF is attractive to engineers because its equations are probably easier to understand compared to those of the UKF and in many situations the differences in performance are rather small (Hegyi et al., 2006; Gross et al., 2012); however, the performance of the EKF might differ and even diverge from that of the UKF in situations when the process to be estimated is *highly* nonlinear.

It is important to notice that the computational cost of the UKF is of the same order than the EKF (Wan and Van Der Merwe, 2000) and it captures the posterior mean and covariance reliably to the third order of a Taylor series expansion when propagated through the nonlinear system. This feature might be a decisive factor when selecting the kind of filter to implement depending on the type of system.

Control

The differences between the proposed AIC schemes are the hold mechanism (either system-matched or tapping) and the comparison made by the triggering mechanism to determine event times (using predicted states $\mathbf{x}_p$ for a system-matched hold and closed loop states $\mathbf{x}_c$ for the tapping case) as explained in section 3.2.5.

An adaptive system-matched based IC is a natural extension of the non-adaptive versions presented in (Gawthrop and Wang, 2009a). This version in particular has been used in many other publications of the same authors. Its formulation is intuitive in the sense that the inter-sample behaviour is dictated by a dynamical system that would evolve in the same way as the closed-loop system in the absence of perturbations. Therefore, the hold is automatically determined by the underlying continuous design process which can be achieved by traditional linear control methods. On the contrary, adaptive tapping IC which is based on (Gawthrop and Gollee, 2012) uses orthogonal functions known as Laguerre polynomials to form the hold. As a result, this controller uses an extra set of design parameters compared to the system-matched version such as the p parameter and the order N of the Laguerre matrix $\mathbf{A}_p$ defined in 3.2.3.

The following two chapters describe different applications of intermittent controllers to real-time systems. The next chapter presents how multivariable IC was used to balance a humanoid robotic structure, focusing on clock and event-driven versions of IC. This implementation constitutes the first use of IC (as described in this chapter) as a control framework for a multi-segmental robot. This is followed by a chapter that discusses the implementation of Adaptive IC on a rotational inverted pendulum system, in simulation and experimentally, showcasing some of the main properties and advantages of using this method.

Chapter 4

Intermittent control of a humanoid robot

4.1 Introduction

The extent of continuous control (CC) in human motor control covers two main ideas: i) it replicates the mean human response to stimuli and, ii) variability in human control (like the irregular body sway pattern in human balance) can be fitted by adding suitably filtered random noise to the motor and sensory states (Levison et al., 1969; Kleinman et al., 1970; Baron et al., 1970). However, the paradigmatic advance in explanation has been limited (Loram et al., 2014), as it does not include smooth control comprised of a sequence of sub-actions as it was observed in human balance (Loram and Lakie, 2002) or in more complex movements such grasping objects or even handwriting.

For this type of movements, the continuous (inner) feedback loop requires a nested control structure with a discontinuous outer feedback loop to (initiated by a triggering event or condition) automate the change from one system state to another. For example, in the case of balance, switching from an ankle strategy to a hip strategy. This requires a predefined list of states plus its triggering condition for each transition. The intermittent behaviour provides a convenient way of implementing state transitions, and the open-loop interval provides the conceptual benefit of movement and response planning inside the control loop.

The IC framework possesses several more inherent properties that could be beneficial for robotics. In the presence of (static) friction or backlash, a CC approach is not always necessary, especially when a certain residual error is acceptable. As a solution, a dead-

band can be introduced in the control scheme, as for example done in motor control (Wescott, 2010). Because of the trigger thresholds, IC provides a way to implement controller dead-bands. It is also a means to influence energy expenditure, because of the typical control pulses that can arise from the triggering. In case of delayed communication (which in humans is present due to the nervous system) smart usage of triggers and only using feedback when needed is a way of preventing instabilities and reducing the network load. For remotely operated robots, this is acknowledged and investigated by robotics and control researchers (Santos et al., 2012, 2014).

In the field of humanoid robotics, real-time constraints make online optimal control and model predictive control (MPC) challenging tasks. Because of the hybrid, nonlinear system with possibly many degrees of freedom, computing the optimal trajectory can take a lot more time than a single time step of the real-time system. Therefore, a lot of effort has been put into the development of fast, efficient algorithms, leading to promising results (Diehl et al., 2006; Todorov and Li, 2005; Li and Todorov, 2004). Though, it is not essential to complete such complex calculations in short intervals. This is why some recent work has suggested the use of a predictor to deal with computational time delay (Neunert et al., 2014; Koenemann et al., 2015). The delay in the IC framework can be used to model computational delays (Ronco et al., 1999). As a result, the framework is able to naturally deal with using computationally heavy control policies.

This chapter is organised as follows: first, an overall description the software and hardware in TUlip is given. Then, the details of the real-time experiment are introduced along with the controllers that were designed and implemented on the robot. This is followed by an overview of the analysis methods and the experiment results. An overall discussion is presented after the results to end the chapter.

4.2 TUlip humanoid robot

In a collaboration with the Technical University of Delft, specifically the Delft Centre for Systems and Control (DCSC) and Delft Biorobotics Laboratory, two visits were made to Delft, Netherlands, in order to implement and asses IC on a humanoid robot. Since 1995, Delft Biorobotics Laboratory have focused on designing balancing and walking robot prototypes as well as other bio-inspired robotic applications (Wisse et al., 2005; Hobbelen et al., 2008; Hobbelen and Wisse, 2008; de Boer, 2012).

One of the most iconic and important prototypes they have developed is a humanoid robot named TUlip, which was designed based on two fundamental goals: i) to develop

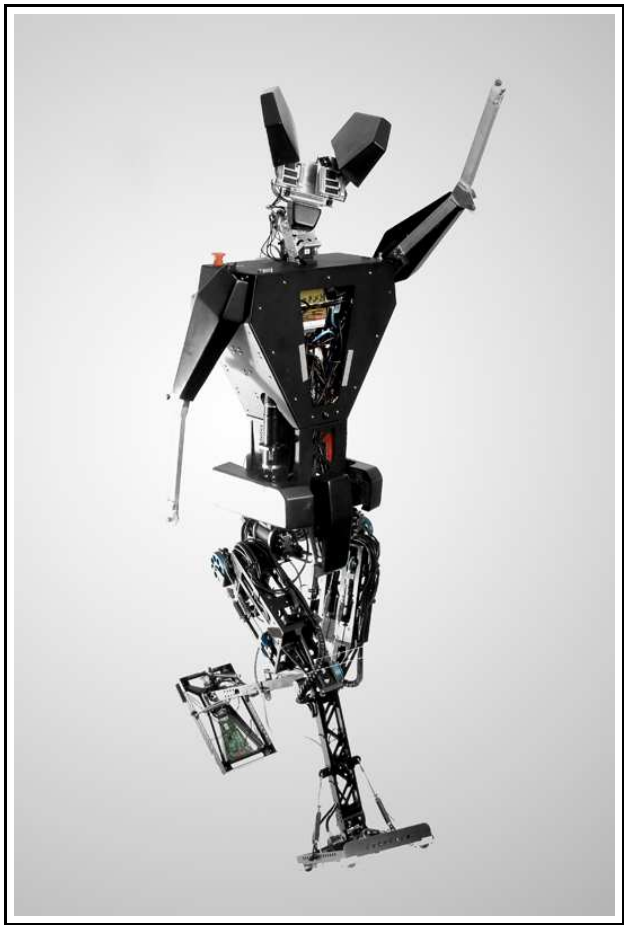


Figure 4.1: TUlip humanoid robot. Complete version of the final prototype, which was designed based on a predecessor bidpedal robot named Flame (Hobbelen et al., 2008) In the picture, the robot is presented with a head and two arms; however, these components were removed in order to run the experiments reported in this chapter.

a testing platform for energy-efficient gait controllers and, ii) to compete regularly at the robotic soccer competition known as RoboCup. Fig. 4.1 shows an image of TUlip.

These goals led the mechanical design in such a way that the prototype had human-like features, particularly the lower extremities, focusing on generating light limbs that allowed the use of low power motors. From the actuation perspective, TUlip uses a concept introduced by Pratt and Williamson (1995) called *series elastic actuation* (SEA), which has been used successfully in other platforms.

4.2.1 Series elastic actuation

The purpose of SEA is to create compliant joints by connecting the motor to the joint through an elastic element, normally a tension string. The amount of torque that is applied by the motor can be then controlled by measuring the elongation of the elastic

element. This configuration is particularly helpful to absorb impacts, extending the life of the internal components of the motors. Moreover, it enables the measurement of energy at the joint without considering the energy consumed by the motor. On the other hand, a potential disadvantage of this approach is that when large torques are demanded, motor saturation might happen since high speeds are required due to the large elongation of the spring, leading to low control bandwidth (de Boer, 2012). Fig. 4.2 shows a simplified diagram of a SEA joint.

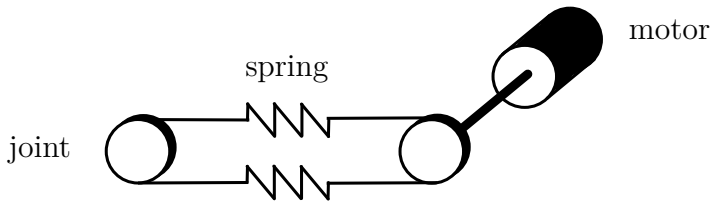


Figure 4.2: SEA Joint. Schematic of the actuation principle in TUlip. Joints are connected to the motors through elastic elements. SEA produces high torques by introducing a gearbox in the drive chain, which in turn reduces the speed and impedance due to increased inertia, friction and backlash. The role of the elastic element placed between the actuator and the load is to introduce compliance, with the benefit of improving force control and output impedance (Pratt and Williamson, 1995)

In TUlip, both the joint and the motors are equipped with incremental encoders to measure position, then by measuring the difference in orientation between them, the extension of the elastic element can be calculated, this provides a measurement of the torque applied on that particular joint.

In terms of human balance, this actuation principle is conceptually similar to the dynamic bias model used by (Loram et al., 2005). This model represents the human body as an inverted pendulum, where the gastrocnemius and soleus muscles act on the ankle joint by means of a spring-like element. It is also known that in IC, the fact that the open-loop interval allows for in-the-loop optimisation, imposes a trade-off in terms of control bandwidth and stability margins (Loram et al., 2014), rendering low bandwidth controllers compared to those of CC. For these reasons, TUlip's actuation system makes it an ideal candidate to test IC and its potential benefits to robotics.

4.2.2 Mechanical properties and hardware

The legs in TUlip have six degrees of freedom (DoF) each, with two DoF in the ankle joint, allowing roll and pitch rotations, one DoF in the knee joints and three per each hip, with a total of 12 DoF. Fig. 4.3 presents a diagram of all segments and DoF in TUlip.

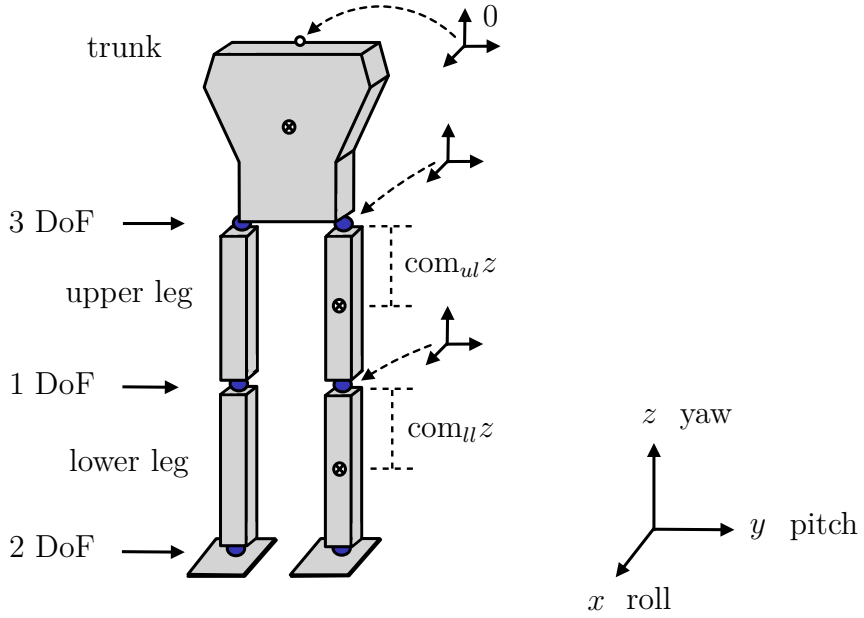


Figure 4.3: TUlip DoF diagram. Ankle and hip joints allow pitch and roll rotations, the knee is restricted to pitch rotations only. CoM position offsets are measured according to the individual coordinate frames

For this study, the DoF that were considered are the ones corresponding to pitch rotations for ankles, knees, and hip. These joints are all actuated based on the SEA principle. The motors that power all joints are Maxon RE 30 motors with HEDS 5540 encoders mounted on them to obtain the angular position of the shaft. The joints are also equipped with incremental encoders (Scancon 2RMHF) to measure the joint angle and the displacement of the elastic elements in the SEA chain. Mounted in the chest of TUlip, there is a 1GHz Diamond Systems Poseidon board used as a target computer to monitor and control the robot.

Table 4.1 shows the length and mass of each section of TUlip, which is 1.095 m tall and weighs 17.98 kg, including the position offsets in x and z directions of the CoM, which are the ones that have a greater influence on a standing position. For this particular experiment, the robot was controlled to perfrom movements only in the sagittal plane.

Table 4.1: TUlip mechanical parameters: mass, length, and CoM offsets.

Section	Mass [kg]	Length [m]	CoM in x [m]	CoM in z [m]
Trunk	11.654	0.5	0.06	-0.202
Upper legs	4.28	0.275	0.05	-0.097
Lower legs	2.046	0.320	0.0375	-0.15

Calibration

In order to execute any task with TUlip, it must be mounted on a metal rig to start a calibration procedure. This is required since the joint encoders are incremental and the reference position is lost every time the target computer is restarted. Fig. 4.4 shows TUlip in its current state, mounted in a calibration rig.

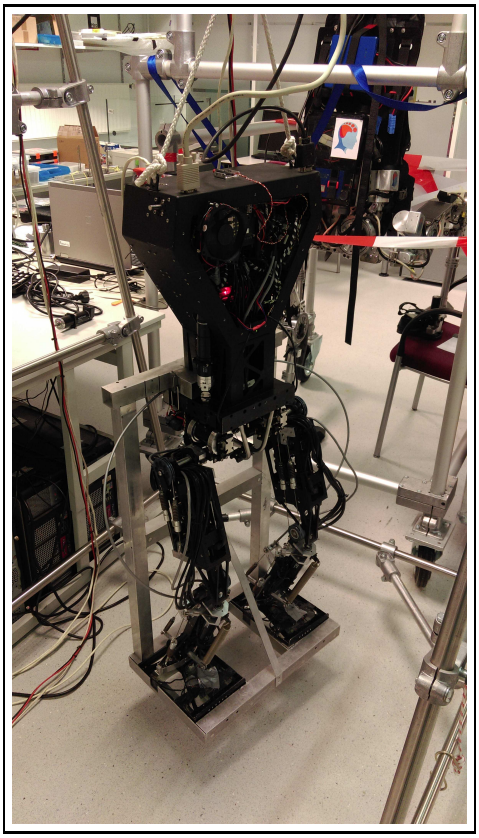


Figure 4.4: TUlip robot in calibration rig.

The calibration procedure imposes several constraints to use TUlip as an experimental platform. It normally takes between 15 and 20 minutes to position the robot in the rig, perform the calibration, to then take it out and find a stable standing configuration. This particular feature had an impact in the way the experiments were designed.

4.2.3 Software

Previous implementations in TUlip used Linux based real-time frameworks to establish communication channels with the sensors and actuators in the system. However, in order to allow quick implementations of motion controllers, the control framework was migrated to Simulink, relying on the Real-Time Windows Target (MathWorks, Inc.) toolbox to generate executable code.

The software architecture comprises low level controllers that apply appropriate PWM signals to each of the actuators. These signals are derived from high level torque references generated by motion controllers running in real-time.

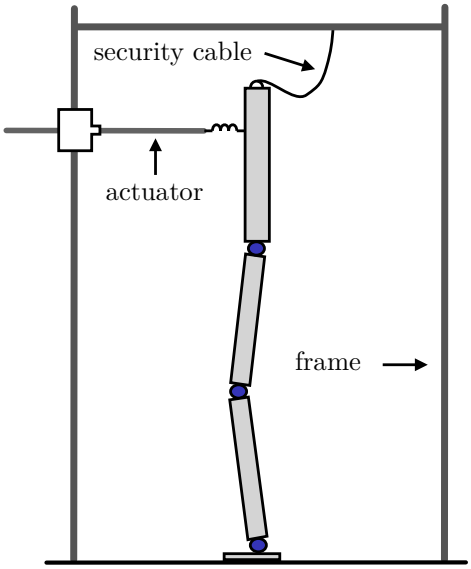
4.3 Balance experiment

The experiment performed on TUlip was a balance test (standing configuration), where the robot was perturbed by means of an actuator during different control regimes such as continuous control (CC), clock-driven IC (ICc) and event-driven IC (ICe). The purpose of this task is to evaluate how effective the different controllers are when they have to compensate for an external perturbation, and to measure for the first time the effects of a system-matched hold IC on the angles and control signals of humanoid robotic platform.

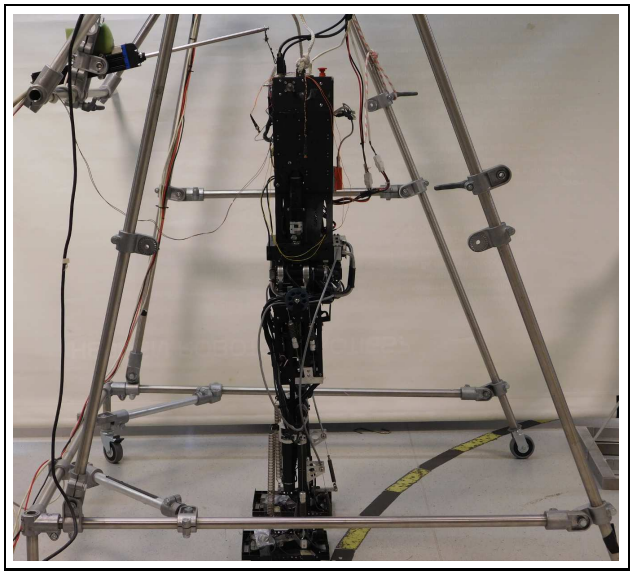
4.3.1 Experiment setup

Fig. 4.5 shows the experiment setup for the balance test, represented both as a diagram and with an image of the actual testing rig. In Fig. 4.5b, TUlip is shown standing sideways, attached to a frame via a security cable, that catches the robot in case of a sudden fall from the effects of a perturbation or from instabilities. To deliver perturbations, a linear actuator is attached to a clamping point in the trunk of the robot by means of a small spring and a cable. The role of the linear actuator is to pull TUlip in the x direction (sagittal plane), for 0.5 secs, at a constant velocity profile that was imposed after considering the transient response of the actuator, forcing the controller in place to compensate in order to bring it back to a predefined stable configuration. The clamping point was positioned in the front of the robot, at -2.5 cm from the top edge of TUlip's trunk in the z direction and at the geometrical centre in the y direction, dividing the trunk by half. This location would enforce the largest angular momentum around the ankle joints. The pull from the actuator had the same displacement profile throughout all trials.

The robot was kept at the same distance relative to the test rig for all trials, facing the linear actuator in order to ensure that the pull would apply the same amount of force. The linear actuator was fixed to the test rig at a height of 1.12 m, slightly above the edge of the trunk, in order to avoid direct contact in the case of a fall.



(a) Experiment diagram.



(b) Side view of TUlip in the test rig.

Figure 4.5: TUlip experimental setup.

4.3.2 Experiment protocol

The robot needs a control action to be applied all the time in order to maintain balance; in other words, in the absence of a stabilising controller, the robot falls to the ground quickly. Therefore, even when no perturbations are applied, the controllers must be operational. To test the effects of CC, ICc and ICc under the same calibration settings, the trials were designed in such a way that the controllers would operate sequentially with one perturbation being applied for each of them.

The trials were divided into three groups of 90 sec, allowing 30 sec of operation for the three controllers. For each group, three perturbations were applied. After the first group of perturbations, the controllers are redesigned using different LQR design parameters to then start a second group of perturbations. At the end of the second group, the controllers are redesigned one more time to run the final group. A total of 9 perturbations are applied during the trial. Two different controller sequences were tested within each group: 1) CC - ICc - ICe, and 2) ICc - ICe - CC. Fig. 4.6 shows a graphical representation of the full procedure for the first sequence.

The controller redesign mentioned above involves a new computation of the state-feedback gains $\mathbf{k}$ shown in (3.11), considering different weights for matrix $\mathbf{R}_c$ in the LQR approach. Specifically, the controllers of groups 1 and 3 are designed using scaled versions of a nominal $\mathbf{R}_c$ matrix used for group 2. The details of the control design used for all trails are given in the following sections.

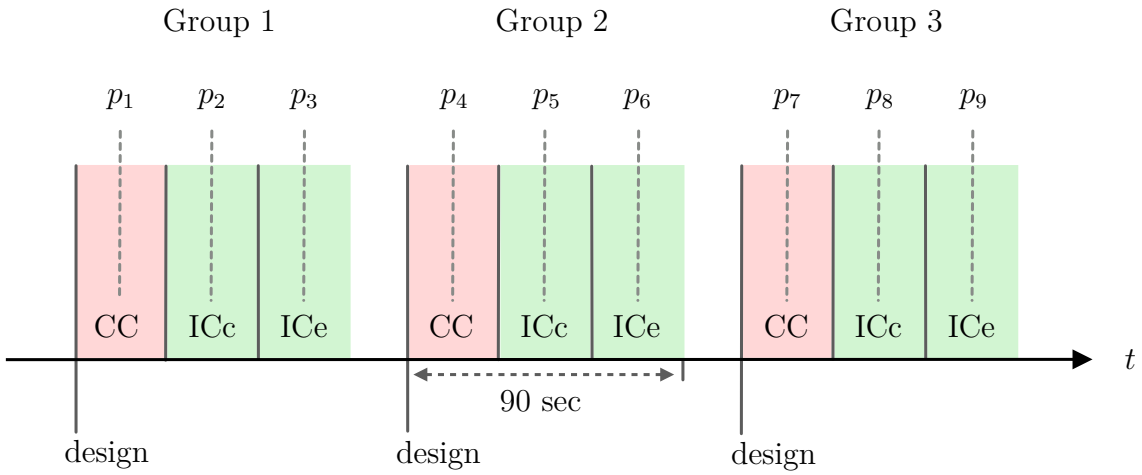


Figure 4.6: TUlip trial. Three different groups of three perturbations each are tested sequentially. Within each group, CC, ICc, and ICe controllers are used to keep TUlip in balance and reject perturbations (denoted by p) coming from the actuator. At the start of each group, the three controllers are designed using a specific LQR weight matrix $\mathbf{R}_c$, which changes for each group, meaning that the only difference between groups is the set of feedback control gains $\mathbf{k}$ that is used as a part of the underlying control design. The values of $\mathbf{R}_c$ used for the different trials are discussed in the following sections.

Data acquisition

The most relevant signals coming from the robot were logged using a data acquisition system and the Real-Time Windows Target toolbox in Simulink. For each trial, 360 sec at a sampling rate of 0.5 kHz were recorded, giving enough time to position the robot after calibration, run the disturbance rejection test and then manually save the recorded data.

4.4 Controller design

This section is dedicated to review the design considerations that were made to implement continuous and intermittent controllers in TUlip. First, the modelling of the robot in a state-space formulation is introduced including a nonlinear steady-state design procedure. Then, an explanation of the control architectures is given, concluding with the LQR and timing parameters used for the real-time experiments.

4.4.1 Dynamical model

It is common to represent a robotic system with n degrees of freedom using n generalised coordinates $\boldsymbol{q} \in \mathbb{R}^n$, that define the joint positions. This notation commonly leads to

motion equations of the form

$$\mathcal{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathcal{C}(\dot{\boldsymbol{q}}, \boldsymbol{q})\dot{\boldsymbol{q}} + \mathcal{G}(\boldsymbol{q}) = \mathcal{B}\mathbf{u}(t)\,, \tag{4.1}$$

where $\mathcal{M}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $\mathcal{C}(\dot{\boldsymbol{q}}, \boldsymbol{q}) \in \mathbb{R}^{n\times n}$ the Coriolis matrix, and $\mathcal{G}(\boldsymbol{q}) \in \mathbb{R}^n$ the gravity effects. Matrix $\mathcal{B} \in \mathbb{R}^{n\times n_u}$ maps the control inputs to the degrees of freedom and $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^{n_u}$ are the n_u control inputs, bounded by actuator limits. These equations can be derived manually using standard methods as in for example (Spong et al., 2006; Murray et al., 1994). Alternatively, they can be computed with efficient algorithms (Featherstone, 2008, 2010a,b). Given the equations of motion of (4.1), the state space equations can be formulated in such a way where $\mathbf{x} \in \mathbb{R}^{n_x}$, with $n_x = 2n$, represents the system state.

$$\dot{\mathbf{x}}(t) = \boldsymbol{f}(\mathbf{x}(t), \mathbf{u}(t))\,. \tag{4.2}$$

The nonlinear motion equations described in (4.2) are often linearised around an equilibrium point, with the purpose of using linear techniques to derive appropriate controllers. Also for the purpose of understanding human balance, single or multiple degree of freedom pendulum models are often linearised (Gawthrop et al., 2015) (Alexandrov et al., 2005) (van der Kooij et al., 2001) (Günther et al., 2012). This step greatly simplifies the analysis or control problem at hand. Assuming the linearisation point $\boldsymbol{\alpha}$ to be an equilibrium of the system, the linearisation of the equations of motion (4.1) yields

$$\bar{\mathbf{M}}\ddot{\boldsymbol{q}} + \bar{\mathbf{K}}\boldsymbol{q} = \bar{\mathbf{B}}\mathbf{u}\,, \tag{4.3}$$

where $\bar{\mathbf{M}} \in \mathbb{R}^{n\times n}$ is a mass matrix and $\bar{\mathbf{K}} \in \mathbb{R}^{n\times n}$ a stiffness matrix due to gravity. The actuator matrix $\bar{\mathbf{B}} \in \mathbb{R}^{n\times n_u}$ remains the same as that of equation (4.1).

Linearisation

Considering the left hand side of (4.1), the following nonlinear function is established

$$\boldsymbol{f}\left(\ddot{\boldsymbol{q}}, \dot{\boldsymbol{q}}, \boldsymbol{q}\right) = \mathcal{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathcal{C}(\dot{\boldsymbol{q}}, \boldsymbol{q})\dot{\boldsymbol{q}} + \mathcal{G}(\boldsymbol{q})\,. \tag{4.4}$$

It is possible to linearise the nonlinear function $\boldsymbol{f}\left(\ddot{\boldsymbol{q}}, \dot{\boldsymbol{q}}, \boldsymbol{q}\right)$ by using the Taylor series expansion method and neglecting the higher order terms that result from it. This linearisation is done around the operating point α, which is assumed to be an equilibrium where $\boldsymbol{q} = \boldsymbol{\alpha}$, $\dot{\boldsymbol{q}} = \dot{\boldsymbol{\alpha}} = \mathbf{0}$, and $\ddot{\boldsymbol{q}} = \ddot{\boldsymbol{\alpha}} = \mathbf{0}$.

$$\boldsymbol{f}\left(\ddot{\boldsymbol{q}}, \dot{\boldsymbol{q}}, \boldsymbol{q}\right) \simeq \boldsymbol{f}\left(\mathbf{0}, \mathbf{0}, \boldsymbol{\alpha}\right) + \left.\frac{\partial \boldsymbol{f}}{\partial \ddot{\boldsymbol{q}}}\right|_{\alpha} \left(\ddot{\boldsymbol{q}} - \ddot{\boldsymbol{\alpha}}\right) + \left.\frac{\partial \boldsymbol{f}}{\partial \dot{\boldsymbol{q}}}\right|_{\alpha} \left(\dot{\boldsymbol{q}} - \dot{\boldsymbol{\alpha}}\right) + \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}}\right|_{\alpha} \left(\boldsymbol{q} - \boldsymbol{\alpha}\right)\,. \tag{4.5}$$

Generally, only natural equilibrium points are considered. In other words, no control effort should be required to maintain the equilibrium position. By doing so, the first term of the series: $\boldsymbol{f}(\mathbf{0}, \mathbf{0}, \boldsymbol{\alpha})$, can be neglected, which for mechanical systems comes down to assuming $\mathcal{G}(\boldsymbol{\alpha}) = 0$. For an inverted pendulum with equal mass distribution, for example, the upright position is such an equilibrium. Also, define $\boldsymbol{q}_\alpha = \boldsymbol{q} - \boldsymbol{\alpha}$ as the generalised coordinate with respect to the equilibrium and consequently, $\ddot{\boldsymbol{q}}_\alpha = \ddot{\boldsymbol{q}} - \ddot{\boldsymbol{\alpha}}$, $\dot{\boldsymbol{q}}_\alpha = \dot{\boldsymbol{q}} - \dot{\boldsymbol{\alpha}}$. It is then found that

$$\begin{aligned} \boldsymbol{f}(\ddot{\boldsymbol{q}}, \dot{\boldsymbol{q}}, \boldsymbol{q}) &\simeq \mathcal{M}|_\alpha \, \ddot{\boldsymbol{q}}_\alpha + \left(\frac{\partial \mathcal{C}}{\partial \dot{\boldsymbol{q}}} \dot{\boldsymbol{q}} + \mathcal{C} \right)\Bigg|_\alpha \dot{\boldsymbol{q}}_\alpha + \left(\frac{\partial \mathcal{M}}{\partial \boldsymbol{q}} \ddot{\boldsymbol{q}} + \frac{\partial \mathcal{C}}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}} + \frac{\partial \mathcal{G}}{\partial \boldsymbol{q}} \right)\Bigg|_\alpha \boldsymbol{q}_\alpha \\ &= \mathcal{M}|_\alpha \, \ddot{\boldsymbol{q}}_\alpha + \left. \frac{\partial \mathcal{G}}{\partial \boldsymbol{q}} \right|_\alpha \boldsymbol{q}_\alpha \, . \end{aligned} \tag{4.6}$$

After evaluating the individual terms at $\boldsymbol{\alpha}$, most of the terms can be neglected. Note that the Coriolis matrix with $\dot{\boldsymbol{q}} = \mathbf{0}$ also equals zero, for all values of $\boldsymbol{q}$. Therefore, the linearisation of equation (4.1), including the right hand-side, is defined as

$$\bar{\mathbf{M}} \ddot{\boldsymbol{q}} + \bar{\mathbf{K}} \boldsymbol{q} = \bar{\mathbf{B}} \boldsymbol{u} \, , \tag{4.7}$$

with

$$\bar{\mathbf{M}} = \mathcal{M}|_\alpha \qquad \bar{\mathbf{K}} = \left. \frac{\partial \mathcal{G}}{\partial \boldsymbol{q}} \right|_\alpha \qquad \bar{\mathbf{B}} = \mathcal{B} \, ,$$

where the mass matrix $\bar{\mathbf{M}}$ is the original mass matrix at $\boldsymbol{\alpha}$ and where stiffness matrix $\bar{\mathbf{K}}$ is the Jacobian of the original gravity vector, also evaluated at $\boldsymbol{\alpha}$. The actuator matrix remains the same and, dropping the subscripts, the generalised coordinates $\boldsymbol{q}$ are now with respect to $\boldsymbol{\alpha}$.

4.4.2 State-space realisation

The linearised equations of motion can be expressed in state-space form as follows

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \, , \tag{4.8}$$

where the state vector is $\mathbf{x}(t) = [\boldsymbol{q} \, , \dot{\boldsymbol{q}}]^T$, the dimensions are $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$ and $\mathbf{B} \in \mathbb{R}^{n_x \times n_u}$, and the system matrices are defined as

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\bar{\mathbf{M}}^{-1}\bar{\mathbf{K}} & \mathbf{0} \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{M}}^{-1}\bar{\mathbf{B}} \end{bmatrix} . \tag{4.9}$$

The system in (4.9) has six states ($n_x = 6$) corresponding to joint positions and angular velocities, and three inputs ($n_u = 3$), one per joint. The control input based on (4.9)

is applied to the corresponding actuator in both legs, the reason for this is that the model assumes that TUlip is a three-segment inverted pendulum. In (4.9), the inputs are mapped according to $\bar{\mathbf{B}} = \mathbf{I}_{n_u \times n_u}$, and the corresponding values of $\bar{\mathbf{M}}$ and $\bar{\mathbf{K}}$ are

$$\bar{\mathbf{M}} = \begin{bmatrix} 6.0593 & 3.383 & 1.426 \\ 3.383 & 2.036 & 0.930 \\ 1.426 & 0.930 & 0.504 \end{bmatrix} \qquad \bar{\mathbf{K}} = \begin{bmatrix} -83.323 & -41.280 & -15.198 \\ -41.280 & -41.280 & -15.198 \\ -15.198 & -15.198 & -15.198 \end{bmatrix}. \tag{4.10}$$

The knee joints have restricted motion in TUlip, where negative angles are not feasible due to physical constraints. The knee can not rotate backwards, to prevent hyper-extension the same way as in humans. The implemented controllers were designed without any constraints in terms of the joint angles that could be achieved based on the control signals, for this reason, the knee joint was "locked" via software to restrict motions that could damage the joint, while allowing small rotations.

4.4.3 Nonlinear steady-state design

In section 3.2.1, a general procedure to introduce a setpoints is explained, which is a completely linear approach. In this case, our reference is always going to be the vertical equilibrium position, which in a linear inverted pendulum model would ensure that the system stays static in the absence of any perturbation. TUlip however, is under the effect of position offsets for the CoM of each link, which generate torques due to gravity. In order to counteract these effects, a different approach was taken to calculate the steady-state versions of both inputs and states. Based on the work from Featherstone (2008) on rigid body dynamics and spatial vectors, the inverse kinematics problem for a robotic platform can be solved by means of a recursive Newton-Euler algorithm, that considers the inertial properties of all bodies in the kinematic chain. This method was used to calculate the joint torques that yield zero acceleration when the system is in the upright position.

Using the system matrices in (4.9), the steady-state state $\boldsymbol{x}_{ss}$ and control input $\boldsymbol{u}_{ss}$ can be computed, given a desired system output $\boldsymbol{y}_{ss}$ and the fact that the rate of change of the state is zero. $\boldsymbol{y}_{ss} \in \mathbb{R}^{n_{ss} \times n_{ss}}$ is an identity matrix that corresponds to an equilibrium task space. The matrix $\mathbf{C}_{ss} \in \mathbb{R}^{n_{ss} \times n_x}$ maps the equilibrium state space $\boldsymbol{x}_{ss} \in \mathbb{R}^{n_x \times n_{ss}}$ to this task space. Note that $\mathbf{C}_{ss}$ is not necessarily equal to $\mathbf{C}$. The matrix $\boldsymbol{u}_{ss} \in \mathbb{R}^{n_u \times n_{ss}}$ is the control space. By defining a steady-state output matrix $\mathbf{C}_{ss}$, it is possible to obtain $\mathbf{X}_{ss}$ if the following inverse is computed

$$\mathbf{X}_{ss}(t) = \mathbf{C}_{ss}^{-1} \boldsymbol{y}_{ss}(t)\,, \tag{4.11}$$

then $\mathbf{X}_{ss}(t)$ is multiplied by the setpoint to obtain

$$\boldsymbol{x}_{ss}(t) = \mathbf{X}_{ss}(t)w(t) \,, \tag{4.12}$$

where $\boldsymbol{x}_{ss}(t)$ can be partitioned as

$$\boldsymbol{x}_{ss}(t) = \begin{bmatrix} \boldsymbol{q}_{ss} & \dot{\boldsymbol{q}}_{ss} \end{bmatrix}^T \,, \tag{4.13}$$

using $\boldsymbol{q}_{ss}$, $\dot{\boldsymbol{q}}_{ss}$, and $\ddot{\boldsymbol{q}}_{ss} = 0$, it is then possible to solve (4.1) for the steady-state joint torques $\boldsymbol{u}_{ss}$

$$\mathcal{M}(\boldsymbol{q}_{ss})\ddot{\boldsymbol{q}}_{ss} + \mathcal{C}(\dot{\boldsymbol{q}}_{ss}, \boldsymbol{q}_{ss})\dot{\boldsymbol{q}}_{ss} + \mathcal{G}(\boldsymbol{q}_{ss}) = \mathcal{B}\boldsymbol{u}_{ss} \,. \tag{4.14}$$

Once $\boldsymbol{u}_{ss}$ is obtained, a continuous control law can be written as follows

$$\begin{aligned} \mathbf{u}(t) &= -\mathbf{k}\left(\mathbf{x}(t) - \boldsymbol{x}_{ss}w(t)\right) + \boldsymbol{u}_{ss}w(t) \\ &= -\mathbf{k}\mathbf{x}_w(t) + \boldsymbol{u}_{ss}w(t) \,. \end{aligned} \tag{4.15}$$

To solve (4.14), the recursive Newton-Euler algorithm discussed previously can be used, as shown in (Featherstone, 2010b). This algorithm only needs joint positions, velocities, and accelerations to compute joint torques as well as a description of the robotic structure that includes the number of bodies in the mechanism, the connectivity between them, and an array of spatial inertias for each body. A description of the equations behind the algorithm can be found in Appendix (B).

4.4.4 Control architectures

The role of all the controllers (CC, ICc and ICe) is to provide high-level joint torque references $\tau_{j,ref}$ for the series elastic actuators. This overall structure is shown in Fig. 4.7, where the SEA stage is comprised by a torque controller, that takes as inputs the joint torque references generated by either CC or IC controllers, and the measured joint torque τ_j.

The output of the torque controller is the required motor torque τ_m (i.e. current) to achieve the desired configuration. The joint torque τ_j is proportional to the measured difference between the motor angle θ_m and joint angle θ_j.

The high-level controllers, continuous control (CC) and intermittent control (ICc and ICe), can be described by the block diagrams in Figs. 4.8 and 4.9 respectively. Both controllers are based on the architectures shown in chapter 3, where the main difference is the fact that there is no need of an observer to estimate angular velocities since they are already provided by the control layers of TUlip.

