

École polytechnique de Louvain

Adaptive control of an active ankle prosthesis

Authors: **Dylan DE COEYER, Alicia LEGRAND**
Supervisor: **Renaud RONSSE**
Readers: **Sophie HEINS, Frédéric CREVECOEUR**
Academic year 2019–2020
Master [120] in Electro-mechanical Engineering
Master [120] in Biomedical Engineering

Context The number of lower-limb amputations worldwide keeps increasing due to the ageing population and the spread of dysvascular diseases. In order to compensate for the missing limb, lower-limb amputees equipped with a prosthesis often develop asymmetric gait strategies which can lead to comorbidities. Different types of prosthetic devices exist to help transtibial lower limb amputees recover mobility, among which active ankle prostheses are a promising technical solution as they are the only ones able to provide a net positive work during the gait cycle. Until now, the lack of efficient control strategies for active prostheses, including real-time adaptation to the users and the environment, prevents them from taking part in some daily life activities and under-exploits the full potential of these devices.

Objectives This thesis is part of a long term project that consists of developing a lightweight, energy efficient and biomimetic active ankle prosthesis named ELSA. The purpose of this work is to develop an efficient control strategy able to provide adaptive assistance to the users in an attempt to recover optimal mobility. This involves compliant actuation technique and reduction of gait asymmetry throughout the walking gait cycle.

Method The explored strategy is based on a bio-inspired control approach. It takes advantage of a cerebellum-like circuit to estimate in real-time the complex inverse dynamics model of the user and perform a compliant actuation. On the other hand, adaptable motor primitives are used to generate and easily modulate the reference joint trajectory with the idea of reducing asymmetries. The latter are monitored using IMUs and adaptive oscillators as signal learning mechanism. A preliminary validation stage is carried out using a simulation environment to show the interest of the proposed control strategy. It consists of four experiments, each of them validating a specific feature : real-time gait analysis, position tracking and learning of the inverse dynamics using LWPR, compliance analysis, and gait asymmetry correction. Also, a few experiments are performed with the latest prototype of the prosthesis for real-time gait analysis validation, and also to show the technical issues encountered.

Results Throughout these experiments, several conclusions can be drawn. Firstly, the real-time gait analysis allows to estimate the phase variable of each leg as well as the temporal position of key gait events. Secondly, the controller is able to learn from the user dynamics and achieve very good trajectory tracking performance. Thirdly, the actuation is much more compliant than using a pure feedback regulator with equivalent tracking performance. Finally, the asymmetry correction mechanism shows mitigate results and cannot be properly validated in the considered simulation environment.

Conclusion The proposed control strategy shows promising results in terms of adaptive compliant actuation learning from the user's dynamics, and real-time gait analysis. However, the limitations induced by the simulation environment prevent from properly validating the asymmetry correction algorithm, meaning that further experiments on real users should be conducted.

ACKNOWLEDGEMENTS

We would first like to express our gratitude to our supervisor, **Mr. Renaud Ronsse**, for his availability, guidance and advice throughout the year. Thank you for offering us the opportunity to visit Ossür in Reykjavik, Iceland.

Besides our supervisor, we are particularly grateful to **Mrs. Sophie Heins** for her valuable help in the development of the control strategy. We hope that our contribution was helpful to you.

We also wish to acknowledge the help provided by **Mr. François Heremans** for his explanations regarding the functioning of the prosthesis and his advice concerning the control strategy.

Then, we would like to offer our special thanks to **Mrs. Christine Detrembleur** who helped us with the use of the **Vicon Nexus** software and the processing of our experimental data from the gait laboratory.

The assistance provided by **Mrs. Sophie Boulet** for her training on the use of the gait laboratory and the help of **Mrs. Virginie Otlet** for the use of the IMUs were greatly appreciated.

We are particularly grateful to **Mr. Pierre Denis** for his participation in the experiment to assess the performance of the ankle prosthesis.

Finally, we would like to thank our **families** for their support during the redaction of this thesis.

	Drawbacks	28
2.5.	Asymmetry measurement	28
2.6.	Reference generation	29
2.7.	Heuristic laws	32
3.	Feedback-error-learning control	34
3.1.	Motivation	34
3.2.	Novel control approach	34
3.3.	LWPR	35
	Working principle	35
	Link with the cerebellum structure	36
4	Preliminary validation	39
1.	Objectives	39
2.	Simulation environment	39
2.1.	Description	39
2.2.	Implementation	40
2.3.	Driven trajectories	41
3.	Real-time gait analysis validation	41
3.1.	Method	42
3.2.	Results	42
3.3.	Discussion	44
4.	Position tracking validation	44
4.1.	Method	45
4.2.	Parameters	45
4.3.	Results	45
4.4.	Discussion	48
5.	Compliance analysis	49
5.1.	Method	49
5.2.	Results	51
5.3.	Discussion	53
6.	Gait asymmetry correction validation	54
6.1.	Method	54
6.2.	Results	56
6.3.	Discussion	58
5	Application on ELSA	61
1.	Challenges and adjustments of the strategy	61
2.	Implementation	62
3.	Experimentation	63
3.1.	Method	64
3.2.	Results	64
3.3.	Discussion	65
4.	Technical issues encountered with ELSA	66
4.1.	Method	66
4.2.	Results	68
4.3.	Discussion	68
6	Conclusion	71

A	Normal gait cycle	73
B	Protocol	75
C	Real-time gait analysis validation with IMUs	79
D	Source code	81
1.	Matlab	81
2.	C++	82
	Bibliography	83

LIST OF ABBREVIATIONS

AI	asymmetry index
AO	adaptive oscillator
ASIS	antero superior iliac spine
CLME	complementary limb motion estimation
COP	centre of pressure
CUSL	cliniques universitaires Saint-Luc
ELSA	Effective Locking Spring Ankle
EMG	electromyography
FF	foot-flat
FSC	finite-state control
FSM	finite-state machine
GPR	gaussian process regression
GRF	ground reaction force
HO	heel-off
HS	heel-strike
IMU	inertial measurement unit
LPS	lockable parallel spring
LWL	locally weighted learning
LWPR	locally weighted projection regression
MBS	multibody system
MMG	mechanomyography
MS	mid-swing
PD	proportional derivative
PID	proportional integral derivative

PR projection regression

PSIS posterior superior iliac spine

RF receptive field

RMSE root mean square error

ROS Robot Operating System

SD standard deviation

SEA series elastic actuator

SVMR support vector machine regression

TO toe-off

VESC Vedder electronic speed controller

1. Context

Each year, the estimated number of amputations worldwide rises to more than 1 million among which 85% represent lower-limb amputations. About 72% of amputations are due to a vascular disease including diabetes and arteriosclerosis complications. The two next common causes of amputation are infections and trauma representing 8% and 7% respectively. The remaining 13% are linked to malignancy or a congenital limb deficiency [1, 2, 3]. Furthermore, the total number of amputations tends to increase with time. This can be explained by several factors such as the ageing population and the increase in the number of patients suffering from a dysvascular disease [4]. Lower limb amputees have a reduced quality of life together with a lower life expectancy and high treatment costs [5]. There are different levels of lower-limb amputations [1, 6] :

- **Foot** amputation, including toes amputation or amputation of a part of the foot,
- **Ankle disarticulation** amputation that consists of the foot and the ankle joint being removed,
- **Transtibial** amputation which refers to a below-knee amputation,
- **Knee disarticulation** amputation that consists of also removing the knee joint,
- **Transfemoral** amputation which refers to a below-hip amputation,
- **Hip disarticulation** amputation for which the entire lower limb is removed, including the removal of the hip joint.

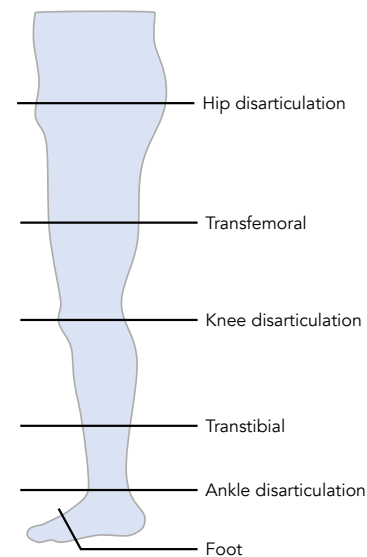


Figure 1.1: Levels of lower-limb amputation.

In this thesis, attention is drawn on transtibial lower-limb amputees.

1.1. Gait disorders after transtibial lower-limb amputation

A stride is characterised by a chain of events which constitutes the gait cycle and which starts when one limb comes into contact with the ground and ends when the same limb is back in contact with the ground. The normal gait cycle is described in Appendix A. In comparison with normal gait, the gait cycle of a lower-limb amputee is often observed as being asymmetric. In order to compensate for the missing limb, amputees develop different compensatory gait strategies that require a higher overall energy consumption and different muscle groups than for normal gait in an attempt to create a smooth walking pattern [7, 8, 9]. The energy requirement increases approximately by 25% and by 40% for traumatic and vascular transtibial amputees respectively [1, 10].

Compensatory motions

Lower-limb amputees adopt compensatory gait strategies that lead to both dynamics and kinematics asymmetries during the gait cycle. These asymmetries result from a reduced weight acceptance control, the fear of falling, or are used as a means of protecting the residual limb [11, 12].

Dynamics asymmetries refer to differences in joint work, vertical ground reaction force (loading) and vertical impulse (push-off) between the sound limb and the prosthetic limb. Indeed one can expect an overall lower joint work provided by the prosthetic limb in comparison with the sound limb resulting from a reduced weight acceptance on the prosthetic side. As a consequence, the sound limb must compensate for the reduced involvement of the prosthetic limb and generates more work (Figure 1.3 (c)). The mean push-off work provided by the prosthetic limb using a passive prosthesis is 0.113 [J/kg] while it is around 0.211 [J/kg] for the non-prosthetic side, which is very similar to the mean push-off work provided by a non-amputee, 0.217 [J/kg] [13]. Vertical and antero-posterior ground reaction force asymmetry can also be observed in lower-limb amputees with an overall lower loading on the prosthetic limb than on the sound limb (Figure 1.2). In addition, according to L. Nolan et al. [12], loading asymmetry can rise up to 23% in the case of lower-limb amputees while it is usually less than 10% for normal gait. This loading asymmetry can be explained by the amputee's higher confidence in the sound limb and by the mass imbalance leading to a potential displacement of the centre of gravity of the body. This asymmetry grows when walking speed increases [11, 12].

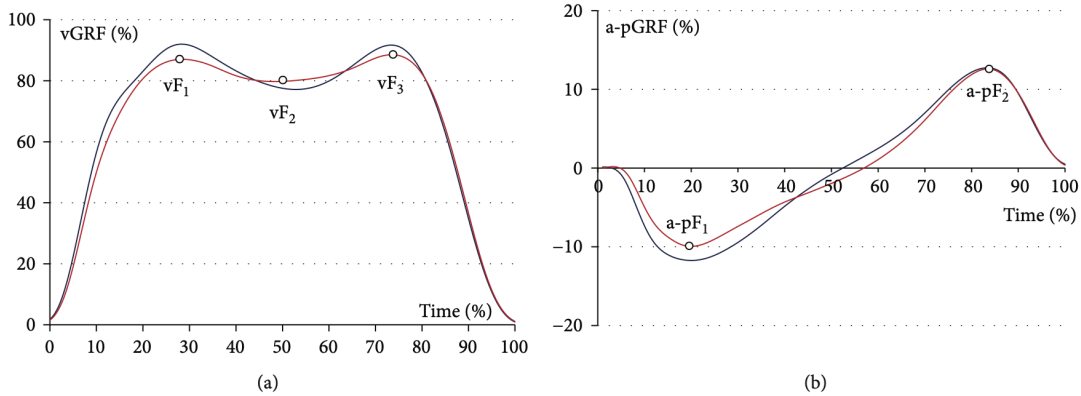


Figure 1.2: (a) Vertical ground reaction force (vGRF), (b) antero-posterior ground reaction force (a-pGRF) components, for the affected side (passive prosthesis, red line) and the unaffected side (blue line) [14].

Kinematics asymmetries refer to a change in the time proportion of the stance phase and the swing phase between both legs. The stance time is often increased on the sound limb while it is decreased on the prosthetic limb [13]. Indeed, as the prosthetic limb is weaker than the sound limb, the amputee tends to reduce the stance time of the prosthetic limb. The heel-strike and heel-off events therefore occur closer in time on the prosthetic side than on the sound side [8, 9, 12]. Moreover, we can observe that amputees' self-selected walking speeds are generally lower than those of non-amputees. This can be explained by the fact that the amount of work required at the ankle joint to walk faster is higher and therefore that the metabolic cost also increases with increasing velocity. In addition, balance deficits may also increase metabolic costs [15].

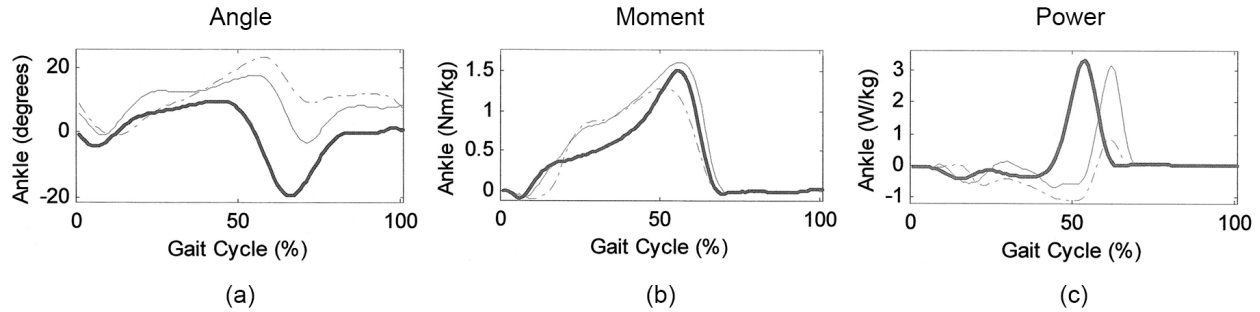


Figure 1.3: (a) Relative angles, (b) Flexion/extension moment, (c) Power, adapted from [16]. *Dotted line* : affected side (passive prosthesis), *Solid line* : unaffected side, *Bold line* : mean normal values.

Comorbidities

As explained above, using a prosthesis causes lower-limb amputees to alter the biomechanics of locomotion. Both dynamics and kinematics gait asymmetries result in a lower gait efficiency which leads to comorbidities constituting a setback in performing daily activities and significantly reduces the quality of life of lower-limb amputees. Major risks imply [10, 12, 17, 18] :

- the inefficiency of the gait strategy that increases energy consumption,
- the balance deficits and the poor postural control that increase the risk of falling,
- musculoskeletal injuries and osteoarthritis in the intact limb due to the intact limb being subjected to increased stress and higher repetitive forces,
- osteoporosis in the residual limb due to insufficient loading of the bones,
- and back pain usually associated to a poor socket fit and alignment, postural changes, or a leg-length mismatch.

1.2. Prosthetic solutions for lower-limb amputees

General purpose

All prostheses share a common goal that consists of replacing the missing joint and mimicking its behaviour to ensure the overall stability of the body and to allow amputees to walk. This way, lower-limb amputees should be able to recover the basic, or even normal function of locomotion, including gait symmetry, speed efficiency and balance. Human locomotion is bipedal and is based on the coordination of the muscles of both legs to store and release the required energy for body support, forward propulsion, etc.

Prosthetic solutions for transtibial lower-limb amputees

The ankle-foot complex is essential in human locomotion as it provides vertical support of the body on the one hand and forward progression on the other hand. Lower-limb amputees' locomotion capabilities are therefore severely impacted and technical solutions to replace the missing limb are required. Different prosthetic devices exist depending on the level of amputation, on the physical condition of the amputee, and on their daily life activities. There are three main types of prostheses (Figure 1.4) : **passive**, **semi-active** and **active** [3, 19]. Passive prostheses are made of passive elements able to store, release and dissipate energy such as springs and dampers, usually in the form

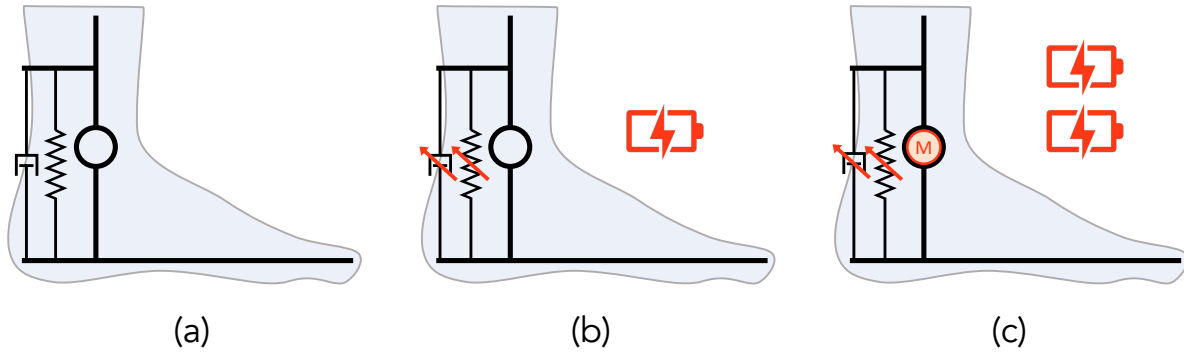


Figure 1.4: Schematic of : (a) Passive prosthesis, (b) Semi-active prosthesis, (c) Active prosthesis.

of a curved blade. Their working principle consists in storing energy during mid-stance and releasing it during push-off, allowing to recover basic standing and walking functions. At most, 0.011 [J/kg] of positive ankle work can be delivered by a passive prosthesis at push-off while about 0.2 [J/kg] of positive ankle work are generated by healthy individuals at their preferred walking speed [3]. On the one hand, passive solutions are extremely robust and require no external energy sources, but on the other hand, they are completely unable to adapt to the task, the user, or the terrain once in use. This is where semi-active prostheses come into play. Their physical properties like damping and stiffness can be actively adapted using a microcontroller, similar to isometric or eccentric muscle functions [19]. Semi-active prostheses can therefore be more reactive but still do not provide the required energy to the user. Their energy consumption is low and mainly driven by the microcontroller. In the case of active or powered prostheses, external power is provided by a motor to assist the amputee during the gait cycle, leading to a higher energy consumption. Many active prostheses combine the active contribution of the motor with passive elements, storing and releasing a part of the required amount of energy during a healthy walking cycle [3, 20]. Although active prostheses are a promising technical solution for lower limb amputees, they still present a lot of issues besides the actuation challenge, including design issues such as their global weight, encumbrance, autonomy and cost but also control issues. In this thesis, the focus is set on the **control** issue of a new active ankle prosthesis, namely the Effective Locking Spring Ankle (ELSA) described in Chapter 2. Indeed active prostheses induce complex control schemes due, among other things, to [3] :

- the need of a reliable interaction between the prosthesis and the amputee,
- the need of adaptation to different locomotive tasks and walking speeds,
- and the complexity of the human walking gait cycle itself.

The lack of efficient control strategies for lower-limb active prostheses significantly reduces the quality of life of amputees and prevents them from taking part in some activities of daily living [21, 22]. The main issue of active prostheses is, therefore, control and more particularly adaptive control. Indeed, advances need to be made to better understand the biomechanics and biological principles that control human motion in order to be able to develop adaptive control strategies for lower-limb prostheses. A major goal is to reduce comorbidities and enhance comfort. Today, it is still very difficult to provide the required assistance for each user no matter the terrain conditions variation, or the speed variation [19, 23]. This is partly due to the need of patient-specific tuning [22].

2. Control strategies for active prostheses/orthoses

2.1. General control framework

Most active lower-limb prosthesis controllers are structured using the generalised framework proposed in Figure 1.5. A hierarchical control structure consists of three different sub-controllers : the **high-level**, the **middle-level**, and the **low-level** controller. High-level control is based on motion intention detection and consists of identifying the current locomotive task, e.g. standing, climbing stairs, slope walking, etc. The middle-level controller converts the user's motion intention at every instant into a desired device state, i.e. the setpoint, by generating patterns and reference signals depending on the current device and user state. Finally, the low-level controller compares the desired device state with the current device state and tries to minimise the error between both by modulating the control commands. The purpose of this thesis is to focus on the middle-level control strategy of the ankle prosthesis prototype ELSA.

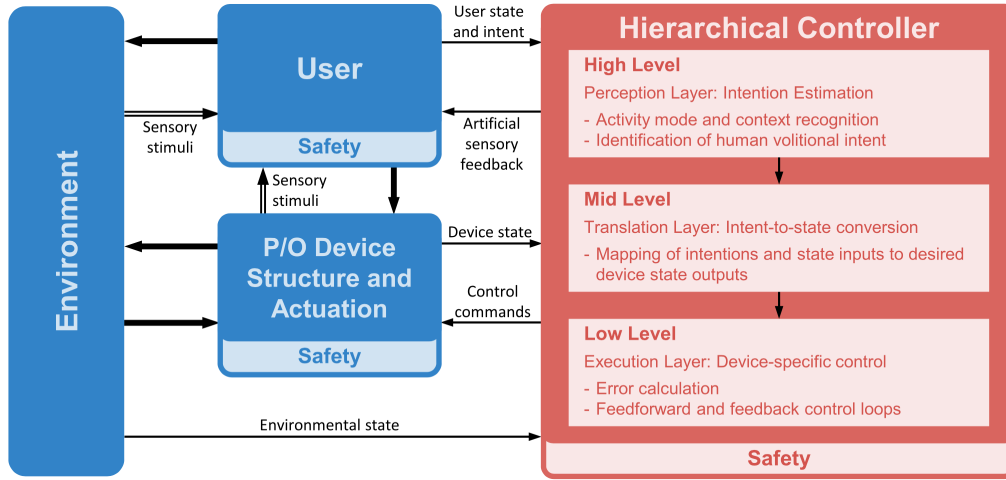


Figure 1.5: Generalised control framework for active lower limb prostheses and orthoses [24].

2.2. State-of-the-art of middle-level control strategies

This section aims at reflecting the state-of-the-art of the middle-level control strategies and is based on a review of active lower extremity prosthetics and orthotics control strategies published by Tucker et al. [24].

Input signals

Different signals can be used as an input for the middle-level controller. Sensors may be **embedded into the prosthesis**, placed on the body surface or directly implanted into the body. Input signals obtained with embedded sensors in the prosthesis allow to determine the device's state. These signals include estimates of joint position (incremental encoders, hall sensors, ...) and joint torque (spring elongation, motor current, ...), angular velocity, ground reaction forces, etc. Other **wearable sensors** may be used for estimating joint positions or limb segment orientations, i.e. goniometers, inclinometers, accelerometers, gyroscopes, magnetometers, and inertial measurement units (IMUs) (Figure 1.7). Besides this, instrumented insoles worn under the foot can be used to sense the ground reaction force (Figure 1.6 (a)). A sensing insole is equipped with several embedded pressure sensors to measure and analyse force distribution on a patient's foot (Figure 1.6 (b)). Ground contact can also be determined by means of foot switches which provide binary information. Estimation of the

centre of pressure (COP) is used in many middle-level control strategies described below such as in finite-state or in virtual constraint controllers. The previously mentioned sensors can be used to determine the progression in the gait cycle, the intention of the user to accelerate or decelerate, etc.

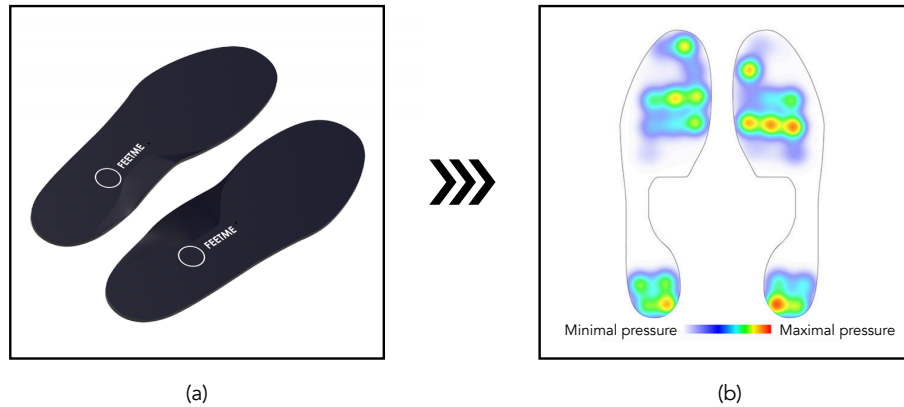


Figure 1.6: (a) FeetMe insole pressure sensor (b) Example of pressure heatmap using FeetMe insole (Source : <https://feetme.fr/en/wearables-devices>).