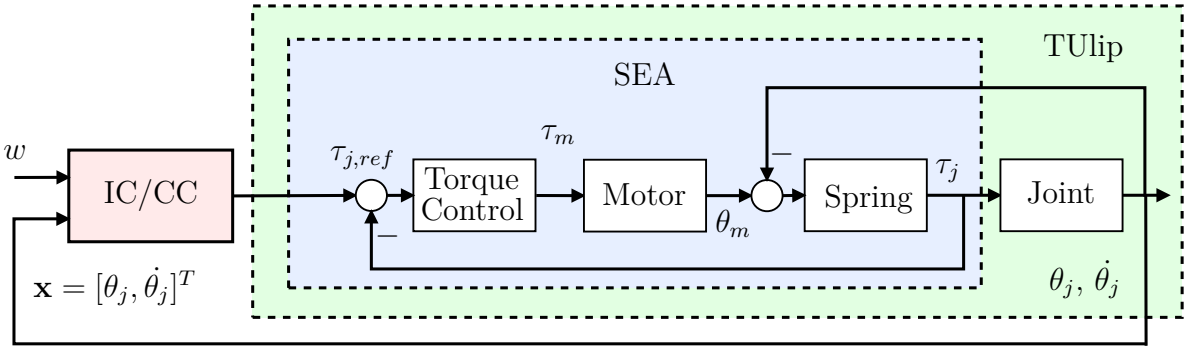


Figure 4.7: TUlip joint torque control scheme. IC/CC controllers generate reference torques $\tau_{j,ref}$ for the low level control stages in order to generate appropriate commands for the motors. Measured joint angles and velocities θ_j, $\dot{\theta}_j$ are used as feedback. The blue colour block represents the series elastic actuation principle applied to the joints. TUlip is represented by the green colour box as the system to be controlled.

It is worth mentioning that the SEA controllers are used by all the high-level controllers currently implemented in TUlip, without changing the parameters or default configuration. The fact that they are independently tuned for all applications allows flexibility in terms of testing any kind of high-level control that provides reference joint torques as an output.

Another important point is that the intermittent controller in Fig. 4.9 can accommodate both clock-driven and event-driven versions, by simply adjusting the triggering thresholds q.

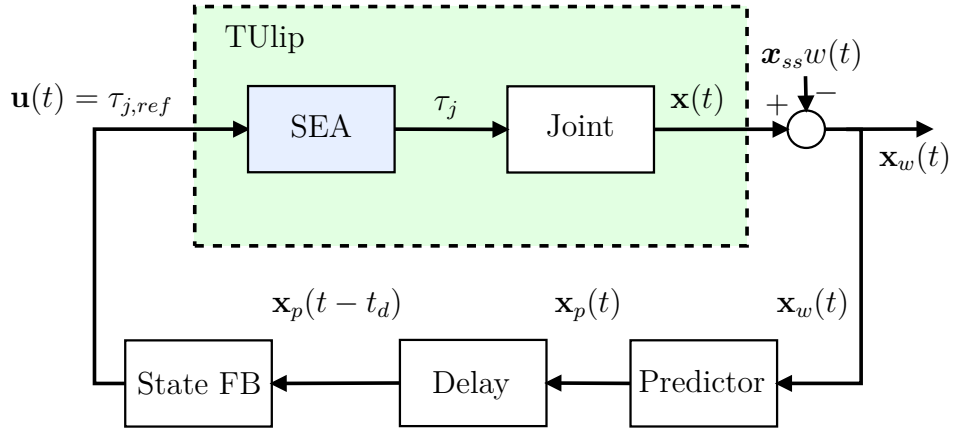


Figure 4.8: TUlip continuous controller (CC). Predicted states $\mathbf{x}_p$ are used to compensate for a deliberately introduced time delay. The output of the *State Feedback* block is a joint torque reference $\tau_{j,ref}$ which is used as high-level for input for TUlip. The signals in this controller flow continuously, as opposed to the IC case where the triggering mechanism samples the states based on events.

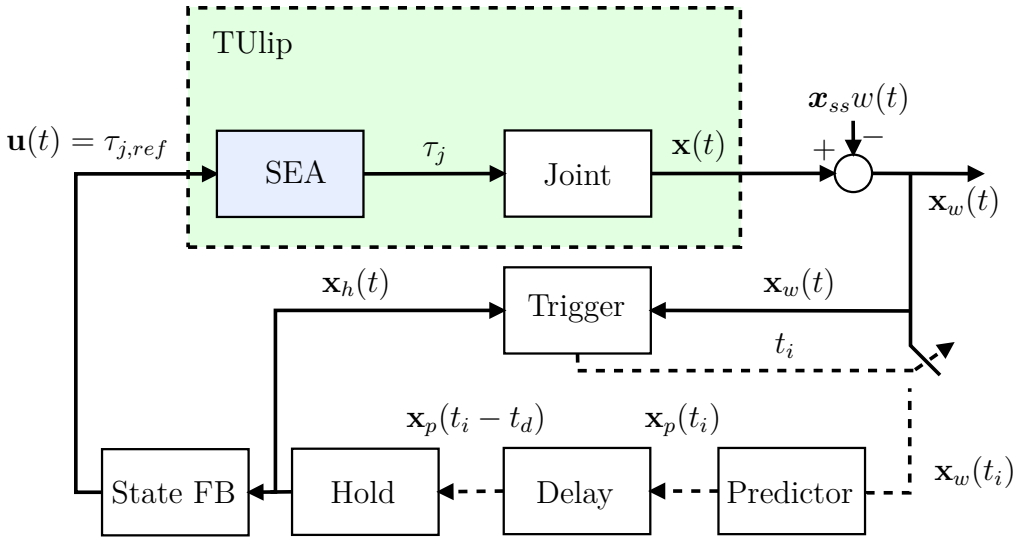


Figure 4.9: TUlip intermittent controller (ICc and ICe). Predicted states $\mathbf{x}_p$ are used to compensate for a deliberately introduced time delay. A trigger mechanism is included to generate events every t_i. Between events, the controller evolves in an open-loop configuration dictated by the system matched hold represented by the *Hold* block. The output of the *State Feedback* block is a joint torque reference $\tau_{j,ref}$.

4.4.5 Controller parameters

The continuous controller (CC) is based on state-feedback, where the gains are obtained using the LQR optimisation approach. This controller also uses a predictor to compensate for a time-delay. This continuous design is the base of the intermittent controller, which takes the same state-feedback approach and adds a triggering mechanism to generate events based on the difference of the hold states and the measured states. In this case, the system-matched hold (SMH) was used to generate the open-loop trajectories.

Each trial is divided in three groups as depicted in Fig. 4.6. Every group would use a controller designed from a different set of state-feedback gains $\mathbf{k}$, where these gains are obtained from optimising the following cost function in the LQR approach

$$J_{LQR} = \int_0^\infty \left[\mathbf{x}(t)^T \mathbf{Q}_c \mathbf{x}(t) + \mathbf{u}(t)^T \mathbf{R}_c \mathbf{u}(t)\right] dt \,. \tag{4.16}$$

From preliminary tests on the robot, the following value of $\mathbf{R}_c$ was found to give an acceptable performance in most situations.

$$\mathbf{R}_{c,nom} = \begin{bmatrix} 1.3 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1.2 \end{bmatrix} . \tag{4.17}$$

For this reason, the matrix in (4.17) was used as the basis to design the controllers of each group, by simply scaling it up or down according to a predefined percentage.

In Table 4.2, relevant parameters used for CC, ICc and ICe controllers are shown. These parameters did not change throughout the trial. The CC case is not affected by any triggering, therefore the minimum open-loop interval Δ_{min} and the threshold q do not apply, but they are relevant for both ICc and ICe. The triggering mechanism of ICe was designed to use all states of the state vector in (4.8) to generate events, based on the comparison of the prediction error defined in (3.62) and the threshold q.

Table 4.2: TUlip controller parameters used for all trials. t_d, Δ_{min}, q, correspond to the time delay, open-loop interval and threshold values. $\mathbf{Q}_c$ is the state weighting matrix from (4.16)

t_d [sec]	Δ_{min} [sec]	q [deg]	$\mathbf{Q}_c$ diag.
0.014	0.032	2	[1,1,1,0,0,0]

Table 4.3 shows the percentage of $\mathbf{R}_{c,nom}$ used for each group in all trials, and the controller sequence within a group. Trials 1 and 2 started the groups with CC, then changing to ICc and finishing with ICe. This is the sequence shown as an example in Fig. 4.6. Trials 3, 4, and 5 use a reversed sequence that starts with ICc, then it switches to ICe, and finishes with CC. A total of 45 perturbations were applied over the 5 trials.

Table 4.3: Percentage values of $\mathbf{R}_{c,nom}$ used for each trial.

Trial	Group sequence	Group 1 (%)	Group 2 (%)	Group 3 (%)
1	CC - ICc - ICe	75	100	125
2	CC - ICc - ICe	75	100	125
3	ICc - ICe - CC	125	100	75
4	ICc - ICe - CC	110	100	90
5	ICc - ICe - CC	105	100	95

The controllers of Group 2 always used the nominal value of $\mathbf{R}_{c,nom}$ shown in (4.17), while groups 1 and 3 used scaled versions of this matrix. The percentages shown in the group columns of Table 4.3 indicate the scaling of $\mathbf{R}_{c,nom}$ used for a particular group.

These percentages lead to a broad classification of the controller gains $\mathbf{k}$ that result from the LQR approach (explained in section 3.2.1, expression (3.11)), using matrices $\mathbf{R}_{c,nom}$ and $\mathbf{Q}_c$. The classification is as follows: in general, we know that if a large value of $\mathbf{R}_c$ is selected, this would penalise the control effort, reducing it in order to keep

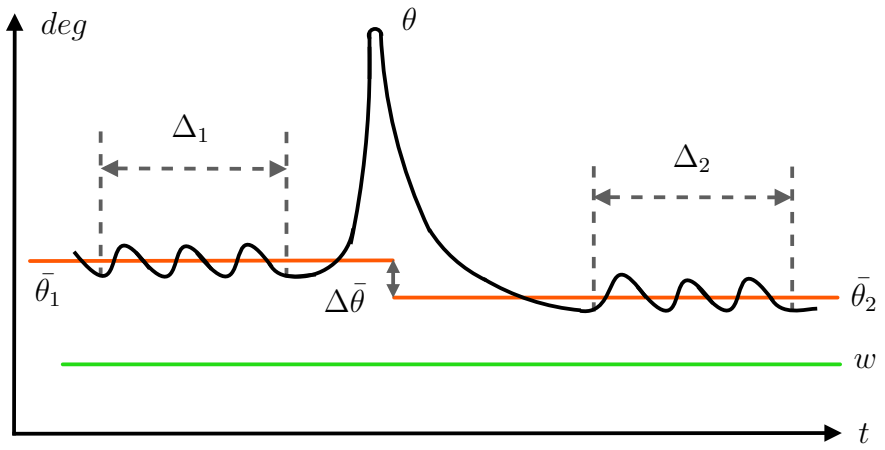


Figure 4.10: Diagram of a perturbation response for the joint angle θ. Δ_1 and Δ_2 are data portions of 8 seconds each, $\bar{\theta}_1$ and $\bar{\theta}_2$ are the corresponding mean values, $\Delta\bar{\theta}$ is the difference between the two means and w is the setpoint.

the optimisation cost in (4.16) small. This scenario produces controller gains $\mathbf{k}$ that are also *low*. A low value of $\mathbf{R}_c$ works in the opposite way, generating more control effort which is associated to a set of *high* gains. Based on this, the gains produced by $\mathbf{R}_{c,nom}$ which applies to group 2 in all trials, are regarded as the nominal gains (or **medium gains**). The groups that worked with $\mathbf{R}_c$ values that are lower than $\mathbf{R}_{c,nom}$ (percentages below 100% in Table 4.3) are considered as **high gain** cases. Conversely, the groups with $\mathbf{R}_c$ values above 100% are **low gain** cases. This classification was used to perform the statistical analysis of the results for all trials, shown in section 4.6.3.

4.5 Analysis methods

The overall results were analysed from a statistical point of view for all of the 45 perturbations applied to the robot. To do this, three different quantities were used to capture the performance of each controller. In Fig. 4.10, a typical joint angle response is presented with the purpose of showing what portions of θ were considered for the analysis. Δ_1 and Δ_2 were defined as a time span of 8 seconds before and after each perturbation, Δ_1 started 8 seconds before the perturbation was triggered and Δ_2 started 4 seconds after the perturbation trigger, allowing the joint angle to return to a steady-state value. Based on this considerations, the following quantities were used to perform the analysis:

- **Joint angle shift:** the joint angle shift is shown in Fig. 4.10 as $\Delta\bar{\theta}$. This value provides information on how capable the controllers were in terms of bringing the joint angles back to the steady-state after the perturbation, by subtracting the mean joint angles in Δ_1 and Δ_2. Defining m as the number of data points

in Δ_1 and Δ_2, and the initial and final data points in each range as Δ_0 and Δ_f respectively, the expression for the joint angle shift is obtained as follows

$$\bar{\theta} = \frac{1}{m} \sum_{i=\Delta_0}^{\Delta_f} \theta_i \tag{4.18}$$

$$\Delta\bar{\theta} = \bar{\theta}_2 - \bar{\theta}_1 \,, \tag{4.19}$$

where $\bar{\theta}_1$ and $\bar{\theta}_2$ correspond to the mean joint angles in Δ_1 and Δ_2.

- **Mean steady-state error:** considering the data points in $\Delta = \Delta_1 + \Delta_2$, and m as the number of data points in this range, the definition for the mean steady-state error is established as

$$e_w = \frac{1}{m} \sum_{i=\Delta_0}^{\Delta_f} \mid \theta_i - w \mid , \tag{4.20}$$

where w is the predefined angle setpoint for the joint.

- **Steady-state error variance:** similarly, the steady-state error variance $\mathrm{Var}(e_w)$ over Δ is defined as

$$\mathrm{Var}(e_w) = \frac{1}{m-1} \sum_{i=\Delta_0}^{\Delta_f} (\mid \theta_i - w \mid - e_w)^2 \,, \tag{4.21}$$

where e_w is the mean steady-state error.

4.6 Results

This section introduces the results of the balance test in two ways: 1) the results of a representative trail are presented first (trial 1 in Table 4.3), showing the evolution of the control inputs (reference joint torques), joint angle, and open-loop intervals for both the ankle and hip joints. 2) the grouped results from all trials are shown based on the analysis methods explained in the previous section. The results for the knee are not included since this joint was restricted to have limited movement via software. The following section introduces the results for the ankle joint during trial 1.

4.6.1 Ankle data

Figs. 4.11, 4.12 and 4.13 correspond to ankle data for groups 1, 2, and 3 respectively. In each of these figures, the results for the three controllers are presented (every column

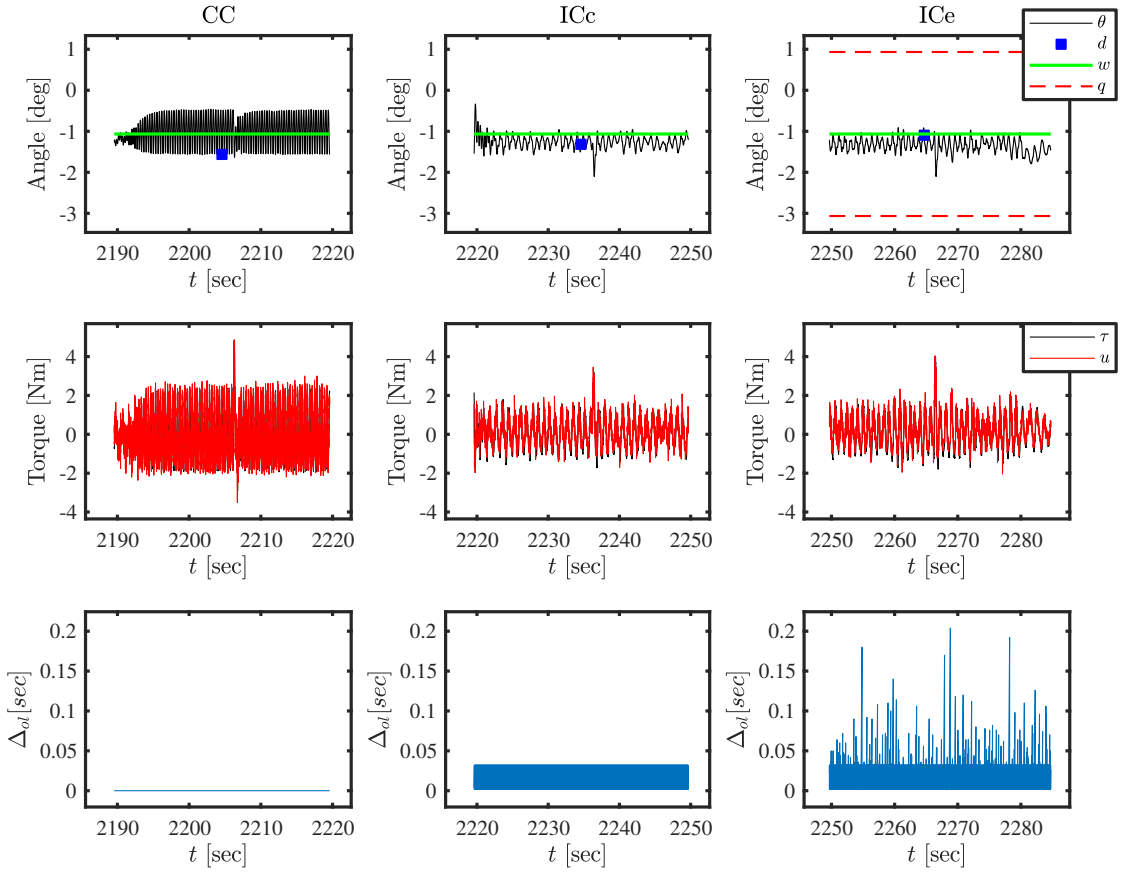


Figure 4.11: Trial 1 - Group 1 - Ankle data. Each column represents a single controller, in the corresponding group sequence (CC-ICc-ICe). First row shows joint angle θ, setpoint w, threshold q, and the perturbation trigger d. Second row shows the reference joint torque u and the measured joint torque τ. The third row corresponds to the open-loop interval Δ_{ol}.

represents a different controller i.e., CC, ICc, or ICe), where the first row shows the joint angle θ, the setpoint defined for that trial w, the moment in time when the pull was triggered d, and the threshold value q that applies only to the ICe case. The second row reveals the control input history during the trial, where $\mathbf{u}$ corresponds to the joint torque references $\tau_{j,ref}$ and τ is the measured joint torque. Finally, the third row shows the open-loop intervals Δ_{ol}, which are relevant only for both intermittent controllers.

For this particular trial, the resulting gains obtained by the LQR design in group 1 are higher than those from group 3, therefore Fig. 4.11 shows the results of all controllers using **high gains** for the ankle joint. It is possible to see that θ oscillates at a higher frequency for CC case compared to both intermittent controllers, which exhibit a quasi-regular sway close to the setpoint value.

A similar trend is observed for the control input for all controllers. The intermittent interval Δ_{ol} in the ICe case raises above the minimum value of 0.032 sec, reaching 0.1

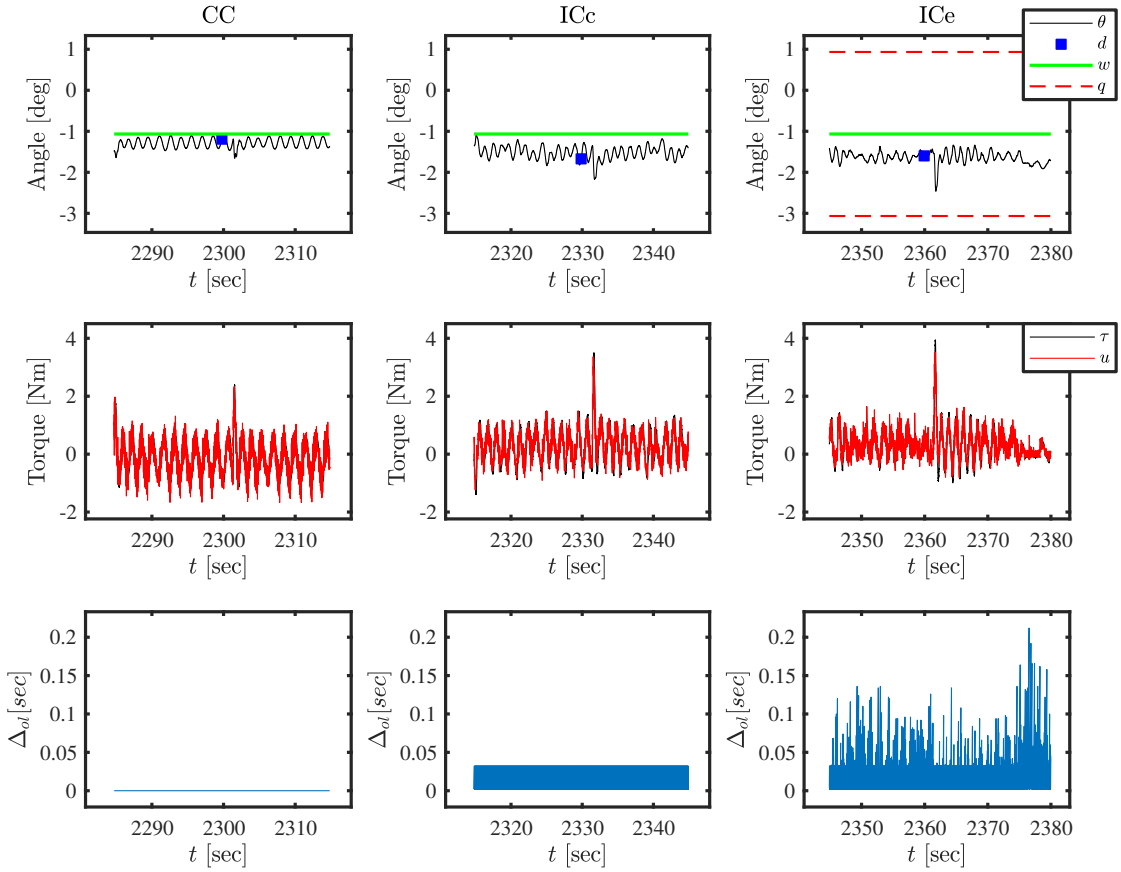


Figure 4.12: Trial 1 - Group 2 - Ankle data.

sec several times and 0.2 sec as a maximum. The ICc triggers periodically every 0.032 sec.

The controllers in group 2 were designed with the nominal value of $\mathbf{R}_c$ (**medium gains**), as shown in Table 4.3. Fig. 4.12 shows the results of group 2 for the ankle joint; for this condition, the continuous controller does not exhibit high frequency oscillations anymore in terms of θ, instead it shows a similar sway pattern compared to those generated by both intermittent controllers.

The control input generated as a consequence of the perturbation in CC is smaller in amplitude compared to ICc and ICe. The intermittent interval for the ICe case still shows some instances where it reaches 0.2 sec, while getting to the 0.05 to 0.1 sec region most of the time.

In group 3, a **low** set of gains (125% of $\mathbf{R}_{c,nom}$) was used to design the controllers. For this condition, the joint angle θ in both intermittent controllers does not return to the state they were in before the perturbation takes place, while CC stays closer to the setpoint w the entire time compared to ICc and ICe. This drift from the reference

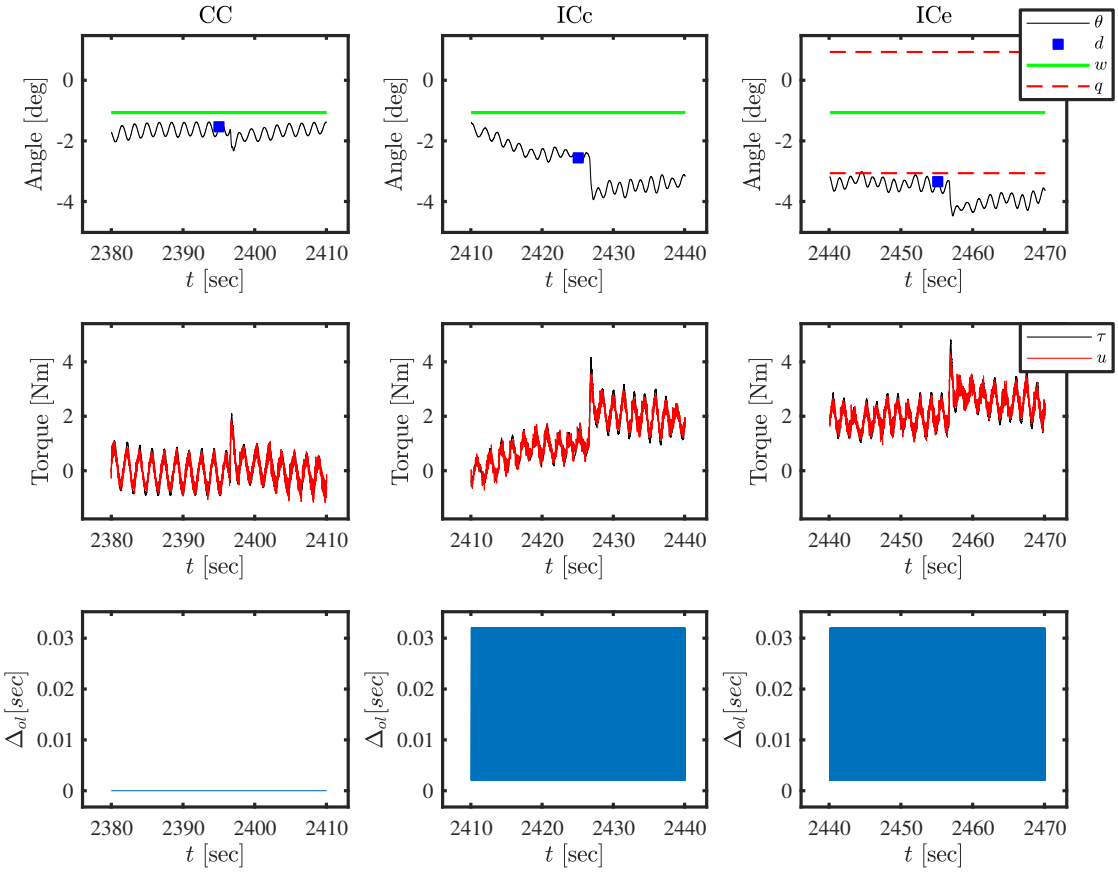


Figure 4.13: Trial 1 - Group 3 - Ankle data.

created by the perturbation generates events at the maximum possible rate, forcing a clock-driven behaviour for ICe, for this reason Δ_{ol} stays fixed at 0.032 sec.

4.6.2 Hip data

The hip joint is particularly difficult to control since it is the heaviest link of the structure; the chest area is where the motherboard and all the amplifiers are located, as well as other electronic components. The presence of these components in the trunk and their particular locations further enhance the effects of the CoM offsets shown in Table 4.1. Figs. 4.14, 4.15, and 4.16 correspond to hip data for groups 1, 2, and 3 respectively.

A very similar pattern compared to the ankle data is observed for this joint, where the **high gain** condition (group 1) results in high frequency oscillations of θ, the nominal gains (group 2) generate acceptable results for all controllers, swaying regularly and rejecting the pull from the actuator, and finally the low gain condition (group 3) shows how the intermittent controllers have difficulties bringing back θ to the values observed before the perturbation, however they are able to maintain a stable upright

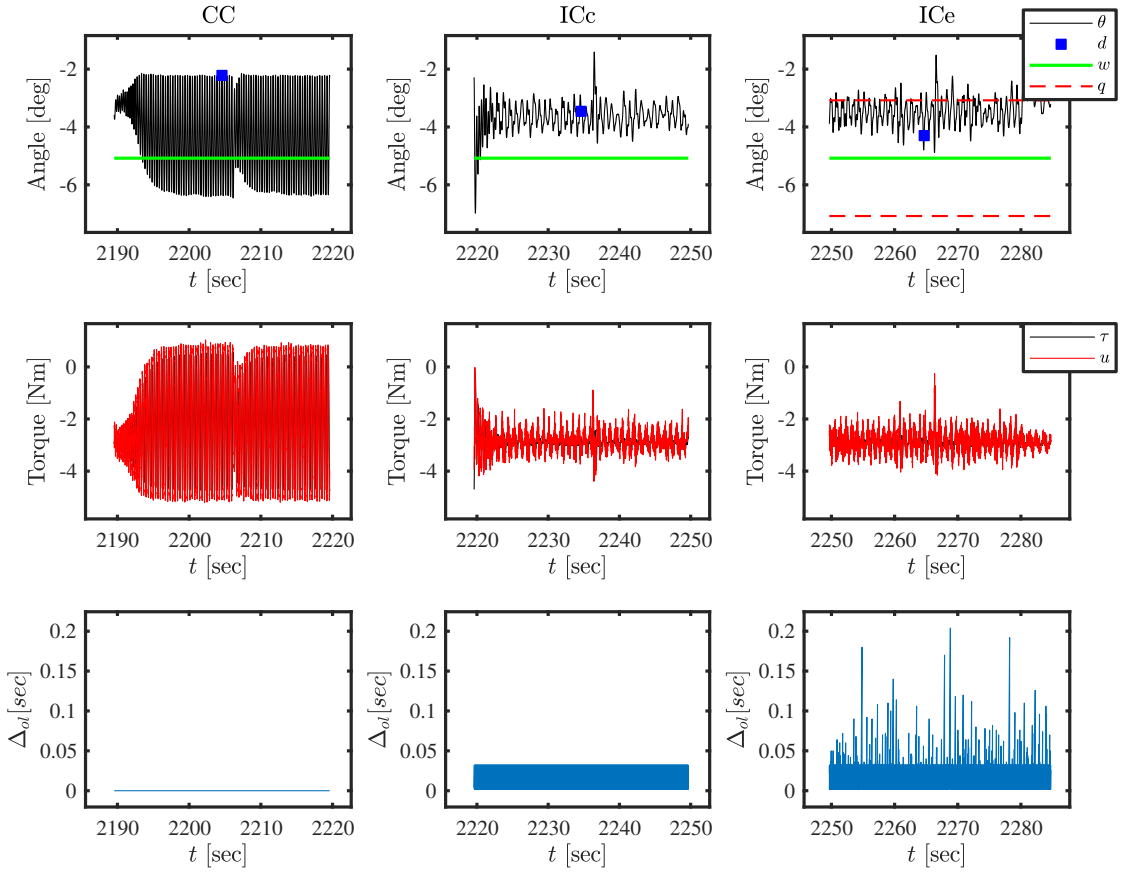


Figure 4.14: Trial 1 - Group 1 - Hip data. Each column represents a single controller, in the corresponding group sequence (CC-ICc-ICe). First row shows joint angle θ, setpoint w, threshold q, and the perturbation trigger d. Second row shows the reference joint torque u and the measured joint torque τ. The third row corresponds to the open-loop interval Δ_{ol}.

configuration.

The aforementioned high frequency oscillations were observed also in trial 2 (shown in Appendix (A), Figs. A.1,A.4), which shared the same design parameters and controller sequence as in trial 1. The control inputs also behaved in a similar way compared to the ankle joint. In reaction to the perturbation, CC showed reference torques of less amplitude compared to those of ICc and ICe. This can be seen in Figs. 4.14 and 4.15.

Overall, the best performance for this particular trial was achieved with the controllers of group 2, rejecting the applied perturbations while swaying quietly around a steady-state value. It can be seen that the response generated by each controller induced steady-state errors (differences between the equilibrium states and the setpoints) for all cases since there is no explicit integral action in them. A particularly interesting result is that the open-loop intervals raised up to 0.2 sec during some instances when the ICe controller was operational, without introducing any negative effects in terms

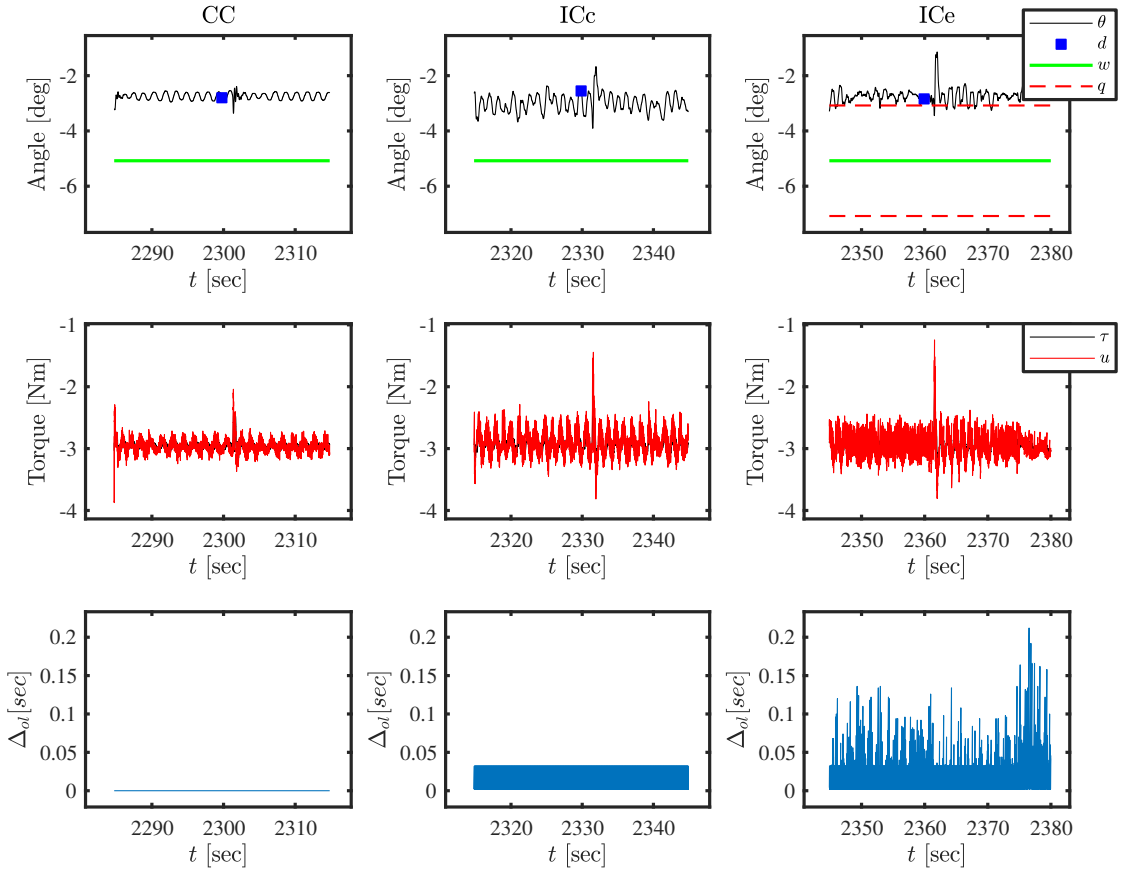


Figure 4.15: Trial 1 - Group 2 - Hip data.

of the amplitudes of joint angles and reference torques.

4.6.3 Grouped results

The grouped results, which consider all the trials in Table 4.3) were analysed using the classification explained in 4.4.5. The data corresponding to cases with high gains were grouped together, the same was done for the medium and low cases. These results are based on the quantities introduced previously in 4.5 and are shown in Figs. 4.17, 4.18, and 4.19, where data for both the ankle (a) and hip (b) joints are included.

First, the results for the **joint angle shift** are presented in Fig. 4.17. Both joints show similar results for the high and medium gain conditions, small joint shifts in terms of degrees for all controllers, where the CC and ICe controllers exhibit slightly less shift compared to that of ICc. Alternatively, the data spread for the low gain condition is higher, as well as the joint shift values.

The **mean steady-state error** results are shown in Fig. 4.18. The ankle joint follows a very similar trend compared to the joint angle shift data in Fig. 4.17, in which the

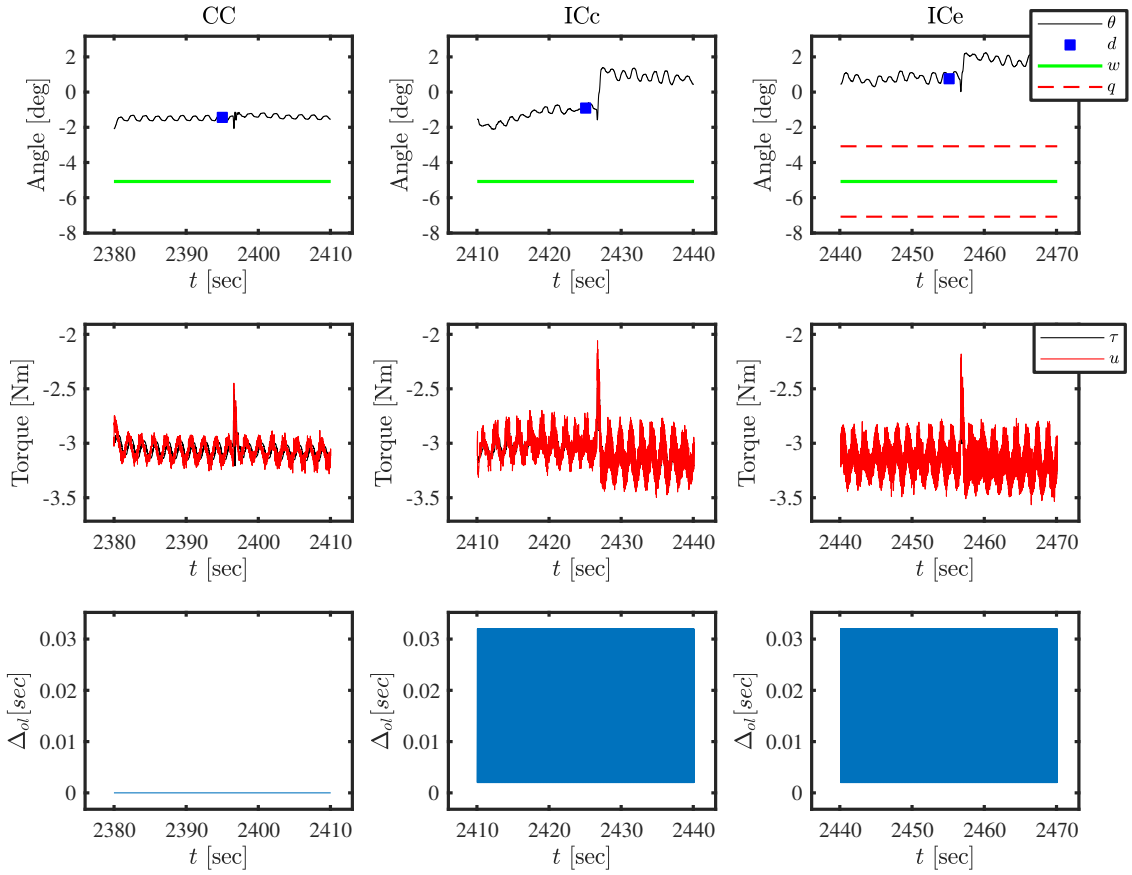


Figure 4.16: Trial 1 - Group 3 - Hip data.

high and medium gain conditions show small mean steady-state errors in contrast with the low gain condition. In this case, CC shows smaller values compared to ICc and ICe, specially for the ankle joint.

Interestingly, the results change for the hip joint. It can be seen that in the high gain condition, the spread of the data is less for the intermittent controllers. This is also the case for the medium gain condition, with the difference that both ICc and ICe controllers registered errors of a smaller magnitude overall compared to CC.

Finally, in Fig. 4.19 the data for the **steady-state error variance** are presented. It is possible to see that the high gain condition shows higher variances overall for both ankle and hip joints; this is a consequence of the high frequency oscillations observed in Figs. 4.11 and 4.14. The medium and low gain conditions registered lower variance values for the two joints; with CC generating values with less spread compared to the two intermittent controllers.

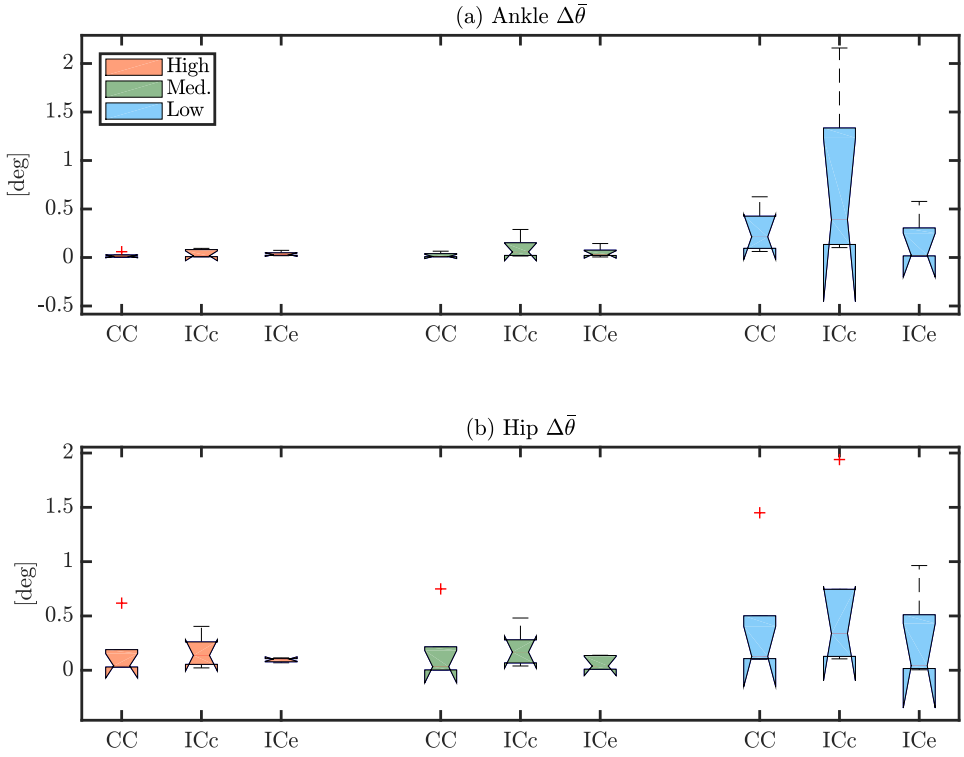


Figure 4.17: Joint angle shift $\Delta\bar{\theta}$ for ankle (a) and hip (b) joints across trials. Results for all perturbations grouped as high, medium, and low gain cases. The box centres, notches, edges, whiskers and crosses show median, confidence in median, inter-quartile limits, range and outliers, respectively.

4.7 Discussion

A balancing experiment using a humanoid robot, in the presence of perturbations, was carried out using multi-input, multi-output continuous and intermittent controllers to maintain stability. The design of all controllers was based on LQR methods, and the intermittent ones used a system-matched hold to generate open-loop trajectories as shown in (Gawthrop et al., 2015). This design consideration allowed us to manipulate the performance of the controllers while retaining control over the amplitude of the control input, in order to keep it within safe limits. In addition to this, it was possible to evaluate the controllers over a range of design values that had a direct influence on the control signal.

The results of the experiment show that it is possible to use intermittent controllers for quiet balancing of a complex multi-segmental robotic structure, resulting in similar performance in terms of stability when compared to a continuous controller designed with the same LQR parameters. This is interesting from the perspective of the computa-

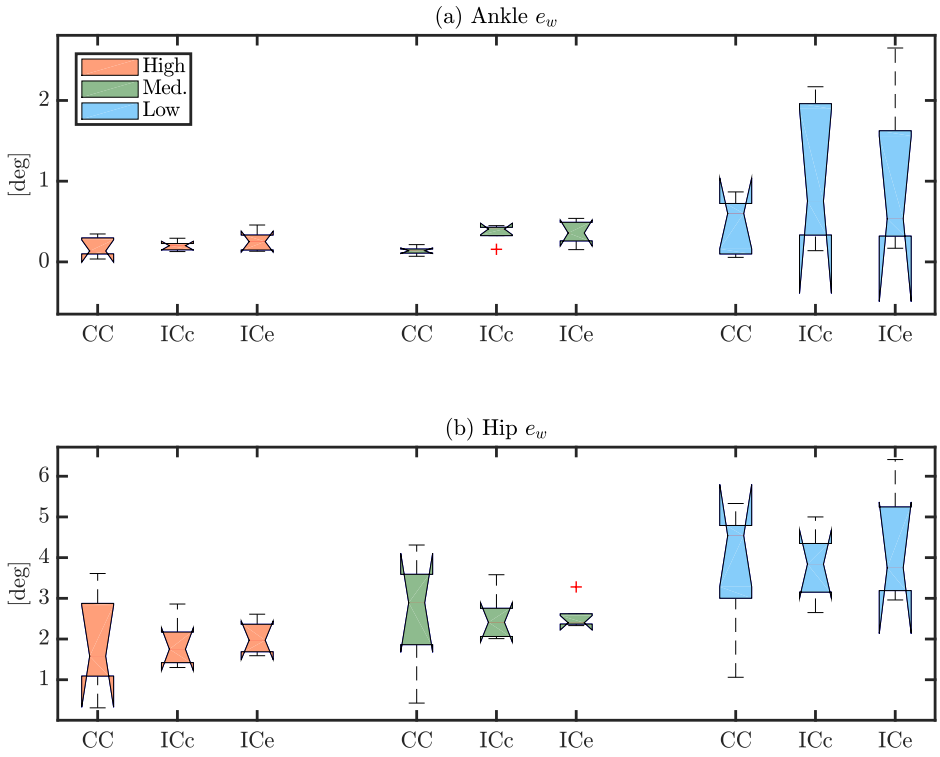


Figure 4.18: Mean steady-state error e_w for ankle (a) and hip (b) joints across trials. Results for all perturbations grouped as high, medium, and low gain cases. The box centres, notches, edges, whiskers and crosses show median, confidence in median, interquartile limits, range and outliers, respectively.

tional resources used to generate an appropriate control action. While these advantages were not directly measured from the experiment, it is possible to argue that the intermittent open-loop intervals observed in the event driven cases (Fig. 4.11) could allow roboticists to implement additional optimisation routines to deal with constraints during more complicated tasks such as walking. This adds flexibility to the design, since other computationally expensive procedures can be computed efficiently within the minimum open-loop interval as opposed to the idea of completing calculations before the next iteration of the real-time loop. Modern robotics is probably one of the fastest evolving fields in engineering and the use of the latest technology in robots is vital to achieve the best performance; however, there are still many robotic structures that have been operating for several years now that could benefit from these ideas. As hardware becomes older, intermittent controllers might provide a consistent paradigm that could enhance the performance of robots that are limited by less powerful processors.

It was expected during the experiment to have steady-state errors to the setpoint of each joint due to the fact that the controllers were implemented without any integral action.

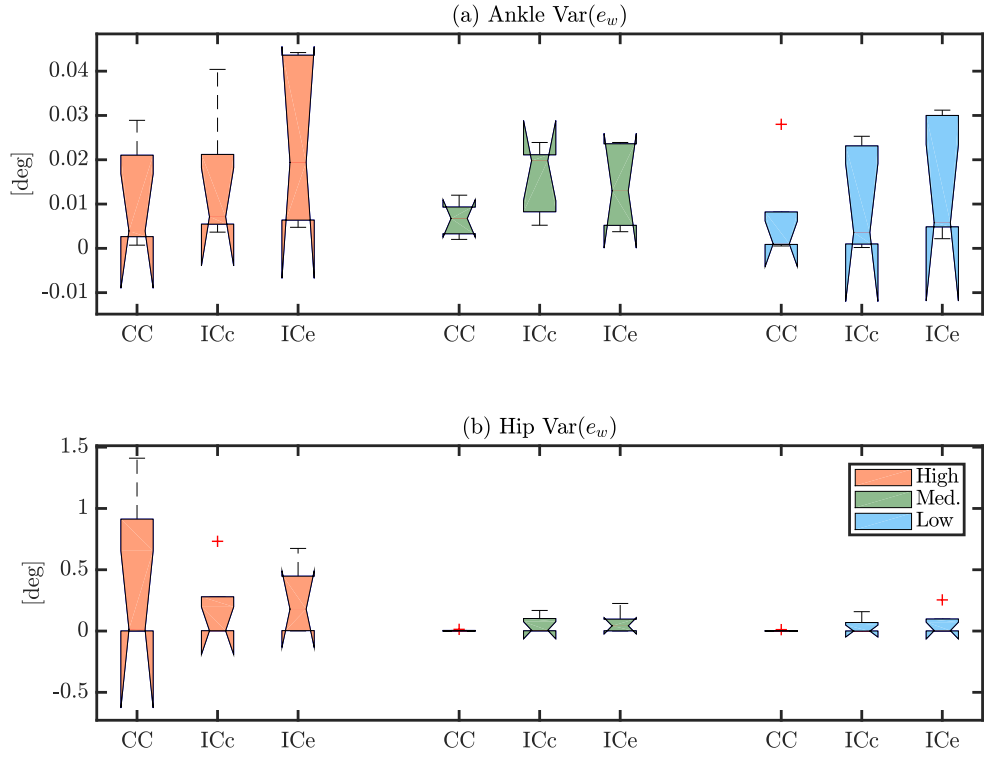


Figure 4.19: Steady-state error variance Var(e_w) for ankle (a) and hip (b) joints across trials. Results for all perturbations grouped as high, medium, and low gain cases. The box centres, notches, edges, whiskers and crosses show median, confidence in median, inter-quartile limits, range and outliers, respectively.

Furthermore, these errors are probably enhanced by the fact that the methods to design the controllers are based on a linear model, which yields dynamics that might differ considerably compared to the true dynamics. In other words, un-modelled dynamics contribute negatively, reducing stability margins. This situation is certainly present in TUlip; for instance, there is friction in every joint and backlash in many mechanical components, also the state-space model described in (4.9) does not contain an explicit description of the SEA joints, which neglects the delays introduced by the springs. With this in mind, the output responses observed from the high gain conditions in trails 1 and 2 (Figs. 4.14, A.4), seem to be related to un-modelled dynamics, where the high frequency oscillations in the continuous control cases arise due to reduced stability margins. The intermittent controllers seems to be able to cope with this mismatch in a better way, since they exhibit regular sway around a steady-state value instead of aggressive oscillations.

In situations where the model is an accurate representation of the plant (i.e., no considerable mismatch), the continuous controller should provide higher frequency band-

width and increased stability margins while sacrificing flexibility, whereas the intermittent controller stays flexible at reduced frequency bandwidth (Gawthrop et al., 2015). However, during the high gain conditions of the experiment, the modelling errors effects were amplified for the continuous controllers. Obtaining accurate models of complex robotic mechanisms is not a simple task; furthermore, wear and deterioration due to environmental conditions might contribute to modelling errors and reduced performance if there is no adaptation mechanism to properly identify and compensate against these changing parameters in real-time. This is another possible situation where robotics can benefit from the use of intermittent controllers since a more robust solution is achieved when uncertainty in the parameters is present.

The results across trials from Figs. 4.17, 4.18 and 4.19, show interesting behaviour for particular situations. While the CC cases exhibit slightly superior performance for *joint angle shift* and *mean steady-state error* measurements for the ankle joint, the intermittent controller performs better when controlling the hip joint, this is particularly evident in Fig. 4.18 specially for the medium and low gain conditions, where the ICe case recorded smaller steady-state errors compared to CC and ICc. This is related to the fact that the control inputs for the intermittent controllers had larger amplitudes compared to CC, applying more energy to the system which in turn helps generating the minimum amount of torque needed to maintain or start motion, overcoming stiction. These larger amplitudes are explained in terms of the triggering mechanism of intermittent control, where an update to the open-loop trajectories determined by the hold only happens after the minimum open-loop period elapses (ICc case) or when an event is generated (ICe case). In case of a sudden perturbation, the controller responds until one of these conditions is met, whereas the continuous controller does not wait in order to compensate against the perturbation.

The controllers used in this experiment were unconstrained, meaning that the design procedure did not consider saturation limits in order to avoid the computation of unsafe control signals. The robot has safety mechanisms in place though, that prevent possible damage of the actuators and amplifiers, but it must be said that a constrained version of these controllers is needed for more complicated experiments or tasks. Normally, push recovery experiments in robotics measure the performance of the controllers in terms of the maximum allowable perturbation while keeping constant the controller parameters. For this experiment, the perturbation was kept constant while changing the controller structures and the gains, evaluating responses for each of them individually. The fact that the controllers were unconstrained had an impact in the strength of the applied perturbations. The perturbations that were applied on TUlip generated control signals that stayed within a safe operating regime in terms of the actuator saturation limits.

Large perturbations were not applied to avoid operating close to these limits, specially since no spare parts were available. Further tests should be carried out to explicitly explore the maximum perturbation values that each of these controllers would tolerate before an imminent fall, based on a constrained controller design.

Chapter 5

Adaptive intermittent control of a rotational pendulum

5.1 Introduction

In this chapter, the two adaptive intermittent controllers that were presented previously are tested in simulation and experimentally. The testing platform is an under-actuated, single input - multiple output (SIMO) rotational pendulum, which means that it is a system that has more degrees of freedom than available actuators. Under-actuated systems in general exhibit interesting properties and can be found in many robotic structures, aircraft and underwater vehicles (Olfati-Saber, 2001), and they are considered particularly challenging from a control point of view. The rotational pendulum is also known in the literature as the Furuta inverted pendulum (Furuta et al., 1992) and it consists of a rotating arm that is controlled by a torque generated with a motor; the arm is coupled with the motor at one end, the other end of the arm is connected to a pendulum, which rotates freely.

The rotational pendulum has been used extensively to test linear and nonlinear controllers, focusing on the swing-up phase first, which makes the arm oscillate until the pendulum "stands up" and reaches a position that is close enough to the vertical (Gordillo et al., 2003; Åström et al., 2008). Then, a second controller takes over to apply fine control actions to keep the pendulum balanced (Ibáñez and Azuela, 2007; Ramírez-Neria et al., 2014). One possible way to deal with uncertainties in the parameters of the rotational pendulum is to propose a robust controller that withstands perturbations (Rigatos et al., 2017); however, an adaptive controller can also identify the plant in real-time and correct its control policy (Chen and Huang, 2014). In this context, an adaptive intermittent controller can be formulated to deal with models of

the pendulum that contain parameters that are incorrect, or that are even changing according to time, in order to stabilise the pendulum at the unstable vertical equilibrium point.

Two different adaptation scenarios are considered in this chapter: 1) a time-varying case is evaluated in simulation, where a defined set of parameters from the rotational pendulum are modified at specific times forcing the controllers to track the variations in order to adjust the control law. 2) an uncertain parameter case is evaluated experimentally, where the initial controllers are designed using an incorrect model, half way during the experiment the redesign stage of the adaptive controllers is enabled in order to reduce the errors caused by the model mismatch.

This chapter is organised as follows: first, a description of the rotational pendulum system is given, this includes an overview of the physical pendulum used for the real-time experiments, the mathematical model of the rotational pendulum and the augmented model needed to formulate state and parameter estimators. Then, the results of the simulation study are presented and followed by the experimental results obtained from the real-time system. The chapter ends with an overall discussion.

5.2 Rotational pendulum

The rotational pendulum system is composed of two elements: a rotating arm of length L_r mounted on a base that contains a motor, and a pendulum of length L_p; both of these elements can rotate 360 degrees in their respective planes. The basic control idea behind this system is to find a controller that would stabilise the pendulum element around the unstable equilibrium point by generating a torque τ that moves the arm. In Fig. 5.1, a diagram of a generic rotational pendulum is shown, including a picture of the physical pendulum used in the real-time experiments reported in this chapter. The equipment consists of a Quanser Consulting SRV-02 servo motor module coupled with a ROTPEN-E module that serves as the rotational arm.

The two outputs of the system are the pendulum angle α and the arm angle θ. Normally, the controller in place should keep both outputs at zero degrees while being able to reject perturbations. It is also possible to keep α close to zero degrees while forcing θ to follow a predefined reference trajectory.

For the real-time experiments, the two outputs were measured by two incremental US Digital optical encoders, with a resolution of 4096 counts per revolution. The position of the arm was controlled through the input provided by a Faulhaber DC

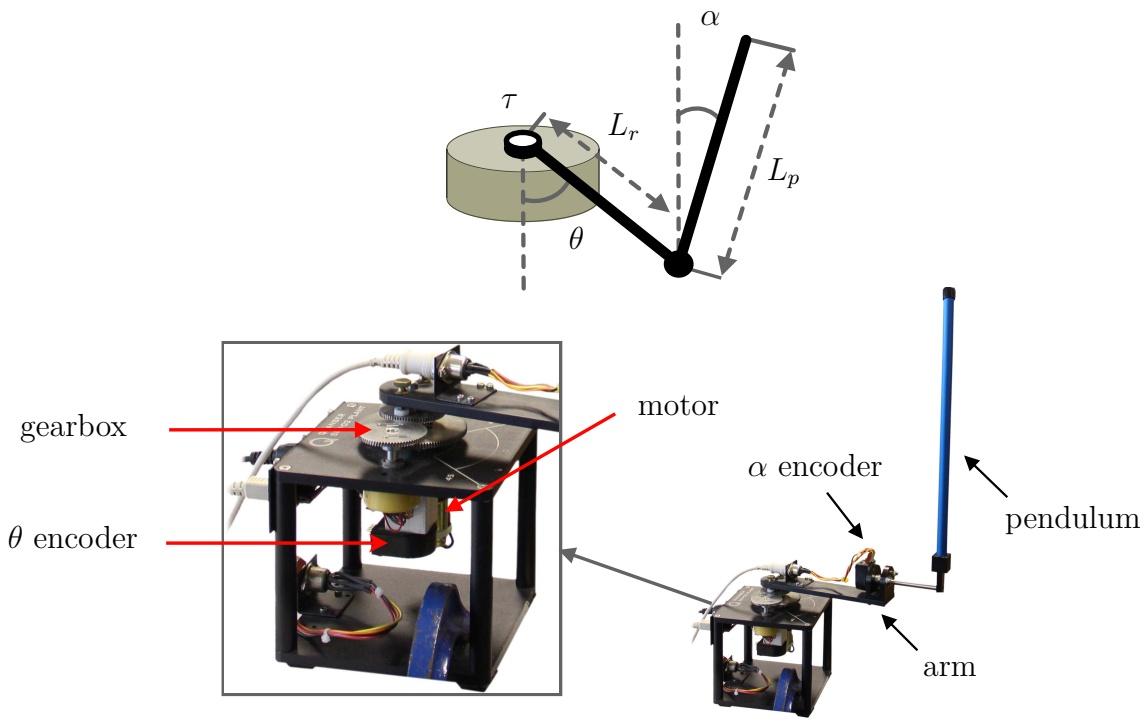


Figure 5.1: Diagram of the rotational pendulum and the Quanser Consulting SRV02 ROTPEN system. A rotating arm of length L_r is attached to the base. At this point, the torque τ is applied, making the arm rotate on the horizontal plane. The other end of the arm is connected to a pendulum of length L_p that rotates freely. The pendulum angle α and the arm angle θ are measured by incremental encoders. An angle of $\alpha = 0$ deg means that the pendulum is positioned at the vertical unstable point. A motor drives the rotating arm which is coupled to the pendulum through a planetary gearbox.

motor (2338S006) equipped with an internal gearbox. The motor shaft drives the external planetary gearbox that rotates the arm. A power amplification stage is used to provide the correct levels of voltage to the motor, which is part of the universal power module provided by Quanser (UPM-15-03).

The data interface between the computer and the rotational pendulum was implemented using a data acquisition card from National Instruments (PCI-6024E). In order to run the experiments, discrete versions of all controllers were implemented using MATLAB/Simulink (MathWorks, Inc.) and the Real-Time windows target, based on a zero-order hold approximation with a sample interval of 1 ms.

The following section introduces the dynamical model of the rotational pendulum including the linear state-space formulation that was used to implement the controllers.

5.2.1 Dynamical model

The nonlinear equations of the rotational pendulum can be written in a compact way as follows

$$\mathcal{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathcal{C}(\dot{\boldsymbol{q}}, \boldsymbol{q})\dot{\boldsymbol{q}} + \mathcal{G}(\boldsymbol{q}) = \mathcal{T}\,, \tag{5.1}$$

where, $\mathcal{M}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $\mathcal{C}(\dot{\boldsymbol{q}}, \boldsymbol{q}) \in \mathbb{R}^{n\times n}$ the Coriolis matrix, and $\mathcal{G}(\boldsymbol{q}) \in \mathbb{R}^n$ the gravity effects. The vector $\mathcal{T}$ corresponds to the torques generated by the actuator and $\boldsymbol{q} = \begin{bmatrix} \theta & \alpha \end{bmatrix}^T$ is the vector of joint positions. Equation (5.1) is the result of applying Euler-Lagrange equations of motion to the rotational pendulum system (Fantoni and Lozano, 2002), which results in the following matrices

$$\mathcal{M}(\boldsymbol{q}) = \begin{bmatrix} m_p L_r^2 + m_p l_p^2 - m_p l_p^2 \cos(\alpha)^2 + J_r & -m_p l_p L_r \cos(\alpha) \\ -m_p l_p L_r \cos(\alpha) & J_p + m_p l_p^2 \end{bmatrix} \tag{5.2}$$

$$\mathcal{C}(\dot{\boldsymbol{q}}, \boldsymbol{q}) = \begin{bmatrix} \left(2m_p l_p^2 \sin(\alpha)\cos(\alpha)\right)\dot{\alpha} & \left(m_p l_p L_r \sin(\alpha)\right)\dot{\alpha} \\ -\left(m_p l_p^2 \cos(\alpha)\sin(\alpha)\right)\dot{\theta} & 0 \end{bmatrix} \tag{5.3}$$

$$\mathcal{G}(\boldsymbol{q}) = \begin{bmatrix} 0 \\ -m_p l_p g \sin(\alpha) \end{bmatrix} \qquad \text{and} \qquad \mathcal{T} = \begin{bmatrix} \tau - D_r\dot{\theta} \\ -D_p\dot{\alpha} \end{bmatrix}, \tag{5.4}$$

where L_r and J_r are the length and moment of inertia of the rotational arm, L_p is the total length of the pendulum with its centre of mass located at $l_p = L_P/2$, a moment of inertia represented by J_p and mass m_p. The viscous damping coefficients for the arm and the pendulum are D_r and D_p respectively. The acceleration due to gravity is depicted as g.

Solving for the acceleration terms and defining $\boldsymbol{z} = \begin{bmatrix} \ddot{\theta} & \ddot{\alpha} \end{bmatrix}^T$, the following state variable representation is obtained

$$\begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \\ \boldsymbol{z} \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \\ \mathcal{M}(\boldsymbol{q})^{-1}\left(\mathcal{T} - \mathcal{C}(\dot{\boldsymbol{q}}, \boldsymbol{q})\dot{\boldsymbol{q}} - \mathcal{G}(\boldsymbol{q})\right) \end{bmatrix}. \tag{5.5}$$

The expression in (5.5) is nonlinear; however, a linear state-space model is needed for the design of adaptive intermittent controllers. Linearisation around the vertical equilibrium point ($\alpha = 0$ deg) yields the following equations for the acceleration terms

$$\ddot{\theta} = \frac{1}{J_T}\left[-\left(J_p + m_p l_p^2\right) D_r\dot{\theta} + m_p l_p L_r D_p\dot{\alpha} + m_p^2 l_p^2 L_r g\alpha + \left(J_p + m_p l_p^2\right)\tau \right] \tag{5.6}$$

$$\ddot{\alpha} = \frac{1}{J_T}\left[m_p l_p L_r D_r \dot{\theta} - \left(J_r + m_p L_r^2\right) D_p \dot{\alpha} - m_p l_p g \left(J_r + m_p L_r^2\right) \alpha - m_p l_p L_r \tau\right], \quad (5.7)$$

where J_T is

$$J_T = J_p m_p L_r^2 + J_r J_p + J_r m_p l_p^2 \,. \quad (5.8)$$

Establishing the state vector $\mathbf{x}(t) = \begin{bmatrix} \theta & \alpha & \dot{\theta} & \dot{\alpha} \end{bmatrix}^T$, and using the following relations $\theta = \mathbf{x}_1$, $\alpha = \mathbf{x}_2$, $\dot{\theta} = \mathbf{x}_3$, $\dot{\alpha} = \mathbf{x}_4$, it is possible to obtain a linear model that approximates the nonlinear dynamics of (5.5) as follows

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (5.9)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \,, \quad (5.10)$$

where

$$\mathbf{A} = \frac{1}{J_T}\begin{bmatrix} 0 & 0 & J_T & 0 \\ 0 & 0 & 0 & J_T \\ 0 & m_p^2 l_p^2 L_r g & -\left(J_p + m_p l_p^2\right) D_r & m_p l_p L_r D_p \\ 0 & -m_p l_p g \left(J_r + m_p L_r^2\right) & m_p l_p L_r D_r & -\left(J_r + m_p L_r^2\right) D_p \end{bmatrix} \quad (5.11)$$

$$\mathbf{B} = \frac{1}{J_T}\begin{bmatrix} 0 \\ 0 \\ J_p + m_p l_p^2 \\ -m_p l_p L_r \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} . \quad (5.12)$$

The linear model in (5.9), based on the system matrices described by (5.11) and (5.12), is both *controllable* and *observable*, and can be used to formulate linear controllers to balance the pendulum around the upright position. This model assumes that the control input $\mathbf{u}$ is a torque signal, which needs to be converted to a voltage V_m that is eventually applied to the servo motor. This conversion is described by the following expression

$$V_m = \frac{R_m \mathbf{u}}{\eta_g \eta_m K_g k_t} + K_g k_m \dot{\theta} \,, \quad (5.13)$$

where η_g and η_m are the gearbox and motor efficiency values respectively, K_g is the gear ratio, k_t is the motor current to torque constant and k_m is the motor back EMF constant.

5.2.2 Augmented model for parameter estimation

In order to implement the proposed adaptive intermittent controllers for the rotational pendulum, the set of parameters φ that the recursive estimator should track must be selected, to then augment the state-vector in (5.5) as shown in (3.77).

Consider the case where the mass m_p and the distance from the rotating joint to the centre of mass l_p of the pendulum are the set of uncertain parameters defined as $\varphi = [\, m_p \quad l_p \,]^T$. Based on this, the new augmented state-vector is defined as

$$\mathbf{x}_{aug}(t) = \begin{bmatrix} \theta & \alpha & \dot{\theta} & \dot{\alpha} & m_p & l_p \end{bmatrix}^T . \tag{5.14}$$

Considering that $\boldsymbol{z} = \begin{bmatrix} \ddot{\theta} & \ddot{\alpha} \end{bmatrix}^T$, $m_p = \mathbf{x}_5$, and $l_p = \mathbf{x}_6$, the state variable system can be written as

$$\dot{\mathbf{x}}_{aug} = \begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \\ \boldsymbol{z} \\ \dot{m_p} \\ \dot{l_p} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_3 \\ \mathbf{x}_4 \\ \boldsymbol{f}(\mathbf{x}_{aug}(t), \tau(t)) \\ 0 \\ 0 \end{bmatrix}, \tag{5.15}$$

where the function $\boldsymbol{f}(\cdot)$ represents the solution for the angular accelerations of the system in terms of the $\mathbf{x}_{aug}$ and the torque τ

$$\boldsymbol{f}(\mathbf{x}_{aug}(t), \tau(t)) = \mathcal{M}^{-1}\left(\begin{bmatrix} \tau - D_r \mathbf{x}_3 \\ -D_p \mathbf{x}_4 \end{bmatrix} - \mathcal{C} \begin{bmatrix} \mathbf{x}_3 & \mathbf{x}_4 \end{bmatrix}^T - \begin{bmatrix} 0 \\ -\mathbf{x}_5 \mathbf{x}_6 g \sin(\mathbf{x}_2) \end{bmatrix} \right) \tag{5.16}$$

and

$$\mathcal{M} = \begin{bmatrix} \mathbf{x}_5 L_r^2 + \mathbf{x}_5 \mathbf{x}_6^2 - \mathbf{x}_5 \mathbf{x}_6^2 \cos(\mathbf{x}_2)^2 + J_r & -\mathbf{x}_5 \mathbf{x}_6 L_r \cos(\mathbf{x}_2) \\ -\mathbf{x}_5 \mathbf{x}_6 L_r \cos(\mathbf{x}_2) & J_p + \mathbf{x}_5 \mathbf{x}_6^2 \end{bmatrix} \tag{5.17}$$

$$\mathcal{C} = \begin{bmatrix} \left(2\mathbf{x}_5 \mathbf{x}_6^2 \sin(\mathbf{x}_2) \cos(\mathbf{x}_2)\right) \mathbf{x}_4 & \left(\mathbf{x}_5 \mathbf{x}_6 L_r \sin(\mathbf{x}_2)\right) \mathbf{x}_4 \\ -\left(\mathbf{x}_5 \mathbf{x}_6^2 \cos(\mathbf{x}_2) \sin(\mathbf{x}_2)\right) \mathbf{x}_3 & 0 \end{bmatrix}. \tag{5.18}$$

The system defined in (5.15) is the model upon which the Kalman based estimator is formulated.

5.2.3 System parameters

The nominal parameters of the rotational pendulum that were used for the simulation study are shown in Table 5.1.

Table 5.1: Rotational pendulum constants

Constant	Value	Units
m_p	0.230	kg
l_p	0.320	m
L_r	0.216	m
J_p	0.008	kg m^2
J_r	0.001	kg m^2
D_r	0.0024	N.m.s/rad
D_p	0.0024	N.m.s/rad

For the real-time experiments, the nominal parameters were the same as in Table 5.1 except from $m_p = 0.127$ kg, $l_p = 0.160$ m, and $J_p = 0.001$ kg m^2. The values associated to the servo motor are shown in Table (5.2)

Table 5.2: Actuator constants

Constant	Value	Units
η_g	0.90	-
η_m	0.69	-
K_g	70	-
k_t	0.0077	N.m/A
k_m	0.0077	V/(rad/s)

5.3 Simulation scenario

Two different cases were used to test the system-matched hold and tapping hold adaptive intermittent controllers presented in the previous chapter. First, a *tracking* case is evaluated, where the controllers force the arm angle θ to follow a reference w in the form of a square function between 0 and 11.5 deg with a period of 10 sec, while keeping the pendulum angle α as close as possible to zero degrees (vertical unstable equilibrium point). The second case uses zero degrees as a reference for both θ and α angles, this situation is commonly known as a the *regulation* case.

During the simulation, the pendulum parameters $L_p = 2l_p$ and m_p are artificially modified to a different value, thus changing the dynamics of the simulated plant at different times. The goal of all controllers is to keep track of these changes and adjust the control law accordingly.

In addition to the adaptive system-matched hold IC (referred in the figures as SMH) and the adaptive tapping IC (shown as ITC), the data generated by an adaptive con-

tinuous controller (CC) is also shown for comparison purposes (the structure of this controller is shown in Fig. 3.10). The adaptive CC implements the following control law

$$\mathbf{u}(t) = -\mathbf{k}\mathbf{x}_w(t) + \boldsymbol{u}_{ss}w(t)\,, \tag{5.19}$$

with the distinction that the gain $\mathbf{k}$ is recalculated continuously (i.e. every simulation step) based on the parameters $\varphi(t)$. This controller can be seen as a result of implementing only the steps in the underlying continuous design stage that serves as a basis for IC. In terms of the state and parameter estimation, all controllers were implemented using both Extended and Unscented Kalman filters.

5.3.1 Controller and estimator design

The details about the control design parameters used for the simulations are presented in the following paragraphs, including the times when the system parameters were modified and the values used to simulate input and output noise.

Timing parameters for AIC: the intermittent controllers were implemented using a minimum open-loop interval of $\Delta_{min} = 0.05$ sec, a delay of $\Delta = 0.003$ sec (this delay is compensated by the intermittent predictor in (3.114)), and $q = 0.5$ deg as the threshold value. The triggering mechanism, which dictates when the events are generated, only uses the measured states θ and α, discarding the respective angular velocities.

LQR design: the following values were used to calculate the state-feedback gains $\mathbf{k}$ using the linear quadratic regulator approach

$$\mathbf{Q}_{c,diag} = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} \qquad \mathbf{R}_c = 1\,, \tag{5.20}$$

where the vector $\mathbf{Q}_{c,diag}$ is the diagonal of $\mathbf{Q}_c$. Both matrices stay constant during the entire simulation, therefore the variation in $\mathbf{k}$ is only modulated by the online computation of the system matrices $\mathbf{A}$ and $\mathbf{B}$.

Tapping hold design: the tapping controller was designed using $p = 15$ and $N = 2$ to generate $\mathbf{A}_p$ following (3.39). This choice yields

$$\mathbf{A}_p = \begin{bmatrix} -p & 0 \\ -2p & -p \end{bmatrix} = \begin{bmatrix} -15 & 0 \\ -30 & -15 \end{bmatrix}. \tag{5.21}$$

The optimisation horizon τ_1 was selected to be as long as the open-loop interval Δ_{ol}, which is 0.05 sec.

Kalman filter parameters: both the EKF and UKF were designed using the same values of the initial error $\mathbf{P}_o$, the process noise $\mathbf{Q}$ and the measurement noise $\mathbf{R}$ covariance matrices. The respective diagonals of $\mathbf{P}_o$ and $\mathbf{Q}$ are described by the following vectors

$$\mathbf{P}_{0,diag} = \begin{bmatrix} 0.1 & 0.1 & 0.1 & 0.1 & 1 & 1 \end{bmatrix} \qquad \mathbf{Q}_{diag} = \begin{bmatrix} 10^{-8} & 10^{-8} & 10^{-8} & 10^{-8} & 10^{-7} & 10^{-7} \end{bmatrix}, \tag{5.22}$$

and $\mathbf{R} = 0.002\ \mathbf{I}_{n_y \times n_y}$, where n_y represents the number of measured outputs in the system, which is 2 in this case. Additionally the vector of initial conditions for both estimators was defined as $\mathbf{x}_o(0) = \begin{bmatrix} 0 & 5.15 & 0 & 0 & 1 & 1 \end{bmatrix}$, which assigns 5.15 deg for the pendulum angle θ and a value of 1 for both l_p and m_p. The design parameters of the UKF, which control the spread of the sigma points involved in the unscented transformation, and the a-priori distribution of $\mathbf{x}$ respectively were set to $\alpha_{ukf} = 0.0001$ and $\beta_{ukf} = 2$ as suggested by Wan and Van Der Merwe (2000).

Time-varying parameters $\varphi(t)$: the nominal value of m_p in Table 5.1 is 0.230 kg, which is used to start the simulation until it is increased by a factor of 2.5 at $t = 7.5$ sec. Similarly, $l_p = 0.320$, was doubled at $t = 17.5$ sec.

Input and measurement noise: as shown in Fig 3.11, v_u and v_y represent input and measurement noise respectively. These were simulated using randomly seeded Gaussian noise with $v_u = 0.01$ and $v_y = 0.001$ as the respective amplitudes.

5.3.2 Results

Before showing the results obtained from the adaptive controllers, it makes sense to present the responses of each of them in two different conditions: i) when the non-adaptive version of the controllers is used with an invariant pendulum model (constant parameters), and ii) when the pendulum parameters do change but the same non-adaptive controllers are used. This simulation was performed in a noise free environment ($v_u = v_y = 0$) and using an EKF to estimate the system state $\mathbf{x} = \begin{bmatrix} \theta & \alpha & \dot{\theta} & \dot{\alpha} \end{bmatrix}$. Notice that in this case, the parameters l_p and m_p are not considered for estimation purposes, therefore the controller is unaware of any variation. The evolution of the arm angle θ and the associated control input are shown in Fig. 5.2.