Surface electromyography (EMG) electrodes provide a measure of the myoelectric signal that characterises the muscle activity (Figure 1.8). Depending on the amputation level and the location of the injury, the muscles controlling the amputated joint may still be present and innervated such that EMG signals can be collected in the remaining limb to detect user's intention and perform control of the artificial joint. EMG is very useful because it provides real-time information that allows to continuously adapt the control to the user and therefore enables speed and terrain adaptability. However EMG signals are not often used as inputs because they present a lot of setbacks that affect their reliability. Indeed, an EMG signal is often very noisy and requires several minutes to be learnt over time. Moreover, for lower-limb amputees, the quality and availability of the signal is highly dependent on the condition of the muscle itself which may have been damaged during the amputation phase, or be atrophied due to a lower activity level, etc. In addition, surface electrodes are not invasive but are less precise than implanted electrodes and can lead to measurement errors. Some control strategies try to solve the estimation precision problem of EMG signals by combining them with other signals such as neuromuscular control described below [24, 21].



Figure 1.7: NGIMU from x-io Technologies (Source : <https://x-io.co.uk/ngimu/>).



Figure 1.8: Surface electromyography electrodes (Source : <https://www.docvadis.fr/>).

As an alternative to electromyography, mechanomyography (MMG) can be used to estimate muscle activity by measuring the vibration of the skin by means of microphones or accelerometers. A significant drawback of this method is the high sensitivity to parasitic noise.

Normalised trajectory control

Normalised trajectory control is based on reference joint trajectories coming from previously recorded gait data and expressed as a function of the gait phase. This way, the reference (typically a desired position pattern) is time independent. The biggest challenge consists in identifying those trajectories.

Echo control

Echo control combines normalised trajectory control to time-based control and consists of reproducing the position trajectory of the sound limb on the prosthetic limb, taking into account some time delay and scaling. Note that some symmetric activities such as sitting-to-standing or standing-to-sitting do not require any time delay. This control strategy is based on the strong assumption that the optimal gait pattern for amputees has to be symmetric. In addition, any compensatory movement of the sound limb is also reproduced in the prosthetic limb although they are not desired.

Complementary Limb Motion Estimation

Complementary limb motion estimation (CLME) uses the position, velocity, or acceleration of intact body parts as an input to predict the intended motion of the affected limb. It is not to be confused with echo control which translates the kinematics of the intact leg to the affected limb with a delay whereas CLME can use any other body part and works in real-time. CLME assumes that there is a strong correlation between the motion of body parts and that it can be used to infer the desired motion of the prosthetic limb. To do so, EMG, angular position or velocity sensors are used on an intact body part, which motion is correlated with the motion of the amputated joint, to record input signals. This technique allows speed adaptability and does not require to implement any delay between the input signals from the intact body parts and the prosthetic limb.

Virtual constraint control

Virtual constraint control uses the COP of the prosthetic foot as a phase variable, as it monotonically increases throughout the stance phase, and the concept of effective rocker shape of the foot. The so-called "effective shape" of the ankle-foot system during walking is defined as the time progression of the COP of the ground reaction force (GRF) in the three-dimensional shank-based coordinate system (Figure 1.9). In the case of an ankle-foot, the effective shape resembles a circular rocker when plotted against the COP phase and is almost invariant with user's weight and walking speed. This effective

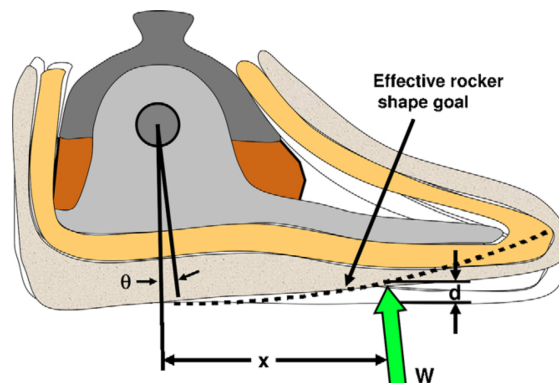


Figure 1.9: Drawing used to model the rotational stiffness needed in a prosthetic single-axis ankle-foot system to mimic effective rocker shapes for able-bodied walking [25].

shape is converted into virtual constraints to determine the actuation of the prosthesis during stance. Another control strategy is used during the swing phase as no COP information is provided when the foot is off the ground [25].

Finite-state control

Finite-state control (FSC) is the most popular middle-level control strategy because it is relatively easy to implement. It applies different control laws (e.g. position or torque control) to each period of the gait cycle. However a different FSC is required depending on the locomotive task detected by the high-level controller. In addition, FSC often requires to tune some parameters in order to be subject-specific and therefore to increase the user's comfort and the gait pattern efficiency. The main disadvantage of FSC is that the number of parameters to be tuned grows exponentially with the number of control laws, the number of periods in the gait cycle, the number of locomotive tasks potentially detected by the high-level controller, etc.

Neuromusculoskeletal model

A neuromusculoskeletal model allows to estimate joints torques based on muscles activity. As the joint has been replaced by a prosthesis, a simple approach consists in directly modulating the actuator's torque based on measured EMG activity of remaining muscles. A dynamic model of the coupled user-device system is required to further estimate the desired joint position, velocity or acceleration. EMG signals can also be used as an additional contribution to the nominal torque provided by another middle-level control strategy.

Impedance control

Impedance is defined as the ratio of a force and a flow, i.e. the ratio of the interaction force between the manipulator and its environment and the robot velocity (eq. 1.1) [26].

$$Z(s) = \frac{\text{Force}}{\text{Velocity}} = \frac{F(s)}{\dot{X}(s)} \quad (1.1)$$

In common cases, impedance is defined based on three parameters, i.e. M , B and K , being the inertia, the damping and the stiffness of the system respectively (eq. 1.2).

$$Z(s) = Ms + B + \frac{K}{s} \quad (1.2)$$

In order to implement impedance control, a dynamic model of the manipulator must first be established. The general model equation is provided in eq. 1.3 where M is the inertia matrix, C refers to the Coriolis effect and the centrifugal force, G is the torques due to gravity while f is the torque due to friction forces, J is the Jacobian relating to the velocity of the considered joint and F_{ext} is the contact force. However in the case of complex systems like an active prosthesis coupled with the body, obtaining an accurate model is not an easy task.

$$M(x)\ddot{x} + C(x, \dot{x})\dot{x} + G(x) + f(\dot{x}) = J^{-T}\tau - F_{ext} \quad (1.3)$$

The second step then consists of determining the desired impedance model based on the specific work requirements (eq. 1.4).

$$M_d(\ddot{x} - \ddot{x}_d) + B_d(\dot{x} - \dot{x}_d) + K_d(x - x_d) = -F_{ext} \quad (1.4)$$

The eq. 1.4 is then substituted in eq. 1.3 to derive a new control law. This way, the effector performs a new dynamic behaviour defined by the impedance model. From a biological point of view, muscles can be seen as impedance adjusters that modify the impedance of the limb depending on the task to perform.

3. Objectives

This master thesis is part of a long-term project that consists of developing a lightweight, energy efficient, and biomimetic active ankle foot prosthesis, ELSA, able to provide sufficient assistance to recover full locomotion faculties. More specifically, this work focuses on the development of an adaptive control strategy aiming at learning from user's dynamics and reducing gait asymmetry throughout the walking gait cycle. To do so, two questions are addressed :

- How can gait asymmetry be measured and reduced in a prosthetic gait?
- How can we perform a compliant actuation with a good tracking performance?

This first chapter described the typical gait disorders encountered after a lower-limb amputation and resultant comorbidities. Also, it briefly introduced the three main types of prostheses and related challenges. Finally, we presented a state-of-the art of the existing control strategies for active prostheses with known advantages and drawbacks.

The second chapter provides a description of the ankle prosthesis ELSA. This includes a description of its mechanical design and its current control strategy as well as the preliminary studies carried out to evaluate its performance and its limitations. This part aims at figuring out the challenges faced with ELSA and outlining the novel control approach.

The third chapter provides a complete description of the proposed novel control strategy with a detailed outline of the real-time gait analysis and the feedback-error-learning parts of the controller. This chapter is therefore a theoretical description of the proposed strategy to both reduce gait asymmetry and perform compliant actuation.

The fourth chapter presents the simplified model developed to perform early validation of the control scheme through multibody simulations with **Robotran**. The goal is to demonstrate the ability of the controller to reduce gait asymmetry and show its adaptive capabilities. The experiments performed in the simulation environment and their main results are provided.

Finally, the fifth chapter raises awareness on the adjustments required to migrate the strategy from the simulation environment to the real prosthetic device. A first-draft implementation based on the Robot Operating System (ROS) is described to show what has already been done and what needs to be done in the future. Additional experiments are also conducted with the prosthesis to validate the real-time gait analysis, and describe the technical issues encountered.

1. Mechanical design of the ELSA prosthesis

This thesis is based on an active ankle-foot prosthesis developed by F. Heremans for transtibial amputees as a Ph.D. thesis [27]. This prosthesis was specially designed to be ultra-lightweight, affordable, energy efficient and biomimetic. It provides one rotational degree of freedom in the sagittal plane, allowing foot plantar- and dorsi-flexion. Its design is bio-inspired and consists of a lockable parallel spring (LPS) which role is to mimic the Achilles tendon and plantar fascia behaviour, as well as a series elastic actuator (SEA) which is intended to provide the additional required energy for push-off (Figure 2.1). It has been demonstrated that combining a lockable parallel spring with a series elastic actuator is globally energy efficient [28]. Given these properties, the prosthesis is designated as ELSA for *Effective Locking Spring Ankle*.

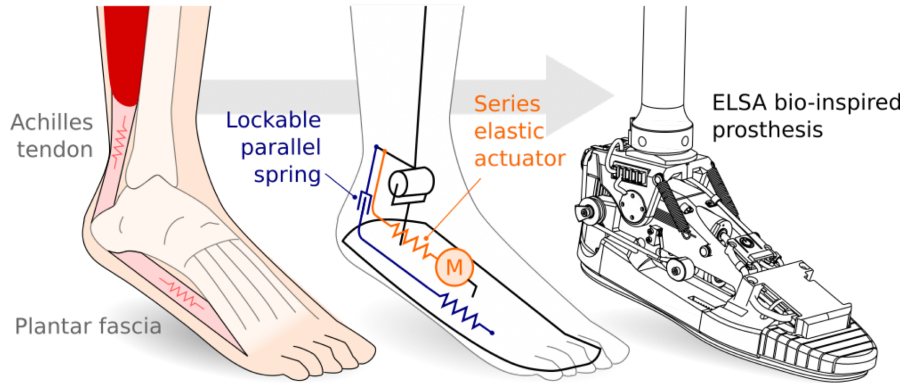


Figure 2.1: Bio-inspired prosthetic foot ELSA [29].

The LPS is used differently in the stance phase and in the swing phase. In the stance phase, the LPS is engaged in order to store and release energy while it is disengaged in the swing phase to allow foot dorsiflexion. The maximum plantarflexion angle following heel strike is used as neutral position for the spring. As shown in Figures 2.2 and 2.3, the rope locking system is composed of a ratchet and pawl mechanism actuated by a micro servo via an elastic link. This way, the timing between actuation and engagement is decoupled. Indeed, as the servo closes, the elastic link and the asymmetric geometry of the teeth allow the pawl to retract as the spring rewinds the rope during the plantarflexion motion following heel-strike. When the joint velocity reverses after maximum plantarflexion, the pawl engages and the system is loaded. This allows the closing actuation time to precede the time at which maximum plantarflexion occurs. In a similar way, when the servo opens, the pawl keeps engaged until the load disappears, due to the friction of the teeth when the elastic link is loaded [27]. Note that the load disappears after push-off when the ankle comes back to the position of maximum plantarflexion reached after heel strike. It is essential to understand this mechanism for the conception of the control strategy.

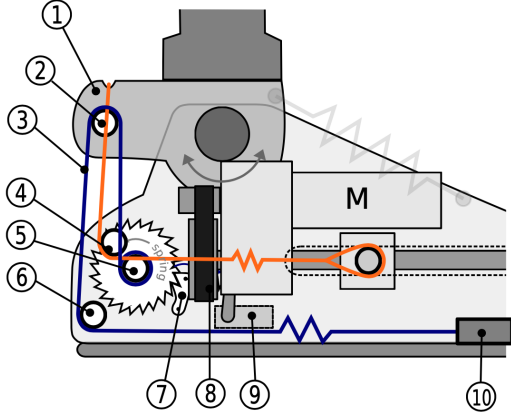


Figure 2.2: Synthetic view of the LPS and SEA : (1) shank lever, (2) shank pulley, (3) nylon elastic rope, (4) ratchet wheel, (5) ratchet drum, (6) heel pulley, (7) pawl, (8) elastic link, (9) micro servo, (10) adjustable clamp [27].

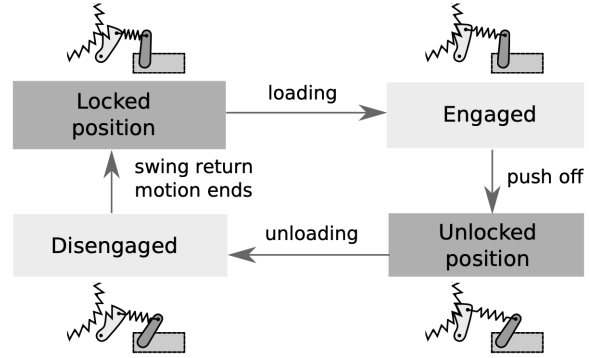


Figure 2.3: Locking states including actuation and engagement [27].

Besides this, the SEA uses a second elastic transmission rope coupled to a ball screw actuator. As depicted in Figure 2.2, a nut is mounted on the screw and linked to the rope. A joint torque is generated by pulling on the elastic rope. The generated torque can be estimated using the measured elongation, i.e. the difference between the nut position and the nut neutral position. More details concerning the mechanical design can be found in dedicated articles published by F. Heremans [27, 28, 29]. The combined action of the LPS and SEA has proved to allow a faithful reproduction of the dynamics of a sound ankle along the gait cycle, while reducing the amount of active energy required. This way, the SEA has to generate only a fraction (about half) of the total torque generated by the system. Note that the prosthesis can be seen as a torque source for higher level controllers, as long as the LPS is engaged and disengaged at the right moment.

2. Current control strategy of ELSA

The control structure developed by F. Heremans and described in [27] uses the general framework presented in Figure 1.5 with a high-level, a middle-level, and a low-level controller.

High-level controller The high-level controller is trivial and requires to manually switch from one locomotion task to another, namely *flat ground walking*, *stairs climbing*, and *stairs descending*. Until now, no research effort has been made to further develop that control level and perform an automatic intention recognition.

Middle-level controller The middle-level controller implemented is structured as a finite-state machine (FSM). The controller switches between three different stages of the gait cycle : early-stance, full-stance and swing, as depicted in Figure 2.4. Early-stance starts when the initial contact between the foot and the ground, i.e. the heel-strike, is detected. The foot then moves to a flat position and reaches the full-stance stage where the push-off is provided. The foot finally enters in the swing stage when the toe-off event is detected, i.e. when the foot leaves the ground. Note that the current version of the controller does not compute any gait phase variable and uses only internal variables (joint angular position, nut position, ...) as a trigger to switch from one state to another.

The middle-level controller provides at every instant a setpoint, holding the desired position of the nut x_{des} , the desired speed v_{des} , current i_{des} and torque at the joint τ_{des} , as well as the desired state for the locking mechanism (Figure 2.3). This vector $[x_{des}, v_{des}, i_{des}, \tau_{des}]$ determines the behaviour of the low-level controller [27]. There is a specific setpoint pattern for each stage of the FSM.

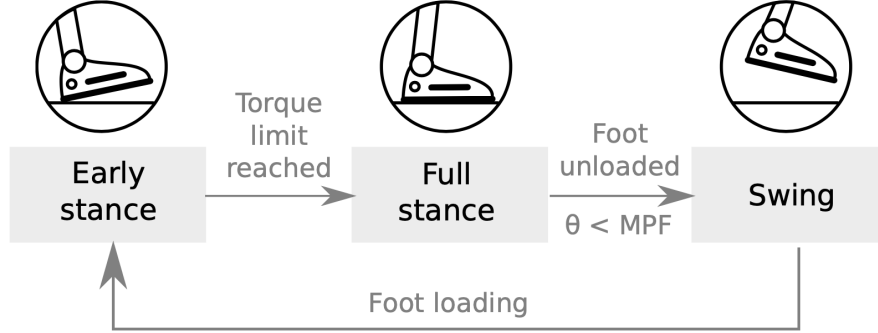


Figure 2.4: State machine of the middle-level controller [27].

Low-level controller The goal of the low-level controller is to reach the desired state of the system. It is implemented as a set of nested loops. A subset of this cascade control architecture can be used to perform either current, speed, position or torque control. The control strategy currently running on the prosthesis performs a position tracking during the swing phase and a current tracking during the stance phase. The setpoint $[\tau_{des}, x_{des}, v_{des}, i_{des}]$ provided by the middle-level controller determines the targeted reference for the chosen control type, i.e. τ_{ref} , x_{ref} , v_{ref} , **or** i_{ref} , and boundary intervals for internal loops. Figure 2.5 depicts the structure of the low-level controller and emphasises these dynamical boundary adjustments. For instance, when performing a position control, the "torque control" dedicated part is bypassed and x_{ref} is directly set to x_{des} . As the position of the nut directly determines the traction force of the series elastic, the reference position is first bounded by the screw hard limits $[x_{min}, x_{max}]$ and then by the desired maximum torque $[x_{-\tau_{des}}, x_{\tau_{des}}]$. The same approach applies at the speed control stage where the nut speed reference v_{ref} provided by the first proportional integral derivative (PID) block is bounded by security limits $[-v_{max}, v_{max}]$ and by the desired maximum speed $[-v_{des}, v_{des}]$. An extra security layer is added in order to reduce the speed to zero as the nut gets close to the screw hard limits. Finally, the current control is very similar with boundaries $[-i_{max}, i_{max}]$ and $[-i_{des}, i_{des}]$. Note that the formula used for torque control can easily be derived from Hooke's law. We are looking for the nut position required for the SEA to produce the desired joint torque (eq. 2.1 and 2.2).

$$\tau_{SEA} = L \cdot F_{SEA} \quad (2.1)$$

$$F_{SEA} = K_{SEA} \cdot (x - x_0) \quad (2.2)$$

with L the lever arm of (1) in Figure 2.2. Inverting this relation provides x_{ref} , the nut position required to produce the reference torque. A geometrical model has been developed and calibrated to estimate the spring neutral position x_0 in real-time, taking into account the dead zone where the rope is not fully stretched. This technique allows to perform torque control very easily without the need for pressure sensors. Since a precise torque control is not required, it has been decided until now to use a current tracking rather than torque tracking to enhance robustness.

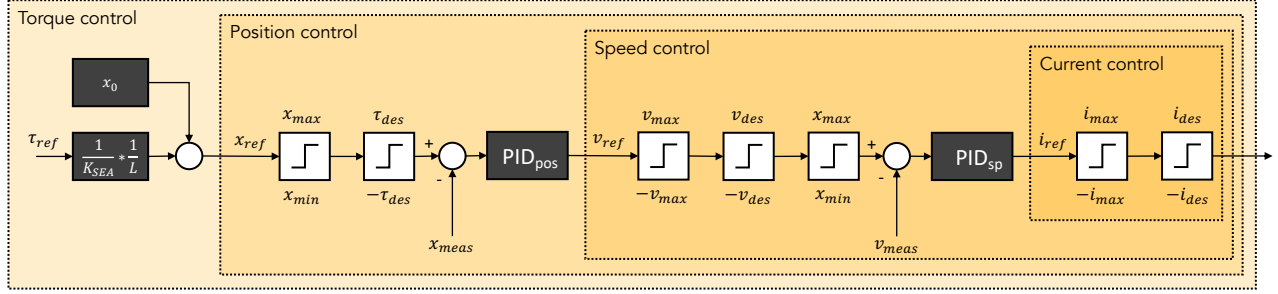


Figure 2.5: Diagram of the low-level controller.

3. Preliminary studies

Several experiments have been performed to evaluate the performance of the prosthesis and its current controller, first using a testbench, then a healthy user adapter, and finally with transtibial amputees.

3.1. Low- and middle-level controllers validation

The controller was first validated by F. Heremans on a testbench to show that typical healthy torque trajectories can be generated by the device. To do so, a typical ankle joint trajectory [30] was imposed on the prosthesis by the testbench. Figure 2.6 shows the normalised prosthesis torques measured by the testbench and compares them to a typical Winter reference torque pattern. One can observe that the prosthesis is able to provide a total torque that closely matches the Winter reference torque throughout the gait cycle and that the SEA generates about half of the total reference torque.

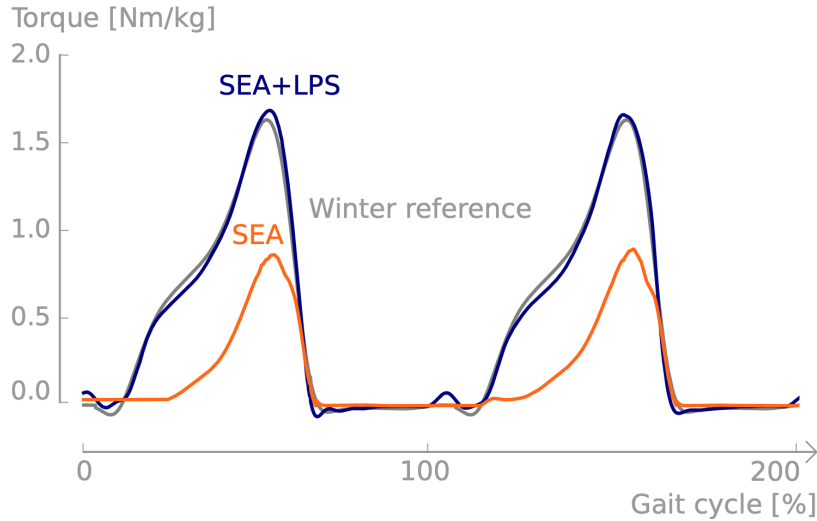


Figure 2.6: Normalized prosthesis torques measured by the testbench compared to a reference torque pattern from Winter (grey) [27].

In a second phase, a healthy user adapter was used to test the device in more realistic conditions and collect quantitative measurements from the prosthesis. As shown on Figures 2.7 and 2.8, the device proved to be capable of faithfully reproducing the kinematics and dynamics of a sound ankle along the gait cycle.

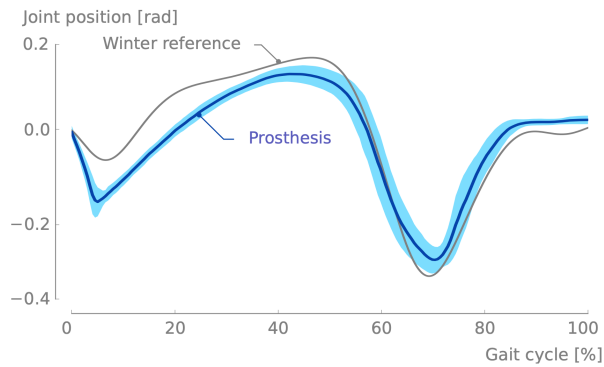


Figure 2.7: Angular position of the prosthesis (blue) throughout the gait cycle compared to a reference position pattern from Winter (grey) [27].