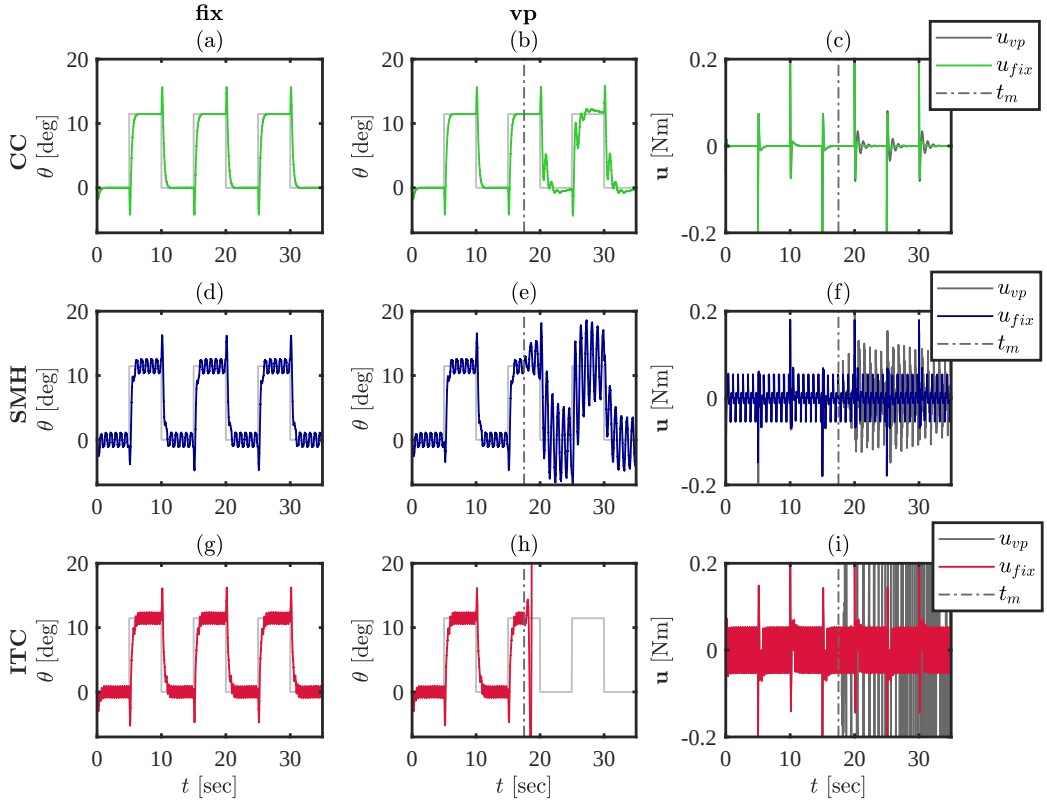


Figure 5.2: Non-adaptive control. Each controller is represented by a row in the figure: CC corresponds to sub-figures (a,b,c) in green, SMH to (d,e,f) in blue, and ITC to (g,h,i) in red. The first column (titled **fix**) shows the evolution of θ when the model is not changed (no mismatch). The second column (titled **vp**) shows θ when the parameter m_p is changed from 0.230 to 0.345 kg at $t = 17.5$ sec. The third column overlaps the control input $\mathbf{u}(t)$ which corresponds to the response from the **fix** column (u_{fix}) and the control input associated to the **vp** column (u_{vp}).

The first case, where the model parameters are fixed (no variation) is represented by the column labelled as **fix**. The column labelled as **vp** (varying parameters) shows the response when the mass of the pendulum m_p is changed from 0.230 to 0.345 kg at 17.5 sec. Lastly, the third column overlaps the control input corresponding to the **fix** column (u_{fix}) and the control input that generated the response of the **vp** column (u_{vp}). Each controller is represented by a row in the figure: sub-figures (a,b,c) correspond to CC, (d,e,f) to SMH, and (g,h,i) to ITC.

These results show how the arm angle trajectory θ is affected by a sudden parameter change at $t = 17.5$ sec (indicated by the vertical dashed line) if there is no adaptive strategy in place. It can be seen from (b) and (e) that the CC response is less affected compared to the SMH, where the oscillations grow considerably. Still, the SMH is capable of withstanding the variation and follow the reference. On the contrary, ITC gives an unstable response soon after m_p is changed. This illustrates the need to track

how m_p is changing over time. The next section shows the results of the tracking case when adaptation is used.

Tracking case

First, the evolution of θ, α, and the control input $\mathbf{u}$ is shown for both the EKF (Fig. 5.3) and UKF (Fig. 5.4).

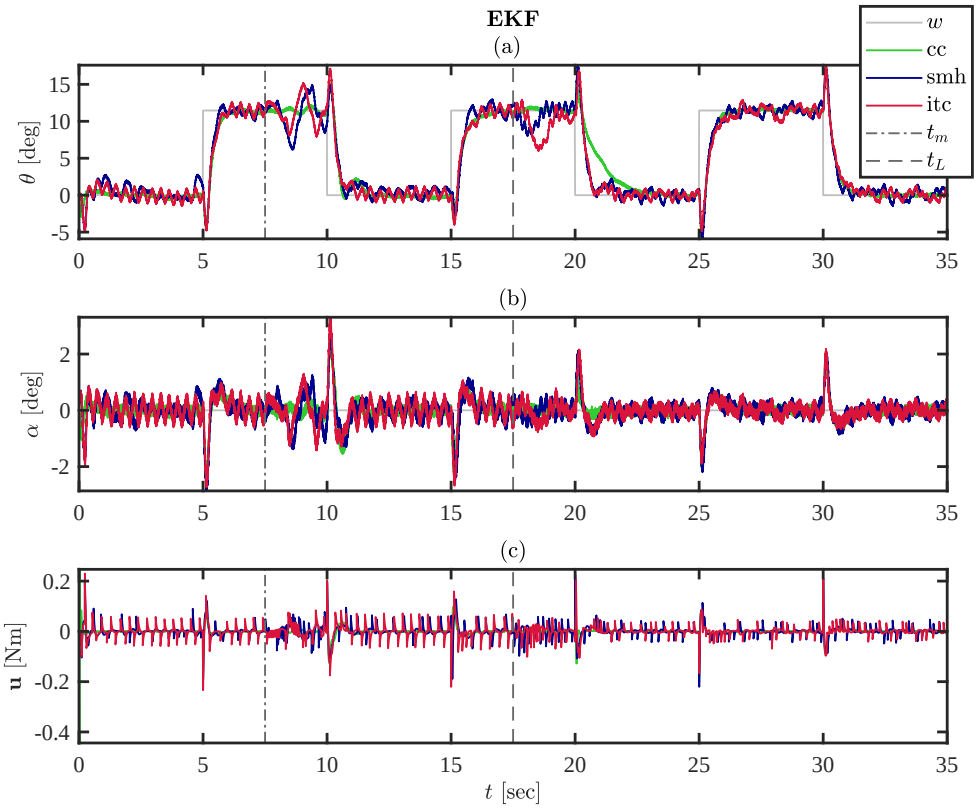


Figure 5.3: Adaptive control - EKF. (a) shows the arm angle θ, (b) is the pendulum angle α, and (c) is the control input $\mathbf{u}$. Each controller is shown in a different colour: green for CC, blue for SMH, and red for ITC. The times when m_p and l_p change are represented by vertical lines by t_m and t_L respectively. The reference w is a square function for θ and 0 deg for α.

The outputs from both filters (EKF and UKF) are similar and follow the same general trend. However, there are small differences between the two:

- The transient during the first seconds of the simulation is shorter for the EKF. This is particularly evident in Fig. 5.4b,c where the pendulum angle α and the control input $\mathbf{u}$ oscillate several times before reaching its steady-state.
- At $t_m = 7.5$ sec, the mass of the pendulum changes and this increases the tracking error for all controllers, being larger for SMH and ITC and being approximately

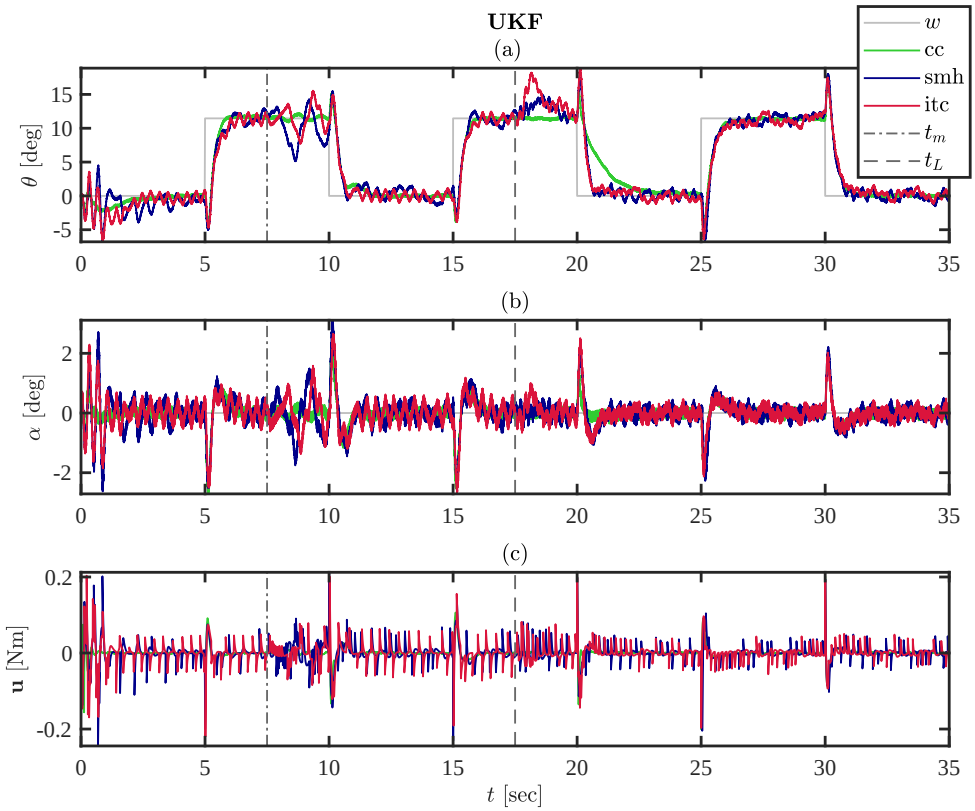


Figure 5.4: Adaptive control - UKF. (a) shows the arm angle θ, (b) is the pendulum angle α, and (c) is the control input $\mathbf{u}$. Each controller is shown in a different colour: green for CC, blue for SMH, and red for ITC. The times when m_p and l_p change are represented by vertical lines by t_m and t_L respectively. The reference w is a square function for θ and 0 deg for α.

similar in amplitude between the EKF and UKF. However, after $t_L = 17.5$ (when the pendulum length changes), θ from Fig. 5.3 responds in the opposite direction compared to the one in Fig. 5.4 for SMH and ITC.

After $t = 20$, the reference w goes from 11.5 deg to 0; the intermittent controllers respond faster to this change compared to CC. This is due to the fact that SMH and ITC have parameter estimates that are closer to the nominal value by the time the reference changes, as shown in Fig. 5.5. The estimated parameters are shown together with the open-loop interval distribution for both intermittent controllers in Fig. 5.5. The first row (a,b) shows a comparison between the *true* parameter m_p and the corresponding estimates obtained with each controller; similarly, the second row (c,d) is a comparison for l_p and finally, the third row (e,f) shows the open-loop intervals for SMH and ITC. The left column in the figure corresponds to EKF based controllers, the one in the right for UKF.

During first seconds of the simulation, the parameter estimates provided by the EKF in

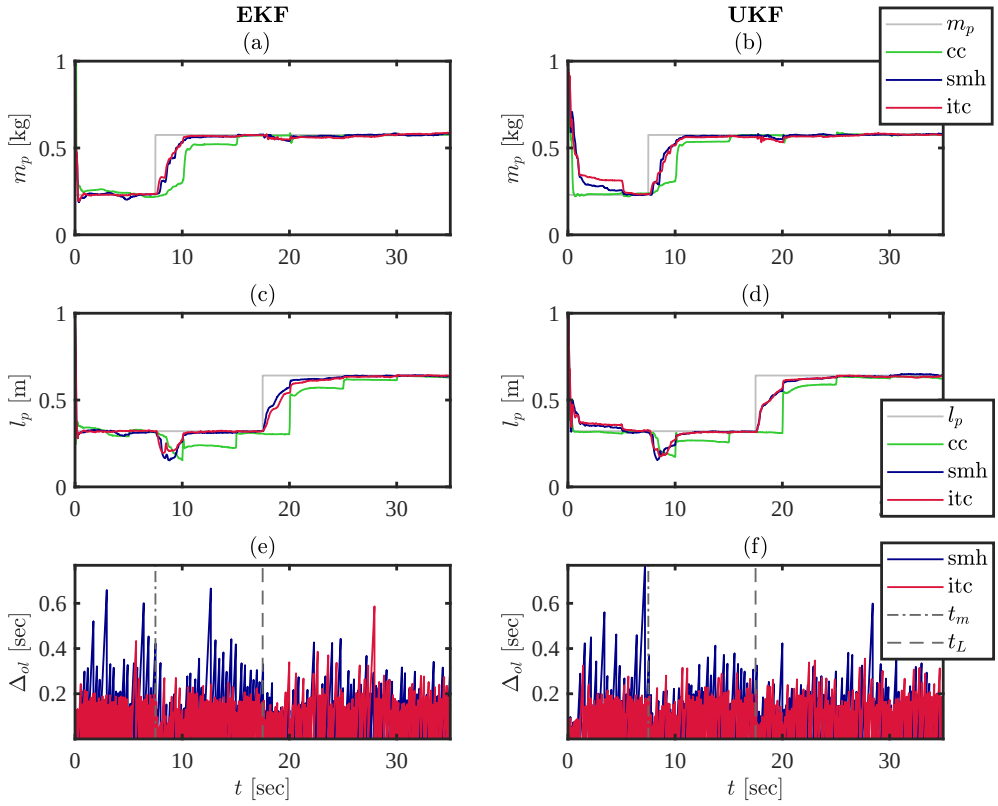


Figure 5.5: Adaptive control - Estimated parameters and open-loop intervals. The first row (a,b) shows m_p estimates obtained with each controller; similarly, the second row (c,d) corresponds to l_p estimates. The third row (e,f) shows the open-loop intervals for SMH and ITC respectively. The left column (a,c,e) corresponds to EKF based controllers, the one in the right (b,d,f) for UKF.

Fig. 5.5a,c converge to the correct values faster compared to the UKF (b,d). This effect is particularly evident for the SMH and ITC, converging only after the first change in reference at $t = 5$ sec. Once m_p and l_p change to their new values, the estimates provided by the SMH and ITC start converging steadily, while the CC estimates have to wait until the next change in reference to catch up. This is due to the fact that the CC control signal does not excite the system enough after the parameter change, resulting in poor estimation. Both intermittent controllers generate control signals that are higher in amplitude in response to the increased tracking error of θ and α, which aids the estimation process.

The open-loop interval Δ_{ol} is presented in Fig. 5.5e,f, showing that the SMH generates a longer Δ_{ol} on average compared to the ITC, specially before $t_L = 17.5$ sec. Right after t_m and t_L, the open-loop intervals reduce considerably for both controllers, which is a consequence of the growing errors in θ and α; the intermittent controllers are forced to close the loop more often, leading to more frequent controller redesigns.

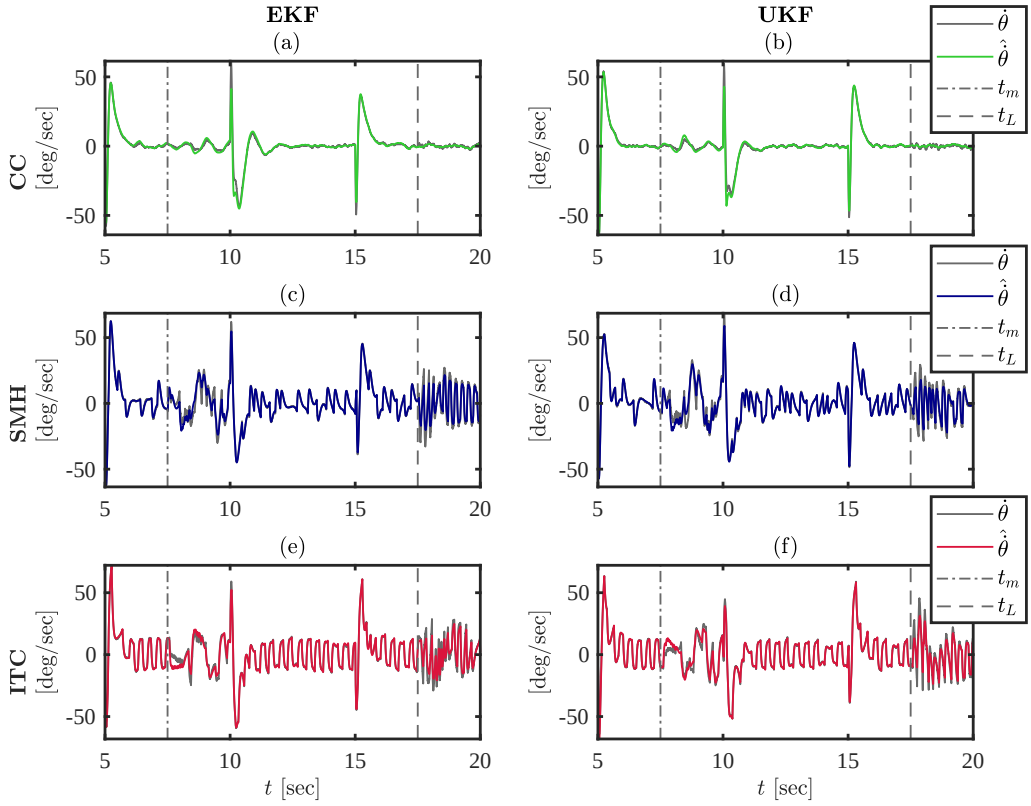


Figure 5.6: Adaptive control - Comparison between the estimated angular velocity $\hat{\dot{\theta}}$ and the simulated value $\dot{\theta}$. Each controller is represented a row: (a,b) for CC, (c,d) for SMH, and (e,f) for ITC. The left column shows data for EKF based controllers, whereas the UKF is displayed on the right side column.

Finally, Fig. 5.6 shows angular velocity $\hat{\dot{\theta}}$ estimates for the arm in comparison with the simulated velocity $\dot{\theta}$. This is in fact a state-variable which is not measured directly, therefore the estimators generate this state based on angle measurements only. The estimation error is small for all controllers, increasing slightly only after the parameters change, converging to the true values a few seconds later.

Regulation case

The regulation case can be seen as part of the tracking case where the reference w is zero for all states. This implies that once the system reaches the steady-state, the control input also decays to zero. The results of this case are presented in a similar way to the tracking case, showing the outputs and inputs for EKF and UKF in Fig. 5.7 and Fig. 5.8 respectively, estimated parameters and open-loop distributions in Fig. 5.9 and estimated arm angular velocity in Fig. 5.10.

As in the tracking case, the responses shown in Fig. 5.7 and Fig. 5.8 share similar

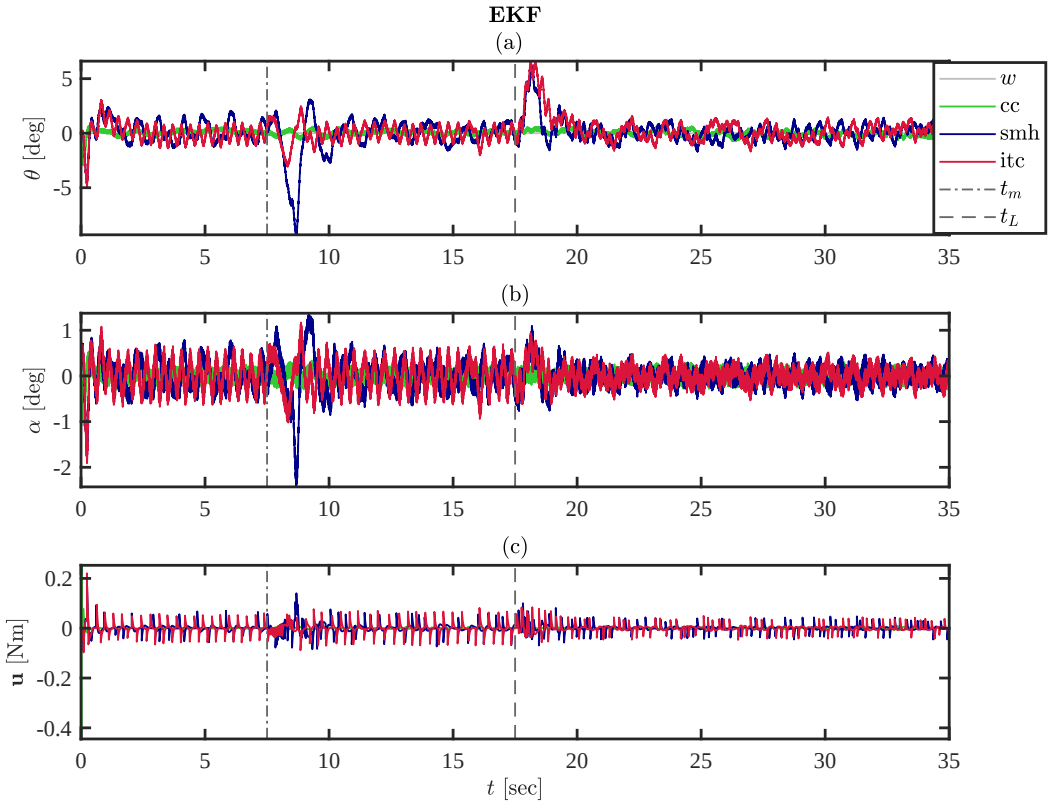


Figure 5.7: Adaptive control - EKF. (a) shows the arm angle θ, (b) is the pendulum angle α, and (c) is the control input $\mathbf{u}$. Each controller is shown in a different colour: green for CC, blue for SMH, and red for ITC. The times when m_p and l_p change are represented by vertical lines by t_m and t_L respectively. Reference w is 0 for θ and α.

trends and are all stable. However, there are important things to notice from both figures:

- After the parameter m_p changes at $t_m = 7.5$ sec, the tracking error of all controllers increases. Interestingly, the EKF based response of the SMH controller in Fig. 5.7 is affected considerably more compared to its UKF counterpart in Fig. 5.8. Moreover, the ITC response shows similar tracking errors regardless the type of estimator.

- Once the length of the pendulum l_p changes at $t_L = 17.5$ sec, the SMH and ITC controllers respond with similar tracking errors, recovering approximately within 5 sec.

- The parameter change at t_L introduces small oscillations to both angles in Fig. 5.7, whereas the response given by the UKF based CC is affected considerably more, with oscillations that grow up approximately to 5 deg for θ and 1.5 deg for α, around $t = 30$ sec.

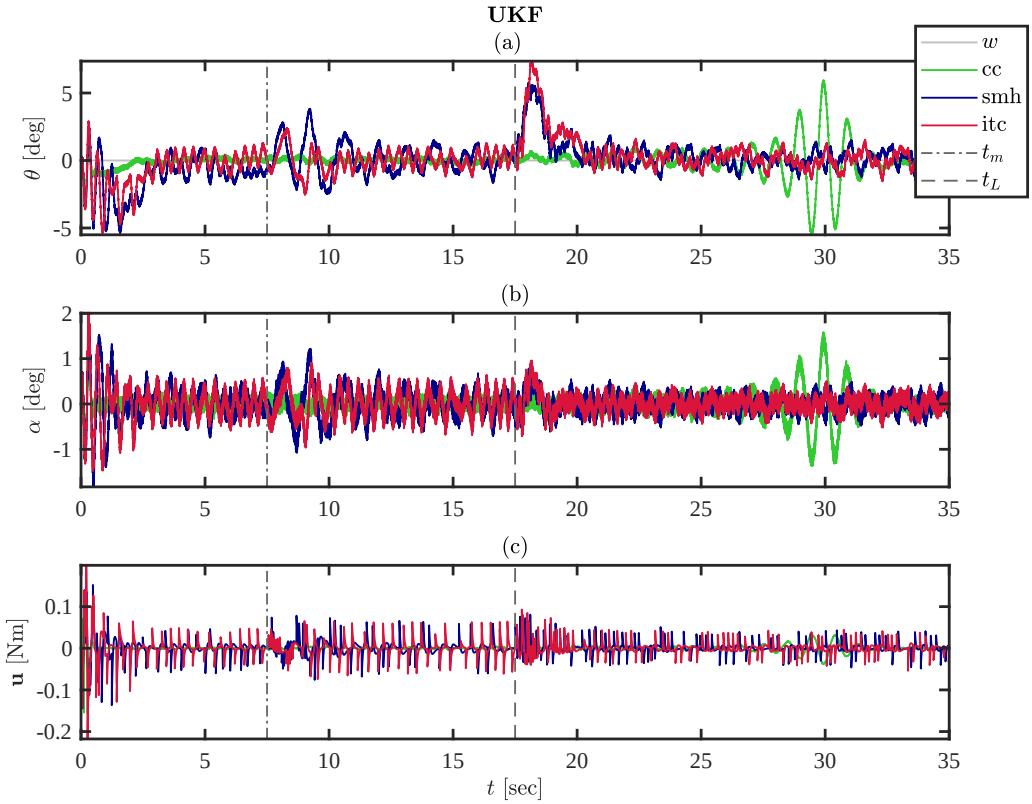


Figure 5.8: Adaptive control - UKF. (a) shows the arm angle θ, (b) is the pendulum angle α, and (c) is the control input $\mathbf{u}$. Each controller is shown in a different colour: green for CC, blue for SMH, and red for ITC. The times when m_p and l_p change are represented by vertical lines by t_m and t_L respectively. Reference w is 0 for θ and α.

Fig. 5.9 shows clearly how the estimated parameters obtained with CC do not converge to the expected value regardless the type of estimator, in fact, the estimation error of the EKF never decreases once the parameters change. The UKF estimation error decreases significantly around $t = 30$ sec as a result of the increasing oscillations observed in Fig. 5.8, these oscillations force the controller to increase the control signal to reduce the tracking error, which improves the estimation slightly. The situation is completely different for the SMH and ITC, where both of them start responding as soon as the parameter changes, converging to the correct values. The open-loop intervals of the intermittent controllers are comparable to the ones in the tracking case, where Δ_{ol} is consistently higher for the SMH specially before the length of the pendulum changes at $t_L = 17.5$.

Finally, the estimated angular velocity $\hat{\dot{\theta}}$ is shown in Fig. 5.10. In contrast with the tracking case 5.6, the estimation error of CC (for EKF and UKF) is now affected by the fact that estimated parameters never converge to the correct values, resulting in an error that increases with time.

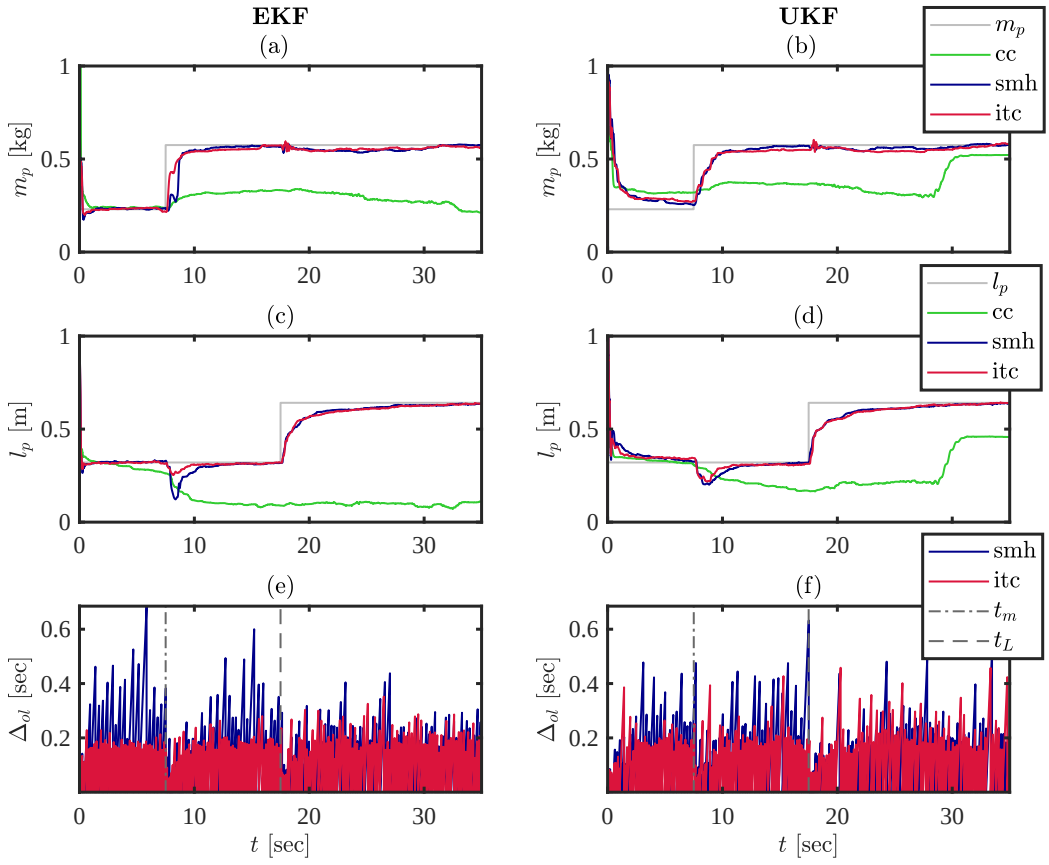


Figure 5.9: Adaptive control - Estimated parameters and open-loop intervals. The first row (a,b) shows m_p estimates obtained with each controller; similarly, the second row (c,d) corresponds to l_p estimates. The third row (e,f) shows the open-loop intervals for SMH and ITC respectively. The left column (a,c,e) corresponds to EKF based controllers, the one in the right (b,d,f) for UKF.

5.4 Real-time experiment

In this section, the experimental results of applying adaptive intermittent controllers on a real rotational pendulum are presented. In this sense, the same controllers were used to balance the pendulum around the equilibrium point; however, the situation is different since in reality, the parameters of the pendulum do not change with time (as in the simulation study). The experiment was designed to consider an initial controller design that is based on parameters that do not match the nominal values. In a way, this is equivalent to designing controllers for a *different* system, which deteriorates the overall response. Since the estimation process runs continuously, an approximation to the *real* values is obtained and then the redesign stage of AIC is enabled in order to update the controllers.

The results are presented in the same style as in the simulation scenario, showing data

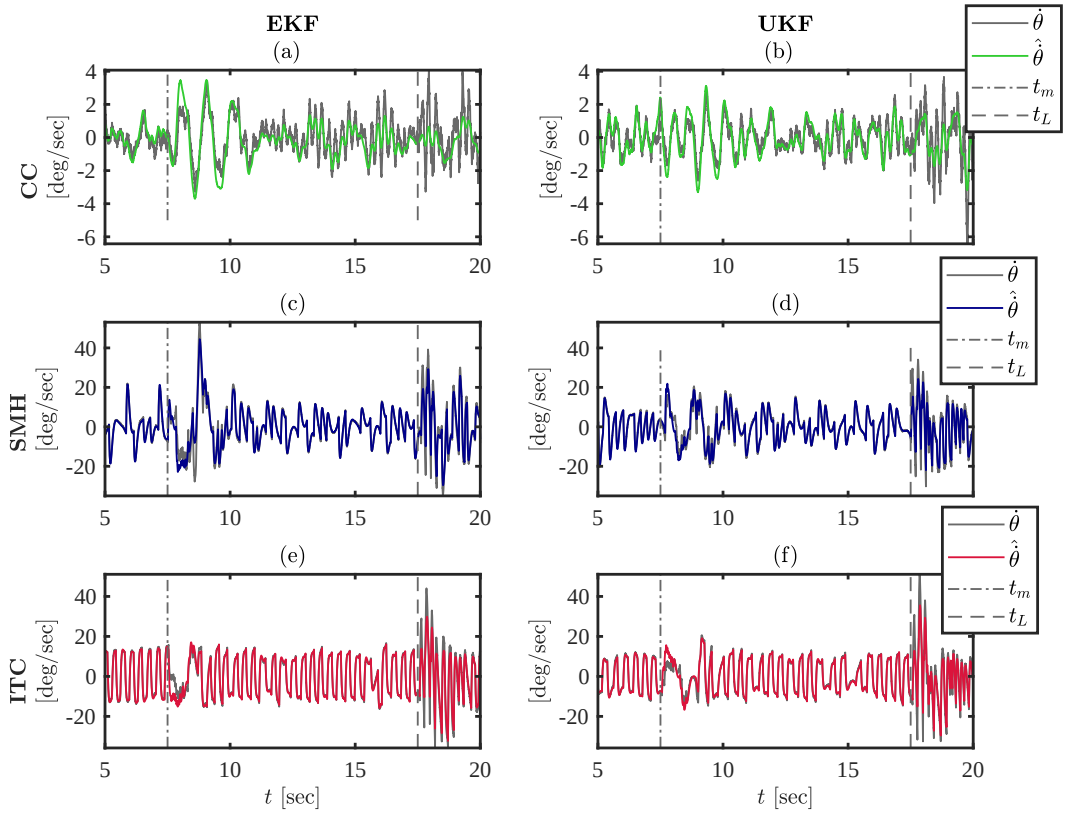


Figure 5.10: Adaptive control - Comparison between the estimated angular velocity $\hat{\dot{\theta}}$ and the simulated value $\dot{\theta}$. Each controller is represented a row: (a,b) for CC, (c,d) for SMH, and (e,f) for ITC. The left column shows data for EKF based controllers, whereas the UKF is displayed on the right side column.

for each controller, using both estimators (EKF and UKF).

5.4.1 Experiment protocol

The experiment involves running three adaptive controllers (CC, SMH, and ITC) to control the pendulum angle α at zero degrees (unstable equilibrium point) while forcing the arm angle θ to follow a square reference signal $w(t)$ that switches between -10 and 10 deg. The total length of the experiment is 60 sec, divided in 30 seconds of evolution using an initial design based on an erroneous pendulum mass parameter $m_p = 0.07$ kg, then the redesign of the controllers based on available parameter estimates is enabled at $t_a = 30$ sec. During the 30 seconds of adaptation, the intermittent controllers use estimates of m_p to update their respective control laws at event times t_i, whereas the continuous controller does this every sample interval.

In order to start the experiment, the pendulum had to be manually rotated to an angle of $\alpha = 12$. At this point, the controllers took over and started balancing the pendulum.

Usually, this can be implemented automatically if a swing-up controller is designed to apply a voltage to the system that would eventually take the pendulum up to angle where the balance controller can start operating. For this experiment, this part is not relevant and was omitted. The fact that the experiment had to be started manually resulted in controllers that started operating in slightly different times; however, after the first change in reference $w(t)$, these differences decrease considerably.

5.4.2 Performance index

To compare the performance of implemented controllers, the following performance indexes are introduced:

- **Root mean squared tracking error (RMS)**: the tracking error $e(t)$ for each controller can be defined for both outputs as

$$e_\theta(t) = \theta(t) - w(t) \qquad e_\alpha(t) = \alpha(t) - w(t)\,. \tag{5.23}$$

  Then, the tracking RMS error is formulated as

$$RMS(e) = \sqrt{\frac{1}{m}\sum_{i=1}^{m} \mid e(t_i) \mid^2}\,, \tag{5.24}$$

  with m being the number of data samples used for the computation.

- **Root mean squared estimation error:** the parameter estimation error can be established as the difference between the nominal parameter value $m_{p,nom}$ and its estimate $\hat{m}_p$

$$e_{m_p}(t) = m_{p,nom} - \hat{m}_p(t)\,, \tag{5.25}$$

  based on this, the estimation RMS error can be computed as in (5.24).

These performance indexes are applied to the recorded data of all controllers before and after the redesign stage is enabled.

5.4.3 Controller and estimator design

The values used to design the controllers are the same as in the simulation study (5.3) except from $\Delta_{min} = 0.01$ sec, the threshold q being set to 1 deg, and the diagonal of the LQR design matrix $\mathbf{Q}_c$ being defined now by the vector [1.5 1.5 0 0].

Since only m_p is being estimated in this experiment, the augmented state vector upon which both Kalman filters were designed is as follows

$$\mathbf{x}_{aug}(t) = \begin{bmatrix} \theta & \alpha & \dot{\theta} & \dot{\alpha} & m_p \end{bmatrix}^T . \tag{5.26}$$

Defining $\theta = \mathbf{x}_1$, $\alpha = \mathbf{x}_2$, $\dot{\theta} = \mathbf{x}_3$, $\dot{\alpha} = \mathbf{x}_4$, $m_p = \mathbf{x}_5$, and using expression (5.26), it is possible to write

$$\dot{\mathbf{x}}_{aug} = \begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \\ \boldsymbol{z} \\ \dot{m}_p \end{bmatrix} = \begin{bmatrix} \mathbf{x}_3 \\ \mathbf{x}_4 \\ \boldsymbol{f}(\mathbf{x}_{aug}(t), \tau(t)) \\ 0 \end{bmatrix}, \tag{5.27}$$

where $\boldsymbol{z} = \begin{bmatrix} \ddot{\theta} & \ddot{\alpha} \end{bmatrix}^T$ and $\boldsymbol{f}(\cdot)$ being

$$\boldsymbol{f}(\mathbf{x}_{aug}(t), \tau(t)) = \mathcal{M}^{-1}\left(\begin{bmatrix} \tau - D_r\mathbf{x}_3 \\ -D_p\mathbf{x}_4 \end{bmatrix} - \mathcal{C}\begin{bmatrix} \mathbf{x}_3 & \mathbf{x}_4 \end{bmatrix}^T - \begin{bmatrix} 0 \\ -\mathbf{x}_5 l_p g \sin(\mathbf{x}_2) \end{bmatrix} \right), \tag{5.28}$$

and

$$\mathcal{M} = \begin{bmatrix} \mathbf{x}_5 L_r^2 + \mathbf{x}_5 l_p^2 - \mathbf{x}_5 l_p^2 \cos(\mathbf{x}_2)^2 + J_r & -\mathbf{x}_5 l_p L_r \cos(\mathbf{x}_2) \\ -\mathbf{x}_5 l_p L_r \cos(\mathbf{x}_2) & J_p + \mathbf{x}_5 l_p^2 \end{bmatrix} \tag{5.29}$$

$$\mathcal{C} = \begin{bmatrix} \left(2\mathbf{x}_5 l_p^2 \sin(\mathbf{x}_2)\cos(\mathbf{x}_2)\right)\mathbf{x}_4 & \left(\mathbf{x}_5 l_p L_r \sin(\mathbf{x}_2)\right)\mathbf{x}_4 \\ -\left(\mathbf{x}_5 l_p^2 \cos(\mathbf{x}_2)\sin(\mathbf{x}_2)\right)\mathbf{x}_3 & 0 \end{bmatrix}. \tag{5.30}$$

The initial condition for the Kalman filters was set to $\mathbf{x}_o(0) = [0\ 0\ 0\ 0\ 2]^T$, and the state covariance matrix $\mathbf{Q}$, was defined by the following vector in its diagonal:

$$\mathbf{Q}_{diag} = \begin{bmatrix} 10^{-7} & 10^{-7} & 10^{-3} & 10^{-3} & 10^{-6} \end{bmatrix} . \tag{5.31}$$

5.4.4 Results

In Figs. 5.11 and 5.12, the outputs θ and α along with the control input $\mathbf{u}$ are presented. In both cases, the oscillations around the setpoint $w(t)$ for θ (shown in (a)), decrease after the redesigns start taking place at t_a. In terms of amplitude, the oscillations are very similar for SMH and ITC, matching closely when the reference is at -10 deg. For CC, the oscillations are slightly smaller compared to both intermittent controllers. The pendulum angle α, shown in (b), also shows a reduction in the amplitude of the oscillations; however, CC and ITC have larger errors specially when the reference changes suddenly.