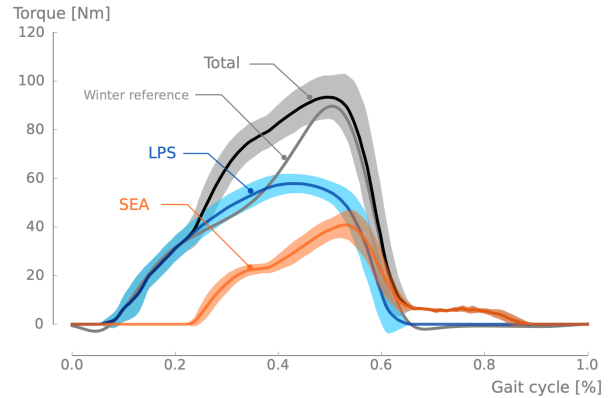


Figure 2.8: Torque profiles (total (black), LPS (blue), SEA (orange)) of the prosthesis throughout the gait cycle compared to a reference torque pattern from Winter (grey) [27].

Given these satisfactory performances, two experiments were performed on transtibial amputees and are described hereafter.

3.2. Experiment 1 : Ossür

A first experiment was conducted on the 31st of October 2019, with a transtibial amputee at the Ossür motion capture laboratory in Reykjavik, world leader in non-invasive orthopaedic solutions. Taking part in the experiment was a good opportunity for us to better understand how the prosthesis works and the challenges related to prosthetic devices in general. During this experiment, no specific experimental protocol was used. The goal was to present the conceptual idea and prototype of ELSA (Figure 2.9) to Ossür and to convince them about its innovative mechanical design for further collaboration. Unfortunately, no quantitative results were recorded due to mechanical issues with the prosthesis, but the user provided feedback about his first impressions. This was the first time that he could wear an active prosthetic device due to his stump being too long to be able to wear any



Figure 2.9: Prototype of the ELSA prosthesis made by F. Heremans.

existing active ankle prosthetic device. Throughout the experiment, it resulted that he could freely wander around and appreciated the active contribution of the prosthesis. It turned out that ELSA provided him enough assistance to allow him to walk on flat ground (Figure 2.10) and up (Figure 2.11) and down (Figure 2.12) the stairs. However, some parameters had to be manually tuned to reach a satisfactory behaviour, especially the angular threshold to trigger the active push-off, and the user was not very confident despite his wide experience with his passive prosthetic foot. This last point emphasises the fact that the behaviour of any prosthesis must be fully predictable for the user to feel confident. It means that any adaptation mechanism developed in this thesis must have a sufficient inertia of adaptation.

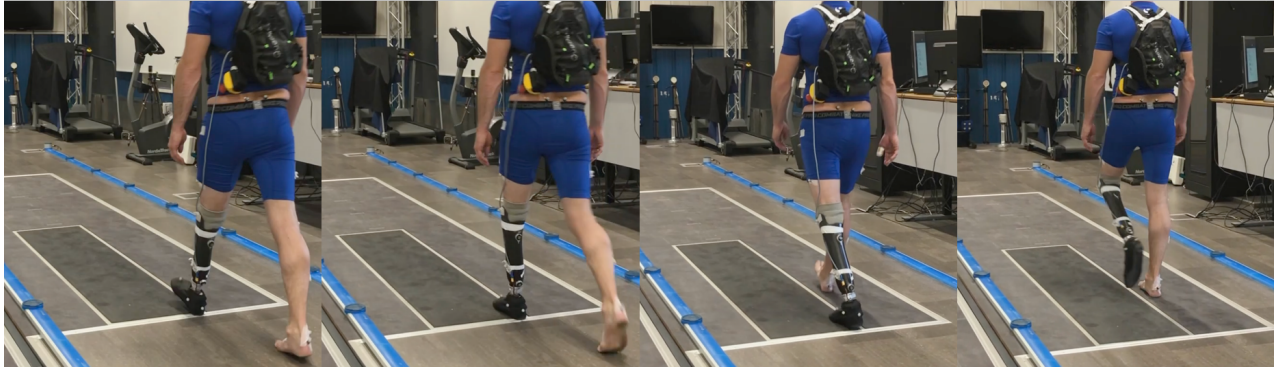


Figure 2.10: Transtibial amputee walking on flat ground with ELSA, Ossür, Reykjavik.



Figure 2.11: Transtibial amputee climbing stairs with ELSA, Ossür, Reykjavik.



Figure 2.12: Transtibial amputee walking down stairs with ELSA, Ossür, Reykjavik.

3.3. Experiment 2 : Cliniques Universitaires Saint-Luc

In order to record quantitative data, we conducted a second experiment with another transtibial amputee on the 19th of December 2019. The objective of the experimental study was to evaluate the performance of the device with an amputee. In particular, the control aspects were evaluated in order to identify both the strength and limitations of the current control strategy.

Experimental protocol

The experiment was carried out at the gait laboratory of the cliniques universitaires Saint-Luc (CUSL). This laboratory benefits from motion capture equipments and a treadmill with force sensors which allow to measure both the body kinematics and interaction forces with the ground thanks to the **Vicon Nexus** software. The motion capture system requires reflective markers to be placed on the body of the amputee to automatically reconstruct a bipedal model of the person. Different models can be used for the markers placement. In our case, we used the *lower limb and shoulder* model with 18 markers (Figure 2.13). Among these 18 markers, 14 are placed symmetrically on the body, i.e. 1 on each : acromion (prominence between the humeral head and the scapula), antero superior iliac spine (ASIS) (the prominence of the anterior edge of the pelvic bone), posterior superior iliac spine (PSIS) (the prominence of the posterior edge of the pelvic bone), lateral condyle of the femur, heel, malleolus and second metatarsal bone. Four other markers are placed on the right and left legs asymmetrically in order to allow the software to make the difference between the right and left legs : 1 on the upper third of the thigh and 1 on the upper third of the lower leg and, 1 on the lower third of the thigh and 1 on the lower third of the lower leg, for the right leg and left leg respectively.

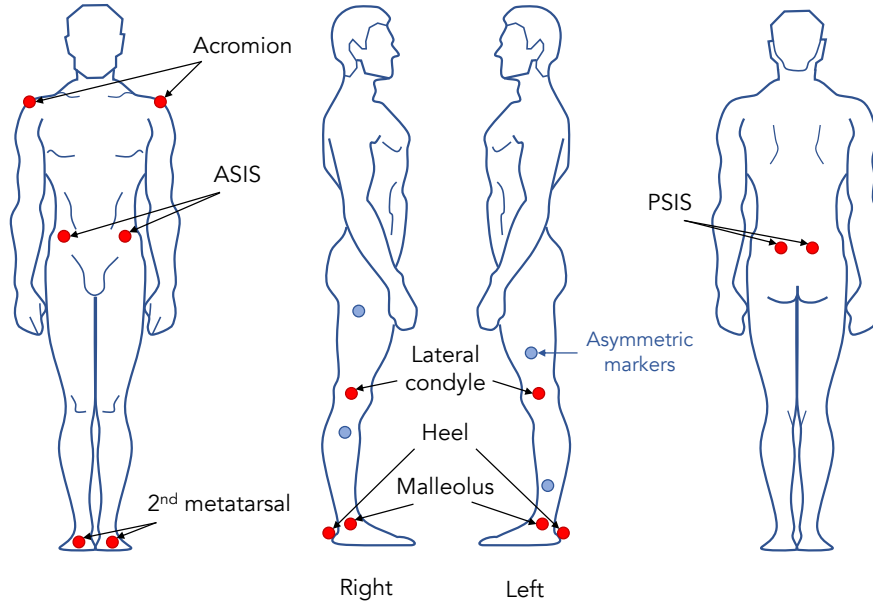


Figure 2.13: Position of the markers for motion capture.

The complete protocol to test the ELSA with an amputee is provided in Appendix B. The main schematic summarising the steps of the experiment is visible in Figure 2.14. The experiment was first performed with the amputee's daily life passive prosthesis and consisted of three identical trials on the treadmill, the only difference being the walking speed at which the amputee walked, i.e. the amputee's natural speed $\pm 15\%$. For each trial, a sequence of 30 seconds was performed without recording to allow the amputee's gait to reach a steady state. Then, the next 30 seconds were recorded. The

whole experiment was then repeated with the ELSA prosthesis and the current control strategy. Figure 2.15 is a sequence of pictures showing a complete gait cycle of the amputee walking with the ELSA prosthesis on the treadmill using the current control strategy.

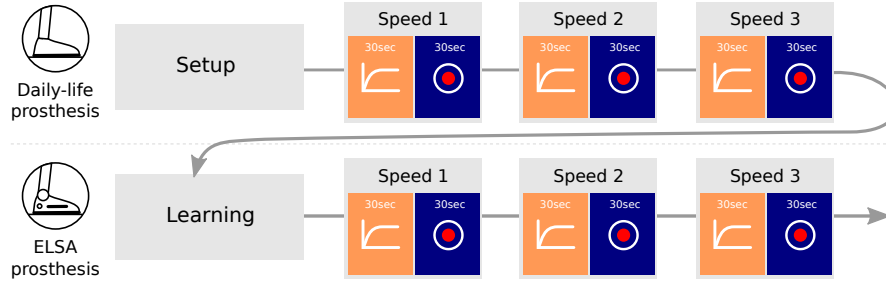


Figure 2.14: Experimental protocol to test the ELSA prosthesis with an amputee (Appendix B).

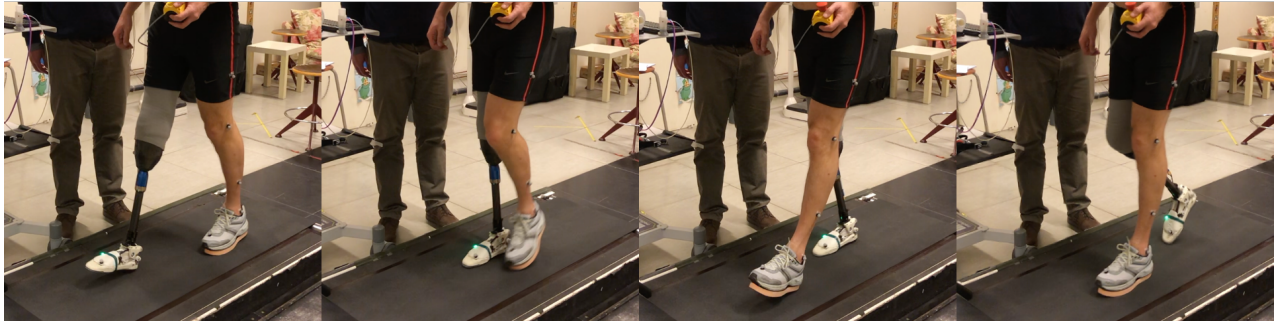


Figure 2.15: Transtibial amputee walking on a treadmill with ELSA, CUSL.

Data analysis

Based on markers trajectory and ground reaction forces, joints angular displacement, moments and power were approximated using a musculoskeletal model developed by C. Detrembleur. Average results are presented in Figure 2.16 for a natural walking speed of 4 [km/h] on the treadmill. From this figure, one can observe that the torque profile generated by the active prosthesis closely matched the average sound profile and did not become negative after heel-strike like the daily life prosthesis. The

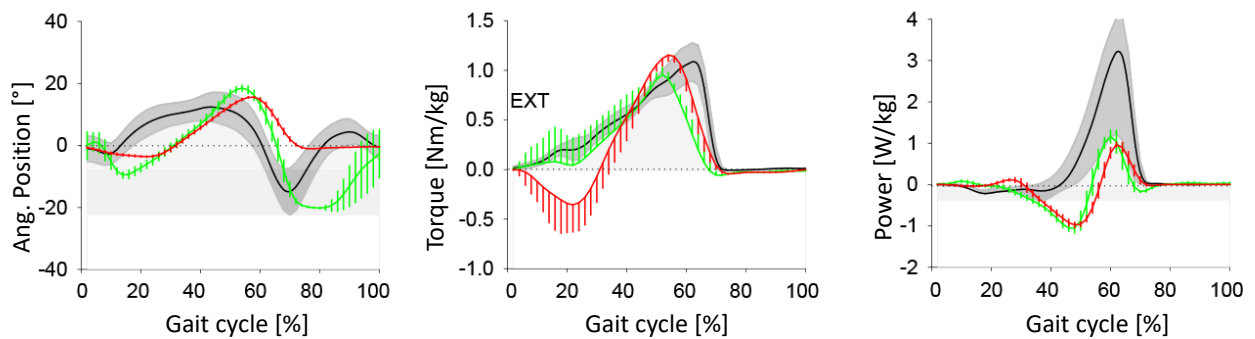


Figure 2.16: Data measured at the gait laboratory for walking speed of 4 [km/h], compared to averaged sound profiles measured at the lab. Daily life passive prosthesis - red, active prosthesis ELSA - green, reference pattern - grey, mean $\pm$ std.

amount of positive power provided by the active prosthesis was also slightly bigger, even though the difference was negligible. However, one can observe that the active assistance provided during push-off faded a bit too early, and the dorsiflexion following toe-off was way too slow. As a consequence, the amputee was afraid of stumbling. Some parameters would have required to be fine tuned, such as the trigger condition to leave the stance phase and the nut velocity during swing phase.

3.4. Advantages and limitations of the current control strategy

The current control strategy has proved to work properly and should already be sufficient to recover basic locomotion. Still, it has several major drawbacks :

- the torque reference used during the stance phase is based on healthy patterns provided by D. Winter [30] (averaged Winter torque profile) to ease comparison with other existing devices,
- no evidence that this pattern is optimal for amputees,
- no inter-subject adaptation,
- no speed adaptation,
- no gait asymmetry correction mechanism,
- many parameters which require manual patient-specific tuning within the finite-state machine.

In order to introduce adaptability in the current middle-level control strategy, many parameters would require to be dynamically modified, such as trigger values to switch from one state to another, speed limits, or torque reference. A possible approach would be to perform parameters adaptation using heuristic laws, highly based on laborious experimental testing. The main challenge would be to provide an adaptive current or torque reference pattern during the terminal stance (between heel-off and toe-off events). A purely heuristic based approach is not ideal since it is established on a rigid scheme where a specific user behaviour leads to a specific parameter variation, with no assurance that this modification is suitable for the actual user. For this reason, it would be convenient to minimise the number of manually tuned parameters and heuristic laws. With all of this in mind, we have decided to opt for a different control structure, more suitable for adaptation, while maintaining the FSM control for the lockable spring. Security mechanisms have to be considered to ensure a proper engagement and disengagement of the LPS, despite the evolving behaviour of the SEA.

1. General idea of the control strategy

Let us summarise the targeted goals and introduce the explored solutions. As explained in Section 1.1. of Chapter 1, gait asymmetry is responsible for comorbidities and a reduced quality of life. For this reason, the assumption is made that a healthy gait pattern for lower-limb amputees should be symmetric. In order to measure asymmetry, we have decided to monitor and compare the kinematics of the two legs using IMUs. The signal of these sensors are learned by adaptive oscillators (AOs) for low pass filtering, and the derivation of a gait phase variable. Then, we believe that the asymmetry reduction could be performed through a kind of position control. An interesting way of representing and adapting a reference signal is to use motor primitives as a function of the gait phase [%]. Simple heuristic laws are used to adapt the motor primitives based on measured asymmetries. In addition, we believe that an efficient asymmetry reduction algorithm could in fact allow speed and slope adaptation by progressively mimicking the behaviour of the sound leg. Concerning actuation, a common position control method based only on feedback is not compliant nor appreciated by users. As explained by A. Calanca [31], dealing with unstructured environments, as for human-robot interactions, requires compliant control strategies. For this reason, a novel method using real-time inverse dynamics model estimation is experimented. This novel method is designed to learn from the user's behaviour and from the environment and is based on the locally weighted projection regression (LWPR) non-linear function approximator.

The idea of the proposed control strategy is to perform an adaptive **compliant position control** which tries to minimise gait asymmetry using adaptive oscillators for gait analysis and a linear combination of adaptable motor primitives as a reference signal. The control scheme is presented in Figure 3.1 and is composed of the nine following functional blocks :

Kinematics Acquisition of the angular velocity of both shins in the sagittal plane. This is the raw signal that is used for gait analysis.

Ground contact Binary detection of ground contact.

Adaptive oscillators Continuous learning of the angular velocity signal by adaptive oscillators to provide a smooth signal approximation, as well as a gait phase linearly increasing from 0 to 100 [%].

Gait events detection Analysis of the learned angular velocity signal to detect key events throughout the gait cycle based on a robust event detection algorithm.

Asymmetry measurement Comparison of phase variable values at each event between both legs. Asymmetry indexes are generated.

Heuristic laws Adaptation of the motor primitives based on the sign and magnitude of the asymmetry index.

Reference generation Generation of the prosthesis reference position based on parametric gaussian motor primitives.

Adaptive cerebellar contribution Incremental learning of the user-specific inverse dynamics model. Provides a real-time estimation of the torque required to reach a desired state based on current state and gait progression. The estimation is used as a feedforward action.

Proportional derivative (PD) regulator Compensation of the remaining error. Although very small, this contribution is essential to introduce adaptation to the user (body mass, dynamics, walking speed, ...) and to the environment (slope, ...) by allowing to update the inverse model.

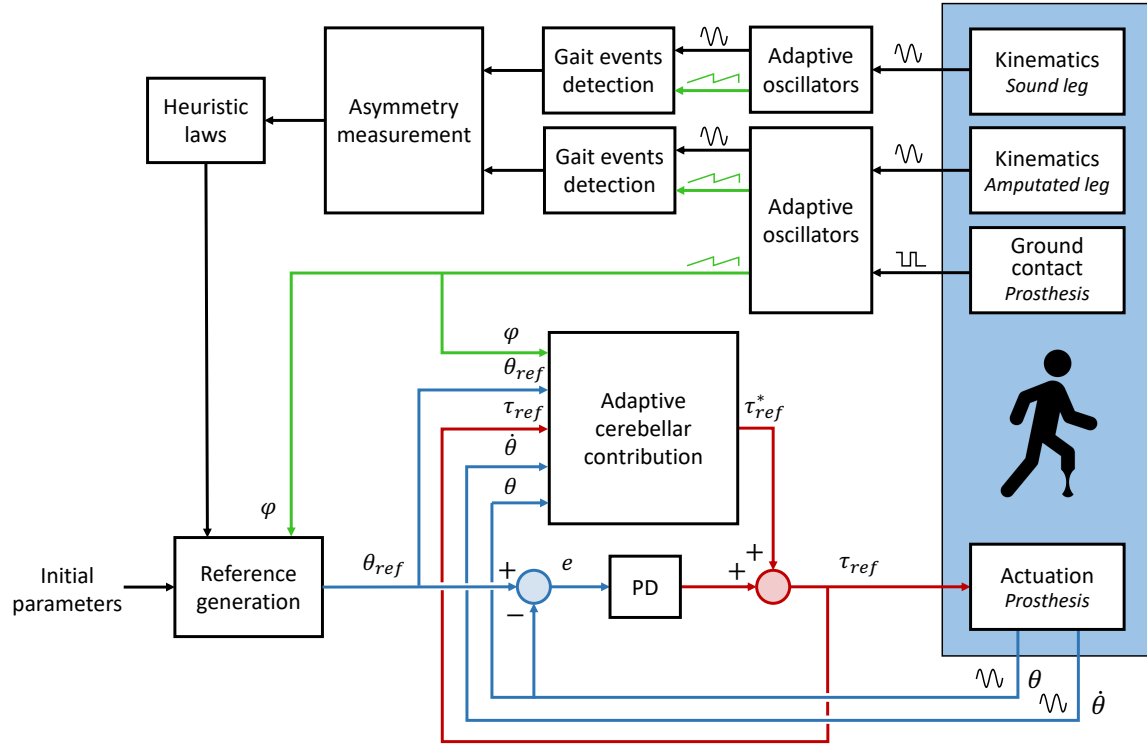


Figure 3.1: Block diagram of the control structure.

Let us now dive into the detailed implementation of these functional blocs, organised in two main sections : real-time gait analysis and feedback-error-learning control.

2. Real-time gait analysis

2.1. Kinematics acquisition

Real-time gait events detection and parameters estimation can be performed using IMUs, a combination of accelerometers and gyroscopes, placed on strategic locations. When dealing with the lower-limb, most methods use shank- and/or foot-mounted IMUs. These devices are used in the frame of this thesis to gather information about the kinematics of the two legs.

The main advantage of IMUs is that they can be worn on different parts of the body and represent a minor discomfort to the user as they are small and lightweight. IMUs can either be placed on the

thigh, the shank or the foot. However, a previous study claims that a placement of the IMU on the shank offers some advantages and may provide more accurate measurements [32]. Indeed a shank-mounted IMU leads to less signal variability compared with a foot-mounted one and less tissue movement compared with a thigh-mounted one. It has been shown by A. Laudanski [33] that foot-mounted IMUs are more likely to be affected by foot deviations from the sagittal plane, i.e. the progression plane, during the gait cycle which results in a part of the acceleration being lost and therefore in poorer estimations of forward velocities. In addition, IMUs placed on the shank have proven to detect gait events with higher accuracy for both healthy and prosthetic gaits [32].

Figure 3.2 shows two possible locations for the IMUs, i.e. shank-mounted and foot-mounted, as well as the relation between the local coordinate system $\{t,n\}$ and the world frame $\{x,y\}$. The gyroscope axes lay orthogonally to the plane defined by these axes. Some sources [32, 33, 34, 35, 36] claim that an efficient gait event detection algorithm can be implemented using only the angular velocity of the limb in the sagittal plane. To visualise the expected measured signals for the two considered positions, Figure 3.3 compares the angular velocity of the shank and of the foot along a complete gait cycle. One can observe that the gait event detection algorithm would be slightly different depending on the position of the IMU. The angular velocity of the foot nearly reaches 0 at foot-flat whereas the angular velocity of the shank is near 0 when the shank is vertical, i.e. at mid-stance. Note that if the gait is perfectly symmetric, the angular velocity is exactly the same for both the right and left shanks at corresponding states of their gait cycle, and therefore for the right and left feet [33].

For these reasons, two IMUs are used in total, one on each leg. The first one is placed on the lateral side of the shank, right above the ankle of the sound limb while the other one is placed on the lateral side of the rod connecting the prosthetic foot to the socket, at the same level. This placement configuration allows to provide the angular acceleration of the shank in the sagittal plane, i.e. z -axis [33]. Note that the IMUs are configured to integrate the signal and directly provide the angular velocity $[\text{°/s}]$ of the shank, as shown in Figure 3.3.

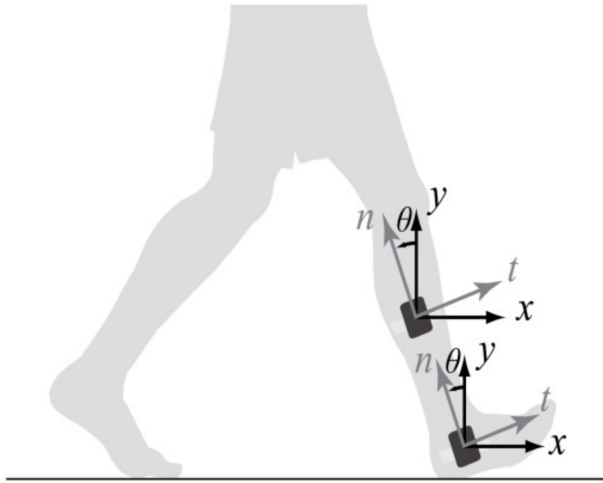


Figure 3.2: IMUs placed on the lateral side of the shank and the foot. The n and t -axes measure the normal and tangential accelerations respectively. The x and y -axes refer to the world coordinate system xoy , with the y -axis being aligned with gravity. The angle θ is defined as the angle between the n -axis and the y -axis [33].

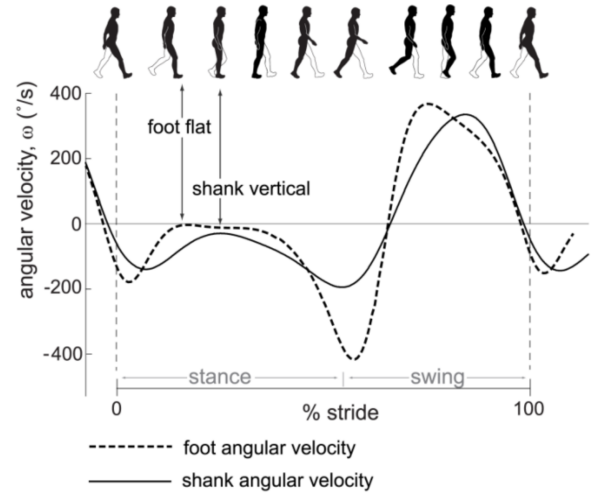


Figure 3.3: Angular velocity ω as a function of the percentage of advancement in the gait cycle. The dotted line and the solid line refer to the foot angular velocity and the shank angular velocity of the right leg respectively [33].

2.2. Ground contact

Either an instrumented insole or foot switch is used to provide binary information of prosthesis contact with the ground. This variable is mainly used for accurate synchronisation of the estimated gait phase variable with the physical heel-strike. Only the phase variable used as input for the angular position reference generation must be precisely synchronised with the physical heel-strike. On the other hand, asymmetry measurement does not require precise phase synchronisation as the comparison is relative between both legs. Finally, the binary information of ground contact is essential to choose the appropriate set of gains and weights for the PD regulator and adaptive cerebellar contribution since the impedance of the environment changes drastically between the swing and the stance phases.

2.3. Adaptive oscillators

Adaptive oscillators are a mathematical tool which goal is to approximate a raw signal by a linear combination of sinusoidal waves, progressively learning its features (phase, frequency, amplitude). This approach is therefore suitable in the case of periodic and stable tasks, like walking in stationary mode. By their internal structure, adaptive oscillators filter out the raw signal and provide a signal estimate that is phase-synchronised, i.e. no delay between the raw signal and the signal estimate [37, 38]. Besides signal learning, the interest of adaptive oscillators is to derive a phase variable which indicates the current progression within the current cycle. The phase synchronisation strongly depends on the ability to robustly identify a biomechanical event in the raw signal to indicate the start of a new cycle. Regarding lower-limb kinematics, the heel-strike event is usually chosen as reference for a new gait cycle. We have decided to stick to this convention.

Signal learning

The paper published by T. Yan et al. [37] provides a method to implement the adaptive oscillators. The input signal, i.e. the angular velocity from each IMU, is decomposed into a finite number of harmonics N . We have chosen to use 3 harmonics which is a good compromise to approximate the shape of this signal. Each oscillator learns the phase, frequency, and amplitude of its associated harmonic to obtain the best estimation of the input signal. The evolution of the system is driven by the error between the input signal and the current approximation. The integration of equations 3.1 to 3.4 over time allows to compute the previously mentioned features for each time step. A diagram of the system is depicted in Figure 3.4.

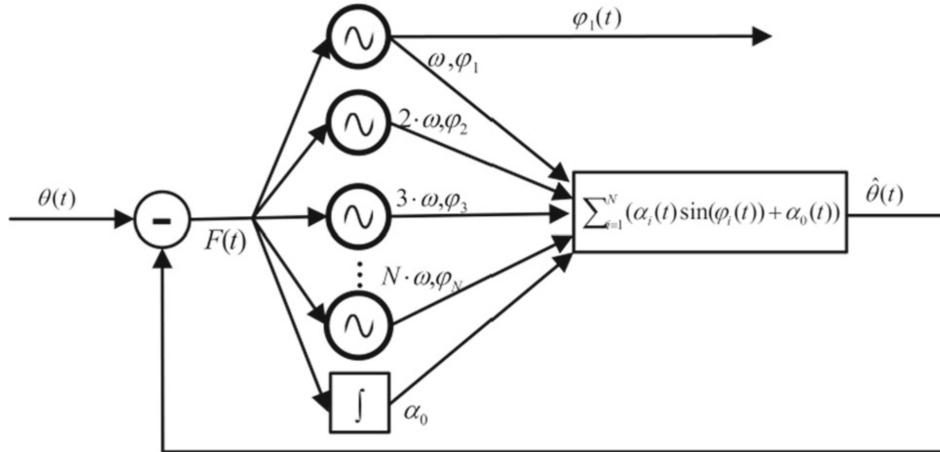


Figure 3.4: AO block diagram [37].

$$\dot{\phi}_i(t) = \omega(t) \cdot i + \nu_\phi \frac{F(t)}{\sum \alpha_i} \cos(\phi_i(t)) \quad (3.1)$$

$$\dot{\alpha}_i(t) = \eta F(t) \sin(\phi_i(t)) \quad (3.2)$$

$$\dot{\omega}(t) = \nu_\omega(t) \frac{F(t)}{\sum \alpha_i} \cos(\phi_1(t)) \quad (3.3)$$

$$\dot{\alpha}_0(t) = \eta F(t) \quad (3.4)$$

Where ω is the fundamental frequency, ϕ_i and α_i are the phase and amplitude of the i th oscillator, and α_0 is the unique offset component. $F(t) = \theta(t) - \hat{\theta}(t)$ is the difference between the input signal and the estimated signal. The latter is computed as a sum of the N harmonics (eq. 3.5).

$$\hat{\theta}(t) = \alpha_0(t) + \sum_{i=1}^N \alpha_i(t) \sin(\phi_i(t)) \quad (3.5)$$

Three relationships (3.6, 3.7 and 3.8) have been established by R. Ronsse et al. [38] to tune the gains of equations 3.1 to 3.4 in order to make the oscillator amplitude α and frequency ω converge with desired time constants τ_α and τ_ω .

$$\nu_\omega = \frac{20}{\tau_\omega^2} \quad (3.6)$$

$$\nu_\phi = \sqrt{24.2\nu_\omega} \quad (3.7)$$

$$\eta = \frac{2}{\tau_\alpha} \quad (3.8)$$

In this thesis, the frequency and amplitude learning time constants have been set based on R. Ronsse et al. [38] to $\tau_\omega = T$ and $\tau_\alpha = 2.8T$, with T the step duration.

Gait phase estimation

A continuous phase variable is determined based on the time elapsed since previous gait event and the phase of the fundamental harmonic ϕ_1 learned by the adaptive oscillators. It is expected to increase linearly from 0 to 100 [%] for a complete gait cycle between two heel-strike events.

$$\phi_n(t) = \text{mod}(\phi_1(t), 2\pi) \quad (3.9)$$

However, this does not ensure that $\phi_n(t_k) = 0$ for t_k , the time at which the heel-strike event is detected. For this reason, a phase error $P_e(t_k)$ is computed as the error between the learned phase $\phi_n(t)$ and 0 or 2π depending on whether ϕ_n has been reset before or after the detected event :

$$P_e(t_k) = \begin{cases} -\phi_n(t) & \text{if } 0 \leq \phi_n(t_k) < \pi \\ 2\pi - \phi_n(t) & \text{if } \pi \leq \phi_n(t_k) < 2\pi \end{cases} \quad (3.10)$$

The phase error $P_e(t_k)$ is updated to avoid inconsistencies when $\phi_n(t_k)$ fluctuates around π rad :

$$\tilde{P}_e(t_k) = \begin{cases} P_e(t_k) - 2\pi & \text{if } P_e(t_k) > \frac{\pi}{2} \text{ and } \tilde{P}_e(t_{k-1}) < -\frac{\pi}{2} \\ P_e(t_k) + 2\pi & \text{if } P_e(t_k) < -\frac{\pi}{2} \text{ and } \tilde{P}_e(t_{k-1}) > \frac{\pi}{2} \\ P_e(t_k) & \text{otherwise} \end{cases} \quad (3.11)$$

Now that the error is known, a phase error learning mechanism is required to correct the estimated phase variable ϕ_n . To do so, the state variable ϕ_e learns the phase error signal $\tilde{P}_e(t_k)$ by integrating expression 3.12 over time and continuously corrects the estimated phase variable ϕ_n to enhance its accuracy.

$$\dot{\phi}_e = \epsilon(t_k)\omega \exp(-\omega(t - t_k)) \quad (3.12)$$

The coefficient $\epsilon(t_k) = k_p(\tilde{P}_e(t_k) - \phi_e(t_k))$ is updated at each new event detection. The proportional gain has been set to $k_p = 0.2$ in this application, as advised by R. Ronsse et al. [38]. Finally, the gait phase expression is provided in eq. 3.13.

$$\phi(t) = \text{mod}(\phi_n(t) + \phi_e(t), 2\pi) \quad (3.13)$$

In the frame of this work, three phase variables are generated. Two variables (one for each leg) are used in the following stage to measure gait asymmetry. Each of them is synchronised by the heel-strike event detected on the raw signal by the detection algorithm presented below in Section 2.4. The third phase variable only concerns the prosthetic leg and is used to generate the prosthesis angular trajectory. This variable is based on the same oscillators but reliably synchronised with the physical heel-strike event using the ground contact sensor.

2.4. Gait events detection

In order to measure gait asymmetry, we have decided to compare the timing of equivalent events between both legs. The idea is to process the learned angular velocity signal to detect heel-strike (HS), foot-flat (FF), heel-off (HO), toe-off (TO), and mid-swing (MS), and record the value of the phase variable when these events occur. In addition, heel-strike detection is required by the adaptive oscillators to set the phase reference.

A simple algorithm has been proposed by H. Silva et al. [35] to robustly detect gait events including the inner-stance events on level ground walking. As soon as an event is detected, it is used to update a finite-state machine (Figure 3.5) which determines the next event to look for.

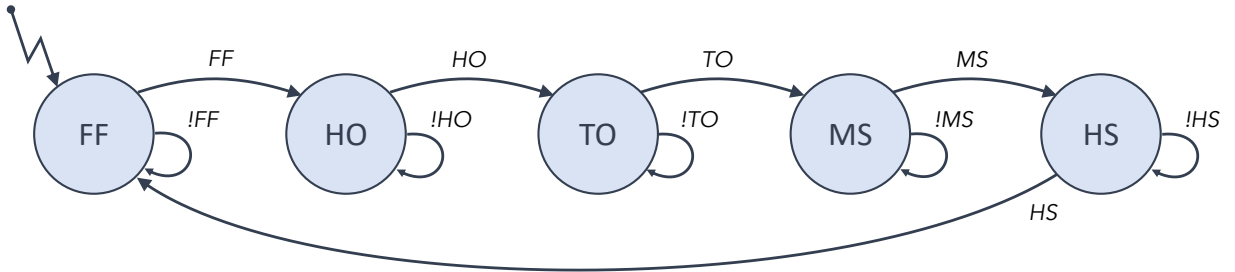


Figure 3.5: Events detection algorithm finite-state machine, adapted from [35]. FF - foot-flat, HO - heel-off, TO - toe-off, MS - mid-swing, HS - heel-strike.

Several conditions need to be satisfied for a specific event to be detected. The first one is a security condition allowing to avoid possible incorrect detection. It verifies whether the elapsed time between the possible new time t at which an event is detected and the previous time t_k at which this event occurred is larger than a fraction ρ of the current stride period T estimated by the adaptive oscillators [37]. It is assumed that the user is walking in steady-state with slow walking cadence variation. The coefficient ρ allows to take this slow variation into account. The value of $\rho \in [0, 1]$ has been set to 0.75 for this application through trial and error.

$$t - t_k \geq \rho \cdot T(t) \quad (3.14)$$

In addition, each event is identified through one or more specific condition(s) based on the current and two previous values of the estimated angular velocity signal, $\hat{\theta}_i$ and $\hat{\theta}_{i-1}$ and $\hat{\theta}_{i-2}$ respectively. These conditions are provided in eq. 3.15 to 3.19. Remember that the condition associated with an event is evaluated only if the FSM is in the right state, looking for this event. Figure 3.6 shows the location of the five key events on the angular velocity profile throughout the gait cycle. One can observe that events are either identified by a local minimum or maximum, or a zero crossing.

1. **Heel-strike** characterised by a zero crossing.

$$\hat{\theta}_{i-2} > 0 \text{ and } \hat{\theta}_{i-1} > 0 \text{ and } \hat{\theta}_i < 0 \quad (3.15)$$

2. **Foot-flat** characterised by a local minimum.

$$\hat{\theta}_{i-2} > \hat{\theta}_{i-1} \text{ and } \hat{\theta}_{i-1} < \hat{\theta}_i \quad (3.16)$$

3. **Heel-off** characterised by a local minimum with an additional condition on the amplitude.

$$\begin{aligned} \hat{\theta} &< -100 \\ \hat{\theta}_{i-2} &> \hat{\theta}_{i-1} \text{ and } \hat{\theta}_{i-1} < \hat{\theta}_i \end{aligned} \quad (3.17)$$

4. **Toe-off** characterised by a zero crossing.

$$\hat{\theta}_{i-2} < 0 \text{ and } \hat{\theta}_{i-1} < 0 \text{ and } \hat{\theta}_i > 0 \quad (3.18)$$

5. **Mid-swing** characterised by a local maximum with an additional condition on the amplitude.

$$\begin{aligned} \hat{\theta} &> 150 \\ \hat{\theta}_{i-2} &< \hat{\theta}_{i-1} \text{ and } \hat{\theta}_{i-1} > \hat{\theta}_i \end{aligned} \quad (3.19)$$

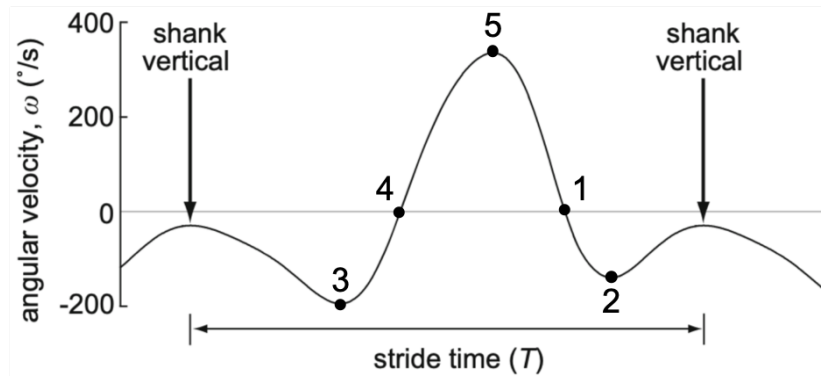


Figure 3.6: Trajectory of the IMU on the shank adapted from [39]. 1 - Heel-strike, 2 - Foot-flat, 3 - Heel-off, 4 - Toe-off, 5 - Mid-swing.

Drawbacks

As it is, the algorithm requires a minimal walking speed to detect heel-off and mid-swing due to the fixed thresholds on the amplitude of the signal, not reached at slow walking. A solution might be to adapt these conditions with the estimated walking speed.

Also, it is important to note that the accuracy of this algorithm is not ensured as there is no direct link between the foot contact state with the ground and the tibia angular velocity. The algorithm is based on average kinematics of healthy subjects walking at normal pace. The assumption is made that this inaccuracy does not produce a detrimental effect on the asymmetry measurement as it should likely be similar for both legs. However, it is problematic if the heel-strike event is used as reference for the gait phase variable affecting reference signal generation (Figure 3.1). This is the reason why an extra ground contact sensor (switch or pressure insole) is required on the prosthetic foot.

2.5. Asymmetry measurement

Gait asymmetry is a measure of the difference between the properties P of the right side and the left side during a locomotive task [10]. In the case of a lower-limb amputee, the asymmetry is rather assessed by comparing the difference between the intact limb and the prosthetic limb. In the frame of this work, we have decided to compare the gait phase value ϕ at which the detected events occur on each leg. These values are expressed as a proportion of gait cycle [%]. It is desirable to obtain an asymmetry index (AI) for each event which can then be used for asymmetry correction. This index should hold two pieces of information : the amplitude and the direction of asymmetry. Common scales are provided in Table 3.1 and are illustrated in Figure 3.7 [17, 10].

	Expression of AI [%]	Symmetry value [%]	Linearity
(a)	$P_{\text{prosthetic limb}} - P_{\text{sound limb}}$	0	Linear
(b)	$\frac{P_{\text{prosthetic limb}} - P_{\text{sound limb}}}{0.5 \cdot (P_{\text{prosthetic limb}} + P_{\text{sound limb}})} \cdot 100$	0	Non-linear
(c)	$\frac{P_{\text{prosthetic limb}}}{P_{\text{sound limb}}} \cdot 100$	100	Linear
(d)	$\ln \frac{P_{\text{prosthetic limb}}}{P_{\text{sound limb}}} \cdot 100$	0	Non-linear

Table 3.1: Most commonly used asymmetry indexes, with the symmetry value obtained when a perfect symmetry is reached, and the linearity of the expression w.r.t. $P_{\text{prosthetic limb}}$.

Figure 3.7 allows to visualise the evolution of the AI in each case for a fixed value of $\phi_{\text{sound limb}} = 50$ [%] and $\phi_{\text{prosthetic limb}}$ ranging from 40 to 60 [%]. As a reminder, we are interested in an index that is proportional to the distance between $\phi_{\text{prosthetic limb}}$ and $\phi_{\text{sound limb}}$, and providing information about the direction of the asymmetry. In a quest of simplicity of the algorithm, we have decided to use the linear expression (a), i.e. a difference between both values, providing both the intensity and direction. Based on this simple asymmetry measurement, heuristic laws can be developed to modify the angular reference position and reduce asymmetry.

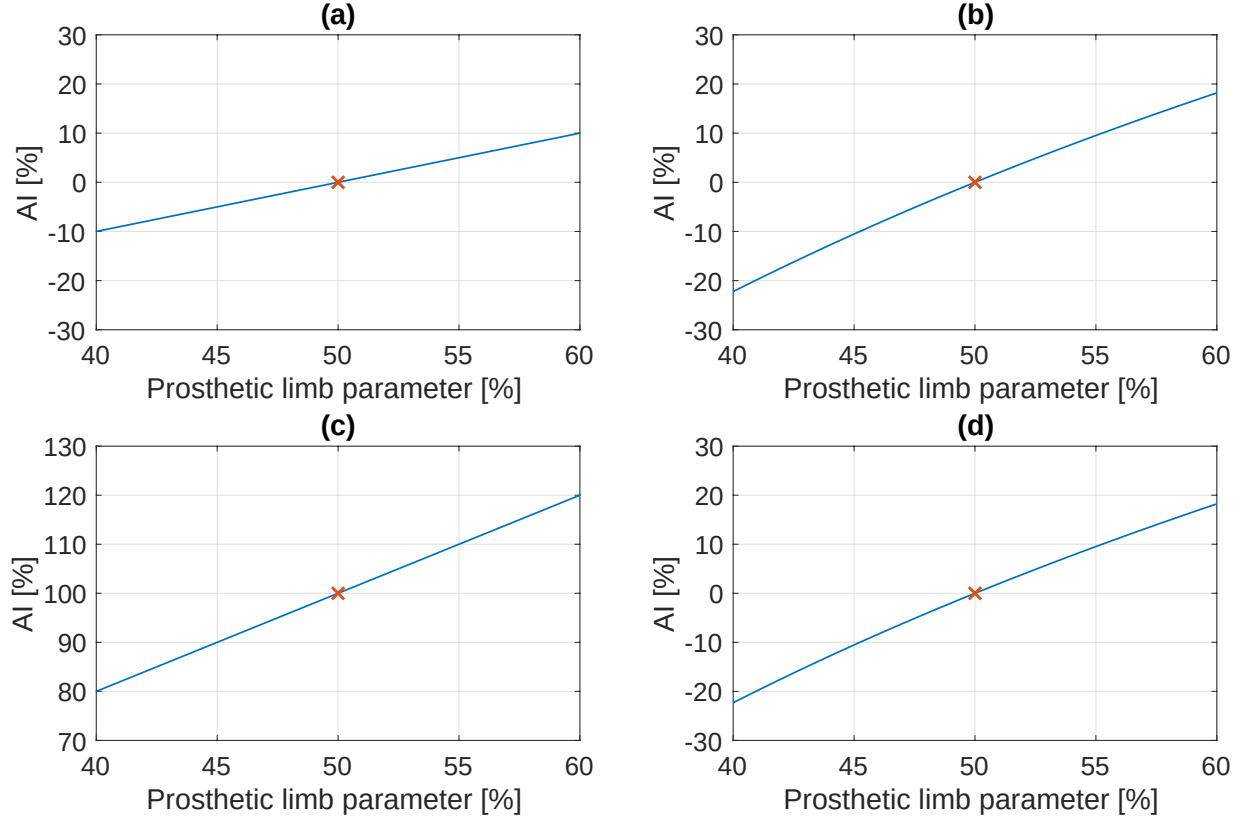


Figure 3.7: Evolution of the asymmetry index for a constant sound limb phase of 50 % and prosthetic limb phase varying from 40 to 60 % based on expressions (a), (b), (c), and (d) from Table 3.1.

2.6. Reference generation

In order to control the prosthesis angular trajectory and work on gait asymmetry reduction, a reference angular position pattern is tracked. This pattern needs to be parametric in a way that an adaptation can easily be performed based on asymmetry indexes. The chosen solution consists in using a linear combination of simple motor primitives to reproduce a biological signal such as the ankle angular position. This approach has already been tested by R. Guarate et al. where motor primitives were combined to adaptive oscillators to control an exoskeleton for different locomotive tasks [40].

Human body movement is governed by a redundant neuromusculoskeletal network with numerous neural mechanisms and muscles. In this network, motor primitives play an important role, namely muscles activation and coordination signals. These are functions that aim at representing the neural mechanisms responsible for coordinated body movements in a simplified way. The high-dimensional and redundant set of biological signals is simplified by means of a small number N of motor primitives (eq. 3.20). Motor primitives can be seen as a set of basic signals that can be linearly combined to generate a large set of muscle stimulations, corresponding to different locomotion tasks [40, 41].

$$f(x) = a_0 + \sum_{i=1}^N a_i f_i \quad (3.20)$$

A possible choice for the shape of motor primitives f_i is the Gaussian function characterised by a mean value μ and a standard deviation σ (Figure 3.8). This function has the advantage to

be symmetric, easily adjustable (two parameters), almost equal to zero away from $x = \mu$ (possibly few overlapping), and progressively converging to unity when getting close to μ (normalised). These characteristics allow to easily approximate signals made of peaks with a reduced number of primitives. Figure 3.9 shows a linear combination of three gaussian primitives to approximate such a signal.

$$f_i(x) = \exp\left(-\frac{1}{2}\left(\frac{x - \mu_i}{\sigma_i}\right)^2\right) \quad (3.21)$$

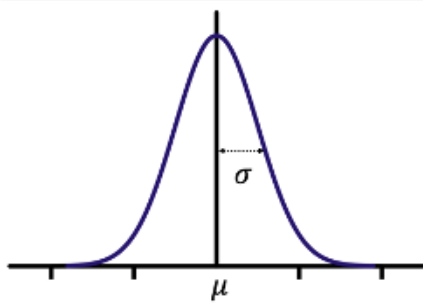


Figure 3.8: Gaussian function and associated parameters.

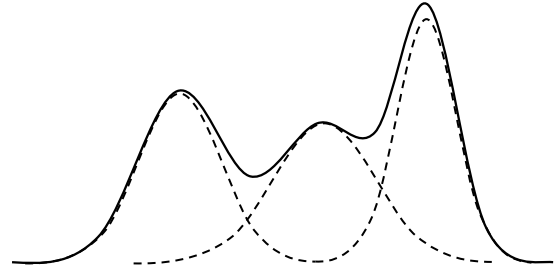


Figure 3.9: Linear combination of gaussian motor primitives.

In the purpose of this thesis, the approximated signal is the reference angular position pattern for the ankle prosthesis, as a function of the gait phase $\phi(t)$. In order to set the number of required primitives and initial values for coefficients, a sound reference pattern coming from the literature (D. Winter [30]) has been used. This pattern is depicted in Figure 3.10.