The control inputs in Figs. 5.11c, 5.12c show very clearly the effect of the parameter mismatch (before t_a), high control values are generated for some portions of the experiment, these being particularly higher for the intermittent controllers. Once the redesigns start, the control inputs decrease significantly since the estimates of m_p are used to improve the controllers.

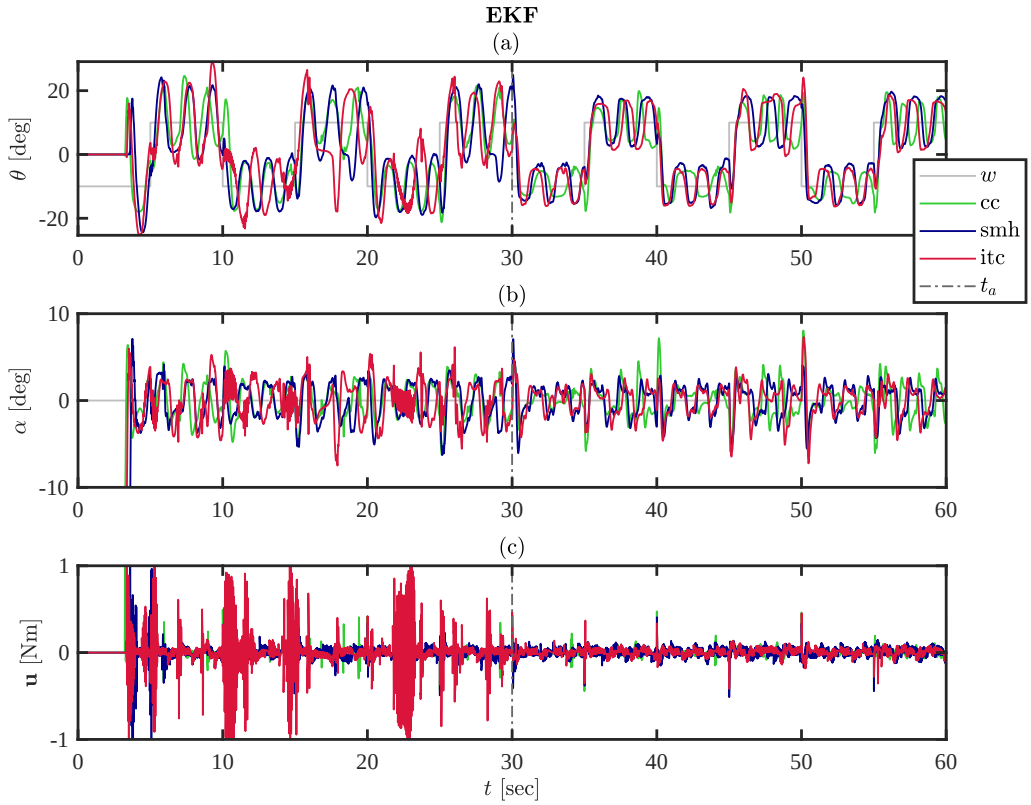


Figure 5.11: Adaptive control - EKF. (a) shows the arm angle θ, (b) is the pendulum angle α, and (c) is the control input $\mathbf{u}$. Each controller is shown in a different colour: green for CC, blue for SMH, and red for ITC. The time when adaptation is enabled t_a, is represented by a vertical line. The reference w is a square function for θ and 0 deg for α.

The EKF and UKF estimates of m_p, along with the open-loop interval distributions of the intermittent controllers are shown in Fig. 5.13. Comparing (a) and (b), it is clear that the variance of the EKF estimates is higher than the estimates obtained with the UKF. Also, the UKF estimates take longer to converge to the nominal value of m_p. In (b), the estimates for CC and ITC start deviating, increasing the error; however, the SMH estimates stays close to the nominal value the entire time.

The EKF and UKF open-loop interval distributions in 5.13c and 5.13d, show a small increase in the overall duration of Δ_{ol} after the redesigns start at t_a. Throughout the entire experiment, Δ_{ol} reaches values between 0.1 and 0.2 sec consistently, which is

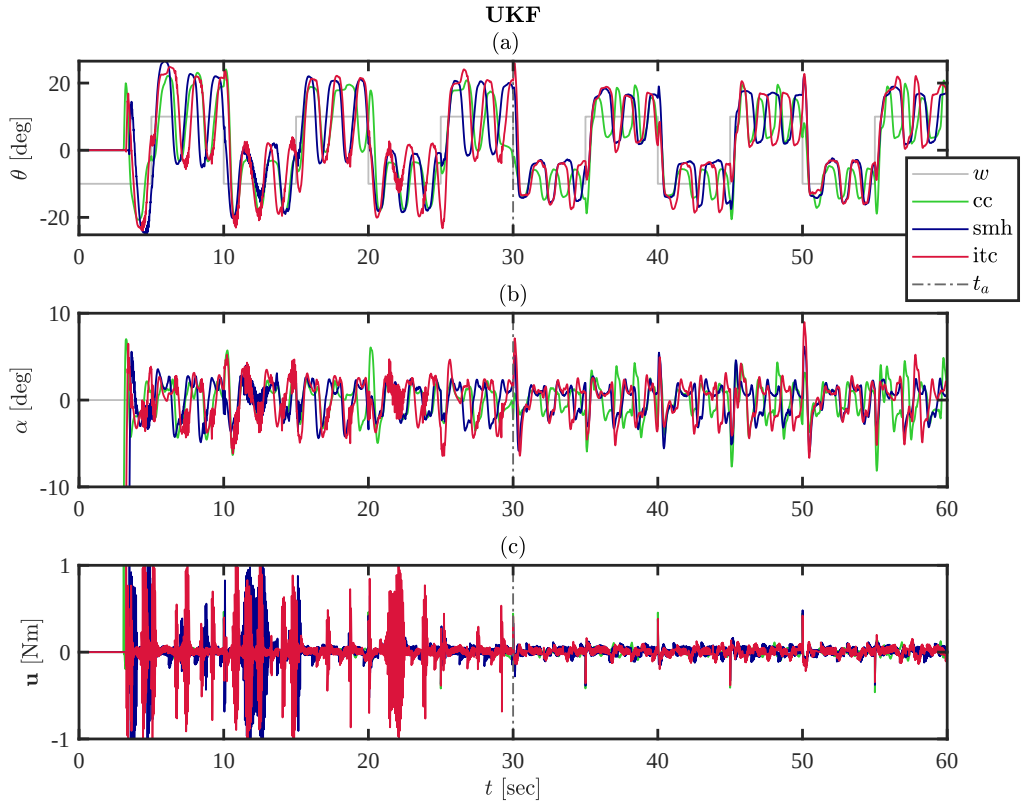


Figure 5.12: Adaptive control - UKF. (a) shows the arm angle θ, (b) is the pendulum angle α, and (c) is the control input $\mathbf{u}$. Each controller is shown in a different colour: green for CC, blue for SMH, and red for ITC. The time when adaptation is enabled t_a, is represented by a vertical line. The reference w is a square function for θ and 0 deg for α.

approximately 10 times greater the imposed minimum open-loop interval.

In Fig. 5.14, the phase planes for θ and α are shown. The data for all UKF based controllers is represented by row, with the first row (a,b) corresponding to CC, the second row (c,d) to SMH, and (e,f) to ITC. The figure shows trajectories before t_a in grey colour. The yellow trajectories reflect the corrections made by all controllers after the redesigns start at t_a. The left column (a,c,e) displays two yellow circles that correspond to the oscillations made around the setpoint $w(t)$ at 10 and -10 deg. In (c) and (e), it is possible to see how the parameter mismatch of the initial controller design affects the intermittent controllers before t_a (grey colour trajectories). Once the redesign takes place, the trajectories become very similar in shape to those displayed in (a), corresponding to CC.

The trajectories for α in right column (b,d,f) are centred around 0 deg, which is the equilibrium point for the pendulum. The CC case in (b) shows how these trajectories actually move away from the centre and grow in magnitude after t_a. In contrast, the

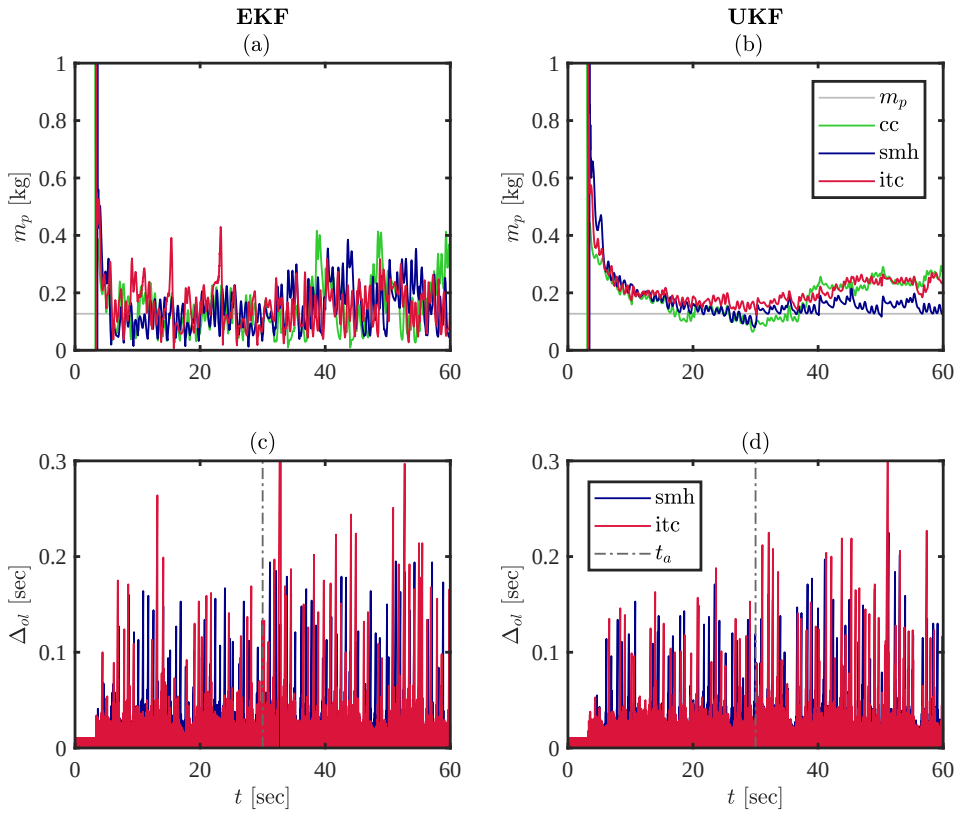


Figure 5.13: Adaptive control - Estimated parameter and open-loop intervals. The first row (a,b) shows m_p estimates obtained with each controller. The second row (c,d) shows the open-loop intervals for SMH and ITC respectively. The left column (a,c) corresponds to EKF based controllers, the one in the right (b,d) for UKF.

two intermittent controllers reduce the trajectories as a result of adaptation.

Performance index results

In Table 5.3, the results of applying the performance indexes described in 5.4.2 are presented. The table shows the RMS tracking error for both θ and α, labelled as $\boldsymbol{\theta_{EKF}}$, $\boldsymbol{\theta_{UKF}}$, $\boldsymbol{\alpha_{EKF}}$, and $\boldsymbol{\alpha_{UKF}}$, with all of them measured in degrees. Also, the RMS parameter estimation error is shown as $\boldsymbol{m_{p,EKF}}$ and $\boldsymbol{m_{p,UKF}}$, measured in Kg. Two columns are presented for each of these performance indexes, t_a^- represents data before the redesign of all controllers, specifically between 5 and 30 sec, while t_a^+ corresponds to the data after adaptation is enabled, from 30 to 60 sec.

The overall results show how all the controllers are capable of reducing the tracking error once adaptation is enabled. The CC reaches smaller tracking errors for θ ($t_a^+ = 7.35$ for EKF and $t_a^+ = 6.72$ for UKF), in comparison with both intermittent controllers, independently of the estimation procedure; however, the tracking error is smaller when

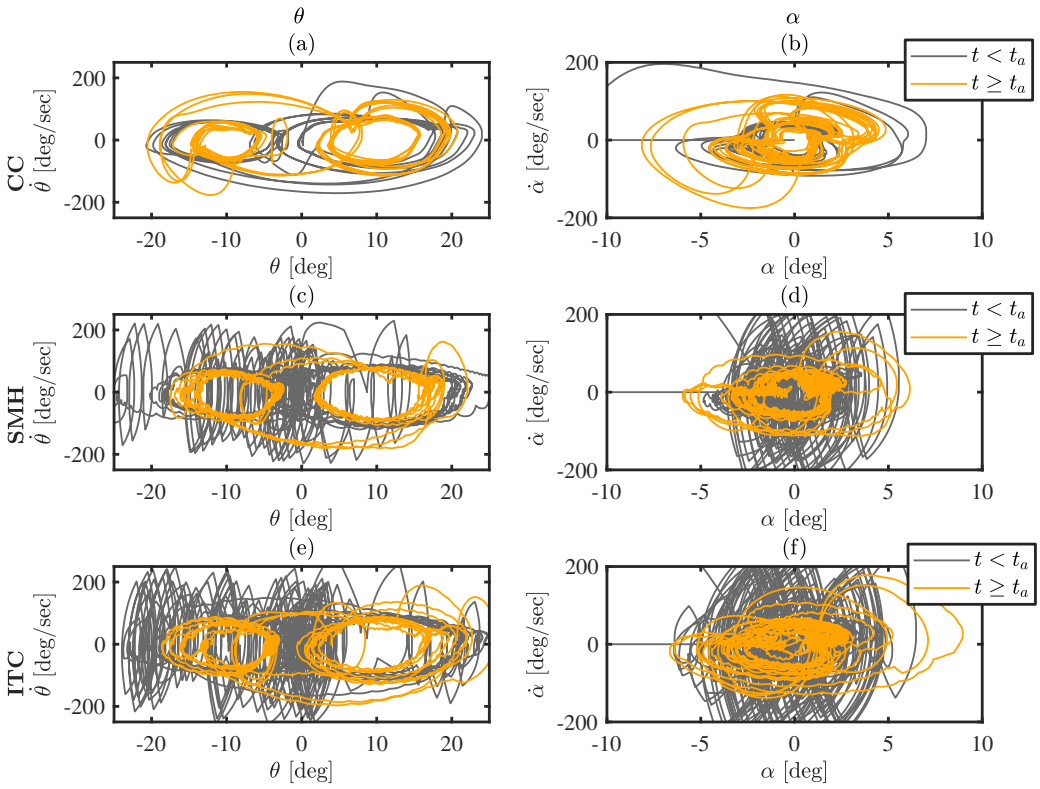


Figure 5.14: Adaptive control - Phase planes. The first row shows phase planes for θ (a) and α (b) for CC. The second (c,d) and the third rows (e,f) show phase planes for SMH and ITC respectively.

Table 5.3: RMS tracking and parameter estimation errors

	$\boldsymbol{\theta_{EKF}}$		$\boldsymbol{\theta_{UKF}}$		$\boldsymbol{\alpha_{EKF}}$		$\boldsymbol{\alpha_{UKF}}$		$\boldsymbol{m_{p,EKF}}$		$\boldsymbol{m_{p,UKF}}$	
	t_a^-	t_a^+	t_a^-	t_a^+	t_a^-	t_a^+	t_a^-	t_a^+	t_a^-	t_a^+	t_a^-	t_a^+
CC	8.47	7.35	9.20	6.72	2.05	1.95	1.96	1.98	0.05	0.10	0.06	0.09
SMH	9.49	7.50	9.04	7.81	2.17	1.81	1.97	1.82	0.05	0.08	0.08	0.03
ITC	9.58	7.51	10.67	7.98	2.31	1.85	2.35	2.15	0.07	0.06	0.07	0.09

the UKF is used. The tracking error for α shows a different trend, with SMH reaching smaller errors ($t_a^+ = 1.81$ for EKF and $t_a^+ = 1.82$ for UKF) compared to CC and ITC. This is consistent also before t_a.

In terms of parameter estimation errors, the UKF based SMH controller is capable of reducing the estimation error from $t_a^+ = 0.08$ to $t_a^+ = 0.03$. On the contrary, CC and ITC deviate from the nominal values, resulting in an error of $t_a^+ = 0.09$ for both of them.

5.5 Discussion

The results presented in this chapter show that it is possible to formulate adaptive intermittent controllers based on Kalman filters for multivariable real-time systems. The Kalman filter framework allows a straight forward formulation based on a state-space representation of the system, which uses a state-vector that is augmented with system parameters as extra states. The self-tuning architecture of these controllers provides flexibility in terms of having independence between the control redesign stage, and the estimation part of the problem. This is also the first time that adaptive intermittent controllers (based on the non-adaptive versions in Gawthrop et al. (2011); Gawthrop and Gollee (2012)) are implemented on a real-time system.

The simulation results show how the proposed adaptive intermittent controllers can be used in situations where some of the physical properties of the plant change with time. These results belong to an extreme case, that can only be implemented in simulation, since the varying parameters that are being tracked by the controllers would never change in such way in a real pendulum. However, the value of the simulation resides in the fact that it shows how the intermittent controllers can provide a feasible solution to changing environments by exploiting the benefits of switching between open and closed-loop. The idea of not applying a control action for a period of time to better understand causality between inputs and outputs has been suggested by (Loram et al., 2011, 2012), arguing that when a control signal is applied continuously it is more difficult to distinguish between the effects of disturbances, motor noise, and parameter variations. In contrast, not applying a control input allows errors to grow exponentially, therefore revealing the possible sources of it and more importantly, clarifying the next steps in terms of correction.

Loram et al. (2011) evaluated this ideas in a visuo-manual tracking task to control a virtual single inverted pendulum by using either continuous or intermittent contact with a joystick. Essentially, the intermittent strategy implemented in their work is based on the the concept of a tap, which is an impulse-like control signal that the subject generates by "tapping" the joystick, decaying quickly to zero, thus applying zero control for the rest of the open-loop period. In fact, this is a feature of intermittent control in general, the production of impulsive actions (determined by the trigger mechanism) helps with the excitation of higher frequencies in the system. This can be seen in Figs. 5.3c, 5.4c, 5.7c, and 5.8c), where the control input of the two adaptive intermittent controllers is shown for the different cases and estimators. The impulse-like behaviour mentioned by Loram et al. (2011) is observed in these figures, whereas the input from the continuous controller is zero once the output reaches the steady-state.

This characteristic is important for estimation purposes, in a way it can be seen as if the adaptive intermittent controllers had a built in mechanism to probe the system in order to obtain more information about it, which relates to the concept caution and probing established by Fel'dbaum (1965) and discussed by Bar-Shalom (1981).

These features come with a cost, the presence of noise, disturbances, varying parameters or even neglected dynamics, influence the system greatly, causing the open-loop behaviour imposed by the generalised hold to differ from the closed-loop response. This difference results in more events, which if seen from the estimation perspective might be beneficial; but it also means that the outputs evolve with a certain degree of steady-state error, as seen in Figs. 5.2d,g. This trade-off can be regulated directly by adjusting the thresholds, where a threshold of zero results in a clock-driven configuration, with events generated at a rate imposed by the minimum open-loop interval Δ_{min}. If Δ_{min} is set to zero, then the response is similar to the continuous control case from Fig. 5.2a.

Another interpretation of this trade-off between a perfect steady-state response and the ability to probe the system to detect possible variations can be made in terms of the *stability-plasticity dilemma* (Carpenter and Grossberg, 1988), which poses the question of how a system can be designed to remain plastic (or adaptive) when facing changes or new environments, while retaining stability gained from previous knowledge. The case of the biological controllers that the human brain implements is a good example of a system that actively regulates this trade-off: achieving adaptation in response to changes in a previously learned task (or even learning a new one), and exploiting stability for when there are no changes and high precision control is needed. In this sense, the adaptive intermittent controllers presented in this chapter are capable of detecting possible changes by tracking differences between a built-in ideal system (i.e. the hold mechanism) and the estimated states, to then generate control signals that constantly excite the system.

The Kalman filter as a joint state and parameter estimator allows the implementation of adaptive intermittent controllers. Certainly, this is not the only way to implement them; but its formulation is consistent with the state-space framework of intermittent control, and provides an intuitive way to implement a state observer and a parameter estimator in one single algorithm. Also, an extended Kalman filter which is based on the linearised version of the nonlinear equations is convenient in terms of its relative simple implementation, and as seen in the simulations, can perform as well as an unscented Kalman filter. This might be a result of the type of system, in this case the rotational pendulum, meaning that the nonlinearities might not be as strong as in other systems. Still, having the alternative to choose between the two configurations

is a powerful advantage, specially when considering that the computational cost of the UKF is of the same order than the EKF (Wan and Van Der Merwe, 2000).

In terms of real-time estimation, the results obtained by using EKF based controllers are not significantly different from those of the UKF, both of them produce very similar responses qualitatively speaking, which was expected. We can argue on the side of ease of implementation then and use the EKF configuration over the more complex UKF; however, based on the estimated parameter results shown in Table 5.3 and in Fig. 5.13b, the estimates that the UKF provided were closer to the nominal values even when the controllers started adjusting their respective control laws. The EKF estimates (as seen in Fig. 5.13a) seem to be more sensitive to the reduced levels of excitation in the control input as a result of the redesigns. In addition, the variance of the EKF estimates is higher compared to the UFK counterpart. These differences suggest that the UKF might be a more suitable solution for the adaptation problem.

The two adaptive controllers proposed in this chapter are different from each other in the way the open-loop behaviour is implemented. This might be related to the differences observed in terms of performance (Table 5.3), where the SMH produced smaller tracking and estimation errors compared to the ITC. It is possible to argue that a better tuning of the ITC controller might have resulted in improved performance; on the other hand, this in itself can be seen as a disadvantage for ITC since the tuning of the SMH controller involves less design variables, with the LQR matrices $\mathbf{Q}_c$ and $\mathbf{R}_c$ being the most influential to the response. The Laguerre function approach on which the ITC bases its open-loop behaviour requires the tuning of two extra parameters: p which controls the shape of the tap and N which is the order of the resulting matrix $\mathbf{A}_p$ defined in (3.39), for this experiment, the selection of these parameters was done by trial and error, since there is no systematic procedure to obtain values that result in the desired response.

The rotational pendulum used for the real-time experiment uses a planetary gearbox driven by the motor shaft. This gearbox is coupled with the pivot of the arm and transmits the torque generated by the motor. This design introduces backlash between the gears and also adds friction to the system, resulting in steady-state errors and oscillations. This nonlinear characteristics were not considered when the controller and the estimators were designed and no active strategy to compensate against them was implemented. Also, the controllers did not have any integral action in the design, which would have contributed to reduce the steady-state errors even further. The oscillations around the setpoint, enhanced by the aforementioned reasons, might contribute positively to the parameter estimation since the control signal is actively trying

to compensate for them. We can speculate that this is the reason why the CC case does not show the behaviour seen in the simulation results of Fig. 5.5, where the CC parameter estimates are updated only when the setpoint changes suddenly, and stay constant when there is no excitation provided by the control input.

The results of these experiments suggest that the intermittent controllers, specially the SMH in combination with a UKF, are viable alternatives to implement adaptation within the IC framework. Moreover, they provide the first real-time evidence of an adaptation process being driven and improved by IC. This improvement is directly associated with the shape and impulsive nature of the control signal generated by the IC. Although this is a simple, but illustrative experiment, further experiments must be carried out in order to confirm this idea.

Chapter 6

Conclusions and future work

This chapter presents the overall conclusions of this PhD thesis, including a discussion, the limitations, and a future work section.

6.1 Discussion

The discussion is presented in terms of the application of intermittent controllers to robotic structures first, to then focus on adaptation in an intermittent control context.

6.1.1 Intermittent control and humanoid robotics

IC has been proposed as an explanation to the *sensory analysis - response selection - response execution* problem that is present in human motor control (van de Kamp et al., 2013a). These ideas provide an explanation to the selection of competing actions in a redundant system with multiple sensory information streams. The argument relies on placing the response planner mechanism within the feedback loop, in a serial configuration, to solve the redundancy problem. This is different compared to the traditionally accepted idea of a planner mechanism working in parallel with the control loop. The authors provide experimental evidence supporting these claims and suggest that such an architecture could be beneficial for other fields such as humanoid and soft robotics.

The results obtained in chapter 4, constitute the first real-time implementation of intermittent control as understood in (Gawthrop et al., 2015), which incorporates the ideas described by (van de Kamp et al., 2013a). Important observations can be made from these results, where the most relevant is related to the observed behaviour of the robot in terms of the outputs. The angles generated by IC are comparable to the ones obtained with a traditional continuous controller. This can be seen specially in group 2

of the first two trials (Figs. 4.12, 4.15, A.2, A.5), although this applies to all trials in general. Focusing on the ankle angles in Fig. 4.12, the average amplitude is less than 0.5 deg for all controllers, the joint angle shift for CC is 0.009 deg, 0.057 deg for ICc, and 0.026 deg for ICe. The mean steady-state error shows also this trend, with 0.214 deg for CC, 0.449 deg for ICc, and 0.539 deg for ICe. For trial 2 in Fig. 4.12, the story is very similar since the mean steady-state error for CC is 0.138 deg, 0.407 and 0.474 deg for ICc and ICe respectively. These relatively small differences were obtained despite the fact that the system switched between open and closed-loop configurations. This result is particularly interesting for roboticists, since in order to perform complicated motion patterns with humanoid robots, multiple optimisation routines and calculations should be performed every time step of the real-time execution. Solving these routines is computationally expensive in most applications. For this reason, the open-loop interval introduced by IC provides extra resources to finish these calculations over multiple time-steps without affecting considerably the overall performance of the system. (Fig. 4.14, Fig. A.4)

Moreover, these results suggest that IC could be applicable to other robotic structures as well, specially for cases where the hardware imposes restrictions in terms of the computational power, limiting the available bandwidth of the controllers. Since industrial robots have been present for many years now, some applications might benefit from a scheme like IC, where the extra computational resources could be used to implement adaptation to counteract performance variations due to wear and ageing of the physical components in a robot.

The high frequency oscillations observed in the high gain groups of trials 1 and 2 (Fig. 4.14, Fig. A.4) suggest that IC might be a robust alternative in situations where the model used to design the controller does not include important system dynamics. The amplitude of these oscillations at the hip, generated by CC, was 4 deg on average for trial 1 and almost 2.5 deg for trial 2, whereas for ICc the values were 1.2 deg in trial 1 and less than a quarter of a degree for trial 2. The CC and IC controllers were based on a model that did not consider the full actuation principle in TUlip, resulting in effects that were amplified during the high gain cases. However, IC seems to be affected much less compared to CC. This idea still needs to be validated in a simulation environment and tested in other systems in order to generalise the result. Nevertheless, the evidence presented here can be used as a starting point to prove this feature of IC.

The experimental work on TUlip allowed us to evaluate IC in the presence of perturbations. More work should be carried out in order to evaluate the range of allowable

disturbances before a fall is imminent. Also, the location of the perturbation was not evaluated, since it was always applied from the front of the robot in the from of a pull. This decision was related to the fact that the model used to design the controllers assumed that TUlip was a three-segment inverted pendulum. In order to evaluate different types of perturbations such as lateral pulls or pushes, a model that considers both legs and the involved degrees of freedom should be generated. Similarly, there are many areas in which IC could be improved to make it a better fit for motion control in robots. For instance, classic robotic measurements could be used as additional sources of information to generate events or to modulate the control signal, such as the *centre of mass position* or the *instantaneous capture point (ICP)*, which defines the place on the ground where the robot has to take a step instantaneously to avoid a fall. If the ICP is kept within a defined area known as the *support polygon* (de Boer, 2012), then a control method like IC can be used to generate joint torques to maintain balance, if the ICP moves outside the polygon, then a step must be taken by the robot. This information might be used to switch to a walking strategy or to a different type of controller.

A more detailed analysis on the effects of the different IC parameters such as delays and thresholds should be carried out to fully explore the limitations of this method. Although theory and intuition tells us that the performance would deteriorate as they grow, an evaluation should still be made in order to find out the range of admissible values, since this has a direct impact on the open-loop intervals.

6.1.2 Adaptation in the context of intermittent control

Adaptation is a central part of our daily lives. It provides a basic mechanism of survival and it leads to learning new skills and behaviours. Scientists and engineers have been trying to formulate models that capture this complex phenomenon for many years, both experimentally and theoretically. In this sense, different models have been used to explain human motion from a computational point of view (Craik, 1947; McRuer and Jex, 1967; Johansson et al., 1988; Kooij and Vlugt, 2007; Loram et al., 2009, 2011; van de Kamp et al., 2013a; Gollee et al., 2017), with many of them relying heavily on control engineering concepts. In particular, the results in (Loram et al., 2011; Gawthrop and Gollee, 2012; Gawthrop et al., 2015) posed interesting questions on the applicability of IC to the adaptation problem, and the possible benefits of this framework in the context of human motor control and engineering systems in general.

The adaptation results presented in this thesis were greatly motivated by the aforementioned ideas and they constitute the first implementation of AIC in a real-time

environment. Two different scenarios were evaluated: a time-varying case where physical parameters of a rotational pendulum were modified in order force the adaptive controllers to adjust accordingly, and a case where the controller design assumed a model with a parameter that was considerably different from the nominal or *true* value, to then identify the system and redesign the controller. The main observation from these results is the fact that the hybrid nature of IC (i.e., alternating between closed and open-loop behaviour) provides a level of balance in terms of how capable the controller is to react to parametric changes and how close it keeps the outputs to their references. This feature was not always observed in the continuous adaptive controllers that were used as a comparison, since they reduce the output error and the control effort to the minimum, which makes the scheme unaware or insensitive of parametric variations until more control effort is applied to excite the system or until the effects of the variations have an effect in the dynamics. This effect is observed clearly in Fig. 5.9, where the IC controllers converge to the real parameter value after 2.5 sec for the mass m_p and after 5 sec for the pendulum length l_p. The CC case reaches m_p and l_p after approximately 8 sec.

The two scenarios discussed above were evaluated using two different versions of IC: *tapping IC* and *system-matched IC*, which work similarly except for the way the open-loop behaviour is generated. It has been argued by Gawthrop and Gollee (2012) that tapping IC might be helpful for systems with friction and backlash since the shape of the control signal can be modified by adjusting the Laguerre functions behind the open-loop behaviour. However, the results from the real-time experiment in chapter 5 showed that steady-state error of the system-matched, UKF based IC was 7.81 for the arm angle θ and 1.82 for the pendulum angle α, which are slightly smaller compared to those of tapping IC (7.91 for θ and 2.15 for α). The parameter estimation error for m_p shows a similar trend, where tapping IC had an error of 0.09 and the system-matched IC value was 0.03, being the only controller capable of reducing the error once adaptation was enabled.

The joint configuration of the Kalman filters allowed us to perform state and parameter estimation using only one algorithm, and to extend the IC framework to include nonlinear estimation techniques. In the real-time experiments, the unscented version of the filter resulted in parameter estimates that were closer to the true values with less variance compared to the extended version, specially after the redesign was enabled.

Further testing of AIC on other engineering platforms is needed to fully grasp the potential of this technique. The combination of AIC with constrained versions of intermittent control would provide a robust solution engineering systems. A constrained

solution is always desirable in the sense of keeping the outputs, states, and inputs of the system under predefined values (Mayne et al., 2000; Gawthrop, 2004; Gawthrop et al., 2013). This requirement, in combination with the hardware capabilities and the dimensions of the system at hand, might impose strict timing requirements to obtain feasible control inputs. Continuous redesign (every sampling instant) of the controller, as in adaptive CC, would contribute negatively to this bottleneck. The adaptive intermittent controllers provide time in the form of open-loop intervals that could be used to solve optimisation routines while redesigning the controller only when events are detected.

6.2 Conclusions

The overall aim of this research was to formulate multivariable adaptive intermittent controllers for real-time structures and to investigate whether intermittent control could be applied to multi-segmental, autonomous, humanoid robots. The results obtained from the experiments in this thesis allow us to establish the following conclusions:

- IC is a viable alternative to traditional control methods in the field of humanoid robotics. For a balancing task, its application resulted in output levels that were comparable to the ones obtained with continuous predictive control, with the added benefit of providing extra computational resources through the open-loop interval, while rejecting small perturbations.

- The results from the adaptation experiments suggest that the typical control signal that IC generates and its triggering mechanism provide a balanced solution in terms of system excitation and steady-state error compared to a continuous adaptation approach.

- Adaptive intermittent control, based on the system-matched hold, generated smaller steady-state and parameter estimation errors when compared to the tapping hold method.

- The Kalman filtering framework was used as the basis to implement state and parameter estimation routines. The adaptation experiment shows that the use of unscented Kalman filters results in smaller steady-state and parameter estimation errors when compared to the extended Kalman filter. This suggests that the way in which the unscented version deals with the system nonlinearities is better suited for adaptation environments where parametric uncertainty is present.

As an overall conclusion, the controllers that were used in this thesis could help in the investigation of neural mechanisms behind adaptation in a human motor control context, while being applicable also to generic engineering systems.

6.3 Limitations

Although intermittent control was applied on different real-time multivariable systems, showing interesting properties and results, there are some limitations about the framework that were identified, which are discussed below:

- IC is a model based technique, meaning that in order to have a working controller, a reliable and accurate model of the system should be used. The adaptive intermittent controllers introduced in chapter 3 aim to reduce the problems that are generated when the model of the system is not accurate or even partially unknown; however, they rely on some degree of knowledge about the system in order to be implemented. In reality, having a good model of the plant is not always feasible and in some scenarios it is not possible to capture the entire range of complex dynamics behind most applications. Therefore, for a real-time implementation, time and effort should be destined to obtain a detailed model of the system or to perform system identification tests.

- The IC framework is based on a linear state-space representation of the system. This means that the stability is guaranteed only if the linearity assumptions are not violated and the system always operates within the considered range. Many systems can be approximated by a linearisation approach given that the operating regimes are well known; however, if the system starts operating outside the assumed range, or close to the boundaries, the performance would decrease considerably. Many techniques are available to deal with these limitations, one of them is the unscented Kalman filter described in chapter 3, which attempts to reduce the effects of the neglected high order derivative terms introduced by the traditional linearisation process of the extended Kalman filter. Even though the use of such a filter would be beneficial in some situations, the rest of the IC framework is still formulated exclusively in linear terms.

- The AIC scheme shown in this thesis exploits the advantage of a control signal that behaves in an impulse-like fashion. As argued in chapter 5, such a control input helps to excite the system constantly, making it more aware of possible changes in its parameters. On the other hand, this characteristic might not be

ideal for real-time systems with tight saturation limits or with actuators that are sensitive to abrupt changes in the control signal.

- The adaptation controllers described in chapter 3 belong to the class of parametric adaptation schemes. Such controllers can handle parameter uncertainty or variations only, but would not perform well for systems that experience structural changes with time. A more complete version of AIC should compensate for both parametric and structural changes efficiently.

6.4 Future Work

The work carried out during this PhD project resulted in control schemes that could serve as the basis for other intermittent adaptation mechanisms or even learning intermittent control. Some follow up ideas that could be explored based on the results presented in this thesis are:

- Formulation of AIC using a direct self-tuning architecture.
- Formulation of AIC based on multiple model structures.
- Evaluation of the explanatory power of AIC as presented in chapter 3 in human motor control tasks.
- Evaluation of power consumption and computational features associated to IC.
- Performance evaluation of AIC in humanoid robotic structures.

6.4.1 Formulation of AIC using a direct self-tuning architecture

The adaptive schemes in this thesis are formulated as indirect self-tuning controllers which provide the advantage of decoupling the control and estimation stages. This gives flexibility in terms of testing and implementation. On the other hand, direct self-tuning adaptive controllers might reduce the number of involved computations since an explicit design stage is not required. The parameter estimation procedure is implemented in such a way that it provides the controller parameters that should be adjusted directly, instead of the model parameters, leading to simpler control formulations. This particular scheme has been related to reinforcement learning algorithms previously (Sutton et al., 1992), suggesting a possible path to establish learning adaptive controllers that exploit this architecture.

6.4.2 Formulation of AIC based on multiple models

Multiple model schemes have been formulated in the past as an alternative to systems with parametric and structural changes, and for situations where the variations in the system model are large and fast (Narendra and Balakrishnan, 1997; Landau et al., 2011). The basic idea behind this approach relies in the formulation of a network or bank of models, which are selected based on the measurement of the system inputs and outputs, and an error criterion. Different controllers can be formulated for each of the models in the bank. A fixed model is selected in real-time by a switching mechanism and the corresponding predefined control law is applied. At this point, parametric adaptive control methods, like the ones presented in this thesis, can be applied to further improve the parameters of the selected controller. The parameters of the selected model can be updated if the performance improves in terms of the error criterion.

This scheme would cover a wider range of possible parameter variations and changes in the operating conditions of the system. At the same time, it would require a higher degree of knowledge of the process that is being controlled, previous testing, and the derivation of controllers for all the models in the bank. Some difficulties can be introduced by the mechanism that switches between models, since a smooth transition might not always be feasible depending on the variations of the plant.