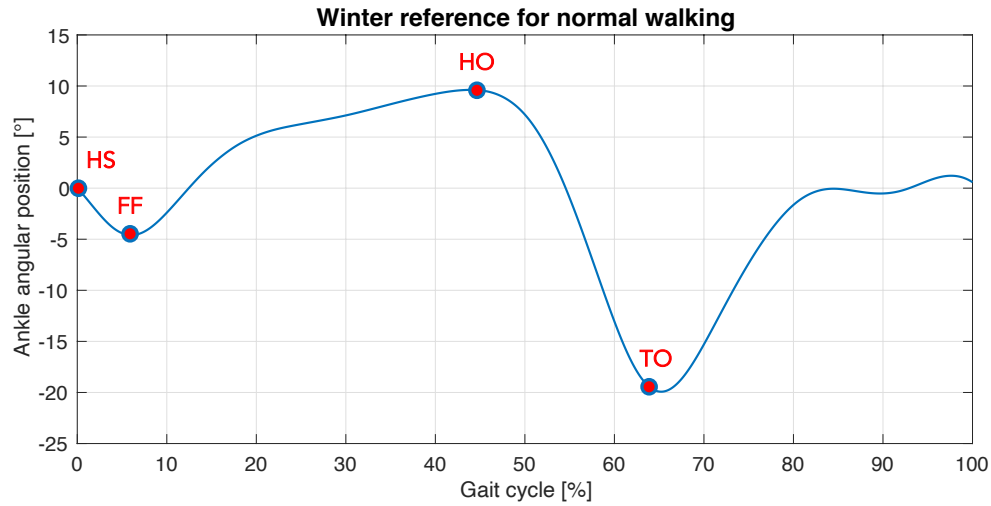


Figure 3.10: Sound ankle position trajectory (Winter), adapted from [30]. HS - Heel-strike, FF - foot-flat, HO - heel-off, TO - toe-off.

Let us now determine the minimal number of gaussian primitives required to approximate this pattern with good precision. At first glance, one can observe on the pattern of Figure 3.10 that at least two primitives are required to capture the general shape of the curve. In addition, a maximum of three primitives are considered for simplicity of the algorithm. The question is now to evaluate the precision improvement of the approximation using either two or three primitives.

The **Curve Fitting Toolbox** of **Matlab** has been used to find the optimal set of parameters values first with two then with three primitives. The algorithm is based on a non-linear least squares analysis. Figures 3.11 and 3.12 show the approximation of the Winter ankle angular position reference for a normal walking period of 1.2 [s] as a function of the gait cycle with these parameters. The fitting parameters and the root mean square error (RMSE) in each case are provided in Table 3.2. We can observe that the approximation with 3 primitives better captures the shape of the Winter reference curve than the approximation with 2 primitives. Proof of this is the RMSE being 5.98 times larger in the case of 2 primitives than in the case of 3 primitives. For this reason, three motor primitives should be used with initial parameters values provided in Table 3.2.

	a_0	a_1	a_2	a_3	μ_1	μ_2	μ_3	σ_1	σ_2	σ_3	RMSE
2 MP	5.46	-10.01	-24.83	/	7.16	67.14	/	5.57	7.52	/	2.81
3 MP	-0.04	-6.01	9.79	-24.62	7.79	43.68	65.38	3.89	18.29	7.00	0.47

Table 3.2: Parameters and RMSE of the motor primitives.

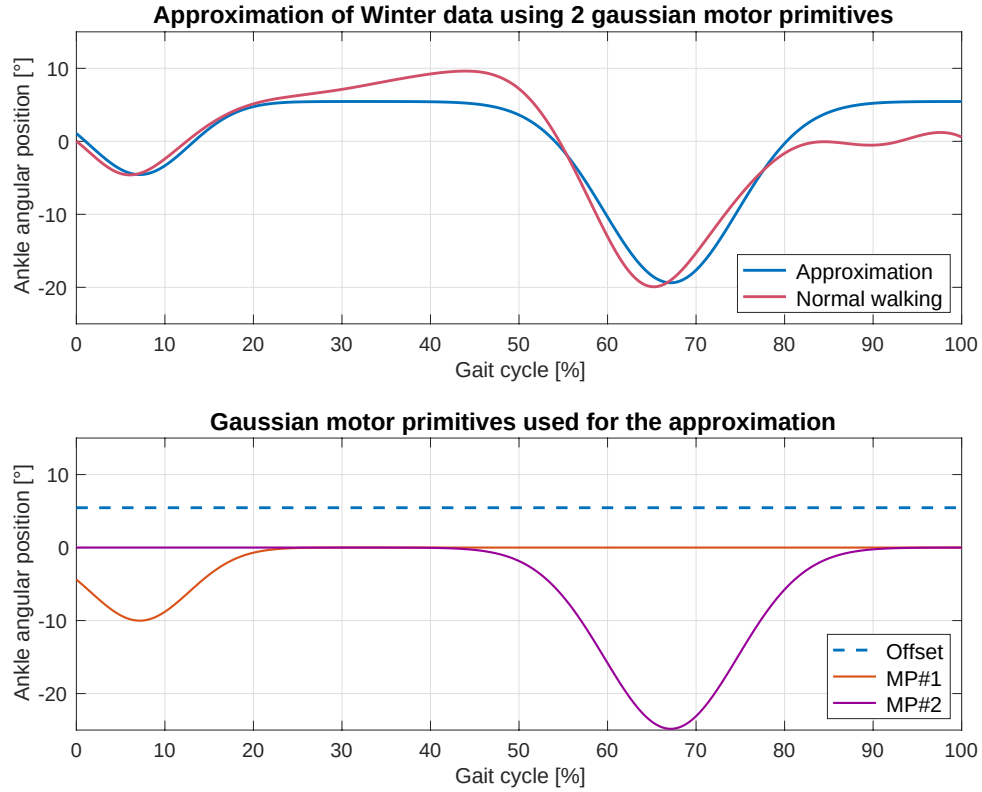


Figure 3.11: (a) Approximation of Winter ankle angular position reference using 2 gaussian motor primitives as a function of the gait cycle. (b) Gaussian motor primitives and offset patterns.

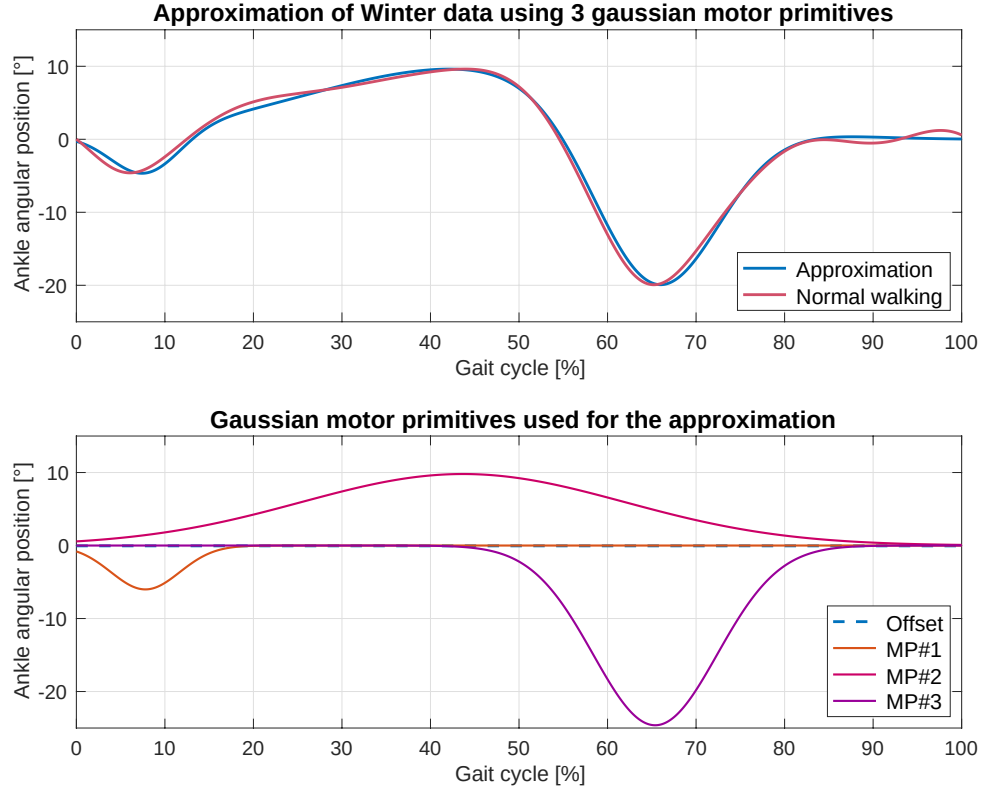


Figure 3.12: (a) Approximation of Winter ankle angular position reference using 3 gaussian motor primitives as a function of the gait cycle. (b) Gaussian motor primitives and offset patterns.

2.7. Heuristic laws

Heuristic laws are developed in an attempt to reduce gait asymmetry and are intended to modify the motor primitives mentioned above. As a reminder, the chosen asymmetry indexes are temporal measurements, i.e. a difference in gait phase between both legs for a specific event. For this reason, we have decided to adapt the temporality (horizontal position) of the three primitives as a correction mechanism. Strictly speaking, only the weights a_i should be adapted by convention, but we have also decided to act on the positions μ_i . Each motor primitive is associated to a specific event, and is modified when an asymmetry is measured for this event. Indeed, one can observe from Figure 3.10 and Figure 3.12 that the major contribution for the foot-flat, the heel-off and the toe-off events comes from the first, second and third motor primitives respectively. The idea is not to mimic exactly the behaviour of the sound ankle but rather to induce a reaction of upper joints that results in a reduction of asymmetry between key gait events of both legs. However, there is no scientific evidence that indicates that this strategy could efficiently reduce gait asymmetry.

The proposed correction mechanism consists of a horizontal shift of a motor primitive which direction and magnitude are determined by the sign and the magnitude of the AI respectively. If a specific gait event occurs $X\%$ earlier (later) for the amputated leg than for the sound leg on the gait cycle, then the parameter μ of the corresponding motor primitive is shifted by $X/N\%$ to the right (left) such that the event should occur a bit later (earlier) at the next step. A qualitative example is provided in Figure 3.13 in the case of an asymmetry in the toe-off event. The higher the value of N , the higher the inertia and the predictability of the controller. The shift of primitives is performed at heel-strike based on asymmetries measured during the previous stride.

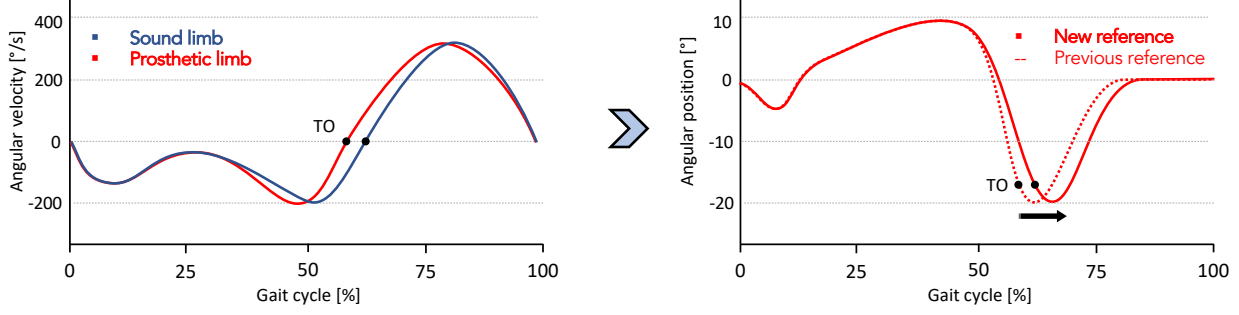


Figure 3.13: (a) Ankle angular velocity profiles of the sound limb and the prosthetic limb with an asymmetry in the toe-off (TO) event, (b) New (solid line) and previous (dotted line) references for the ankle angular position profile of the prosthetic leg.

However, a horizontal shift of a motor primitive induces a significant amplitude variation of the three peaks due to primitives overlapping. For illustration, Figure 3.14 (a) shows the evolution of the ankle angular trajectory for three values of μ_3 . As one can observe, the amplitude of the third peak increases as the negative third primitive moves away from the positive second primitive (Figure 3.12 (b)). To counteract this undesired effect, the weights a_i are updated at every step to minimise the difference between the new and nominal peaks amplitude. These new values x are the solution of a small optimisation problem $\min_x f(x)$ with initial conditions being the nominal weights. One can observe in Figure 3.14 (b) that the peak amplitude remains constant when the weights are adapted using this method.

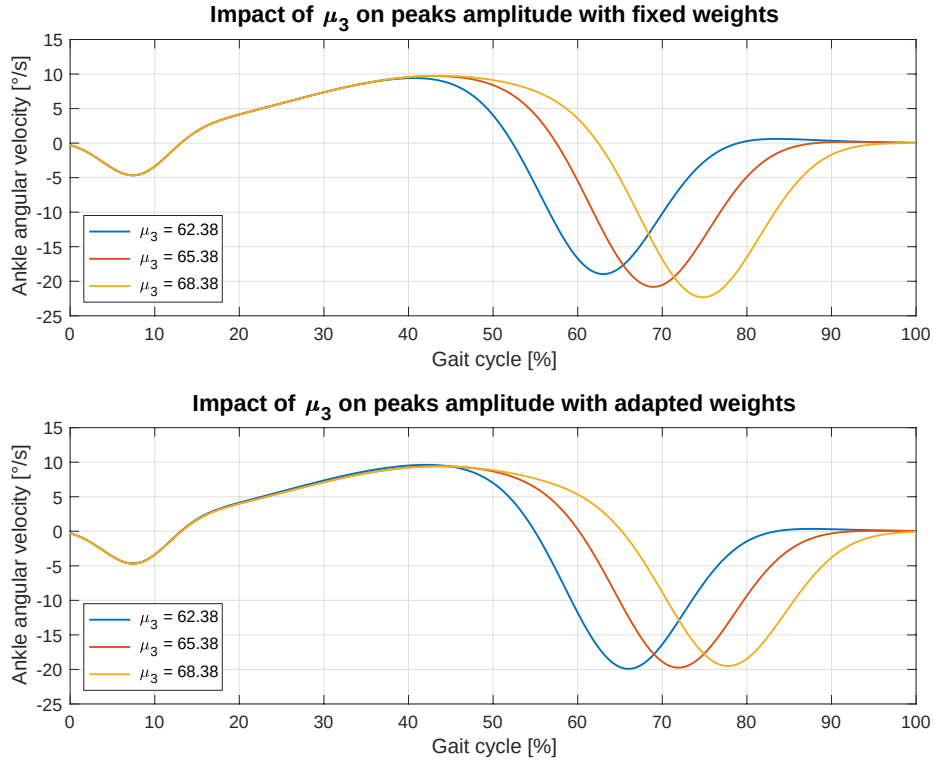


Figure 3.14: Impact of μ_3 variation on the peaks amplitude with fixed weights (a) and adapted weights (b). The equation and parameters used to generate the ankle angular velocity profiles are provided in Section 2.6. of Chapter 3.

For some gait events such as heel-strike or mid-swing, the modification of the ankle angular profile is not meaningful. First, the heel-strike event is the reference event and corresponds to the 0 [%] gait phase variable. Then, the prosthesis has no control on the time at which the user decides to end the swing phase and to enter the stance phase with a new heel-strike event. Similarly, modifying the motor primitives at mid-swing would not have an impact on gait asymmetry as the foot is off the ground. However, this latter asymmetry index is still interesting to monitor. For these reasons, only the foot-flat, heel-off, and toe-off events are considered for asymmetry correction. For security purpose, the position of motor primitives μ_i are bounded in the range $[0 \ 100\%]$ with $\mu_1 < \mu_2 < \mu_3$.

3. Feedback-error-learning control

In this section, a novel control approach is presented in order to perform a compliant position control of the ankle prosthesis.

3.1. Motivation

The human body is a very complex non-linear dynamical system made of numerous joints and capable of interacting with external disturbances and uncertainties from the environment. The complexity of the human body together with the complexity of human-prosthesis interactions, therefore, make the development of accurate models very difficult, especially in the case of human locomotion modelling which can be characterised by a high dimensional space, a system behaviour influenced by external disturbances, and non-linearities [42, 43, 44].

Feedback control is usually used to solve these issues by means of high feedback gains in order to reduce tracking errors between the actual state and the desired state and therefore to ensure the robustness of the system. However, this is not suitable for biological systems since high gains lead to system instability and non-compliant movements that can compromise user safety. As explained by A. Calanca et al. [31], compliant control strategies are fundamental to ensure a safe interaction between the user and the prosthetic device. To do so, a compliant control strategy implicitly regulates the energy transfer to the environment. Reducing tracking errors should not be performed at the expense of controller compliance.

3.2. Novel control approach

A novel bio-inspired control approach based on the functionality of the cerebellar circuit aims at enabling accurate motor learning, increasing adaptability to external disturbances, and ensuring precise and compliant movements. This control approach is based on two components that complement each other [42, 43, 44] : a **feedforward action** based on a cerebellum-like controller for prediction and a **feedback contribution** for error compensation, usually in the form of a P(I)D controller. The feedforward action can provide great benefits and is an attempt to reduce the feedback contribution and therefore to enhance compliance while improving task-space tracking. Usually, a dynamic model of the system is inverted to predict the input required to reach a desired output. The more accurate the model, the smaller the feedback gain. Since no accurate model of the walking gait exists, we need a tool that is able to approximate this model based on real-time data collected from the sensors.

The feedforward action works as a simplified model of the cerebellar circuit. The cerebellum plays an important role in movement control, movement coordination and balance maintenance. It learns the internal models of the body parts based on afferent inputs from the sensory system, the spinal cord and other parts of the brain. In the cerebellum, two internal models are used to perform motor control, the **inverse model** which estimates the command to be provided to reach a desired state,

i.e. inverse dynamics, and the **forward model** which predicts the future state based on a specific command, i.e. forward dynamics. In biorobotic control, the cerebellum function can be represented by the LWPR algorithm which is a non-parametric learning method that uses a variable number of locally linear models. The LWPR is particularly suited when the function to be learned is non-linear and when the application requires incremental and real-time adaptation. In the literature, the LWPR algorithm has proven its effectiveness to model the inverse [43, 44] or forward [42, 45] dynamics of complex mechanical systems such as simulated models of a robot arm [44, 46] or a quadruped robot [43]. As far as we know, the LWPR has never been used in the field of human locomotion. In this work, an inverse dynamics model is approximated based on θ , $\dot{\theta}$, θ_{ref} , as well as the gait phase ϕ since the user dynamic response is supposed to change along the stride. Practically, the LWPR learns the 4-D function linking the current and the desired state to the torque. As explained by S. Tolu et al. [44], a complete inverse dynamics model would require to use θ , $\dot{\theta}$, $\ddot{\theta}$, θ_{ref} , $\dot{\theta}_{\text{ref}}$, and $\ddot{\theta}_{\text{ref}}$. In order to facilitate the tuning process, we have made the assumption that a part of the information of $\dot{\theta}_{\text{ref}}$, and $\ddot{\theta}_{\text{ref}}$ is held in ϕ .

The role of the feedback action is then to compensate the remaining error. However its contribution is much lower than with a simple feedback control as the approximated model provides almost the right input to apply through the feedforward action, leading to a small error. This way, lower gains can be used, leading to better stability and compliance. This corrected torque reference is iteratively learned by the LWPR to enhance its internal model. This idea was brought to us by S. Heins who is currently working on a similar project. Therefore, the strategy has been developed in close collaboration with her.

3.3. LWPR

Non-linear function approximation can be performed in two different ways. The approximator algorithm either tries to fit the non-linear function globally or locally. In the first case, gaussian process regression (GPR) or support vector machine regression (SVMR) algorithms are often used. However they are not computationally efficient when new data is added incrementally. Algorithms based on local function approximation techniques, such as locally weighted learning (LWL) or projection regression (PR), are better suited for incremental and real-time learning. However, these techniques present some setbacks. Regarding LWL, it cannot cope with high dimensional input data as it requires a very high number of local models to provide accurate approximations. Despite the fact that PR can cope with high dimensional input data, it struggles to apply projections efficiently on the input space, i.e. find a minimal number of projections to provide an accurate approximation. LWPR has proven to be a fast, efficient and incremental non-linear function approximator by combining the LWL and the PR working principles [47].

In comparison with other learning algorithms such as the GPR or SVMR, the LWPR algorithm presents relatively low computational costs despite highly redundant or irrelevant data inputs. This allows the LWPR algorithm to perform accurate, fast and real-time learning [44, 46], perfectly fitting the requirements of locomotion.

Working principle

As said above, the LWPR is a non-linear function approximator that, given a query point x , provides an approximated function $f(x)$ that is the weighted mean of K local linear models $\psi_k(x)$ (eq. 3.22).

$$f(x) = \frac{1}{W(x)} \sum_{k=1}^K \omega_k(x) \psi_k(x) \quad (3.22)$$

For each input data x , a receptive field (RF), or $\omega_k(x)$ is computed as a gaussian weight that is specific to each linear model $\psi_k(x)$, where D_k is the distance metric and c_k is the centre of the receptive field (eq. 3.23). The validity region, i.e the region of space for which each linear model is active, is determined by the distance metric D_k . Note that the amplitude of the gaussian weights is irrelevant due to the normalisation by $W(x)$.

$$\omega_k(x) = \exp \left(-\frac{1}{2}(x - c_k)^T D_k (x - c_k) \right) \quad (3.23)$$

$$W(x) = \sum_{k=1}^K \omega_k(x) \quad (3.24)$$

K local linear models of N -dimensions are produced based on an initial offset value β_0 , a direction of regression s_i and its corresponding slope β_i (eq. 3.25). Each model $\psi_k(x)$ is adapted to the local dimension of the training data, the regression directions s_i being automatically selected so that the number of dimensions is minimised. This way, the number of directions of regression N is not necessarily equal to the number of inputs. A simple example is provided in Figure 3.15 with $K = 2$ and $N = 1$.

$$\psi_k(x) = \beta_0 + \sum_{i=1}^N \beta_i s_i \quad (3.25)$$

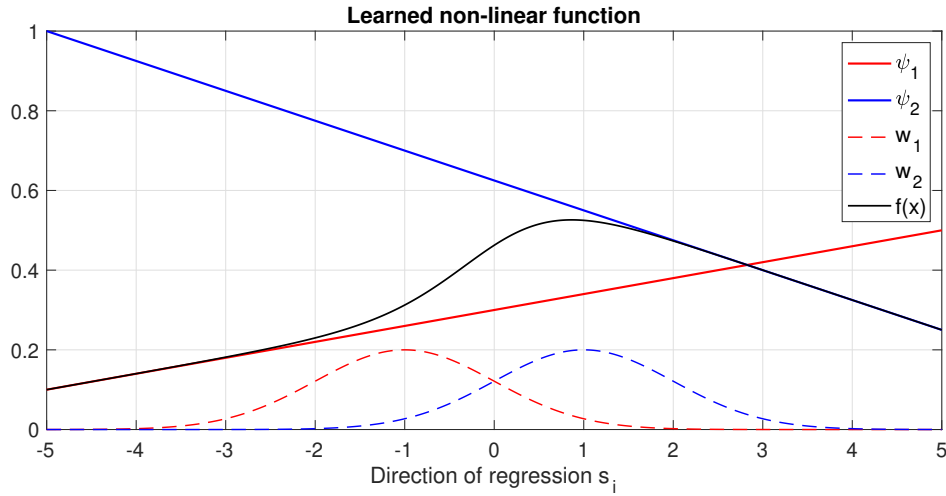


Figure 3.15: Simple example of non-linear function approximation with one direction of regression s_i . The approximation is a combination of two weighted linear models ψ_1 and ψ_2 .

Link with the cerebellum structure

As explained above, the LWPR structure is organised such that it mimics the functionality and morphology of the cerebellum. A comparison between both structures has been established in Table 3.3. Both the cerebellum microcircuit and the LWPR architecture are depicted in Figures 3.16 and 3.17 [43, 44].

Stage	Cerebellum	LWPR
Input acquisition	The sensory input $y(t)$ is transmitted to the granular cells (GC) through the mossy fibers (MF) .	The input $x_{input} = (x_1, \dots, x_n)$ is transmitted to the RF .
Prediction	Based on the input, the GC use the inverse model of the body part to compute a prediction $p_i(t)$.	Based on the input, K linear model predictions y_k are computed.
Weighting	The parallel fibers (PF) collect the information $p_i(t)$ from the GC and transfer a portion of it to the Purkinje cells (PC) at the PF-PC synapse where the information is filtered.	The receptive fields use gaussian weights to determine the region of validity of each model. Each prediction is weighted.
Model update	The PC also receive information from the climbing fibers (CF) which help them to modulate their activity by carrying a teaching signal $e(t)$. The CF allow the cerebellum to update its internal models.	A teaching signal y_{train} is used to update local models and the size of the receptive fields.
Output generation	The deep nuclear cell (DNC) generates the final cerebellar output $z(t)$, result of the weighted $w_i \cdot p_i(t)$ combination.	The output of the LWPR y is the weighted sum of all models predictions.

Table 3.3: Comparison between the cerebellum the LWPR.

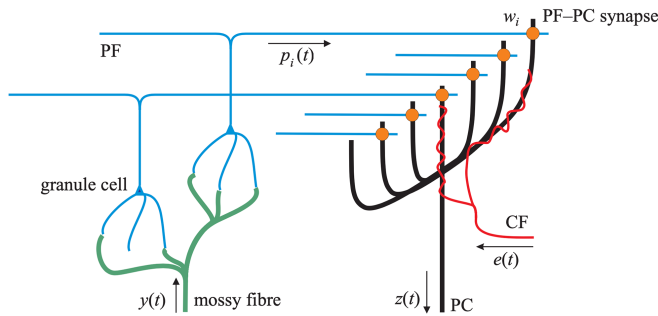


Figure 3.16: Cerebellar microcircuit [48].

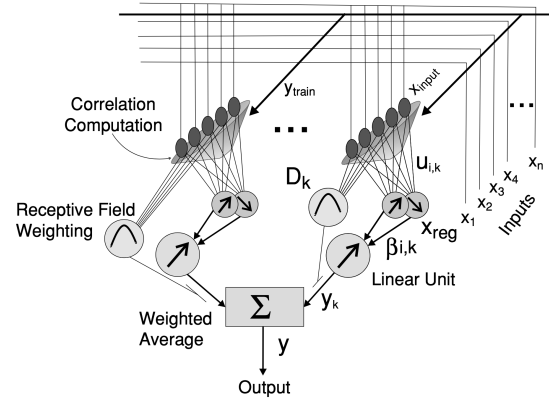


Figure 3.17: LWPR architecture [47].

1. Objectives

In order to evaluate the performance and interest of this novel control strategy, a preliminary validation stage is conducted using a simulation environment. The ultimate goal is to demonstrate through multibody dynamics simulations four important features of the control strategy :

1. the real-time gait analysis allows to estimate the phase variable of each leg as well as the temporal position of major gait events,
2. the algorithm is able to provide a torque reference for the ankle prosthesis leading to an accurate position tracking and adapting to the user dynamics,
3. the actuation is more compliant than using a pure feedback regulator,
4. a typical gait asymmetry can be reduced.

2. Simulation environment

2.1. Description

For the purpose of the simulations, a simple bipedal model of a walking human developed in **Robotran** is used. **Robotran** is a symbolic software developed at UCLouvain to model and analyse multibody system (MBS). The model is based on the work of S. Heins and is used in the frame of her own researches. As depicted in Figures 4.1, it is composed of two simplified legs where body segments (femur, tibia, and foot) are modelled as rigid bodies, and articulations (hip, knee, and ankle) are modelled as ideal revolute joints. Both legs are connected to the pelvis, which is linked to the trunk by a centred revolute joint aiming at modelling the pelvic obliquity movement in the frontal plane. Note that hip articulations compensate for this movement to keep legs in the sagittal plane. A prismatic joint provides to the trunk a single degree of freedom in the vertical direction. As a consequence, the body model is constrained to stay upright and cannot fall down. The trunk trajectory is computed by **Robotran** as the solution of the multibody equations. The simulation time-step is set to 10^{-4} [s]. All body parts dimensions are based on the average segment lengths proposed in [30] (p.83) for a user height of 1.80 [m]. In the same way, segments mass and center of mass are set with respect to the anthropometric data provided in the same document (p.86).

Each foot is equipped with two forces sensors, one at the heel and one at the toe. In fact, in **Robotran** these sensors are position sensors and the vertical force is computed using the "mesh-based" ground contact model (eq. 4.1) provided in [49].

$$F_z = K_{\text{ground}} \cdot (Z_{\text{ground}} - Z_{\text{sensor}}) \cdot \max(0, 1 - B_{\text{ground}} \cdot \dot{Z}_{\text{sensor}}) \quad (4.1)$$

In this model, $K_{\text{ground}} = 77000$ [N/m] is the ground stiffness, $Z_{\text{ground}} = -1.03$ [m] is the ground surface vertical position, $B_{\text{ground}}^{-1} = 0.5$ [m/s] is the maximal velocity used for normalisation, Z_{sensor}

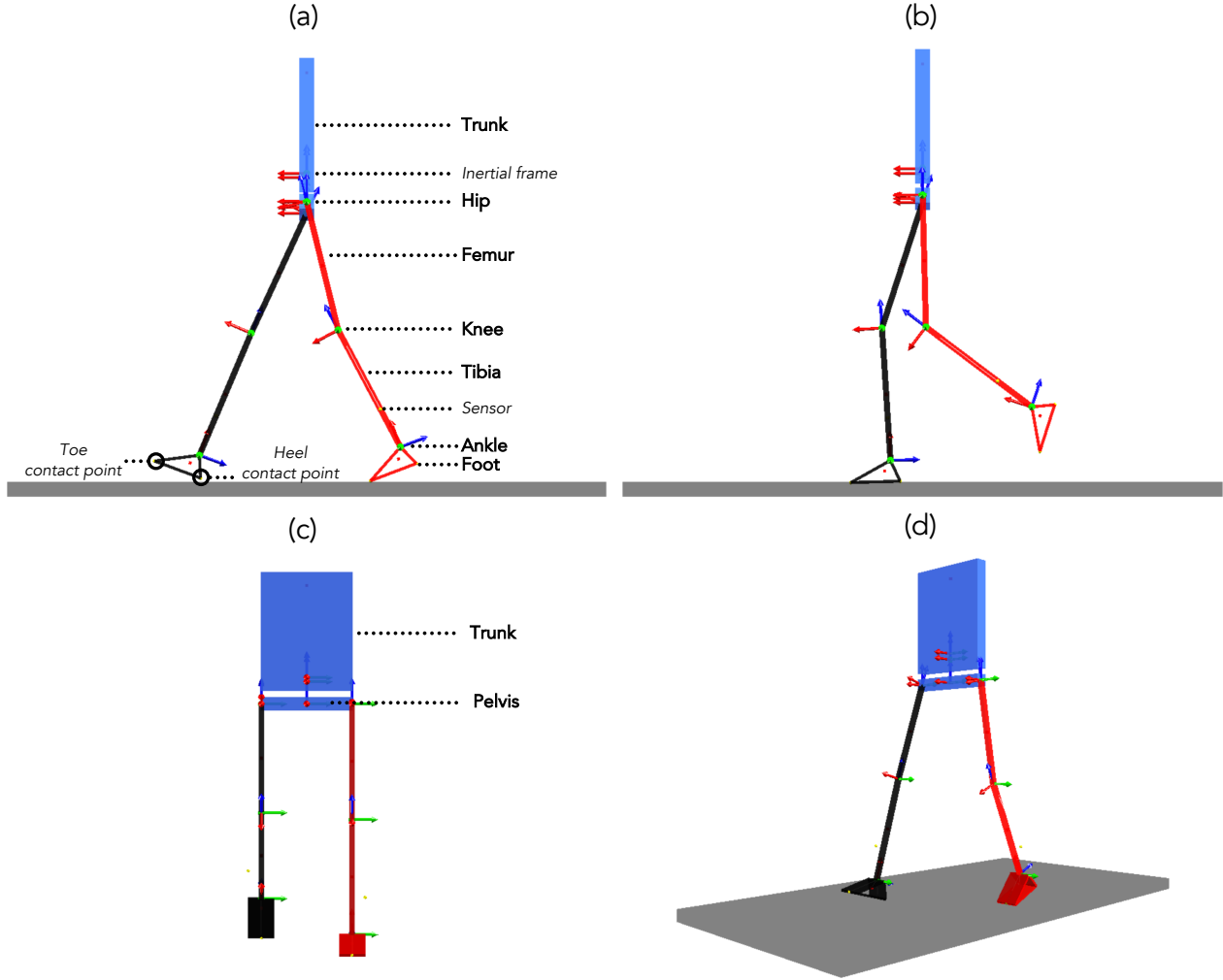


Figure 4.1: **Robotran** simple bipedal model description. (a) Side view with legend. (b) Side view during walking simulation. (c) Front view with legend. (d) 3D view.

$[m]$ is the sensor vertical position, and $\dot{Z}_{\text{sensor}}$ $[m/s]$ is the sensor vertical velocity. Finally, the total ground reaction force is obtained as the sum of these two contributions (heel and toe). In addition, the heel sensor is used as the binary ground contact detector (Figure 3.1).

An extra kinematic sensor is placed on the side of each tibia, at mid-height, as proposed in Section 2.1. of Chapter 3. The goal is to emulate the behaviour of IMUs and acquire the absolute angular velocity of the shins in the sagittal plane.

2.2. Implementation

The control and analysis of this multibody system has been fully achieved in the **Matlab** environment. We successfully implemented the entire control scheme presented in Figure 3.1 in a parametric manner. The code is user-friendly, thoroughly commented, and could be used for further works involving ELSA.

2.3. Driven trajectories

In an attempt to reproduce the kinematics of a walking person, the hip joints, knees, and sound ankle joints are driven to follow predetermined trajectories based on Winter references [30]. However, the imperfect ground contact model introduces undesired phenomena and prevents us from using raw Winter trajectories. First, undesired foot contact with the ground during swing phase can appear, as well as rebounds following heel-strike. To reduce these phenomena, the hip trajectory amplitude has been scaled by a factor of 1.3. Another drawback of the imperfect ground model is the bad synchronisation of the heel-strike which tends to appear at $\sim 8\%$ of Winter trajectories rather than 0% . To bring the heel-strike exactly at 0% , the pelvic oscillation motion is arbitrarily set to a sine wave synchronised with the heel-strike. This way, the leg reaching terminal swing is lowered with respect to the other leg in terminal stance. Both heel-strike and toe-off events are enforced slightly earlier. All driven trajectories are provided in Figure 4.2 for a normal walking period of 1.2 [s] and compared with the average sound profiles. Although the arbitrary trajectory imposed at the pelvis is very different from the biological one, it should have no negative impact on the dynamics and kinematics of the overall model. Finally, ankle designated as prosthetic can either be driven, or torque controlled.

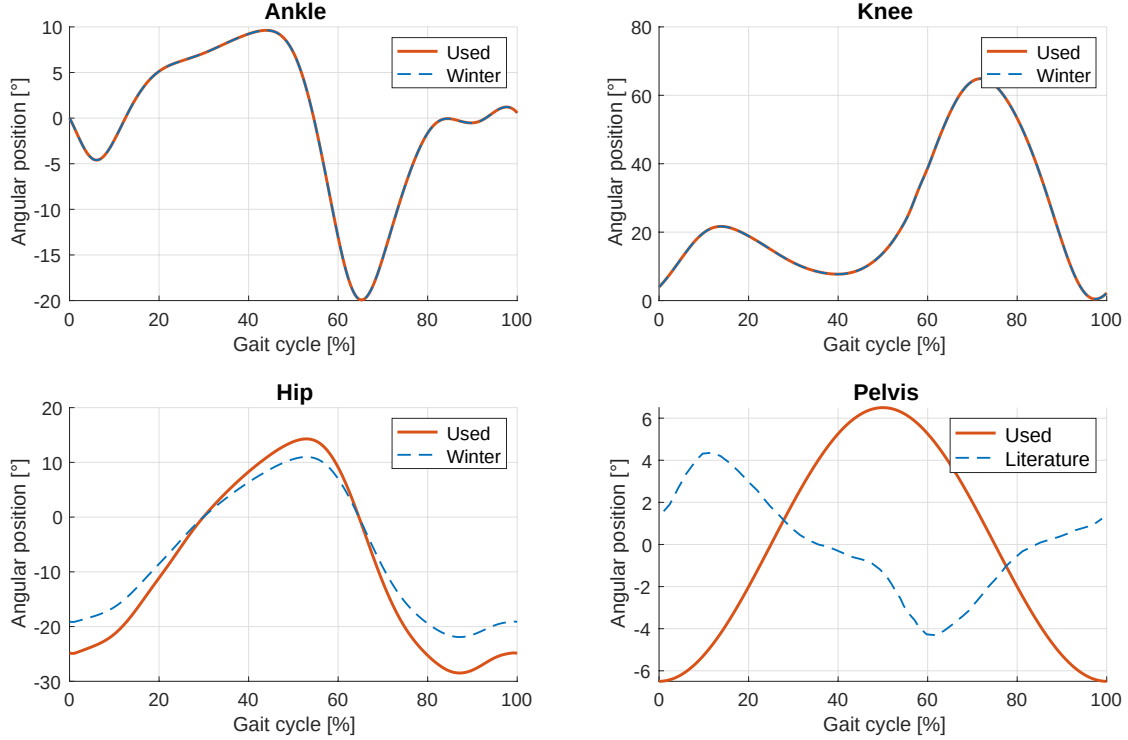


Figure 4.2: Driven trajectories used for the sound ankle, knees, hips, and pelvic obliquity compared to sound patterns found in literature [30] [50].

3. Real-time gait analysis validation

This section focuses on the validation of the real-time gait analysis. This stage comes in two steps. First, demonstrate the learning capacity of the adaptive oscillators to estimate a periodic raw signal and to derive an accurate gait phase variable for both legs. Then, show that the conditions provided in [35] allow to robustly isolate five events per stride on the learned angular velocity signal.

3.1. Method

The experiment consists in 10 simulated steps with a gait period of 1.2 [s]. The kinematics of all revolute joints including the prosthetic ankle are driven to the trajectories provided in Figure 4.2. The raw angular velocity signal measured by the virtual IMUs located on the shins is learned by the adaptive oscillators. Beforehand, 20 steps were simulated to make the adaptive oscillators converge and reach a steady state. A particular attention must be paid to the **synchronisation** between the raw and learned signals, as well as the **smoothness** of the signal to allow an accurate events detection. Also, the synchronisation of the phase variable with respect to the heel-strike event is evaluated based on the average phase error at heel-strike. Secondly, the algorithm developed by H. Silva et al. [35] is applied on the learned angular velocity signal to detect the other gait events. Assuming that this algorithm is accurate, we demonstrate its robustness by showing that the gait events are systematically detected. Finally, a last and similar experiment is conducted at a gait period of 1.0 [s] to see how the adaptive oscillators trained for a 1.2 [s] walking period and the events detection algorithm react to a pace variation.

3.2. Results

First, Figure 4.3 shows the raw and learned signals of the shank angular velocity as well as the estimated gait phase for the right and left legs. One can observe that the learned signal captures well the overall shape of the raw signal. In addition, both signals are well synchronised. Regarding the gait phase variable, it appears to increase linearly from 0 to 100 [%] between two heel-strikes,

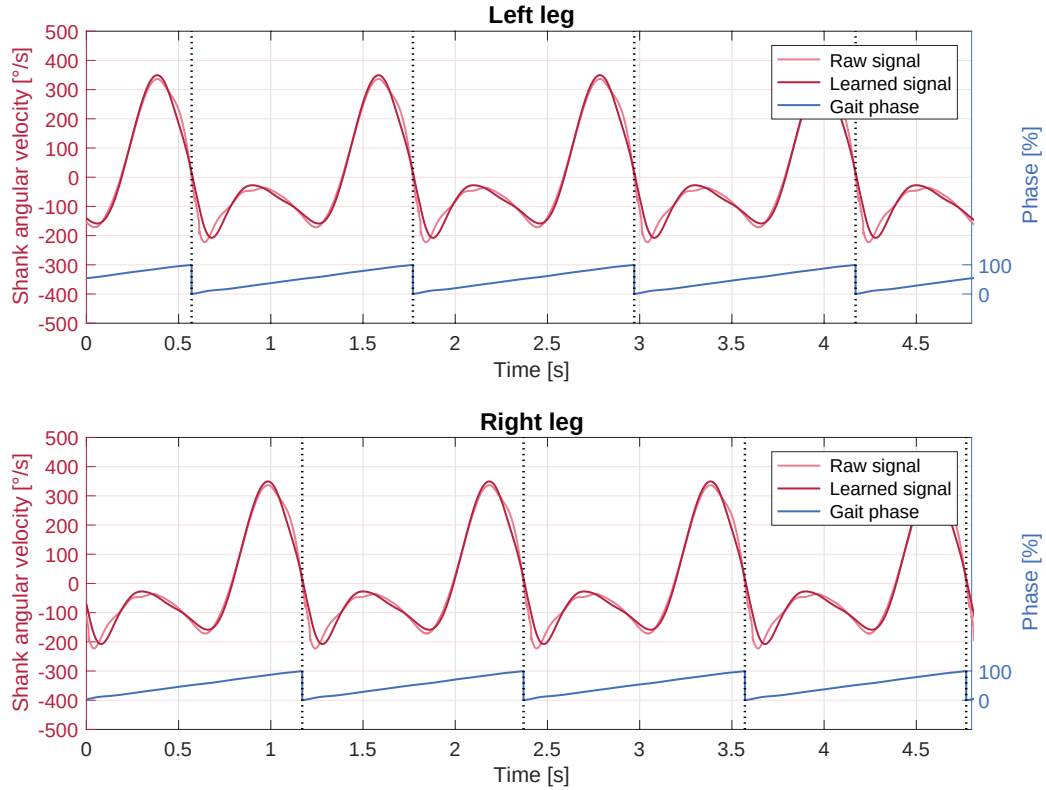


Figure 4.3: Shank angular velocity [$^{\circ}/s$] (raw and learned signals) and phase variable [%] as a function of time [s] for the left (a), and right leg (b), at a walking period of 1.2 [s]. The vertical dotted lines show the time at which the heel-strike events are detected by the algorithm on the raw signal.

denoted by vertical dotted lines. The root mean square error (RMSE) and standard deviation (SD) of the phase error at heel-strike events are 0.0098 and 0.0024 respectively for 10 consecutive steps. The low values obtained for the RMSE and SD demonstrate the reliability of the gait phase variable estimation. Note that very similar results are obtained using the ground physical touchdown as event of reference, except that the phase signal has a slight horizontal offset.

Regarding the gait events detection, results are provided in Figure 4.4. At first glance, the shape of the velocity signal is very similar to the one predicted in Figure 3.6. One can observe that, for each step, the algorithm is able to systematically isolate the five gait events on the learned angular velocity signal. As this signal is very smooth, there is no possible ambiguity on the position of each event. From these results, an asymmetry measurement can easily be performed. Remember that the heel-strike event is detected on the raw signal and appears here on the 0 horizontal axis only because the learned signal is a good approximation.

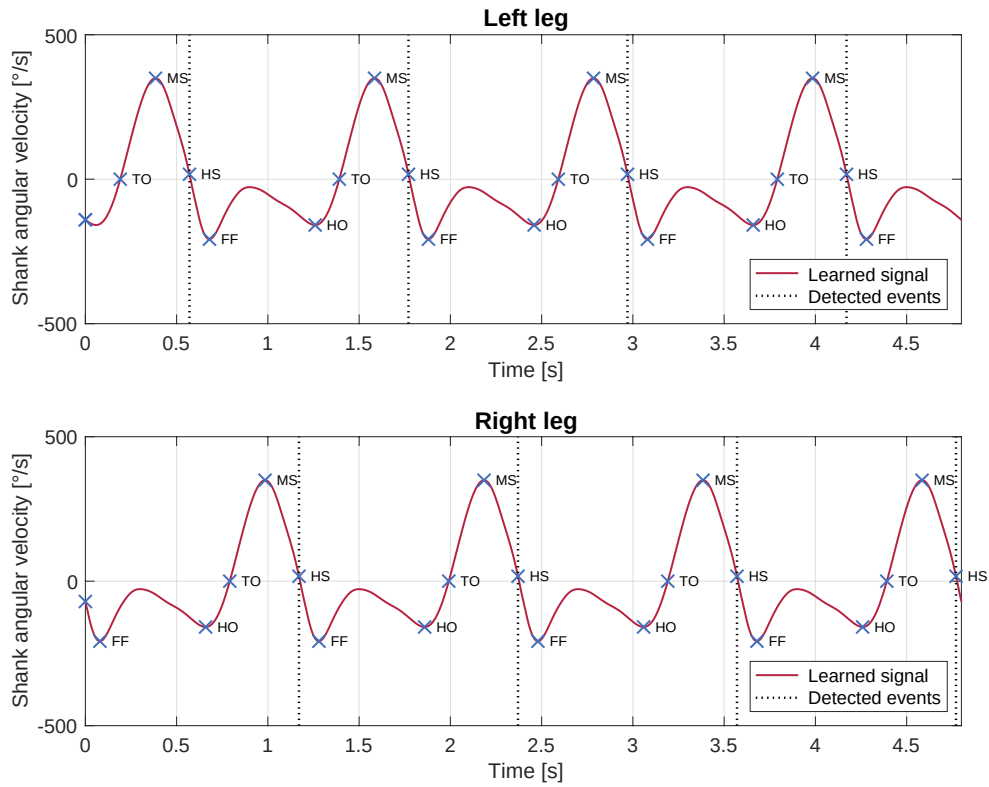


Figure 4.4: Learned signal of the shank angular velocity [°/s] over time for both the left leg (a), and right leg (b), at a walking period of 1.2 [s]. The crosses correspond to an event detection : heel-strike (HS), foot-flat (FF), heel-off (HO), toe-off (TO) and mid-swing (MS). The vertical dotted lines show the time at which the heel-strike events are detected by the algorithm on the raw signal.

Finally, Figure 4.5 shows the behaviour of the adaptive oscillators, gait phase, and events detection when the walking period is suddenly reduced from 1.2 [s] to 1.0 [s]. Two steps are required to recover a satisfactory signal approximation, while four steps are required to re-synchronise the phase signal. During this short transient state, the consistency of events detection is reduced as the learned signal is slightly de-synchronized with the raw signal. One can observe in Figure 4.5 (b) that the heel-strike event is initially quite far from the 0 horizontal axis and then converges towards it as the learned signal refines. However, we assume for now this de-synchronization to be similar between the two legs and thus not problematic.

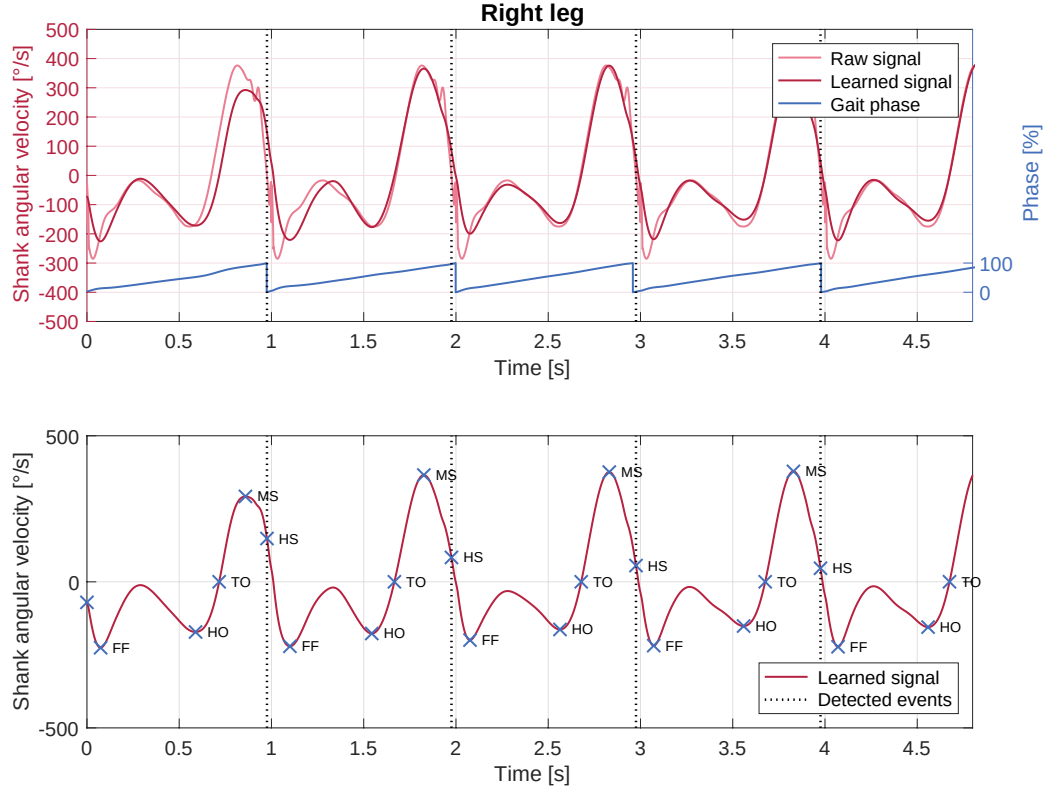


Figure 4.5: (a) Shank angular velocity [°/s] (raw and learned signals) and phase variable [%] as a function of time [s] for the right leg at a walking period of 1.0 [s]. (b) Learned signal of the shank angular velocity [°/s] over time for both the right leg. The crosses correspond to an event detection : heel-strike (HS), foot-flat (FF), heel-off (HO), toe-off (TO) and mid-swing (MS). The vertical dotted lines show the time at which the heel-strike events are detected by the algorithm on the raw signal.