6.4.3 Explanatory power of AIC in human motor control

Many of the human motor control experiments discussed in this thesis involved some form of adaptation and learning. A possible future project could use AIC to evaluate how well they explain the experimental data in comparison with traditional continuous schemes.

The visuo-manual task of balancing an virtual inverted pendulum using a joystick, as in (Loram et al., 2011), is a good starting point to test AIC in a well defined environment. Another possibility is to apply them to a reaching task under the influence of a variable force field as presented by Gawthrop et al. (2015).

6.4.4 Power consumption and computational features of IC

It has been argued in this thesis that IC can provide extra computational resources by enforcing an open-loop intermittent interval. However, this has not been directly

quantified in terms of number of operations and the efficiency of the algorithm. Similarly, the power consumption associated to the IC input signals, which might be higher in amplitude compared to a continuous approach, has not been studied in detail for complex real-time systems. A thorough quantitatively analysis is needed in order to evaluate the tradeoff between increased computational resources and energy consumption caused by the impulsive nature of the IC signals.

6.4.5 Performance evaluation of AIC in humanoid robotics

The evaluation of multivariable IC in the field of humanoid robotics was presented in chapter 4, showing promising results in terms output performance while providing extra computational resources introduced by the open-loop intervals. However, having accurate models is not a simple task for highly complex robotic structures. It is a common problem in robotics to formulate controllers for systems with parametric uncertainty and in some cases, with un-modelled dynamics. AIC could be used to reduce the levels of uncertainty in such systems based on approximate models.

Appendix A

Figures chapter 4

The following figures, show the joint angle, control input (reference torque) and open-loop interval for the ankle and hip, during trials 2, 3, 4, and 5, as shown in Table 4.3.

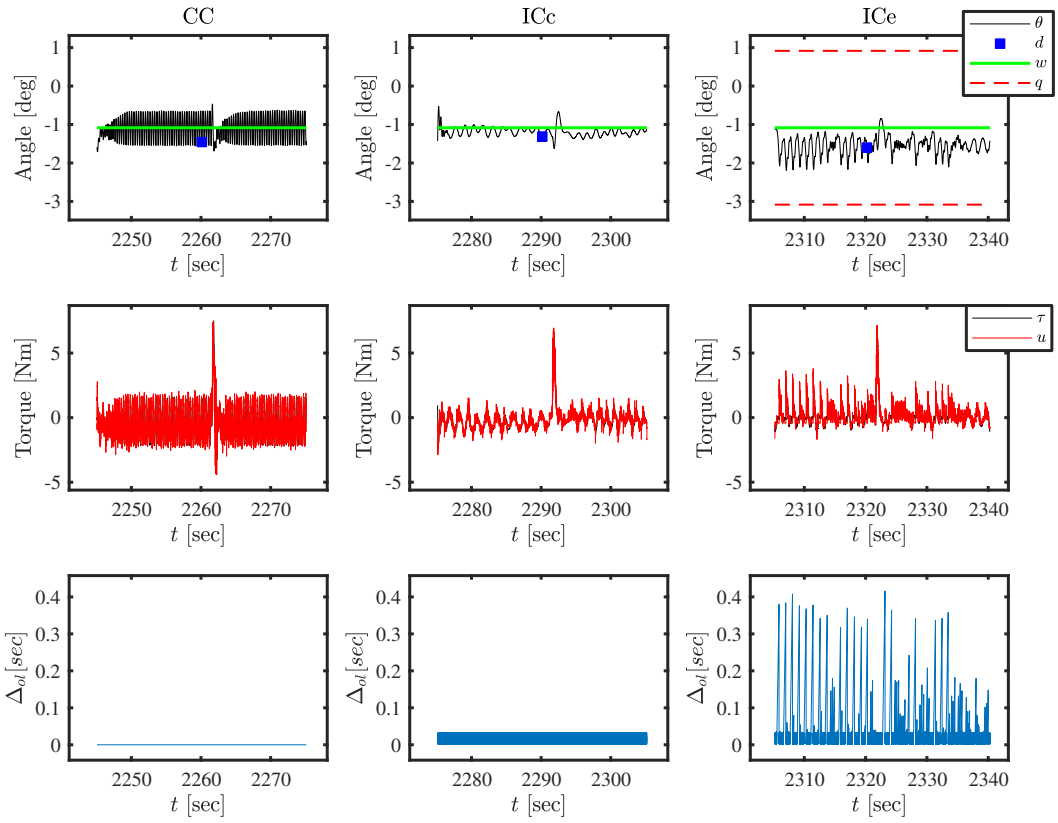


Figure A.1: Trial 2 - Group 1 - Ankle data.

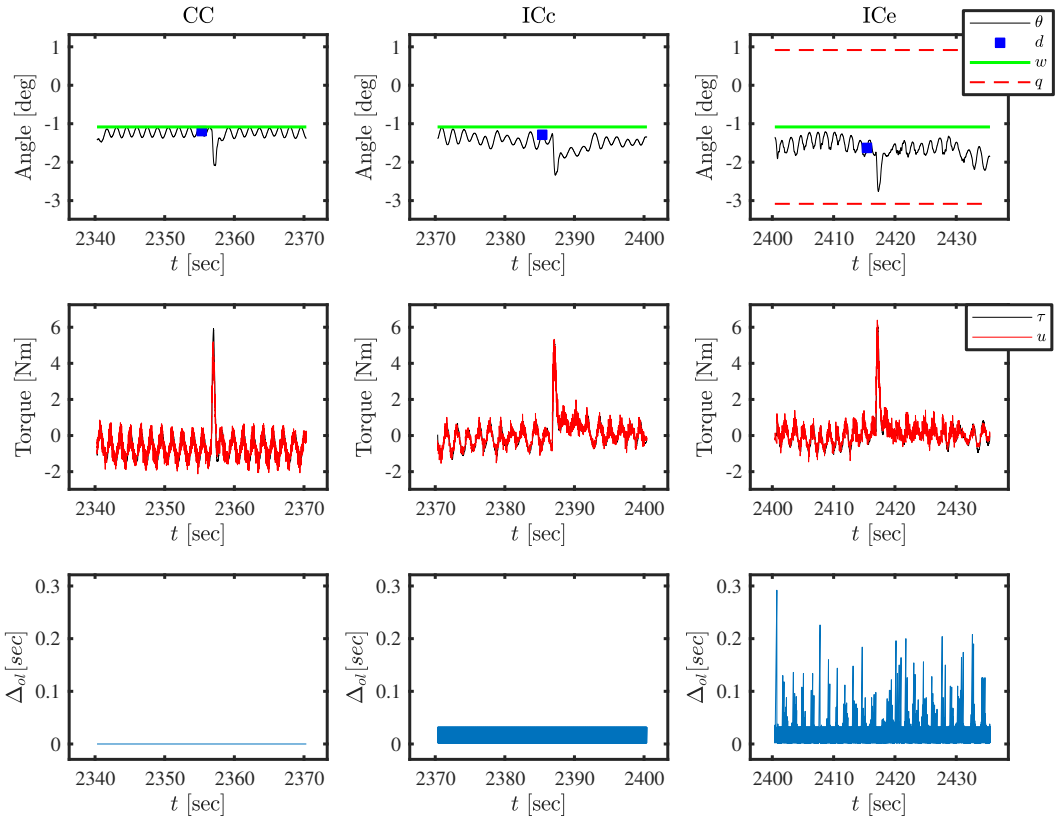


Figure A.2: Trial 2 - Group 2 - Ankle data.

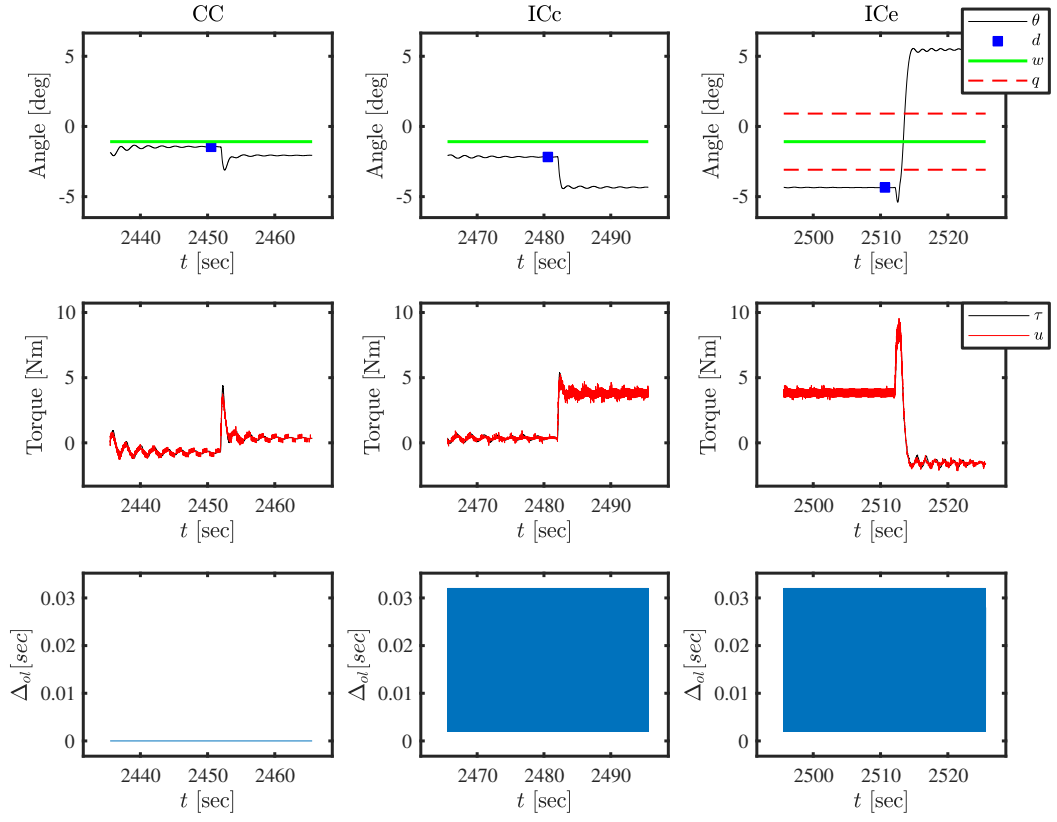


Figure A.3: Trial 2 - Group 3 - Ankle data.

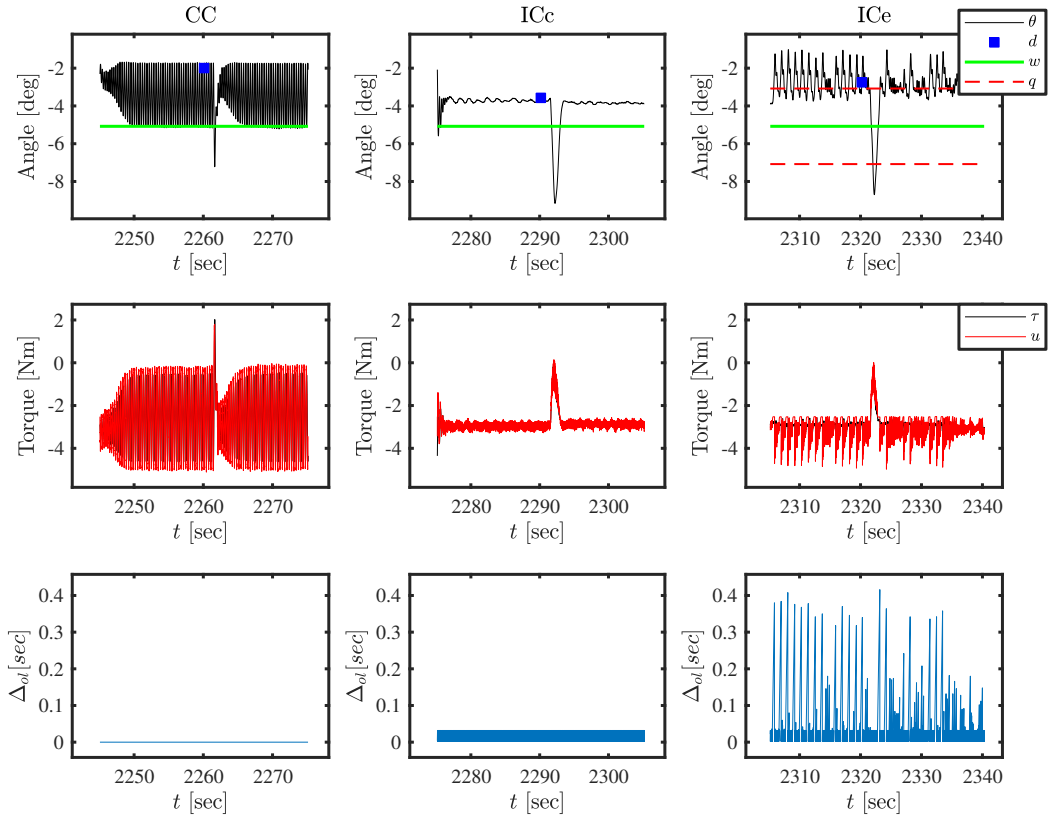


Figure A.4: Trial 2 - Group 1 - Hip data.

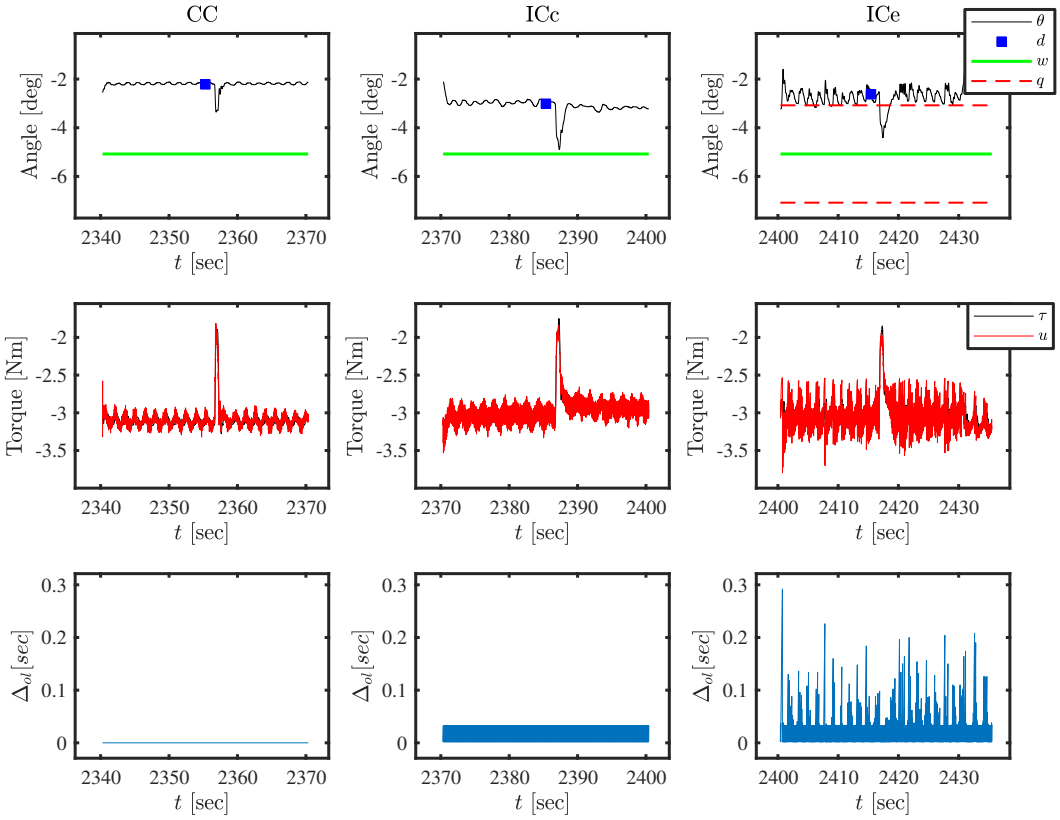


Figure A.5: Trial 2 - Group 2 - Hip data.

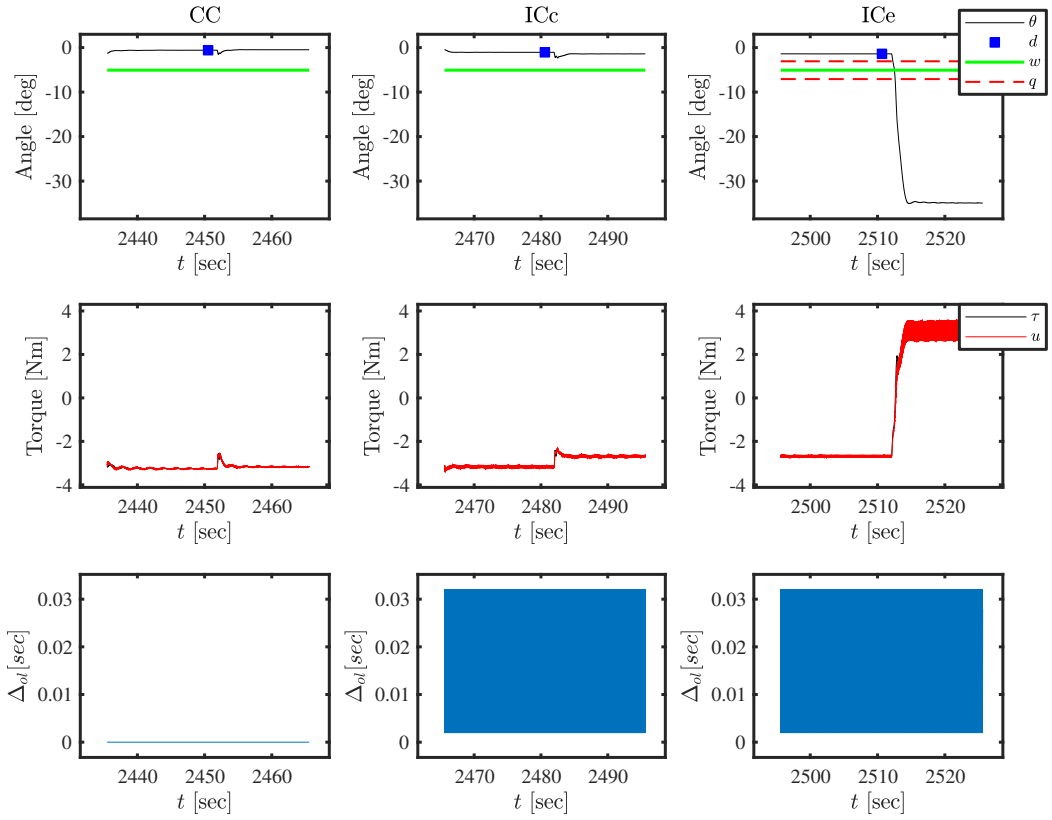


Figure A.6: Trial 2 - Group 3 - Hip data.

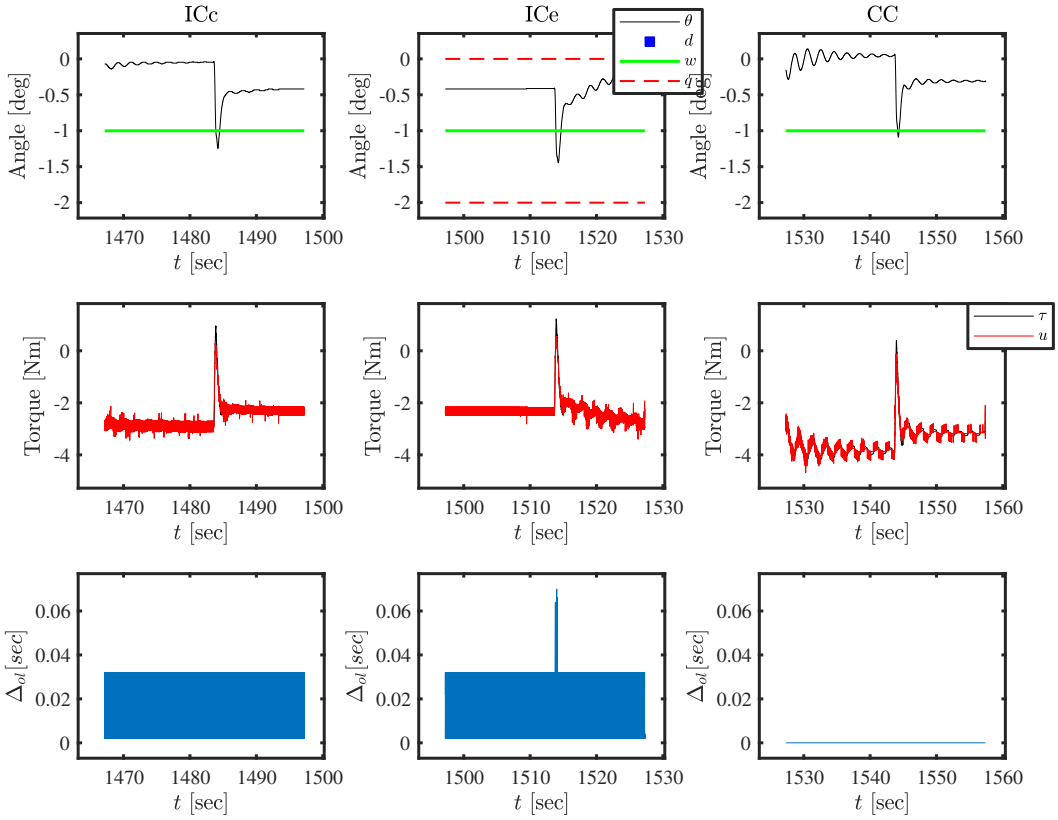


Figure A.7: Trial 3 - Group 1 - Ankle data.

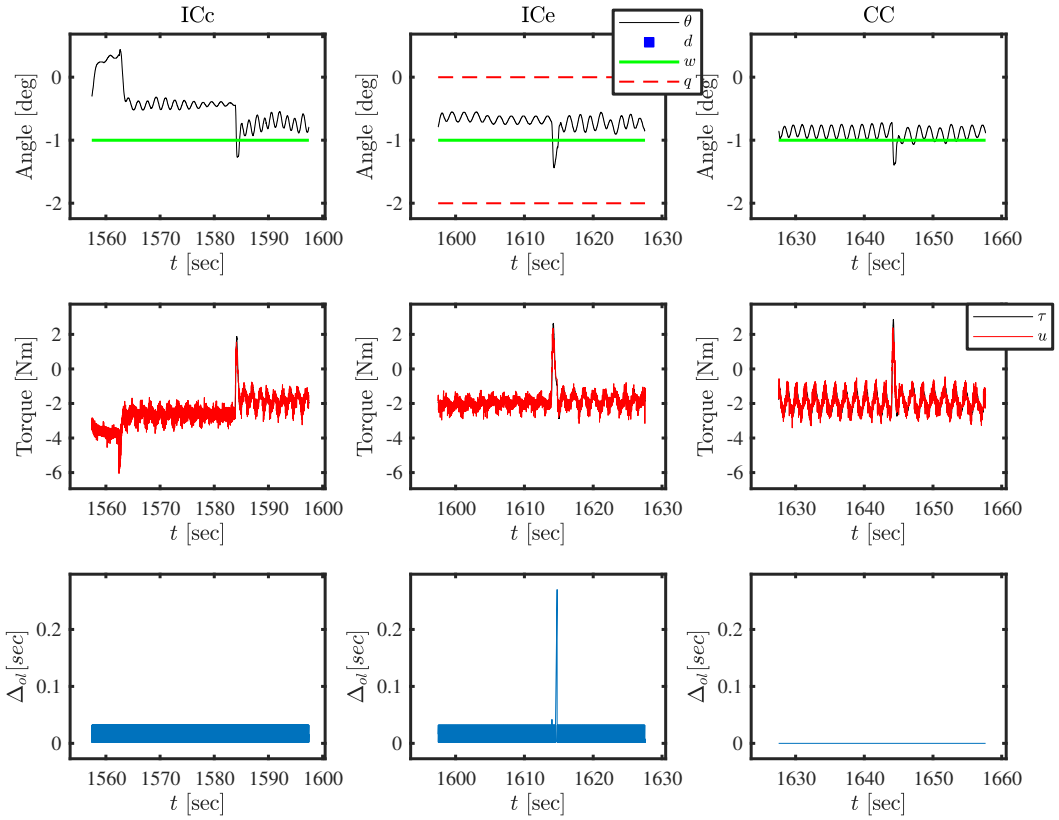


Figure A.8: Trial 3 - Group 2 - Ankle data.

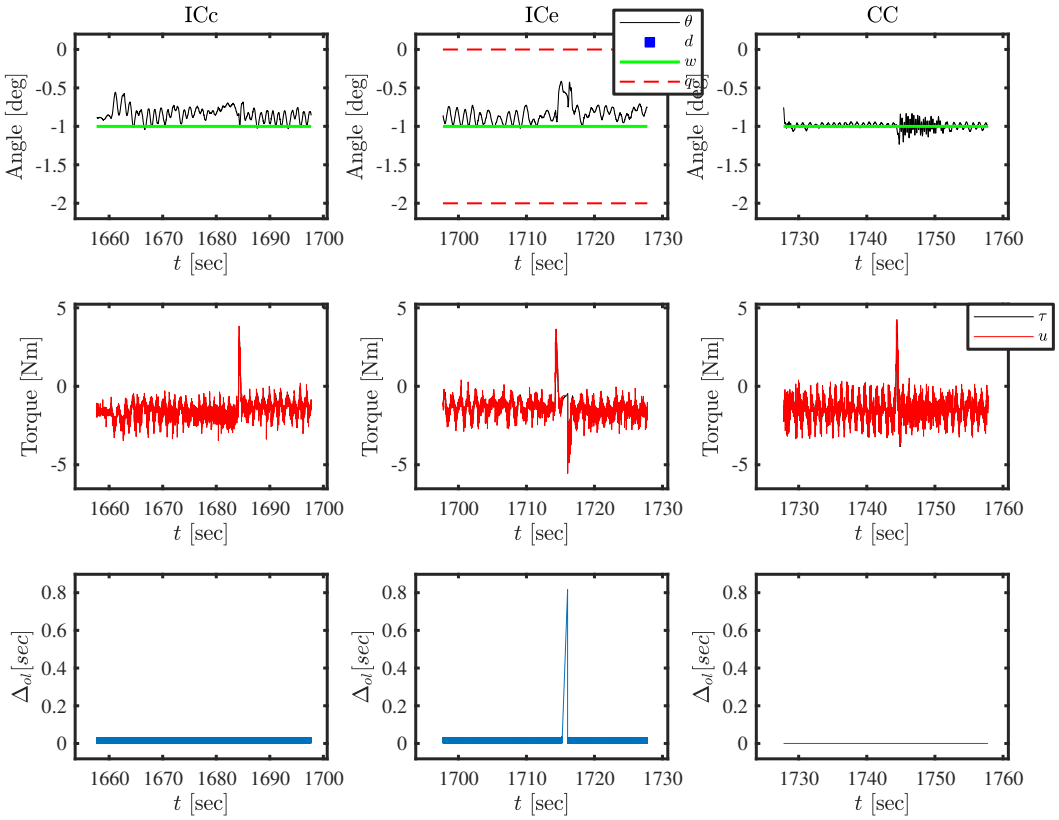


Figure A.9: Trial 3 - Group 3 - Ankle data.

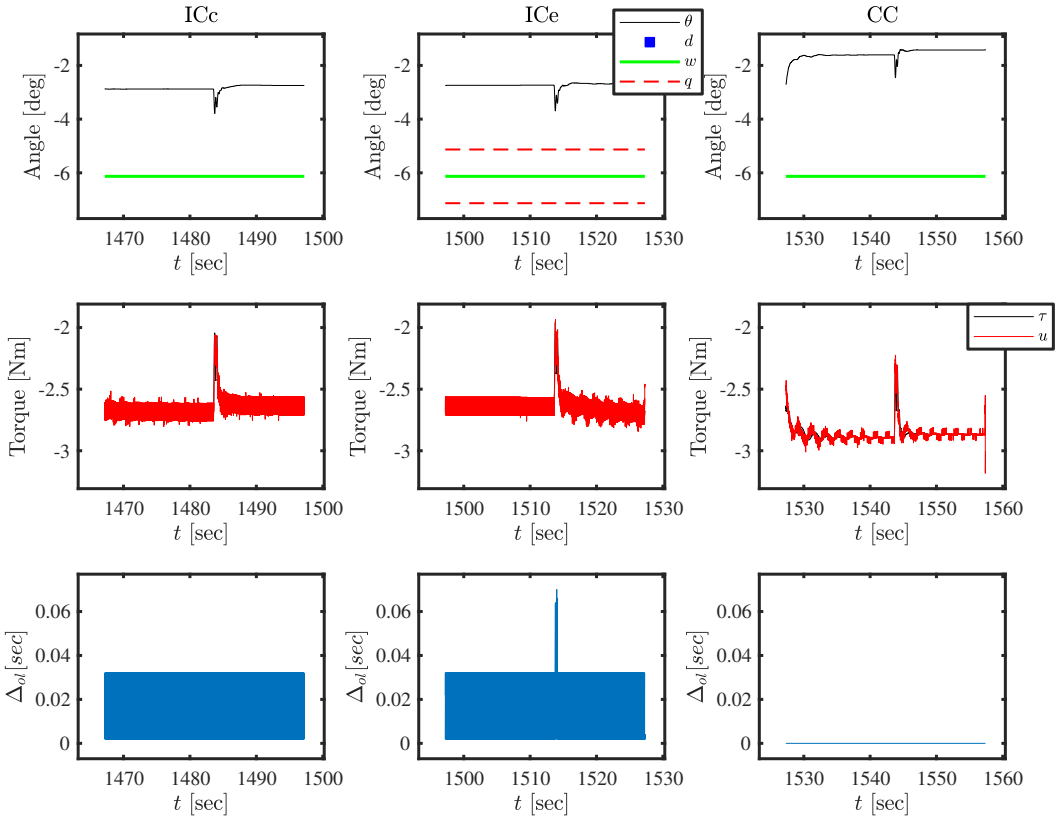


Figure A.10: Trial 3 - Group 1 - Hip data.

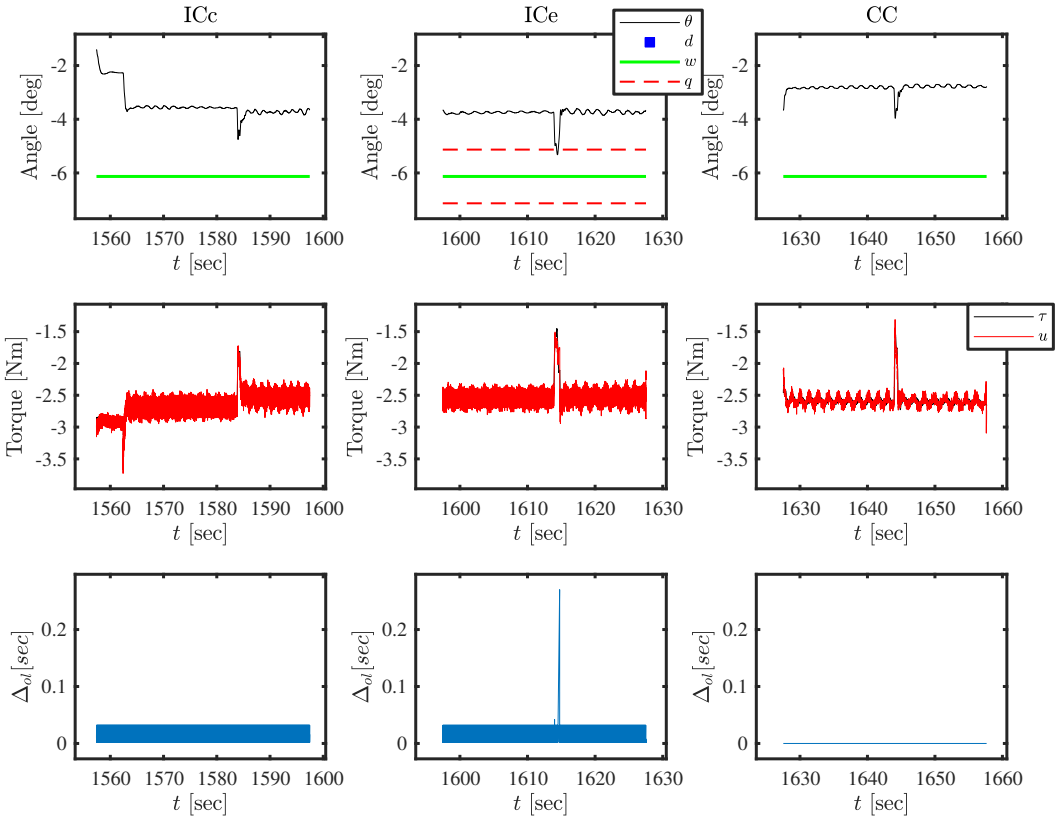


Figure A.11: Trial 3 - Group 2 - Hip data.

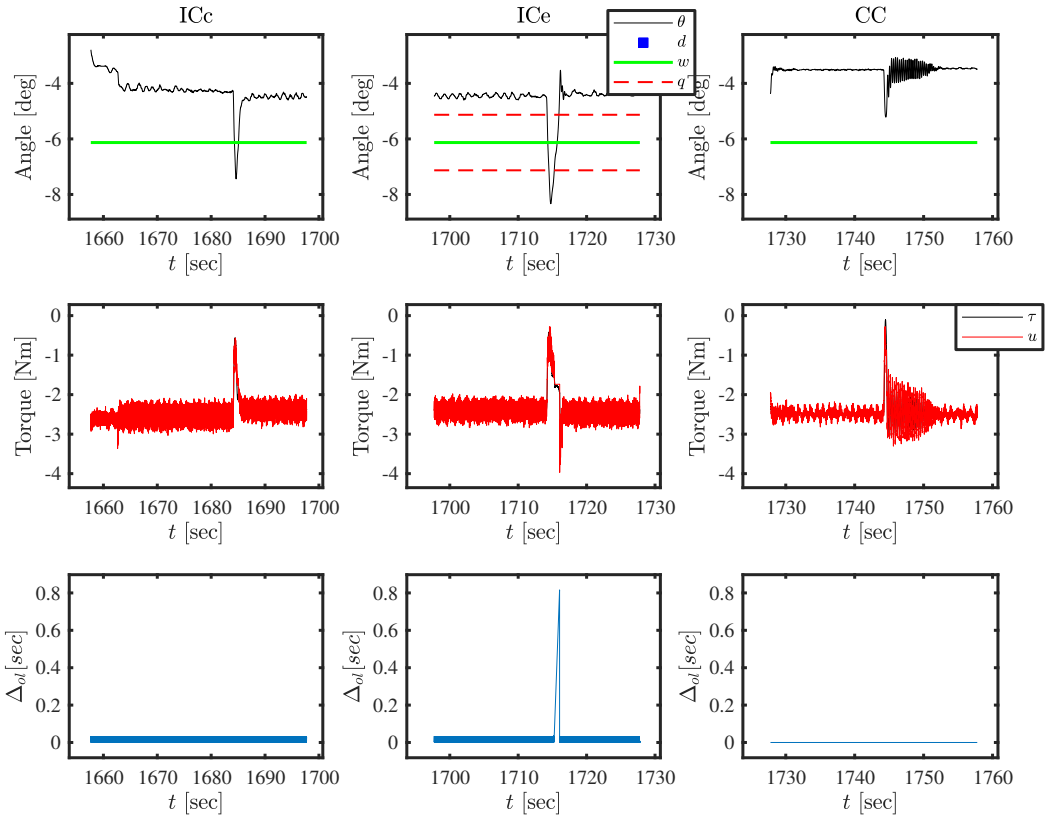


Figure A.12: Trial 3 - Group 3 - Hip data.

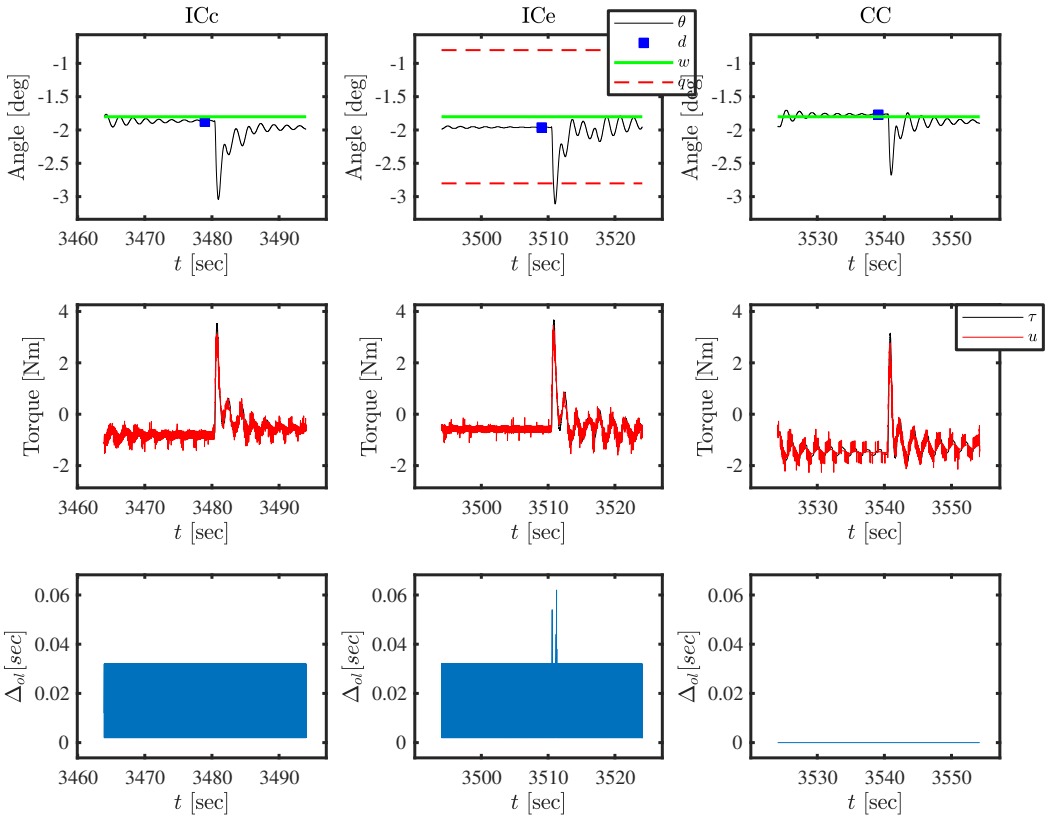


Figure A.13: Trial 4 - Group 1 - Ankle data.