3.3. Discussion

This experiment shows that three adaptive oscillators can approximate a typical tibia angular velocity signal with a sufficient accuracy to perform events detection and asymmetry measurement. Also, we observe that the adaptive oscillators are able to adapt to a sudden change of walking speed within 4 steps. A faster convergence could probably be obtained through fine tuning of τ_α and τ_ω (Section 2.3. of Chapter 3), and we can expect to have less approximation error for continuous and progressive speed variations. However, this method is not ideal for real-life applications as the adaptive oscillators require an initial training stage of 10 to 20 steps to synchronise and reach a steady-state before the experiment. During this period, another control strategy independent of the learned signal must be used.

4. Position tracking validation

This section aims at showing that the control scheme is able to perform accurate position tracking by combining the feedback and feedforward actions. Also, the learning capabilities of the LWPR are assessed in a realistic situation.

4.1. Method

For this experiment, the kinematics of all revolute joints are driven to the trajectories provided in Figure 4.2 except for the prosthetic ankle which is torque controlled. The torque reference is generated by the combination of the feedforward and feedback actions as depicted in Figure 3.1 to track the trajectory provided in Figure 3.12 (a).

As a first step, a blank LWPR model is initialised and has to learn on the fly at an update frequency of 100 [Hz] based on feedback correction. The experiment consists in 15 simulated steps with a gait period of 1.2 [s]. The focus is set on the evolution of the feedback contribution as time goes by to show the learning capabilities of the LWPR. This value should normally be reduced as the LWPR refines its estimation of the required torque.

As a second step, the accuracy of the position tracking is analysed along the 15th gait cycle and compared with a pure feedback based control. The goal is to show the tracking performance that can be obtained with the proposed control scheme when the LWPR is properly trained. From the moment where the LWPR reaches a steady state, all signals (torque, trajectory, reference) are very stable. An averaging of the results over several steps has few interest as the simulation is almost invariant, reason why we focus on a representative gait cycle.

4.2. Parameters

The value of relevant parameters tuned to obtain the following results are provided in Table 4.1. Regarding the PD regulator, different coefficients K_p and K_d are used during the stance and swing phases as the impedance of the environment is completely different in these two cases. The transition between both values is linear to reduce vibrations. The duration of transition is referenced as d . Also, two additional gains not shown in Figure 3.1 are used as weights $W \in [0, 1]$ to set the feedback and feedforward contributions. Finally, the two most important parameters for the LWPR initialisation are provided : init_D is the distance metric assigned to newly created local models and norm_{in} is the expected range of the input data. As you may have noticed, init_D and norm_{in} are three dimensional because the number of LWPR inputs has been reduced to three ($\phi, \theta, \dot{\theta}$). More information about this choice is provided in Section 4.4. below.

Parameter	Value	Parameter	Value	Parameter	Value
$K_{p,\text{stance}}$	6000 [Nm/rad]	$K_{p,\text{swing}}$	100 [Nm/rad]	$d_{\text{swing to stance}}$	0.005 [s]
$K_{d,\text{stance}}$	13 [Nm/rad.s ⁻¹]	$K_{d,\text{swing}}$	3 [Nm/rad.s ⁻¹]	$d_{\text{stance to swing}}$	0.0005 [s]
$W_{\text{LWPR},\text{stance}}$	1 [-]	$W_{\text{LWPR},\text{swing}}$	0.5 [-]	init_D	$250 \cdot \mathbf{I}_{3 \times 3}$
$W_{\text{PD},\text{stance}}$	0.25 [-]	$W_{\text{PD},\text{swing}}$	0.5 [-]	norm_{in}	[1 25° 25°]

Table 4.1: Parameters values.

4.3. Results

The results obtained from the first part of the experiment are provided in Figure 4.6. Only the 12 first steps are displayed as the following ones are similar. On the first subplot, one can observe that the angular position tracking is initially quite bad, but then rapidly improves as the blank LWPR is

updated. The second subplot shows the contribution of feedback and feedforward actions in the total torque applied at the prosthetic ankle. Remember that the total torque is the sum of the feedback and feedforward actions. More significant, the third subplot shows the evolution of the feedback component torque with respect to the global joint torque at each step. One can observe that this contribution starts at 44% and rapidly drops down to 4%. Almost 95% of this reduction is reached after 3 steps only, showing that the LWPR converges sufficiently fast to learn from the user in real-time. This training session leads up to 31 RFs being automatically generated.

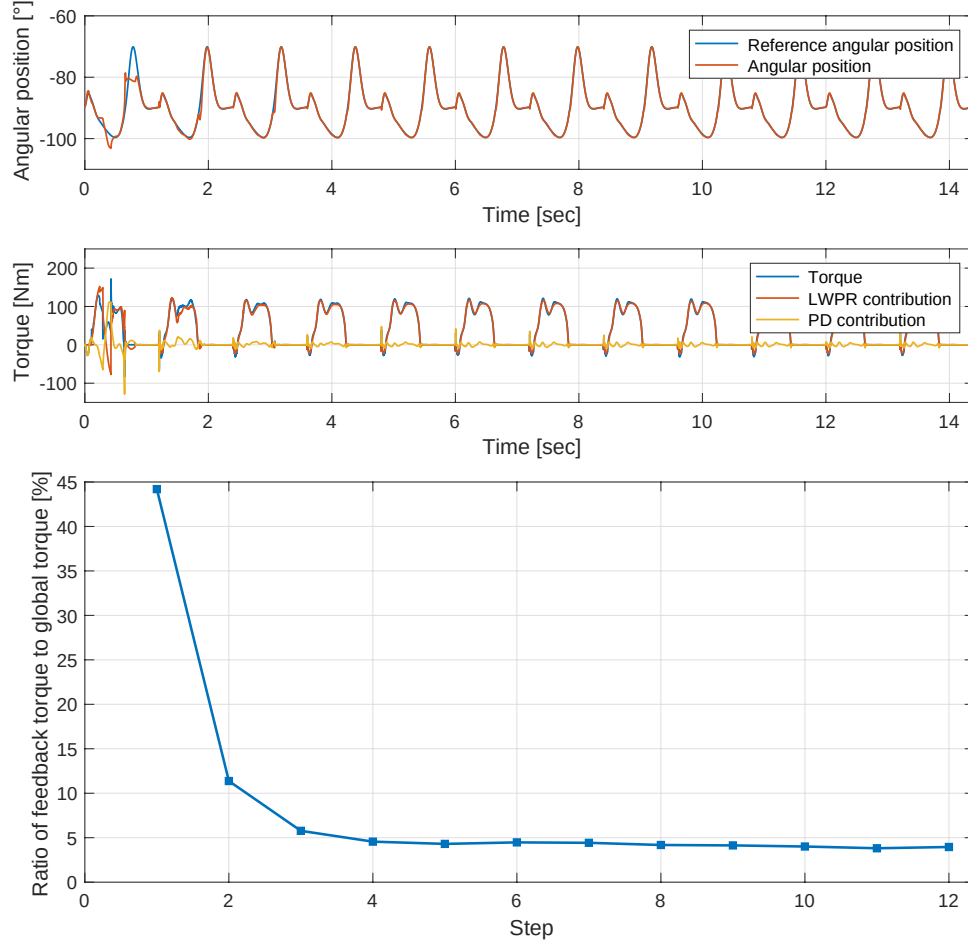


Figure 4.6: Learning process of a blank LWPR model throughout 12 steps with parameters provided in Table 4.1. (a) Ankle angular position tracking. (b) Total torque generated by the prosthesis, with LWPR and PID contributions. (c) Evolution of the ratio of feedback torque to global torque.

In addition, the reference position tracking along the 15th step is analysed in Figure 4.7 and compared to the tracking performance of a pure feedback based-regulator. The root mean square errors between the reference and actual angular positions are 0.15 [°] and 0.57 [°] for combined feedback and feedforward actions and pure feedback respectively.

Despite these encouraging results, the stability of the LWPR seems to be compromised after a certain number of steps as depicted in Figure 4.8. As a consequence, the feedback contribution seems to increase again around the 50th step, even though all simulation parameters are constant. Figure 4.9 focuses on the 60th step and shows the instability following heel-strike with more details. This phenomenon appears only if the learning process is enabled.

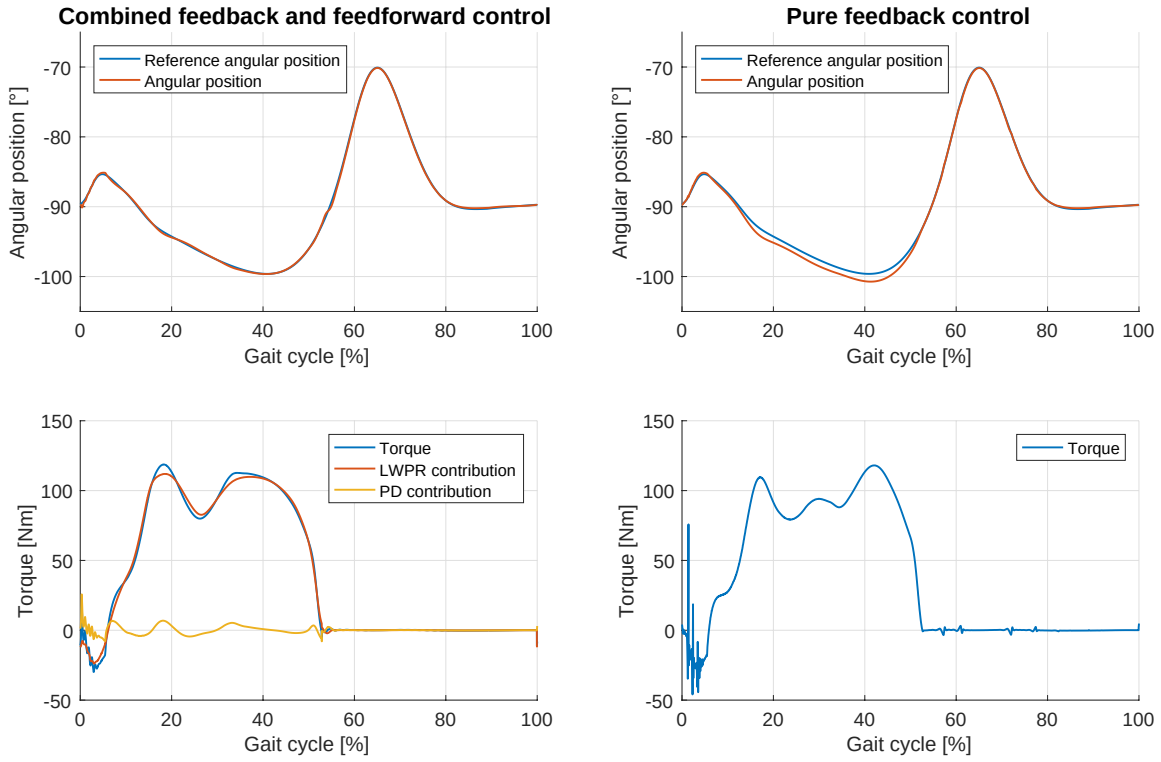


Figure 4.7: Angular position tracking (up) and torque (down) for the ankle prosthesis along the 15th gait cycle compared to a pure feedback based tracking with the same reference.

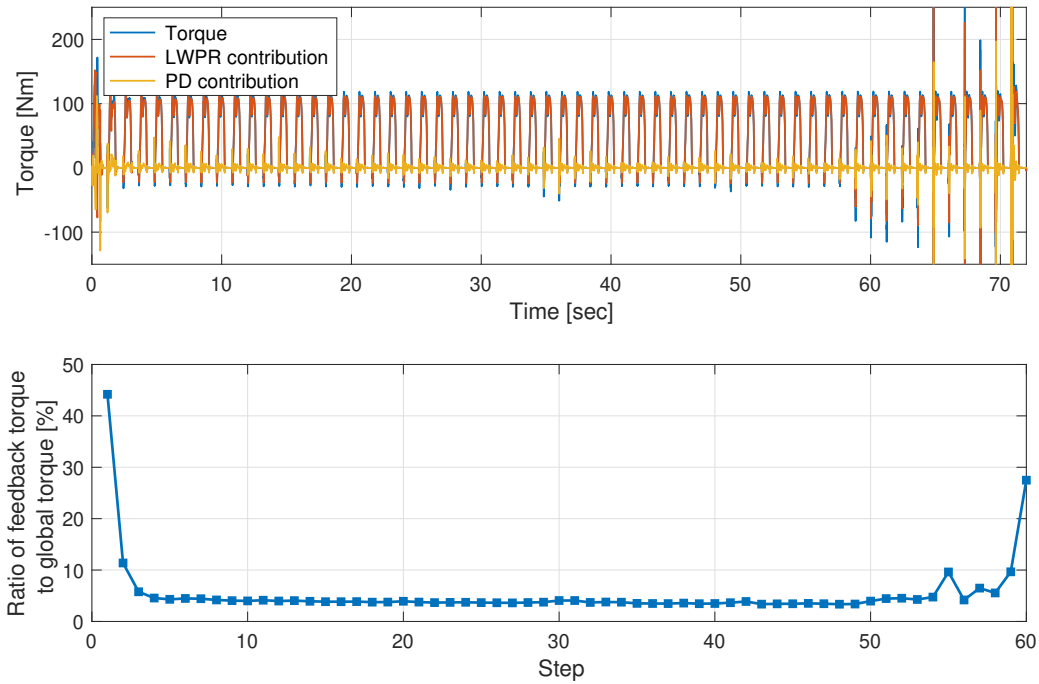


Figure 4.8: Learning process of a blank LWPR model throughout 60 steps with parameters provided in Table 4.1. Emergence of an instability at the 50th step. (a) Total torque generated by the prosthesis, with LWPR and PID contributions. (b) Evolution of the ratio of feedback torque to global torque.

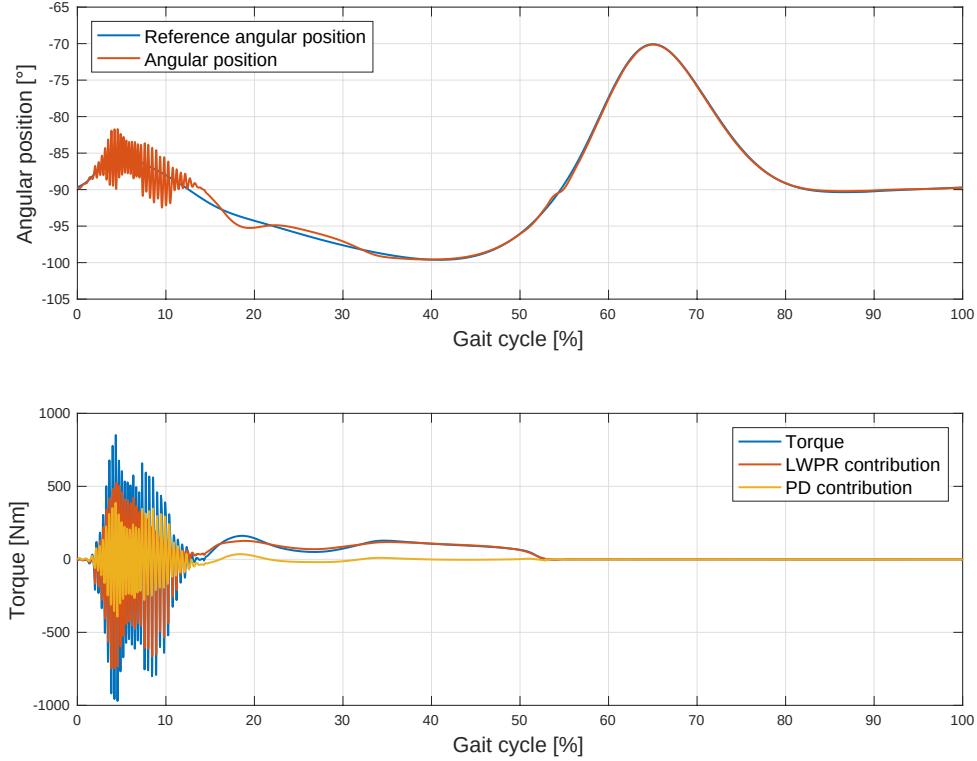


Figure 4.9: Angular position reference tracking (a), and total torque (b), along the 60th gait cycle with the parameters of Table 4.1.

4.4. Discussion

We demonstrate through Figure 4.6 that the LWPR is able to converge in three steps towards a quasi steady-state, sufficiently fast to learn from the user in real-time. The reduced contribution of the feedback regulator ($\sim 5\%$) is a good indicator of the accuracy of the inverse dynamics model estimation and the compliance of the system. Indeed, Figure 4.7 proves that the achievable tracking performances are at least as good as a good feedback regulator. In our case, we observe that the quality of tracking is slightly better for the proposed control scheme. This is due to the fact that the PD requires a minimal error to provide the desired torque, while the LWPR learns the almost exact torque required to reach the reference. On the other hand, the LWPR introduces a slight oscillation at the toe-off event when the torque must be reduced very fast. A higher number of RFs (i.e. smaller distance metrics) might be required to capture this steep slope, at the expense of higher computational costs.

It is important to note that the fast convergence speed at the beginning of the experiment is due to the large initial feedback contribution. This is not necessarily representative of a real-case scenario where a pre-trained model might be used instead of a blank one to maximise performance from the beginning. For a daily use, the model can, for example, be saved at the end of a walking session and reloaded at the beginning of the following one. The goal of this experiment was rather to illustrate the learning features of the LWPR. We have also discovered through Figure 4.8 that the LWPR might become unstable after a certain number of similar steps (~ 50). So far, we have not found any valid explanation to this phenomenon and we suggest to disable the LWPR learning feature when the feedback contribution is sufficiently low or steady from one step to the next.

Regarding the LWPR, Section 3. of Chapter 3 mentions four inputs which are the current angular position θ , the angular speed $\dot{\theta}$, the desired angular position θ_{ref} , and the gait phase ϕ . However, it

appears that this configuration leads to poor results. Indeed, one can observe in Figure 4.10 some undesired peaks in the LWPR contribution following heel-strike. Despite a high number of manual tuning tests (~ 40) for the values of init_D and K_p/K_d , these results are the best that we could obtain. This effect is cancelled by removing the angular velocity $\dot{\theta}$, computed by numerical differentiation of the position signal, from the LWPR inputs. For this entire work, three inputs are therefore used. A possible solution to reduce these instabilities might lie in the init_D parameter. As a reminder, this matrix sets the width of the receptive fields in all the dimensions defined by input variables. Until now, we have used a scalar matrix with identical distance metrics in all dimensions. A line of research might be to modulate each diagonal term of init_D to reduce the impact of destabilising inputs.

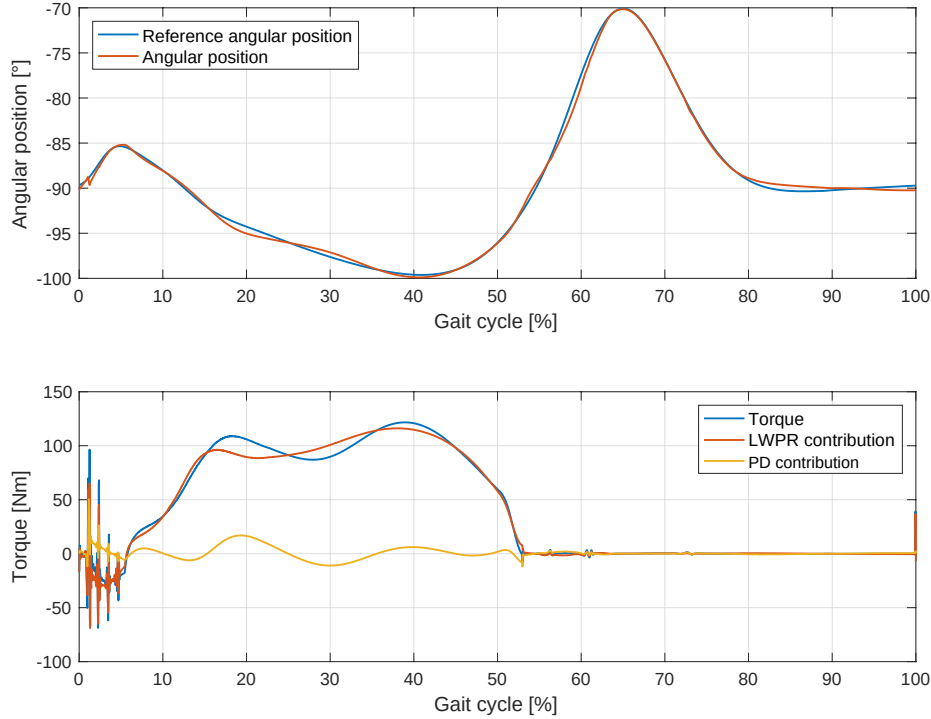


Figure 4.10: Angular position reference tracking (a), and torque (b), along the 15th gait cycle using four inputs for the LWPR.

5. Compliance analysis

This section aims at quantifying the compliance of actuation through an impedance analysis. We expect to demonstrate that the parallel contribution of feedback and feedforward actions based on LWPR leads to a better compliance than a pure feedback-based control.

5.1. Method

As explained in Section 2.2. of Chapter 1, the impedance of a system is defined as the transfer function between the interaction force with the environment and the joint velocity (e.q. 1.1). In this case, the system considered is the controlled ankle prosthesis linked to the user's body. As the only joint controlled by the prosthesis is revolute, force and velocity are replaced by torque and angular velocity, but the concept remains unchanged.

As in the previous experiment, the kinematics of all revolute joints are driven except for the prosthetic ankle which is torque controlled. A unique step is simulated at a gait period of 1.2 [s].

The pre-trained LWPR model is loaded from the previous experiment after 15 steps. However, the learning mechanism is disabled to avoid affecting the compliance analysis. Without any adaptation, there is no point in simulating several steps as all signals are perfectly periodic. Parameters values are provided in Table 4.1. The experiment consists in two stages. First, a sinusoidal torque perturbation of known amplitude (1 [Nm]) and frequency is applied on the virtual prosthesis. This periodic perturbation introduces some vibration of the prosthesis and a torque response from the controller at an equal frequency and variable amplitude. This way, the ratio between the torque and the speed amplitude of oscillation is measured throughout the gait cycle. To capture the amplitude of oscillation, an experiment is first performed and recorded without perturbation (Figure 4.11 (a)). The unperturbed torque and angular trajectory are subtracted from the perturbed signals, providing the additive oscillation theoretically centred around 0. The first solution for amplitude extraction consists in using the Hilbert filter to extract the signal envelope and average along the desired section of the gait cycle (Figure 4.11 (b)). However, we have observed that the angular trajectory and torque pattern could diverge from the unperturbed ones when applying a torque perturbation (oscillations not centred around 0). For this reason, we have decided to use the Fourier transform, extracting the oscillation component of both signals at the frequency of perturbation (Figure 4.11 (c)). As illustration, Figure 4.11 shows a sinusoidal signal of amplitude 0.5 and frequency 1 [Hz] at which is added another sinusoidal signal of amplitude 0.1 ± 0.02 and frequency 50 [Hz]. The Hilbert method provides an average amplitude of oscillation of 0.0999, while the frequency analysis gives 0.0995 at 50 [Hz]. Both are really close to the original amplitude of 0.1. However, the Hilbert method will not work properly if the isolated oscillations are not centred around 0.