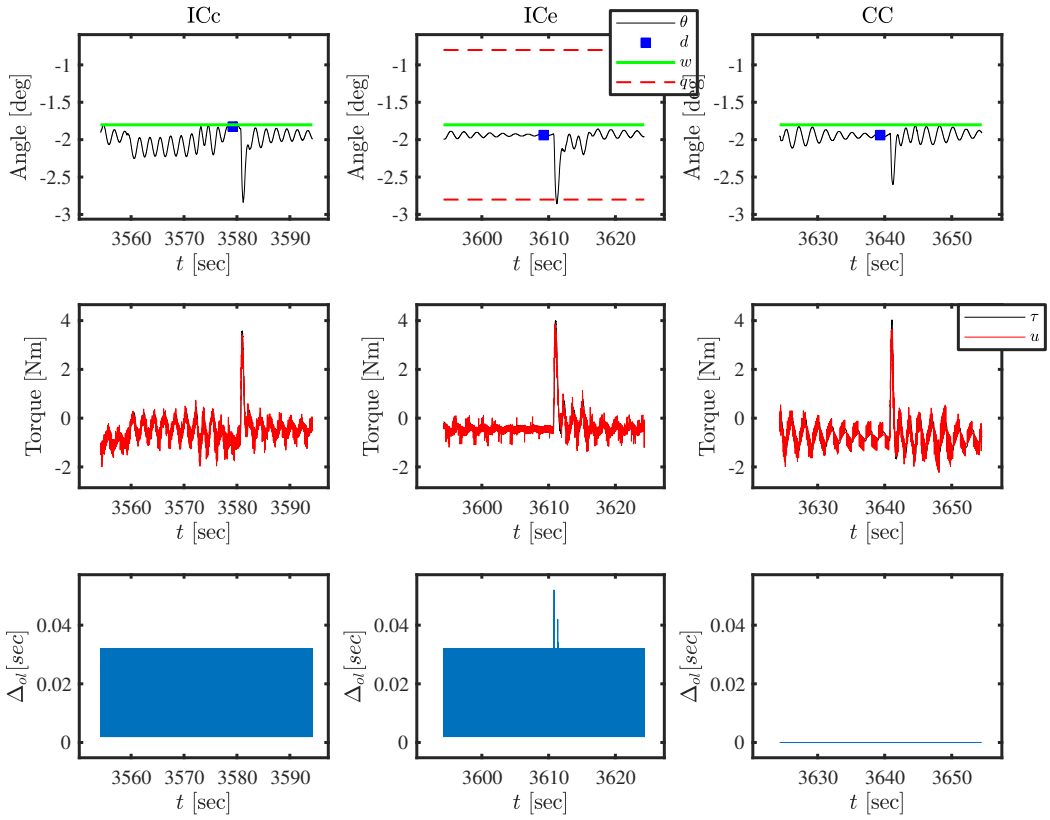


Figure A.14: Trial 4 - Group 2 - Ankle data.

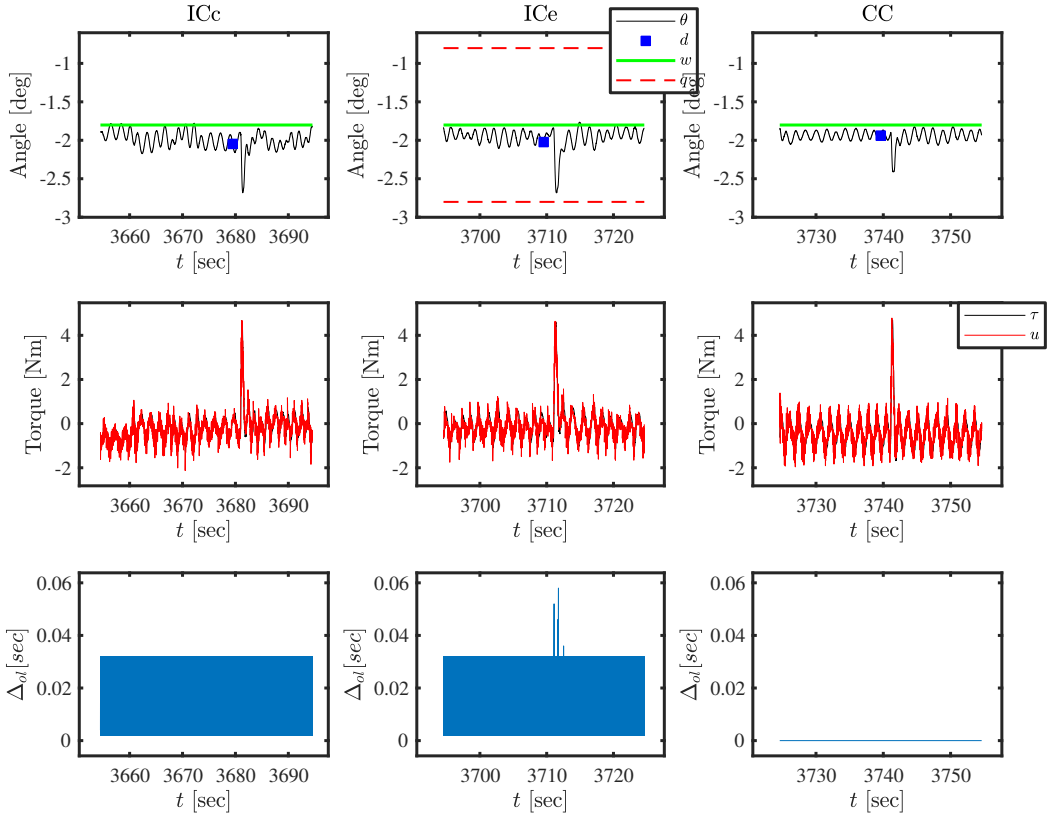


Figure A.15: Trial 4 - Group 3 - Ankle data.

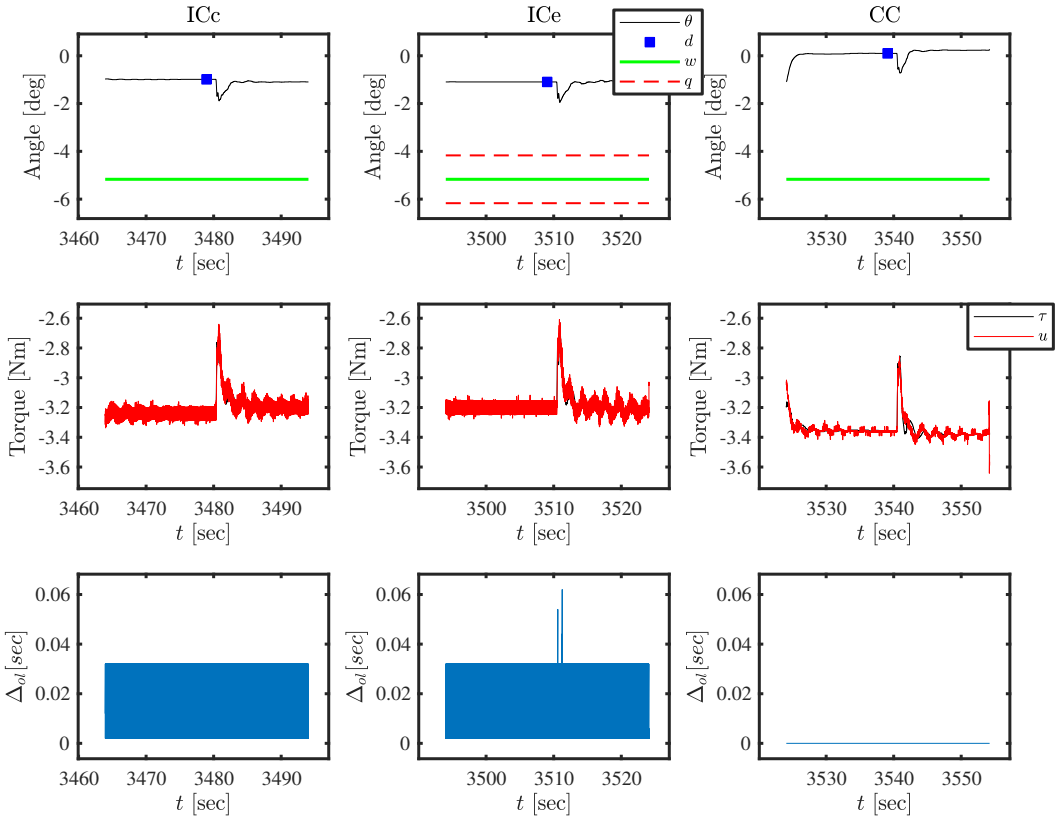


Figure A.16: Trial 4 - Group 1 - Hip data.

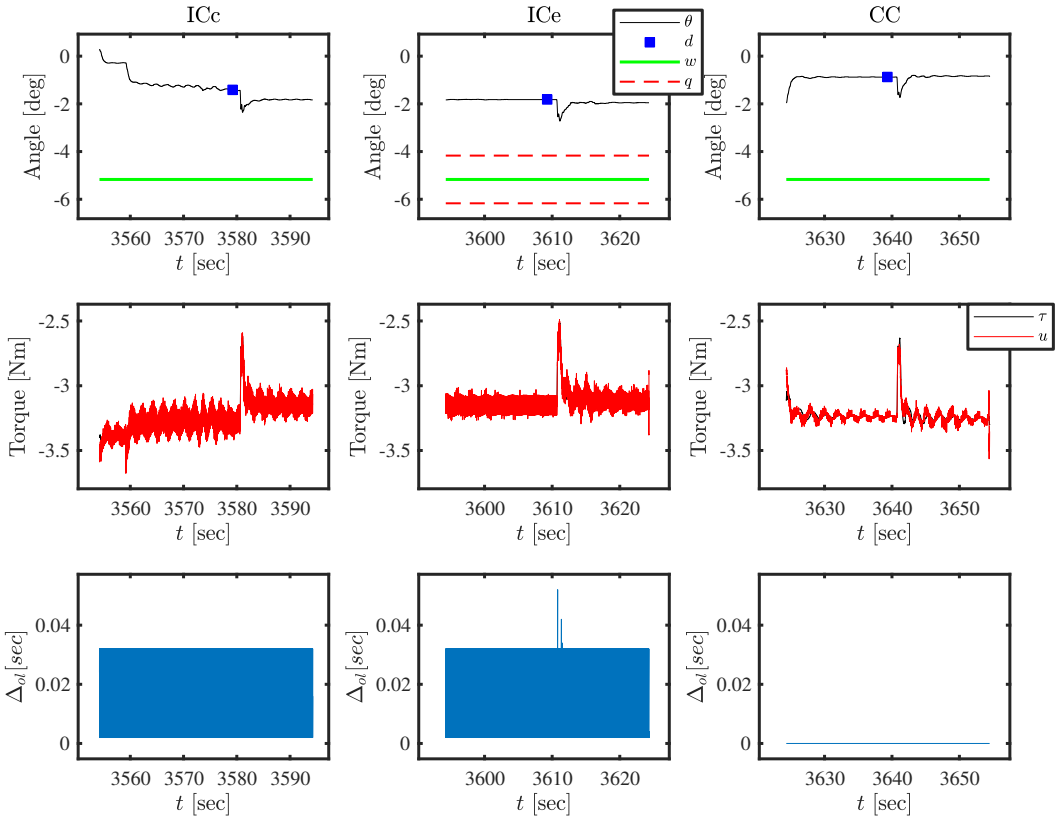


Figure A.17: Trial 4 - Group 2 - Hip data.

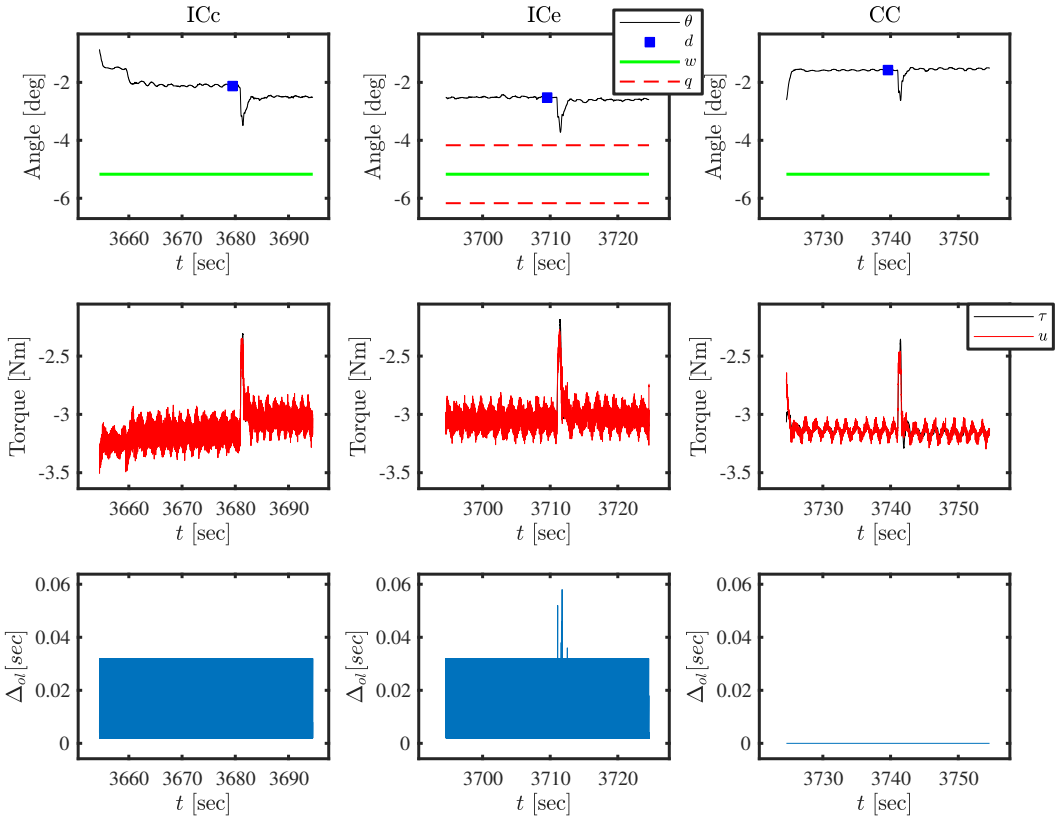


Figure A.18: Trial 4 - Group 3 - Hip data.

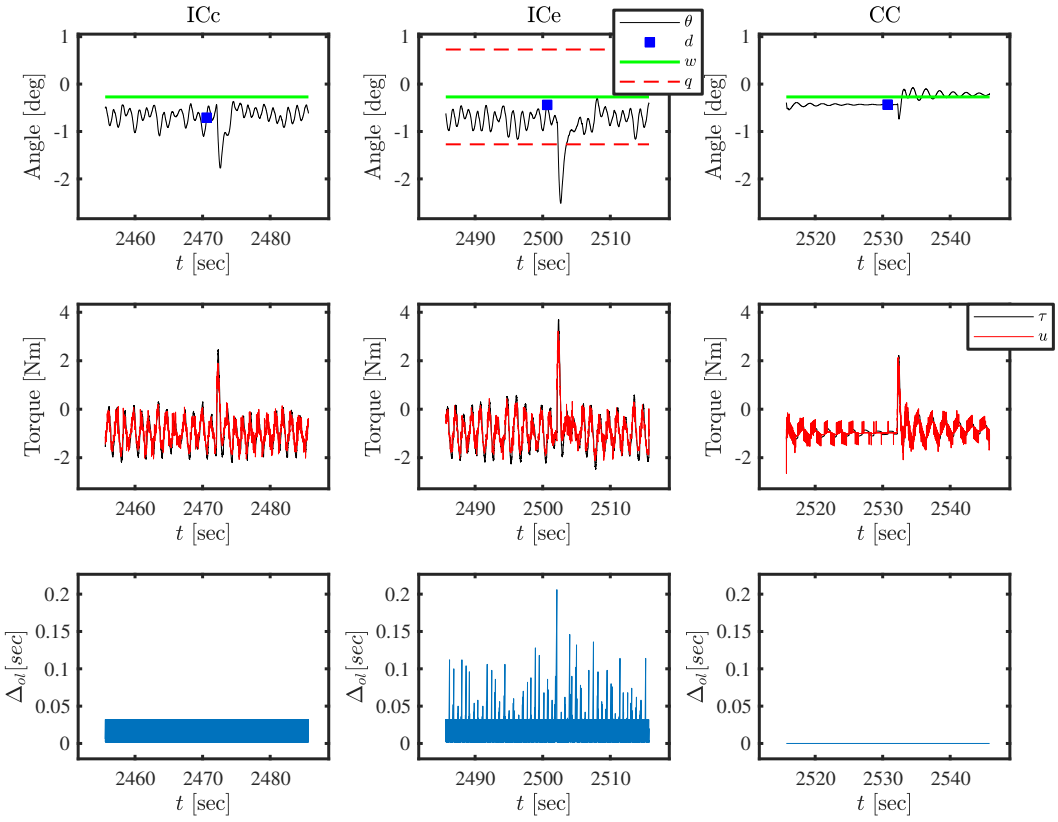


Figure A.19: Trial 5 - Group 1 - Ankle data.

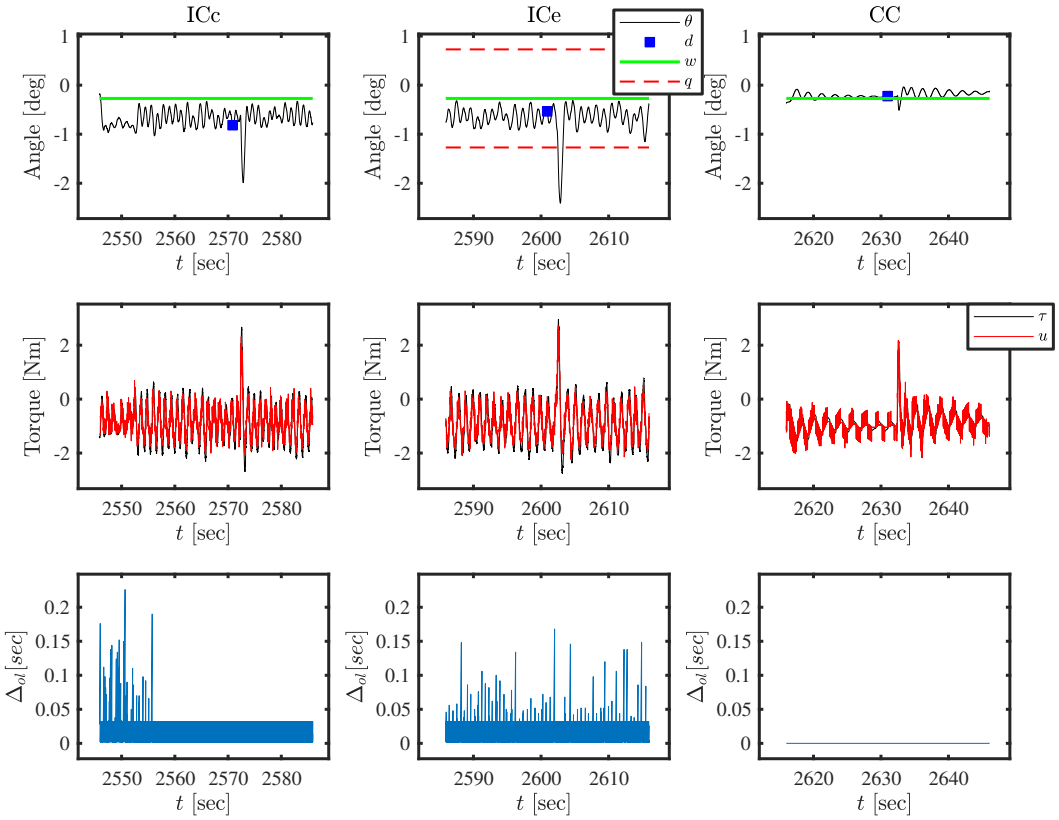


Figure A.20: Trial 5 - Group 2 - Ankle data.

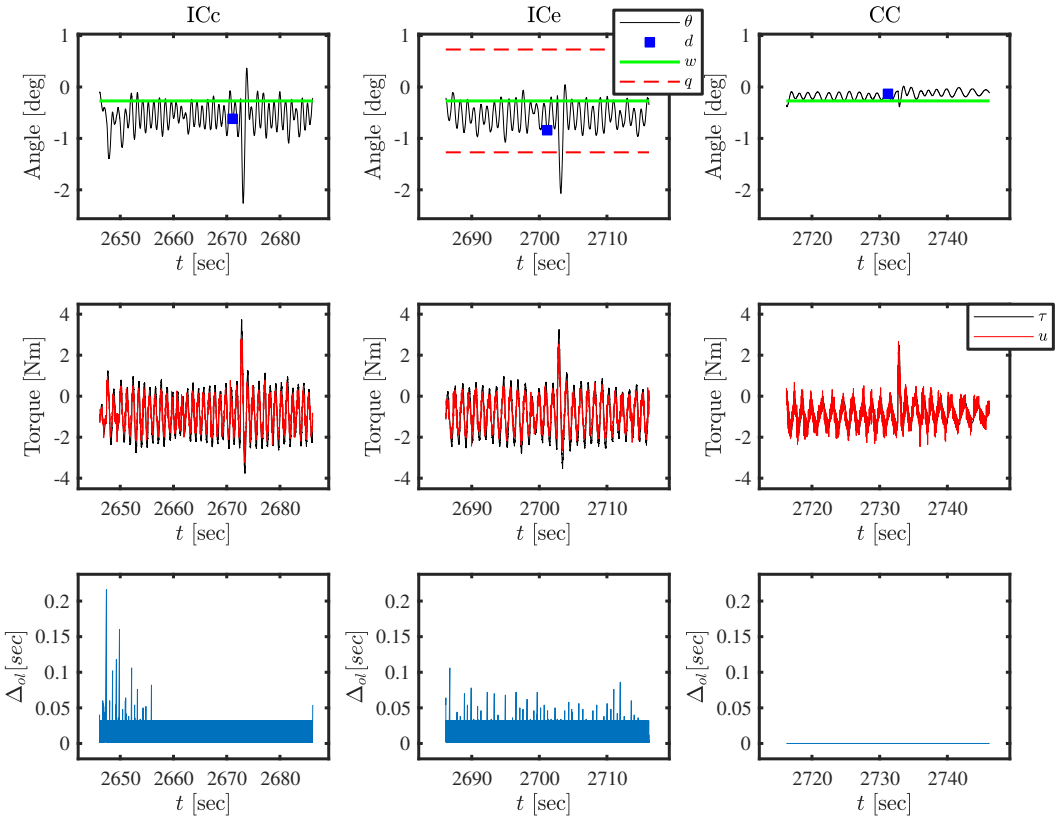


Figure A.21: Trial 5 - Group 3 - Ankle data.

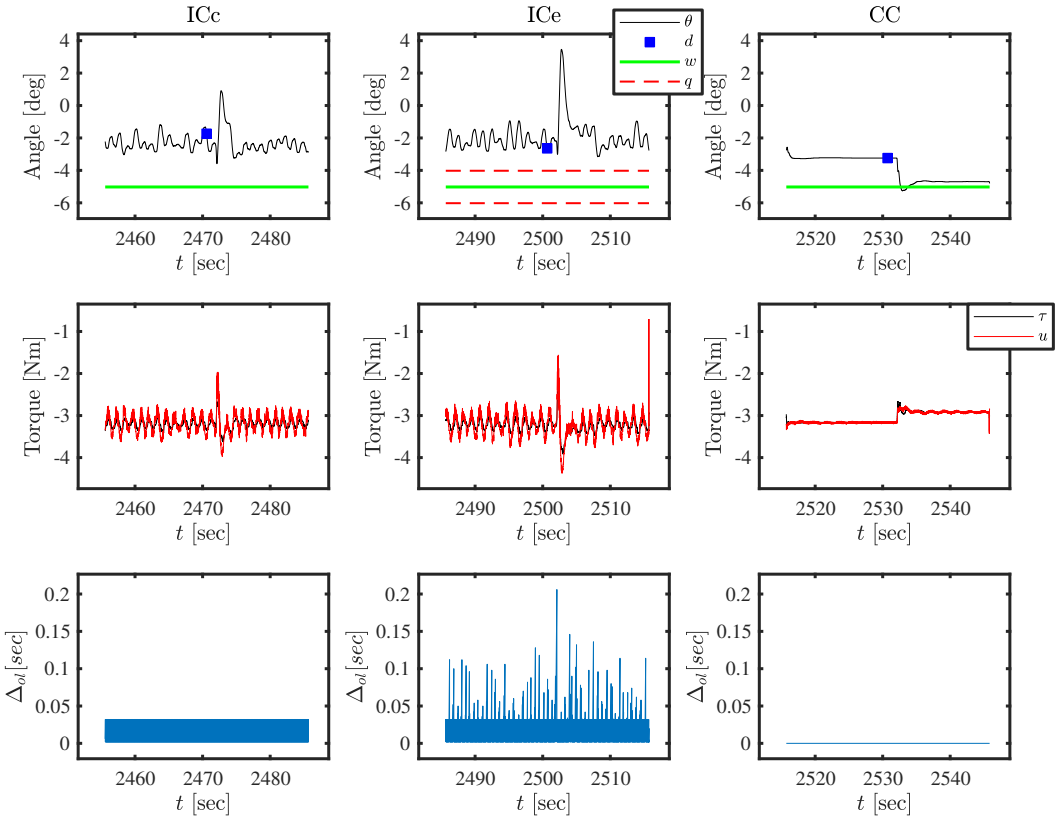


Figure A.22: Trial 5 - Group 1 - Hip data.

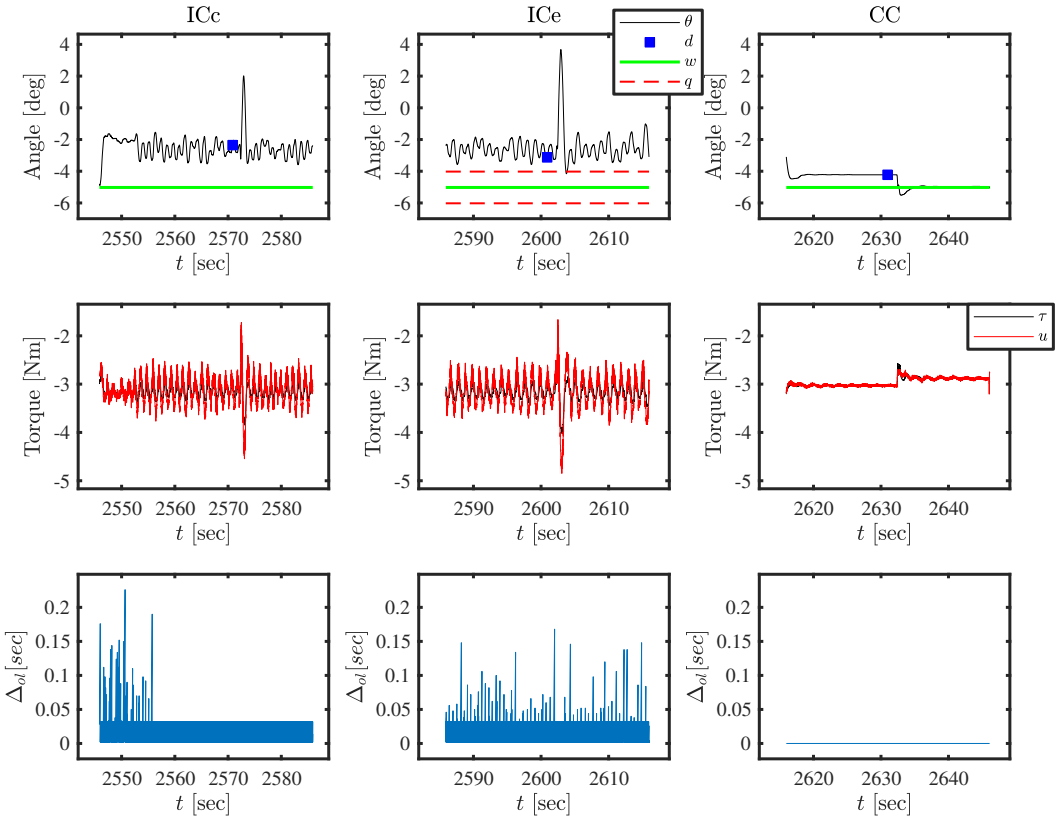


Figure A.23: Trial 5 - Group 2 - Hip data.

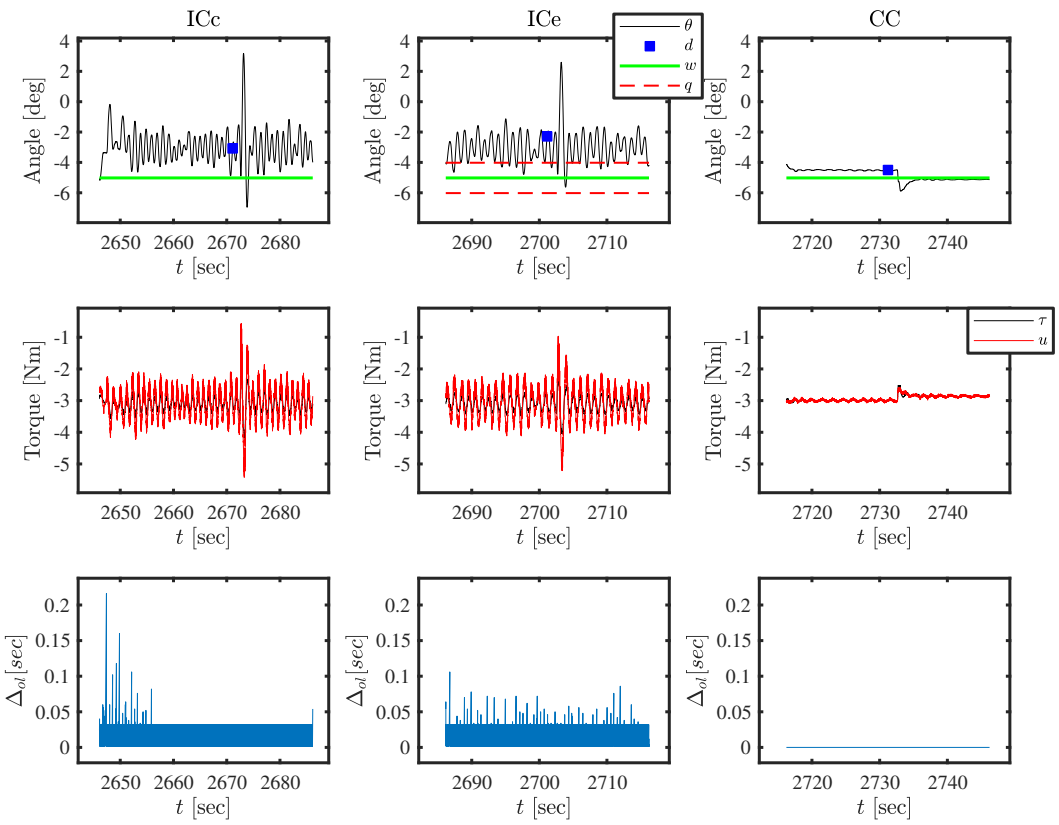


Figure A.24: Trial 5 - Group 3 - Hip data.

Appendix B

Recursive Newton-Euler algorithm

Equation (4.14) can be solved efficiently by means of a recursive Newton-Euler algorithm. These solutions are based on the idea that rigid bodies have six degrees of freedom, and that the motions and forces acting on these bodies can be described using a form of six-dimensional vectors called *spatial vectors* (Featherstone, 2008). This description allows a problem formulation that suitable for recursive solutions and that reduces the number of operations compared to traditional three-dimensional methods. The following paragraphs will provide simple description of the equations involved in the algorithm; however, an in depth, formal introduction to this topic can be found in (Featherstone, 2010a,b).

The joint torques τ_j that would generate a set of desired joint accelerations $\ddot{\boldsymbol{q}}$ can be computed recursively for robotic mechanisms defined by a kinematic tree. These can be modelled as a set of links numbered from 1 to N, a fixed base link, and corresponding joints that connect the links. For instance, joint i connects from link $\lambda(i)$ to link i, considering the link at the base as the start of the tree. In this case, $\lambda(i)$ corresponds to the link number of the parent of link i. This notation is useful for structures with many branches; however, in the case of a single branch (as in a multi-segment inverted pendulum), $\lambda(i) = i - 1$, resulting in a consecutive numbering of the joints and links from the base until the end of the tree. Fig. B.1 shows a diagram of a single branch, multi-link structure.

With this in mind, the velocity of link i (denoted by $\boldsymbol{v}_i$) can be calculated based on the velocity of link $i - 1$ and the velocity across the joint i as follows

$$\begin{aligned} \boldsymbol{v}_i &= \boldsymbol{v}_{i-1} + \boldsymbol{s}_i \dot{\boldsymbol{q}}_i \\ \boldsymbol{v}_0 &= 0 \end{aligned} \tag{B.1}$$

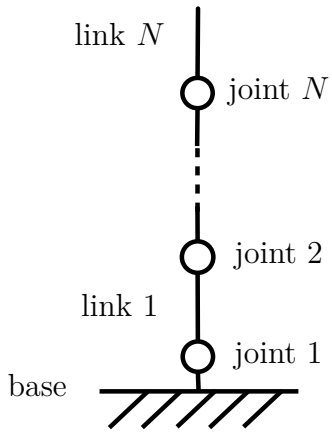


Figure B.1: Multi-link structure diagram. Joints and link are numbered from 1 to N. The base is considered the root link in the chain.

where $\dot{\boldsymbol{q}}_i$ is the $d_i \times 1$ joint velocity vector, s_i represents a $6 \times d_i$ matrix describing the axis of motion for joint i, and $\boldsymbol{v}_0$ is the velocity of the base. In this context, d_i represents the degree of freedom. The link acceleration $\boldsymbol{a}_i$ can be obtained as derivatives with respect to time of the link velocities in (B.1), this yields

$$\begin{aligned} \boldsymbol{a}_i &= \boldsymbol{a}_{i-1} + \dot{\boldsymbol{s}}_i \dot{\boldsymbol{q}}_i + \boldsymbol{s}_i \ddot{\boldsymbol{q}} \\ \boldsymbol{a}_0 &= -g \end{aligned} \tag{B.2}$$

with $\ddot{\boldsymbol{q}}$ being a vector of joint acceleration variables, $\boldsymbol{a}_0$ is the acceleration of the base defined by the gravity, and $\dot{\boldsymbol{s}}_i = \boldsymbol{v}_i \times \boldsymbol{s}_i$. Expressions (B.1) and (B.2) can be used to calculate the spatial force transmitted from link $i-1$ to i across the joint (represented by $\boldsymbol{f}_i$) obtaining

$$\boldsymbol{f}_i = \boldsymbol{f}_{i+1} + I_i \boldsymbol{a}_i + \boldsymbol{v}_i \times I_i \boldsymbol{v}_i \tag{B.3}$$

where I_i is a 6×6 spatial inertia matrix. Finally, it is possible to extract the vector of joint forces τ_i (which corresponds to joint torques in the case of revolute joints) from $\boldsymbol{f}_i$ as follows

$$\tau_i = \boldsymbol{s}_i^T \boldsymbol{f}_i \tag{B.4}$$

Equations (B.1), (B.2), (B.3), and (B.4) are the basis of the Newton-Euler algorithm. Their implementation requires further considerations which were not included in this description. Details on code implementations can be found in (Featherstone, 2010b).

Bibliography

Alamir, M. (2004). New path-generation based receding-horizon formulation for constrained stabilization of nonlinear systems. *Automatica*, 40(4):647–652.

Alamir, M. and Boyer, F. (2003). Fast generation of attractive trajectories for an under-actuated satellite application to feedback control design. *Optimization and Engineering*, 4(3):215–230.

Alamir, M. and Boyer, F. (2006). Re-injecting the structure in nmpc schemes application to the constrained stabilization of a snakeboard. *Fast Motions in Biomechanics and Robotics*, pages 1–20.

Alamir, M. and Marchand, N. (2003). Constrained minimum-time-oriented feedback control for the stabilization of nonholonomic systems in chained form. *Journal of Optimization Theory and Applications*, 118(2):229–244.

Alexandrov, A. V., Frolov, A. A., Horak, F., Carlson-Kuhta, P., and Park, S. (2005). Feedback equilibrium control during human standing. *Biological cybernetics*, 93(5):309–322.

Arnold, W. F. and Laub, A. J. (1984). Generalized eigenproblem algorithms and software for algebraic riccati equations. *Proceedings of the IEEE*, 72(12):1746–1754.

Arulampalam, B. R. S. (2004). *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House.

Asai, Y., Tasaka, Y., Nomura, K., Nomura, T., Casadio, M., and Morasso, P. (2009). A model of postural control in quiet standing: robust compensation of delay-induced instability using intermittent activation of feedback control. *PLoS One*, 4(7):e6169.

Åström, K. (2008). *Event based control. In Astolfi A, Marconi L (eds)*. Analysis and design of nonlinear control systems. Springer.

Åström, K., Aracil, J., and Gordillo, F. (2008). A family of smooth controllers for swinging up a pendulum. *Automatica*, 44(7):1841 – 1848.

Åström, K., Borisson, U., Ljung, L., and Wittenmark, B. (1977). Theory and Applications of Self-Tuning Regulators. *Automatica*, 13:457–476.

Åström, K. and Wittenmark, B. (1995). *Adaptive Control*. Addison-Wesley Series in Electrical Engineering. Addison-Wesley.

Athans, M., Wishner, R., and Bertolini, A. (1968). Suboptimal state estimation for continuous-time nonlinear systems from discrete noisy measurements. *IEEE Transactions on Automatic Control*, 13(5):504–514.

Bar-Shalom, Y. (1981). Stochastic dynamic programming: Caution and probing. *IEEE Transactions on Automatic Control*, 26(5):1184–1195.

Baron, S., Kleinman, D., and Levison, W. (1970). An optimal control model of human response part II: prediction of human performance in a complex task. *Automatica*, 6(3):371–383.

Basmajian, J. and De Luca, C. (1985). *Muscles alive: their functions revealed by electromyography*. Williams & Wilkins.

Bhushan, N. and Shadmehr, R. (1999). Computational nature of human adaptive control during learning of reaching movements in force fields. *Biological cybernetics*, 81(1):39–60.

bin Mohd Taib, M. A. F., Hayakawa, T., and Cetinkaya, A. (2013). Delayed feedback control for linear time-varying periodic systems in act-and-wait fashion. *IFAC Proceedings Volumes*, 46(12):11 – 16. 5th IFAC Workshop on Periodic Control Systems.

Bini, D., Iannazzo, B., and Meini, B. (2011). *Numerical Solution of Algebraic Riccati Equations*. Society for Industrial and Applied Mathematics.

Biswas, S. K., Qiao, L., and Dempster, A. G. (2017). A novel a priori state computation strategy for the unscented kalman filter to improve computational efficiency. *IEEE transactions on Automatic Control*, 62(4):1852–1864.

Bobrow, J., Park, F., and Sideris, A. (2006). *Recent Advances on the Algorithmic Optimization of Robot Motion*, pages 21–41. Springer Berlin Heidelberg, Berlin, Heidelberg.

Bottaro, A., Casadio, M., Moraso, P., and Sanguineti, V. (2005). Body sway during quiet standing: Is it the residual chattering of an intermittent stabilization process? *Human Movement Science*, 24(4):588–615.

Bottaro, A., Yasutake, Y., Nomura, T., Casadio, M., and Morasso, P. (2008). Bounded stability of the quiet standing posture: An intermittent control model. *Human Movement Science*, 27(3):473 – 495.

Bronstein, A. M., Brandt, T., Woollacott, M. H., and Nutt, J. G. (1996). *Clinical disorders of balance, posture and gait*. Arnold London.

Bucy, R. and Joseph, P. (1968). *Filtering for stochastic processes with applications to guidance*. Interscience tracts in pure and applied mathematics. Interscience Publishers.

Buschmann, T., Lohmeier, S., Ulbrich, H., and Pfeiffer, F. (2005). Optimization based gait pattern generation for a biped robot. In *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pages 98–103. IEEE.

Cabodevila, G., Chaillet, N., and Abba, G. (1995). Energy-minimized gait for a biped robot. In *Autonome Mobile Systeme 1995*, pages 90–99. Springer.

Carpenter, G. A. and Grossberg, S. (1988). The art of adaptive pattern recognition by a self-organizing neural network. *Computer*, 21(3):77–88.

Castano, J. A., Zhou, C., Li, Z., and Tsagarakis, N. (2016). Robust model predictive control for humanoids standing balancing. In *International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 147–152. IEEE.

Chalam, V. V. (1987). *Adaptive control systems: Techniques and applications*. Marcel Dekker, Inc.

Chapman, C., Bushnell, M., Miron, D., Duncan, G., and Lund, J. (1987). Sensory perception during movement in man. *Experimental Brain Research*, 68(3):516–524.

Chen, Y.-F. and Huang, A.-C. (2014). Adaptive control of rotary inverted pendulum system with time-varying uncertainties. *Nonlinear Dynamics*, 76(1):95–102.

Chevallereau, C., Formal'Sky, A., and Perrin, B. (1998). Low energy cost reference trajectories for a biped robot. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 2, pages 1398–1404. IEEE.

Chowdhary, G. and Jategaonkar, R. (2010). Aerodynamic parameter estimation from flight data applying extended and unscented kalman filter. *Aerospace science and technology*, 14(2):106–117.

Clark, A. (2013). Whatever next? predictive brains, situated agents, and the future of cognitive science. *Behavioral and brain sciences*, 36(3):181–204.

Clarke, D. W. and Gawthrop, P. J. (1981). Implementation and application of microprocessor-based self-tuners. *Automatica*, 17(1):233–244.

Collins, D., Cameron, T., Gillard, D., and Prochazka, A. (1998). Muscular sense is attenuated when humans move. *The Journal of physiology*, 508(2):635–643.

Cox, H. (1964). On the estimation of state variables and parameters for noisy dynamic systems. *IEEE Transactions on automatic control*, 9(1):5–12.

Craik, K. (1947). Theory of the human operator in control systems I. The operator as an engineering system. *British Journal of Psychology*, 38:56.

Craik, K. (1948). Theory of the human operator in control systems II. Man as an element in a control system. *The Journal of Physiology*, (1):142–148.

Crassidis, J. L. (2006). Sigma-point kalman filtering for integrated gps and inertial navigation. *IEEE Transactions on Aerospace and Electronic Systems*, 42(2):750–756.

Daum, F. (2005). Nonlinear filters: beyond the kalman filter. *IEEE Aerospace and Electronic Systems Magazine*, 20(8):57–69.

de Boer, T. (2012). *Foot placement in robotic bipedal locomotion*. PhD thesis, Delft University of Technology, Department of BioMechanical Engineering.

Deliagina, T. G., Zelenin, P. V., Beloozerova, I. N., and Orlovsky, G. N. (2007). Nervous mechanisms controlling body posture. *Physiology & Behavior*, 92(1):148 – 154.

Denk, J. and Schmidt, G. (2001). Synthesis of a walking primitive database for a humanoid robot using optimal control techniques. In *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, pages 319–326.

Diehl, M., Bock, H. G., Diedam, H., and Wieber, P.-B. (2006). Fast direct multiple shooting algorithms for optimal robot control. In *Fast motions in biomechanics and robotics*, pages 65–93. Springer.

Doeringer, J. A. and Hogan, N. (1998). Intermittency in preplanned elbow movements persists in the absence of visual feedback. *Journal of Neurophysiology*, 80(4):1787–1799.

Donchin, O., Francis, J. T., and Shadmehr, R. (2003). Quantifying generalization from trial-by-trial behavior of adaptive systems that learn with basis functions: Theory and experiments in human motor control. *Journal of Neuroscience*, 23(27):9032–9045.

Duarte, M. and Zatsiorsky, V. M. (2001). Long-range correlations in human standing. *Physics Letters A*, 283(1-2):124–128.

Egardt, B. (1979). *Stability of adaptive controllers*, volume 20. Springer.

Egardt, B. (1980). Stability analysis of continuous-time adaptive control systems. *SIAM Journal on Control and Optimization*, 18(5):540–558.

Englsberger, J., Ott, C., Roa, M. A., Albu-Schäffer, A., and Hirzinger, G. (2011). Bipedal walking control based on capture point dynamics. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 4420–4427. IEEE.

Erez, T., Lowrey, K., Tassa, Y., Kumar, V., Kolev, S., and Todorov, E. (2013). An integrated system for real-time model predictive control of humanoid robots. In *2013 13th IEEE-RAS International Conference on Humanoid Robots*, pages 292–299.

Estrada, T. and Antsaklis, P. J. (2008). Stability of discrete-time plants using model-based control with intermittent feedback. In *2008 16th Mediterranean Conference on Control and Automation*, pages 1130–1136.

Fantoni, I. and Lozano, R. (2002). *Non-linear control for underactuated mechanical systems*. Springer Science & Business Media.

Featherstone, R. (2008). *Rigid body dynamics algorithms*, volume 49. Springer New York.

Featherstone, R. (2010a). A beginner's guide to 6-D vectors (part 1). *Robotics & Automation Magazine, IEEE*, 17(3):83–94.

Featherstone, R. (2010b). A beginner's guide to 6-D vectors (part 2). *Robotics & Automation Magazine, IEEE*, 17(4):88–99.

Fel'dbaum, A. A. (1965). *Optimal Control Systems*. Academic Press, New York.

Feng, G. and Lozano, R. (1999). *Adaptive control systems*. Newnes.

Fishbach, A., Roy, S. A., Bastianen, C., Miller, L. E., and Houk, J. C. (2005). Kinematic properties of on-line error corrections in the monkey. *Experimental Brain Research*, 164(4):442–457.

Franklin, G., Powell, J., and Emami-Naeini, A. (1994). *Feedback control of dynamic systems*. Addison-Wesley.

Friston, K. (2008). Hierarchical models in the brain. *PLoS computational biology*, 4(11):e1000211.

Friston, K. (2010). The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11(2):127.

Friston, K. (2011). What is optimal about motor control? *Neuron*, 72(3):488–498.

Friston, K., Mattout, J., and Kilner, J. (2011). Action understanding and active inference. *Biological cybernetics*, 104(1-2):137–160.

Friston, K. J., Daunizeau, J., and Kiebel, S. J. (2009). Reinforcement learning or active inference? *PloS one*, 4(7):e6421.

Fuller, A. T. (1968). Optimal nonlinear control of systems with pure delay. *International Journal of Control*, 8(2):145–168.

Furuta, K., Yamakita, M., and Kobayashi, S. (1992). Swing-up control of inverted pendulum using pseudo-state feedback. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 206(4):263–269.

Gatev, P., Thomas, S., Kepple, T., and Hallett, M. (1999). Feedforward ankle strategy of balance during quiet stance in adults. *The Journal of physiology*, 514(3):915–928.

Gawthrop, P. (1982). *Continuous-time self-tuning control. Vol 1: design*. Engineering control series. Research Studies Press.

Gawthrop, P. (2004). Intermittent constrained predictive control of mechanical systems. In Petersen, I. R., editor, *Proceedings of the 3rd IFAC Symposium on Mechatronic Systems*, Manly, Australia.

Gawthrop, P. and Gollee, H. (2012). Intermittent tapping control. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 226(9):1262–1273.

Gawthrop, P., Gollee, H., and Loram, I. (2015). Intermittent control in man and machine. In Miskowicz, M., editor, *Event-Based Control and Signal Processing*, Embedded Systems, chapter 14, pages 1–99. CRC press, London.

Gawthrop, P. and Lim, K. (1982). Robustness of self-tuning controllers. In *IEE Proceedings D-Control Theory and Applications*, volume 129, pages 21–29. IET.

Gawthrop, P., Loram, I., Gollee, H., and Lakie, M. (2014). Intermittent control models of human standing: similarities and differences. *Biological Cybernetics*, 108:159–168.

Gawthrop, P., Loram, I., and Lakie, M. (2009). Predictive feedback in human simulated pendulum balancing. *Biological cybernetics*, 101(2):131–146.

Gawthrop, P., Loram, I., Lakie, M., and Gollee, H. (2011). Intermittent control: a computational theory of human control. *Biological Cybernetics*, 104(1-2):31–51.

Gawthrop, P., Neild, S., and Wagg, D. (2012). Semi-active damping using a hybrid control approach. *Journal of Intelligent Material Systems and Structures*, 23(18):2103–2116.

Gawthrop, P., Wagg, D., Neild, S., and Wang, L. (2013). Power-constrained intermittent control. *International Journal of Control*, 86(3):396–409.

Gawthrop, P. and Wang, L. (2006). Intermittent predictive control of an inverted pendulum. *Control Engineering Practice*, 14(11):1347–1356.

Gawthrop, P. and Wang, L. (2007). Intermittent model predictive control. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 221(7):1007–1018.

Gawthrop, P. and Wang, L. (2009a). Event-driven intermittent control. *International Journal of Control*, 82(12):2235–2248.

Gawthrop, P. and Wang, L. (2011). The system-matched hold and the intermittent control separation principle. *International Journal of Control*, 84(12):1965–1974.

Gawthrop, P. J. (1976). *Studies in identification and control*. D.Phil. thesis, Oxford University.

Gawthrop, P. J. (2009). Frequency-domain analysis of intermittent control. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 223(5):591–603.

Gawthrop, P. J. and Wang, L. (2009b). Constrained intermittent model predictive control. *International Journal of Control*, 82(6):1138–1147.

Gelb, A. (1974). *Applied Optimal Estimation*. Applied Optimal Estimation. MIT Press.

Giri, F., M'Saad, M., Dion, J.-M., and Dugard, L. (1991). On the robustness of discrete-time indirect adaptive (linear) controllers. *Automatica*, 27(1):153–159.

Gollee, H., Gawthrop, P. J., Lakie, M., and Loram, I. D. (2017). Visuo-manual tracking: does intermittent control with aperiodic sampling explain linear power and non-linear remnant without sensorimotor noise? *The Journal of physiology*, 595(21):6751–6770.

Gollee, H., Mamma, A., Loram, I., and Gawthrop, P. (2012). Frequency-domain identification of the human controller. *Biological Cybernetics*, 106(6-7):359–72.

Goodwin, G., Graebe, S., and Salgado, M. (2001). *Control System Design*. Prentice Hall.

Goodwin, G., Ramadge, P., and Caines, P. (1980). Discrete-time multivariable adaptive control. *IEEE Transactions on Automatic Control*, 25(3):449–456.

Gordillo, F., Acosta, J. A., and Aracil, J. (2003). A new swing-up law for the furuta pendulum. *International Journal of Control*, 76(8):836–844.

Gordon, A., Forssberg, H., Johansson, R., and Westling, G. (1991). Visual size cues in the programming of manipulative forces during precision grip. *Experimental Brain Research*, 83(3):477–482.

Gross, J. N., Gu, Y., Rhudy, M. B., Gururajan, S., and Napolitano, M. R. (2012). Flight-test evaluation of sensor fusion algorithms for attitude estimation. *IEEE Transactions on Aerospace and Electronic Systems*, 48(3):2128–2139.

Günther, M., Müller, O., and Blickhan, R. (2012). What does head movement tell about the minimum number of mechanical degrees of freedom in quiet human stance? *Archive of Applied Mechanics*, 82(3):333–344.

Hawley, L. and Suleiman, W. (2017). Control strategy and implementation for a humanoid robot pushing a heavy load on a rolling cart. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 4997–5002. IEEE.

Haykin, S. (2001). *Kalman filtering and neural networks*. John Wiley & Sons, Inc.

Hegyi, A., Girimonte, D., Babuska, R., and De Schutter, B. (2006). A comparison of filter configurations for freeway traffic state estimation. In *IEEE 2006 Intelligent Transportation Systems Conference.*, pages 1029–1034. IEEE.

Hobbelen, D., de Boer, T., and Wisse, M. (2008). System overview of bipedal robots flame and tulip: Tailor-made for limit cycle walking. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2486–2491. IEEE.

Hobbelen, D. G. and Wisse, M. (2008). Ankle actuation for limit cycle walkers. *The International Journal of Robotics Research*, 27(6):709–735.

Hsu, S.-H. and Fu, L.-C. (2006). A fully adaptive decentralized control of robot manipulators. *Automatica*, 42(10):1761–1767.

Hu, Y. and Mombaur, K. (2017). Optimal control based push recovery strategy for the icub humanoid robot with series elastic actuators. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 5846–5852. IEEE.

Ibáñez, C. A. and Azuela, J. H. S. (2007). Stabilization of the furuta pendulum based on a lyapunov function. *Nonlinear Dynamics*, 49(1):1–8.

Insperger, T. (2006). Act-and-wait concept for continuous-time control systems with feedback delay. *IEEE Transaction in Controls Systems Technology*, 14(5):974–977.

Izawa, J., Rane, T., Donchin, O., and Shadmehr, R. (2008). Motor adaptation as a process of reoptimization. *Journal of Neuroscience*, 28(11):2883–2891.

Johansson, R., Magnusson, M., and Akesson, M. (1988). Identification of human postural dynamics. *IEEE Transactions on Biomedical Engineering*, 35(10):858–869.

Julier, S., Uhlmann, J., and Durrant-Whyte, H. F. (2000). A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on automatic control*, 45(3):477–482.

Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H. F. (1995). A new approach for filtering nonlinear systems. In *American Control Conference, Proceedings of the 1995*, volume 3, pages 1628–1632 vol.3.

Kagami, S., Kitagawa, T., Nishiwaki, K., Sugihara, T., Inaba, M., and Inoue, H. (2002). A fast dynamically equilibrated walking trajectory generation method of humanoid robot. *Autonomous Robots*, 12(1):71–82.

Kajita, S., Kanehiro, F., Kaneko, K., Yokoi, K., and Hirukawa, H. (2001). The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 1, pages 239–246. IEEE.

Kalman, R. E. (1958). Design of a self-optimising control system. *Trans. ASME*, 80:468–478.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45.

Kalman, R. E. and Bucy, R. S. (1961). New results in linear filtering and prediction theory. *Journal of basic engineering*, 83(1):95–108.

Karniel, A. (2011). Open questions in computational motor control. *Journal of integrative neuroscience*, 10(03):385–411.

Kawato, M. (1999). Internal models for motor control and trajectory planning. *Current opinion in neurobiology*, 9(6):718–727.

Kelly, R. (1997). PD control with desired gravity compensation of robotic manipulators: A review. *Int. J. Rob. Res.*, 16(5):660–672.

Kharitonov, V. L. (2017). Prediction-based control for systems with state and several input delays. *Automatica*, 79:11–16.

Kiemel, T., Zhang, Y., and Jeka, J. J. (2011). Identification of neural feedback for upright stance in humans: stabilization rather than sway minimization. *Journal of Neuroscience*, 31(42):15144–15153.

Klaiman, E. and Karniel, A. (2006). Bimanual adaptation: internal representations of bimanual rhythmic movements. *Experimental Brain Research*, 171(2):204–214.

Kleinman, D. (1969). Optimal control of linear systems with time-delay and observation noise. *IEEE Transactions in Control Systems Technology*, 14:524–527.

Kleinman, D., Baron, S., and Levison, W. (1970). An optimal control model of human response. Part I: theory and validation. *Automatica*, 6(3):357–369.

Koenemann, J., Prete, A. D., Tassa, Y., Todorov, E., Stasse, O., Bennewitz, M., and Mansard, N. (2015). Whole-body model-predictive control applied to the HRP-2 humanoid. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3346–3351.

Kooij, H. and Vlugt, E. (2007). Postural responses evoked by platform pertubations are dominated by continuous feedback. *Journal of Neurophysiology*, 98(2):730–743.

Körding, K. P. and Wolpert, D. M. (2004). Bayesian integration in sensorimotor learning. *Nature*, 427(6971):244–247.

Kowalczyk, P., Glendinning, P., Brown, M., Medrano-Cerda, G., Dallali, H., and Shapiro, J. (2012). Modelling human balance using switched systems with linear feedback control. *Journal of The Royal Society Interface*, 9(67):234–245.

Krause, M., Englsberger, J., Wieber, P.-B., and Ott, C. (2012). Stabilization of the capture point dynamics for bipedal walking based on model predictive control. *IFAC Proceedings Volumes*, 45(22):165–171.

Kwakernaak, H. and Sivan, R. (1972). *Linear optimal control systems*. Wiley-Interscience publication. Wiley Interscience.

Lakie, M., Caplan, N., and Loram, I. (2003). Human balancing of an inverted pendulum with a compliant linkage: neural control by anticipatory intermittent bias. *Journal of Physiology*, 551:357–370.

Lakie, M. and Loram, I. D. (2006). Manually controlled human balancing using visual, vestibular and proprioceptive senses involves a common, low frequency neural process. *The Journal of Physiology*, 577(1):403–416.

Landau, I. D., Lozano, R., M'Saad, M., and Karimi, A. (2011). *Adaptive control: algorithms, analysis and applications*. Springer Science & Business Media.

Lee, J. H. and Ricker, N. L. (1993). Extended Kalman filter based nonlinear model predictive control. In *1993 American Control Conference*, pages 1895–1899.

Lee, Y. W. (1962). Statistical theory of communication. *Naval Research Logistics Quarterly*, 9(2):161–162.

Levison, W. H., Baron, S., and Kleinman, D. L. (1969). A model for human controller remnant. *IEEE Transactions on Man-Machine Systems*, 10(4):101–108.

Levy, N., Pressman, A., Mussa-Ivaldi, F. A., and Karniel, A. (2010). Adaptation to delayed force perturbations in reaching movements. *PLoS one*, 5(8):e12128.

Lewis, F., Xie, L., and Popa, D. (2007). *Optimal and Robust Estimation: With an Introduction to Stochastic Control Theory, Second Edition*. Automation and Control Engineering. CRC Press.

Li, W. and Todorov, E. (2004). Iterative linear quadratic regulator design for nonlinear biological movement systems. In *ICINCO (1)*, pages 222–229.

Liu, C. and Atkeson, C. G. (2009). Standing balance control using a trajectory library. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 3031–3036. IEEE.

Lockhart, D. B. and Ting, L. H. (2007). Optimal sensorimotor transformations for balance. *Nature Neuroscience*, 10:1329–1336.

Loffler, K., Gienger, M., Pfeiffer, F., and Ulbrich, H. (2004). Sensors and control concept of a biped robot. *IEEE Transactions on industrial electronics*, 51(5):972–980.

Loram, I., Gollee, H., Lakie, M., and Gawthrop, P. (2011). Human control of an inverted pendulum: is continuous control necessary? Is intermittent control effective? Is intermittent control physiological? *The Journal of Physiology*, 589(Pt 2):307–24.

Loram, I. and Lakie, M. (2002). Human balancing of an inverted pendulum: position control by small, ballistic-like, throw and catch movements. *Journal of Physiology*, 540(3)(3):1111–1124.

Loram, I., Lakie, M., and Gawthrop, P. (2009). Visual control of stable and unstable loads: what is the feedback delay and extent of linear time-invariant control? *The Journal of Physiology*, 587(Pt 6):1343–65.

Loram, I., van de Kamp, C., Gollee, H., and Gawthrop, P. (2012). Identification of intermittent control in man and machine. *Journal of the Royal Society*, 9(74):2070–84.

Loram, I., Van de Kamp, C., Lakie, M., Gollee, H., and Gawthrop, P. (2014). Does the motor system need intermittent control? *Exercise and Sport Sciences Reviews*, 42(3):117–125.

Loram, I. D., Gawthrop, P. J., and Lakie, M. (2006a). The frequency of human, manual adjustments in balancing an inverted pendulum is constrained by intrinsic physiological factors. *The Journal of physiology*, 577(1):417–432.

Loram, I. D., Maganaris, C. N., and Lakie, M. (2005). Human postural sway results from frequent, ballistic bias impulses by soleus and gastrocnemius. *The Journal of Physiology*, 564(1):295–311.

Loram, I. D., Maganaris, C. N., and Lakie, M. (2006b). Use of ultrasound to make noninvasive in vivo measurement of continuous changes in human muscle contractile length. *Journal of Applied Physiology*, 100(4):1311–1323.

Luenberger, D. (1971). An introduction to observers. *IEEE Transactions on Automatic Control*, 16(6):596–602.

Luenberger, D. (1979). *Introduction to Dynamic Systems: Theory, Models, and Applications*. Wiley.

Madigan, M. L., Davidson, B. S., and Nussbaum, M. A. (2006). Postural sway and joint kinematics during quiet standing are affected by lumbar extensor fatigue. *Human Movement Science*, 25(6):788–799.

Manganiello, P., Ricco, M., Petrone, G., Monmasson, E., and Spagnuolo, G. (2015). Dual Kalman filter based identification and real-time optimization of PV systems. *IEEE Transactions on Industrial Electronics*, 62(11):7266–7275.

Maurer, C. and Peterka, R. J. (2005). A new interpretation of yspontaneous sway measures based on a simple model of human postural control. *Journal of neurophysiology*, 93(1):189–200.

Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814.

Mcgee, L. A., Schmidt, S. F., Mcgee, L. A., and Sc, S. F. (1985). Discovery of the kalman filter as a practical tool for aerospace and. Technical report, Industry, National Aeronautics and Space Administration, Ames Research.

McRuer, D. T. and Jex, H. R. (1967). A review of quasi-linear pilot models. *IEEE Transactions on Human Factors in Electronics*, (3):231–249.

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

Meskin, N., Nounou, H., Nounou, M., and Datta, A. (2013). Parameter estimation of biological phenomena: an unscented kalman filter approach. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 10(2):537–543.

Miall, R., Weir, D., and Stein, J. (1993a). Intermittency in human manual tracking tasks. *J. Motor Behav.*, 25:203–216.

Miall, R., Weir, D. J., Wolpert, D. M., and Stein, J. (1993b). Is the cerebellum a smith predictor? *Journal of motor behavior*, 25(3):203–216.

Miall, R. C. and Wolpert, D. M. (1996). Forward models for physiological motor control. *Neural networks*, 9(8):1265–1279.

Mombaur, K. (2009). Using optimization to create self-stable human-like running. *Robotica*, 27(3):321–330.

Montestruque, L. and Antsaklis, P. (2003). On the model-based control of networked systems. *Automatica*, 39(10):1837–1843.

Morari, M. and Lee, J. H. (1999). Model predictive control: past, present and future. *Computers and Chemical Engineering*, 23(4):667–682.

Morrison, G. and Cebon, D. (2016). Sideslip estimation for articulated heavy vehicles at the limits of adhesion. *Vehicle System Dynamics*, 54(11):1601–1628.

Morse, A. (1980). Global stability of parameter-adaptive control systems. *IEEE Transactions on Automatic Control*, 25(3):433–439.

Murray, R. M., Li, Z., Sastry, S. S., and Sastry, S. S. (1994). *A mathematical introduction to robotic manipulation*. CRC press.

Nair, G. N. and Evans, R. J. (2003). Exponential stabilisability of finite-dimensional linear systems with limited data rates. *Automatica*, 39:585–593.

Narendra, K. S. and Annaswamy, A. M. (1989). *Stable Adaptive Systems*. Prentice-Hall, Inc.

Narendra, K. S. and Balakrishnan, J. (1997). Adaptive control using multiple models. *IEEE transactions on automatic control*, 42(2):171–187.

Nashner, L. M. and McCollum, G. (1985). The organization of human postural movements: a formal basis and experimental synthesis. *Behavioral and brain sciences*, 8(1):135–150.

Navas, T. and Stark, L. (1968). Sampling or intermittency in hand control system dynamics. *Biophysics Journal*, 14:252–302.

Naveau, M., Kudruss, M., Stasse, O., Kirches, C., Mombaur, K., and Souères, P. (2017). A reactive walking pattern generator based on nonlinear model predictive control. *IEEE Robotics and Automation Letters*, 2(1):10–17.

Neilson, P., Neilson, M., and O'Dwyer, N. (1988). Internal models and intermittency: a theoretical account of human tracking behavior. *Biological Cybernetics*, 58:101–112.

Neunert, M., Farshidian, F., and Buchli, J. (2014). Adaptive real-time nonlinear model predictive motion control. In *IROS 2014 Workshop on Machine Learning in Planning and Control of Robot Motion*.

Niculescu, S. (2001). *Delay Effects on Stability: A Robust Control Approach*. Lecture Notes in Control and Information Sciences. Springer London.

Nielson, P. (1999). The intermittency of control movements and the psychological refractory period. *Motor Control*, 3:280–284.

Nomura, T., Oshikawa, S., Suzuki, Y., Kiyono, K., and Morasso, P. (2013). Modeling human postural sway using an intermittent control and hemodynamic perturbations. *Mathematical Biosciences*, 245(1):86–95.

Olfati-Saber, R. (2001). *Nonlinear control of underactuated mechanical systems with application to robotics and aerospace vehicles*. PhD thesis, Massachusetts Institute of Technology.

Park, J. and Chung, W. (2000). Design of a robust H∞ PID control for industrial manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 122(4):803–812.

Pashler, H. (1992). Attentional limitations in doing two tasks at the same time. *Current Directions in Psychological Science*, 1(2):44–48.

Pashler, H. (1994). Dual-task interference in simple tasks: data and theory. *Psychological bulletin*, 116(2):220.

Peterka, R. J. (2000). Postural control model interpretation of stabilogram diffusion analysis. *Biological cybernetics*, 82(4):335–343.

Peterka, R. J. (2002). Sensorimotor integration in human postural control. *Journal of Neurophysiology*, 88(3):1097–1118.

Peterson, B. and Narendra, K. (1982). Bounded error adaptive control. *IEEE Transactions on Automatic Control*, 27(6):1161–1168.

Plomp, K. A., Vidarsdóttir, U. S., Weston, D. A., Dobney, K., and Collard, M. (2015). The ancestral shape hypothesis: an evolutionary explanation for the occurrence of intervertebral disc herniation in humans. *BMC evolutionary biology*, 15(1):68.

Pratt, G. and Williamson, M. (1995). Series elastic actuators. In *Proceedings of the International Conference on Intelligent Robots and Systems 1995*, volume 1, pages 707–712.

Pratt, J., Carff, J., Drakunov, S., and Goswami, A. (2006). Capture point: A step toward humanoid push recovery. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 200–207. IEEE.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1996). *Numerical recipes in C*, volume 2. Cambridge university press Cambridge.

Qu, Z. and Dorsey, J. (1991). Robust tracking control of robots by a linear feedback law. *IEEE Transactions on Automatic Control*, 36:1081–1084.

Ramírez-Neria, M., Sira-Ramírez, H., Garrido-Moctezuma, R., and Luviano-Juárez, A. (2014). Linear active disturbance rejection control of underactuated systems: The case of the furuta pendulum. *ISA Transactions*, 53(4):920 – 928. Disturbance Estimation and Mitigation.

Richalet, J. (1993). Industrial applications of model based predictive control. *Automatica*, 29(5):1251–1274.

Rigatos, G., Siano, P., Abbaszadeh, M., Ademi, S., and Melkikh, A. (2017). Nonlinear h-infinity control for underactuated systems: the furuta pendulum example. *International Journal of Dynamics and Control*.

Ronco, E., Arsan, T., and Gawthrop, P. (1999). Open-loop intermittent feedback control: practical continuous-time GPC. *IEE Proceedings - Control Theory and Applications*, 146(5):426–434.

Roussel, L., Canudas-de Wit, C., and Goswami, A. (1998). Generation of energy optimal complete gait cycles for biped robots. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 3, pages 2036–2041. IEEE.

Ruymgaart, P. A. and Soong, T. T. (1988). *Mathematics of Kalman-Bucy filtering*. Springer.

Santos, C., Mazo, M., and Espinosa, F. (2014). Adaptive self-triggered control of a remotely operated P3-DX robot: simulation and experimentation. *Robotics and Autonomous Systems*, 62(6):847–854.

Santos, C., Mazo Jr, M., and Espinosa, F. (2012). Adaptive self-triggered control of a remotely operated robot. In *Advances in Autonomous Robotics*, pages 61–72. Springer.

Sastry, S. and Bodson, M. (2011). *Adaptive Control: Stability, Convergence and Robustness*. Dover Books on Electrical Engineering Series. Dover Publications.

Schmitt, D. (2003). Insights into the evolution of human bipedalism from experimental studies of humans and other primates. *Journal of Experimental Biology*, 206(9):1437–1448.

Schneider, R. and Georgakis, C. (2013). How to not make the extended kalman filter fail. *Industrial & Engineering Chemistry Research*, 52(9):3354–3362.

Schwartz, A. B. (2016). Movement: how the brain communicates with the world. *Cell*, 164(6):1122–1135.

Shadmehr, R. and Krakauer, J. W. (2008). A computational neuroanatomy for motor control. *Experimental brain research*, 185(3):359–381.

Shadmehr, R. and Mussa-Ivaldi, S. (2012). *Biological learning and control: How the brain builds representations, predicts events, and makes decisions*. MIT Press.

Simon, D. (2006). *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience.

Skoyles, J. R. (2006). Human balance, the evolution of bipedalism and dysequilibrium syndrome. *Medical Hypotheses*, 66(6):1060 – 1068.

Slotine, J. J. E. and Yang, H. S. (1989). Improving the efficiency of time-optimal path-following algorithms. *IEEE Transactions on Robotics and Automation*, 5(1):118–124.

Smith, J. J. (1990). *Circulatory response to the upright posture*. Number 6. CRC Press.

Smith, O. J. M. (1959). A controller to overcome dead-time. *ISA Transactions*, 6(2):23–33.

Spong, M. W., Hutchinson, S., and Vidyasagar, M. (2006). *Robot modeling and control*. John Wiley & Sons New York.

Squeri, V., Masia, L., Casadio, M., Morasso, P., and Vergaro, E. (2010). Force-field compensation in a manual tracking task. *PLoS One*, 5(6):e11189.

Sutton, R. S., Barto, A. G., and Williams, R. J. (1992). Reinforcement learning is direct adaptive optimal control. *IEEE Control Systems*, 12(2):19–22.

Tao, G. (2014). Multivariable adaptive control: A survey. *Automatica*, 50(11):2737–2764.

Taylor, C. J., Chotai, A., and Young, P. C. (2000). State space control system design based on non-minimal state-variable feedback: further generalization and unification results. *International Journal of Control*, 73(14):1329–1345.

Tedrake, R., Manchester, I. R., Tobenkin, M., and Roberts, J. W. (2010). Lqr-trees: Feedback motion planning via sums-of-squares verification. *The International Journal of Robotics Research*, 29(8):1038–1052.

Telford, C. (1931). The refractory phase of voluntary and associative responses. *J. of Exp. Psychol.*, 14:1–36.

Todorov, E. (2004). Optimality principles in sensorimotor control. *Nature Neuroscience*, 7(9):907–915.

Todorov, E. and Jordan, M. I. (2002). Optimal feedback control as a theory of motor coordination. *Nature neuroscience*, 5(11):1226.

Todorov, E. and Li, W. (2005). A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *Proceedings of the 2005 IEEE American Control Conference*, pages 300–306.

Tomei, P. (1991). Adaptive PD control for robot manipulators. *IEEE Transactions on Robotics and Automation*, 7:565–570.

Urata, J., Nshiwaki, K., Nakanishi, Y., Okada, K., Kagami, S., and Inaba, M. (2011). Online decision of foot placement using singular lq preview regulation. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 13–18. IEEE.

van de Kamp, C., Gawthrop, P., Gollee, H., Lakie, M., and Loram, I. (2013a). Interfacing sensory input with motor output: does the control architecture converge to a serial process along a single channel? *Frontiers in Computational Neuroscience*, 7:1–12.

van de Kamp, C., Gawthrop, P., Gollee, H., and Loram, I. (2013b). Refractoriness in sustained visuo-manual control: is the refractory duration intrinsic or does it depend on external system properties? *PLOS Computational Biology*, 9:e1002843.

van der Kooij, H., Jacobs, R., Koopman, B., and Grootenboer, H. (1999). A multisensory integration model of human stance control. *Biological Cybernetics*, 80(5):299–308.

van der Kooij, H., Jacobs, R., Koopman, B., and van der Helm, F. (2001). An adaptive model of sensory integration in a dynamic environment applied to human stance control. *Biological cybernetics*, 84(2):103–115.

Van Der Kooij, H. and Peterka, R. J. (2011). Non-linear stimulus-response behavior of the human stance control system is predicted by optimization of a system with sensory and motor noise. *Journal of computational neuroscience*, 30(3):759–778.

Van Der Merwe, R. (2004). *Sigma-point Kalman filters for probabilistic inference in dynamic state-space models*. PhD thesis, Oregon Health and Science University.

Van Der Merwe, R. and Wan, E. A. (2001). The square-root unscented kalman filter for state and parameter-estimation. In *Proceedings of 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'01)*, volume 6, pages 3461–3464.

Vince, M. (1948). The intermittency of control movements and the psychological refractory period. *British Journal of Psychology*, 38:149–157.

Voss, M., Ingram, J. N., Wolpert, D. M., and Haggard, P. (2008). Mere expectation to move causes attenuation of sensory signals. *PLoS One*, 3(8):e2866.

Vukobratović, M. and Stepanenko, J. (1972). On the stability of anthropomorphic systems. *Mathematical biosciences*, 15(1-2):1–37.

Wahlberg, B. and Mäkilä, P. (1996). On approximation of stable linear dynamical systems using laguerre and kautz functions. *Automatica*, 32(5):693–708.

Wan, E. A. and Van Der Merwe, R. (2000). The unscented kalman filter for nonlinear estimation. In *The Adaptive Systems for Signal Processing, Communications, and Control Symposium (IEEE AS-SPCC).*, pages 153–158. IEEE.

Wang, L. (2009). *Model Predictive Control System Design and Implementation Using MATLAB®*. Advances in Industrial Control. Springer London.

Welford, A. T. (1952). The psychological refractory period and the timing of high-speed performance âĂŤa review and a theory. *British Journal of Psychology. General Section*, 43(1):2–19.

Wescott, T. (2010). Controlling motors in the presence of friction and backlash. *Wescott Design Services, www. wescottdesign. com/articles/Friction/friction. pdf.*

Wieber, P.-B. (2006). Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 137–142. IEEE.

Wieber, P.-B. and Chevallereau, C. (2006). Online adaptation of reference trajectories for the control of walking systems. *Robotics and Autonomous Systems*, 54(7):559–566.

Winter, D. A., Patla, A. E., Prince, F., Ishac, M., and Gielo-Perczak, K. (1998). Stiffness control of balance in quiet standing. *Journal of Neurophysiology*, 80(3):1211–1221.

Wisse, M., Schwab, A. L., van der Linde, R. Q., and van der Helm, F. C. (2005). How to keep from falling forward: elementary swing leg action for passive dynamic walkers. *IEEE Transactions on robotics*, 21(3):393–401.

Witrant, E., Georges, D., Canudas-de Wit, C., and Alamir, M. (2007). *On the Use of State Predictors in Networked Control Systems*, pages 17–35. Springer Berlin Heidelberg.

Wittenmark, B. and Åström, K. J. (1984). Practical issues in the implementation of self-tuning control. *Automatica*, 20(5):595–605.

Wolpert, D. M. (1997). Computational approaches to motor control. *Trends in Cognitive Sciences*, 1(6):209 – 216.

Wolpert, D. M., Ghahramani, Z., and Jordan, M. I. (1995). An internal model for sensorimotor integration. *Science*, pages 1880–1882.

Wolpert, D. M., Miall, R. C., and Kawato, M. (1998). Internal models in the cerebellum. *Trends in cognitive sciences*, 2(9):338–347.

Young, P. (1981). Parameter estimation for continuous-time transfer functions. *International Journal of Control*, 17(1):23–39.

Young, P., Behzadi, M., Wang, C., and Chotai, A. (1987). Direct digital and adaptive control by input-output state variable feedback pole assignment. *International Journal of Control*, 46(6):1867–1881.

Young, P. C., Chotai, A., and Tych, W. (1991). Identification, estimation and control of continuous-time systems described by delta operator models. In *Identification of continuous-time systems*, pages 363–418. Springer.