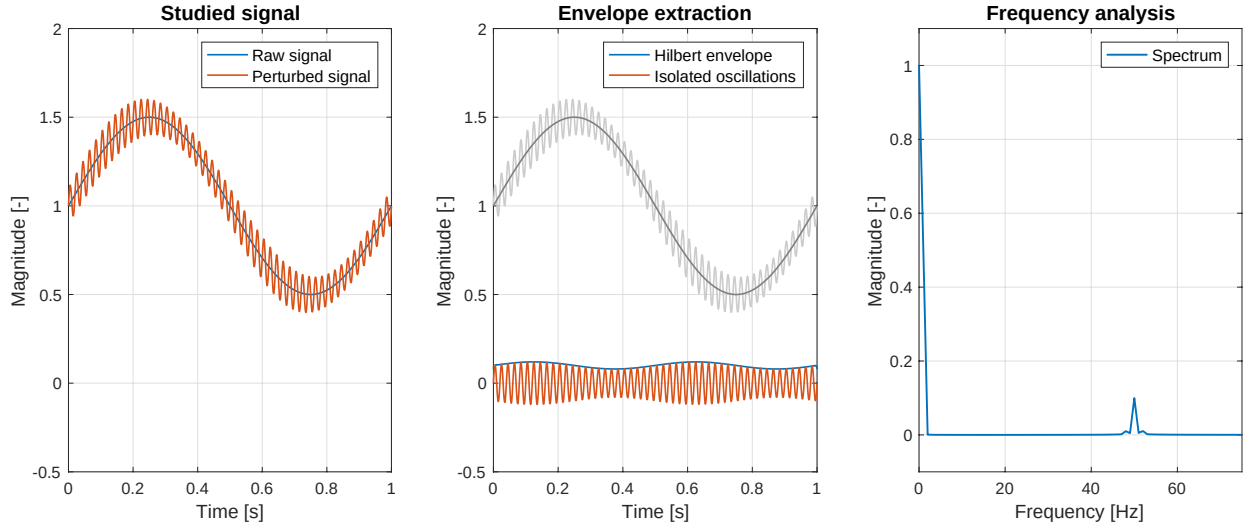


Figure 4.11: Extraction of the average amplitude of oscillation of a perturbed signal.

The impedance is studied separately along the stance and the swing phases as the dynamics of the environment is very different in those two cases. The experiment is repeated for a range of frequency from 1 to 2500 [Hz], i.e. $1/4$ of the simulation frequency (10000 [Hz]) to ensure a sufficient amplitude of perturbation. A Bode diagram is then built and provides the evolution of the impedance magnitude with frequency of the perturbation. We can assume that the observed transfer function is characterised by three parameters, i.e. M , B , and K , being the inertia, damping and stiffness of the system (eq. 4.2). For a revolute joint, units are $[kg \cdot m^2]$, $[\frac{Nm}{rad \cdot s^{-1}}]$, and $[\frac{Nm}{rad}]$ respectively.

$$Z(s) = Ms + B + \frac{K}{s} \quad (4.2)$$

The second stage of the experiment consists in fitting the magnitude of this second order model to the observed data (i.e. finding M , B , and K) using the least-squares method. The analytical expression of the amplitude is derived in eq. 4.5. Given this expression and as explained by H. Vallery et al. [51], we expect to observe a V- or U-shaped Bode diagram defined by a predominant stiffness at low frequencies, damping at intermediate frequencies, and inertia at high frequencies.

$$|Z(s)| = \frac{|Ms^2 + Bs + K|}{|s|} \quad (4.3)$$

$$= \frac{|-M\omega^2 + Bj\omega + K|}{|j\omega|} \quad (4.4)$$

$$= \frac{\sqrt{(K - M\omega^2)^2 + (B\omega)^2}}{\omega} \quad (4.5)$$

Finally, the same process is reproduced with a pure feedback based controller for comparison. All parameters values are provided in Table 4.1. We expect the stiffness to be smaller with the feedforward contribution than with pure feedback. The obtained inertia is compared to the known inertia of the foot for validation.

5.2. Results

The Bode diagram showing the magnitude of impedance during the stance phase is provided in Figure 4.12. A comparison is performed between the proposed controller and a pure feedback-based controller. The second order model approximation based on eq. 4.5 is shown on the same figure. The values of M , B , and K , solution of the least-square optimisation problem are provided in Table 4.2. One can observe in both cases a L-shaped diagram with a bandwidth of 100 [Hz], i.e. the maximum frequency until which the system behaves as a pure spring around the nominal trajectory. Since the rising slope of the U-shaped diagram is not visible, the fitting process is unable to find the right value of M and tends to set it at 0. Despite this, the second order model well captures the shape of the experimental data. Table 4.2 shows that the obtained equivalent stiffness is 3.25 times smaller when using the LWPR than with the pure feedback regulator. The damping is 4.1 times smaller too.

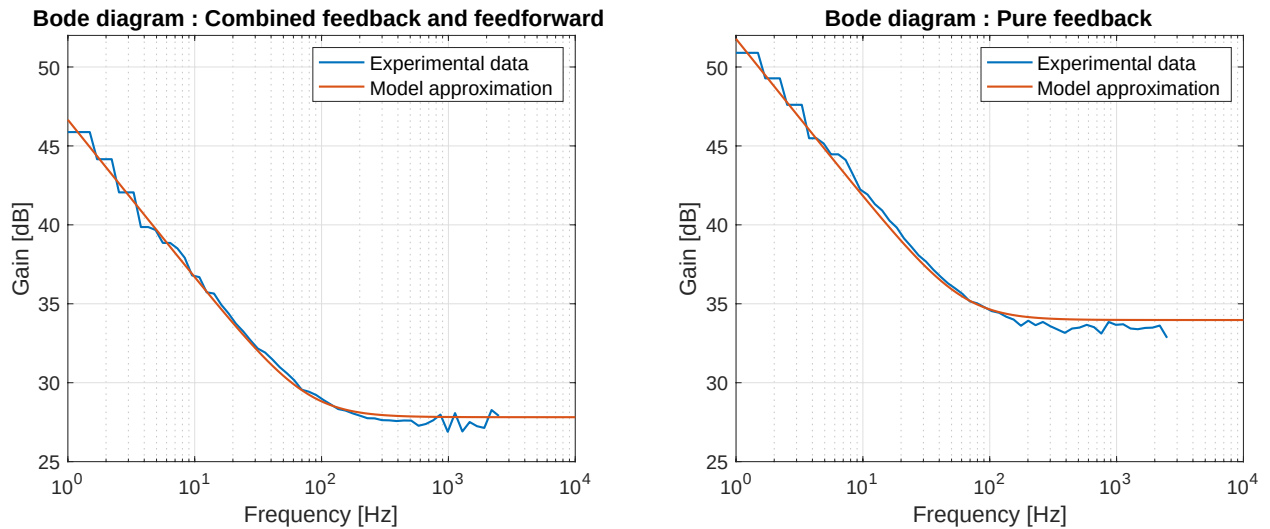


Figure 4.12: Bode diagram of the impedance transfer function $Z(s)$ for the stance phase. Comparison between the combined feedback/feedforward based controller (a), and pure feedback based controller (b). Blue = experimental data obtained through simulation, red = model approximation (eq. 4.5).

Parameter	Combined action	Pure feedback
$K \left[\frac{Nm}{rad} \right]$	46293	150578
$B \left[\frac{Nm}{rad \cdot s^{-1}} \right]$	605	2495
$M \left[kg \cdot m^2 \right]$	? (0)	? (0)

Table 4.2: Fitted parameters values for the impedance model (eq. 4.2) during the stance phase.

The same diagram is provided for the swing phase in Figure 4.13 along with the fitted parameters values in Table 4.3. This time, a U-shaped diagram is obtained for the proposed controller, and a L-shaped one for the pure feedback. However, the fitting process managed to find a non-zero value for M in both cases. The pure stiffness bandwidth is now reduced at 10 [Hz], and the inertia takes over at approximately 1000 [Hz]. This time, the equivalent stiffness and damping are both 2 times smaller when using the LWPR than with the pure feedback regulator. The measured inertia for both controllers are much higher than the value set in *Robotran* to perform the simulations.

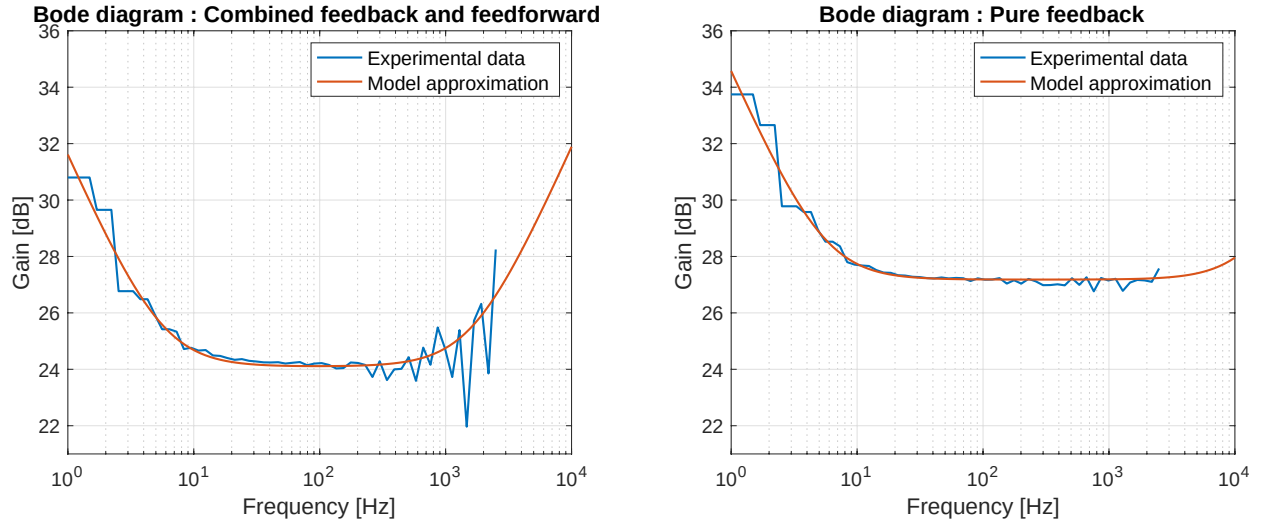


Figure 4.13: Bode diagram of the impedance transfer function $Z(s)$ for the swing phase. Comparison between the combined feedback/feedforward based controller (a), and pure feedback based controller (b). Blue = experimental data obtained through simulation, red = model approximation (eq. 4.5).

Parameter	Combined action	Pure feedback
$K \left[\frac{Nm}{rad} \right]$	1426	2819
$B \left[\frac{Nm}{rad \cdot s^{-1}} \right]$	258	522
$M \left[kg \cdot m^2 \right]$	0.1523	0.0343

Table 4.3: Fitted parameters values for the impedance model (eq. 4.2) during the swing phase.

5.3. Discussion

These results prove that the contribution of the LWPR allows to significantly increase compliance with an equivalent stiffness two to three times smaller than the considered feedback regulator. Also, we have seen that the bandwidth is sufficiently high using the LWPR to compensate typical perturbation components linked with human locomotion (below 10 [Hz]), and damp higher frequency perturbations.

Concerning the stance phase, the L-shaped diagram obtained is probably due to the fact that the corner frequency for the rising slope is higher than 2500 [Hz]. To illustrate this fact, Figure 4.14 shows the same diagram over an extended range of frequency and three variants of the previously obtained model approximation. Only the parameter M is changed, from 0.001 to 0.1 [$kg \cdot m^2$]. One can observe in this illustration that the rising slope directly linked with inertia can emerge at a higher frequency than the maximal observable frequency of 2500 [Hz], and the fitting process would not make any difference between any of these situations.

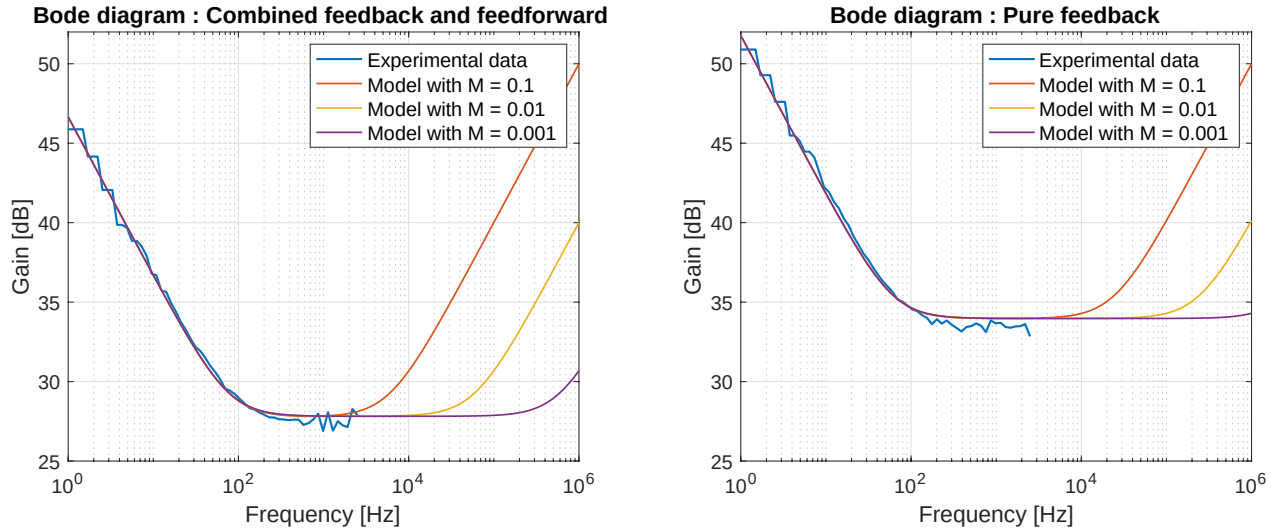


Figure 4.14: Bode diagram obtained for the stance phase, extended to 10^6 [Hz] with three model approximations. Comparison between the combined feedback/feedforward based controller (a), and pure feedback based controller (b).

Concerning the swing phase, the U-shaped diagram is closer to the expectations with a lower stiffness than the one observed in the stance phase. However, we do not know the reason why the computed value of the equivalent inertia above ground is much higher than the real inertia of the foot defined in *Robotran*. This might be due to the low number of samples at high frequencies, reducing the fit quality. Nevertheless, the value of M is not essential in the frame of this work where we stick to low frequencies.

Comparing Tables 4.2 and 4.3, one can observe that the ratio between stiffness of the pure feedback controller (~ 53) is equivalent to the ratio between $K_{p,stance}$ and $K_{p,swing}$ (60). Similarly, the ratio between the damping values (~ 4.8) is equivalent to the ratio between $K_{d,stance}$ and $K_{d,swing}$ (4.3). This makes sense since the feedback and damping of a pure feedback regulator are directly driven by the gains K_p and K_d .

6. Gait asymmetry correction validation

The goal of this experiment is to evaluate the ability of the control scheme to correct an artificially induced gait asymmetry by adapting the reference pattern of the ankle prosthesis, through motor primitives shifting. This stage introduces some tricky challenges. First, how should we generate realistic asymmetries? Most papers only provide trajectories recorded from healthy subjects. Second, how can we correct these asymmetries if all "uncontrolled" joints are driven? There is no ideal answer to these questions and this is a major limitation of the used simulation environment. Given these limitations, we have formulated a protocol in three stages which might provide some interesting information and lines of research.

6.1. Method

As a reminder, the three parameters on which we can play to reduce asymmetry are μ_1 , μ_2 , and μ_3 , i.e. the position of the three motor primitives constituting the prosthesis angular trajectory. The weights a_1 , a_2 , and a_3 are automatically updated to maintain a constant amplitude of the three peaks (Section 2.7. of Chapter 3).

Standard walking data are generally provided for slow, normal, and fast walking speeds (e.g. Winter reference [30]). Trajectories are slightly different from one walking pace to the other. This way, the experiment consists in modifying the walking pace of the bipedal system from normal ($T = 1.2$ [s]) to fast ($T = 1$ [s]). The kinematics of all "uncontrolled" revolute joints are driven to the trajectories provided in Figure 4.15 while the prosthetic ankle is driven to the trajectory generated by the adaptive motor primitives. We have decided to enforce the movement of the virtual prosthesis in order to decouple the problems of trajectory generation and trajectory tracking. The hip trajectory is still scaled by a factor 1.3 with respect to the literature, and the pelvis motion is now fixed.

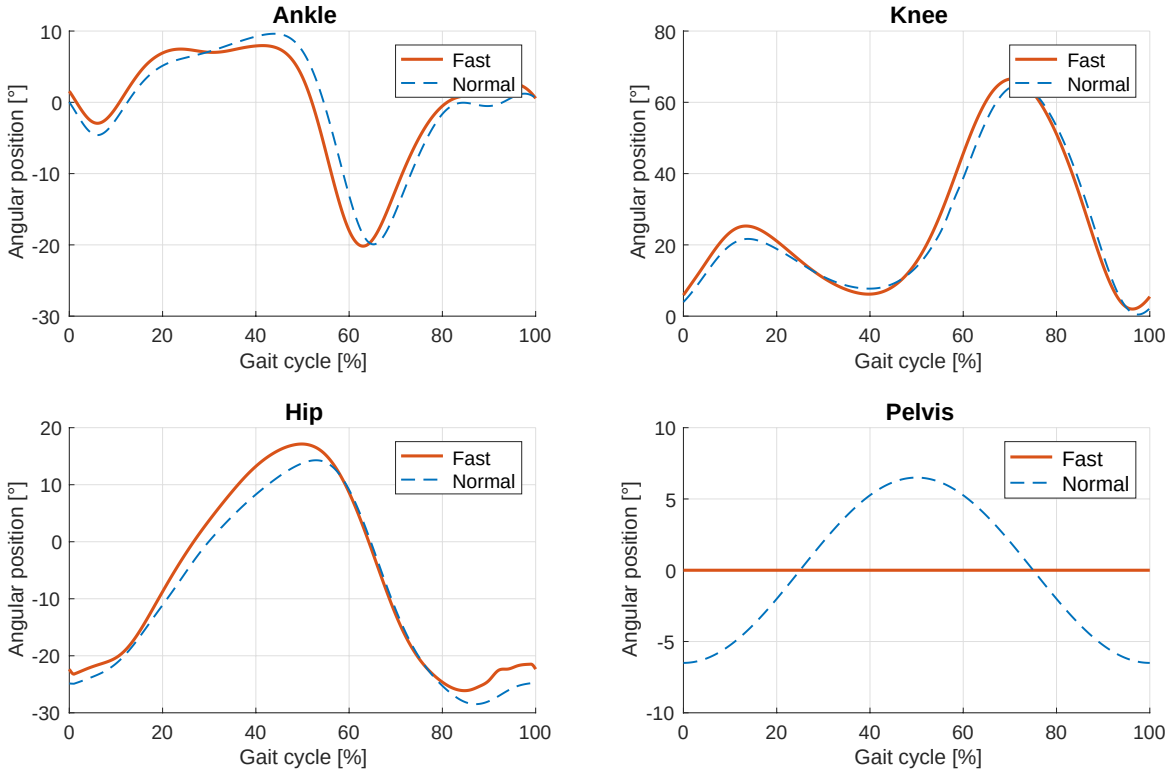


Figure 4.15: Driven trajectories for fast walking compared to normal trajectories previously used.

Even though the 5 driven joints instantaneously converge to the targeted walking pace, the ankle prosthesis is unaware of this change and maintains at first its angular trajectory. This inappropriate behaviour is expected to generate asymmetries between both legs. An asymmetry of foot-flat, heel-off, and toe-off events should be measured and progressively cancelled thanks to the adaptation of the motor primitives. However, as the angular velocity of the shins is fully driven by the (symmetrically actuated) upper joints in the simulation environment, these asymmetries cannot be measured by the virtual IMUs. For the purpose of this simple experiment, the position/force sensors located under the feet are used to detect these three events. The heel sensor was already used to detect heel-strike on the prosthetic side for ground contact detection and adaptive oscillators synchronisation. All events are now detected using the rules provided in Table 4.4 within the same FSM of Figure 3.5.

Event	Rule	Event	Rule
Heel-strike	$F_{\text{heel}} \neq 0$	Heel-off	$Z_{\text{heel}} - Z_{\text{toe}} \geq 0.04, F_{\text{heel}} = 0$
Foot-flat	$Z_{\text{toe}} \leq Z_{\text{heel}}$	Toe-off	$F_{\text{toe}} = 0$

Table 4.4: Rules for events detection based on ground force sensors.

The experiment consists in three stages. First, 15 steps are simulated at this new gait period of 1 [s] with the adaptation mechanism described in Section 2.7. of Chapter 3 fully enabled. For the sake of clarity, the adaption law is provided in eq. 4.6. The experiment is repeated three times with the inertia of adaptation N firstly set to 10, then 15, and 20. During this experiment, the evolution of μ_1 , μ_2 , and μ_3 as well as the events asymmetry are monitored. The former is expected to converge toward a new value close to the nominal one, while the latter is expected to converge toward zero.

$$\begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix} + \begin{bmatrix} \text{AI}_{\text{FF}} \\ \text{AI}_{\text{HO}} \\ \text{AI}_{\text{TO}} \end{bmatrix} \cdot \frac{1}{N} \quad (4.6)$$

After this, a sensitivity analysis is performed to study the effect of μ_1 , μ_2 , and μ_3 , i.e. the position of motor primitives, on the timing of the foot-flat, heel-off, and toe-off events at normal walking speed. To do so, three runs of 10 steps are simulated, with varying μ_1 , μ_2 , and μ_3 respectively, below and above the nominal value (Table 3.2). Since motor primitives overlap each other (Figure 3.12 (b)), we expect to see some correlation between every μ_i and every corresponding event. The obtained sensitivity functions are linearized around the nominal value of μ . The nine slopes are stored in a 3×3 matrix $\mathbf{S}$. Ideally, the diagonal terms, i.e. the slopes $\Delta\phi_{\text{FF}}/\Delta\mu_1$, $\Delta\phi_{\text{HO}}/\Delta\mu_2$, and $\Delta\phi_{\text{TO}}/\Delta\mu_3$ should be close to unity, while all others should be close to zero. Once inverted, the sensitivity matrix can be used for control purpose by providing the variation of μ_1 , μ_2 , μ_3 required to modify the timing of each event as desired (eq. 4.7 and 4.8). However, this method supposes that the effect of each μ is purely additive.

$$\begin{bmatrix} \Delta\phi_{\text{FF}} \\ \Delta\phi_{\text{HO}} \\ \Delta\phi_{\text{TO}} \end{bmatrix} = \mathbf{S} \cdot \begin{bmatrix} \Delta\mu_1 \\ \Delta\mu_2 \\ \Delta\mu_3 \end{bmatrix} \longrightarrow \begin{bmatrix} \Delta\mu_1 \\ \Delta\mu_2 \\ \Delta\mu_3 \end{bmatrix} = \mathbf{S}^{-1} \cdot \begin{bmatrix} \Delta\phi_{\text{FF}} \\ \Delta\phi_{\text{HO}} \\ \Delta\phi_{\text{TO}} \end{bmatrix} \quad (4.7)$$

With this in mind, the third stage consists in simulating 15 more steps at a gait period of 1 [s] while enabling the adaptation mechanism based on the sensibility matrix. A comparison of asymmetry correction performance is performed with results of the first stage.

$$\begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix} + \mathbf{S}^{-1} \cdot \begin{bmatrix} \text{AI}_{\text{FF}} \\ \text{AI}_{\text{HO}} \\ \text{AI}_{\text{TO}} \end{bmatrix} \cdot \frac{1}{N} \quad (4.8)$$

6.2. Results

The results of the first experiment using the simple asymmetry correction are provided in Figure 4.16. The left subplots show the evolution of each asymmetry index throughout the 15 simulated steps. One can observe that the asymmetry starts at 1.056 [%] for foot-flat, -2.523 [%] for heel-off, and 2.508 [%] for toe-off, then converges towards a value closer to zero. However, the remaining asymmetry is still significant. Regarding the position of motor primitives, one can observe that they evolve quasi-linearly and do not converge towards a specific value. A reduction of N induces as expected a faster evolution of the μ_i and a slightly better asymmetry reduction.

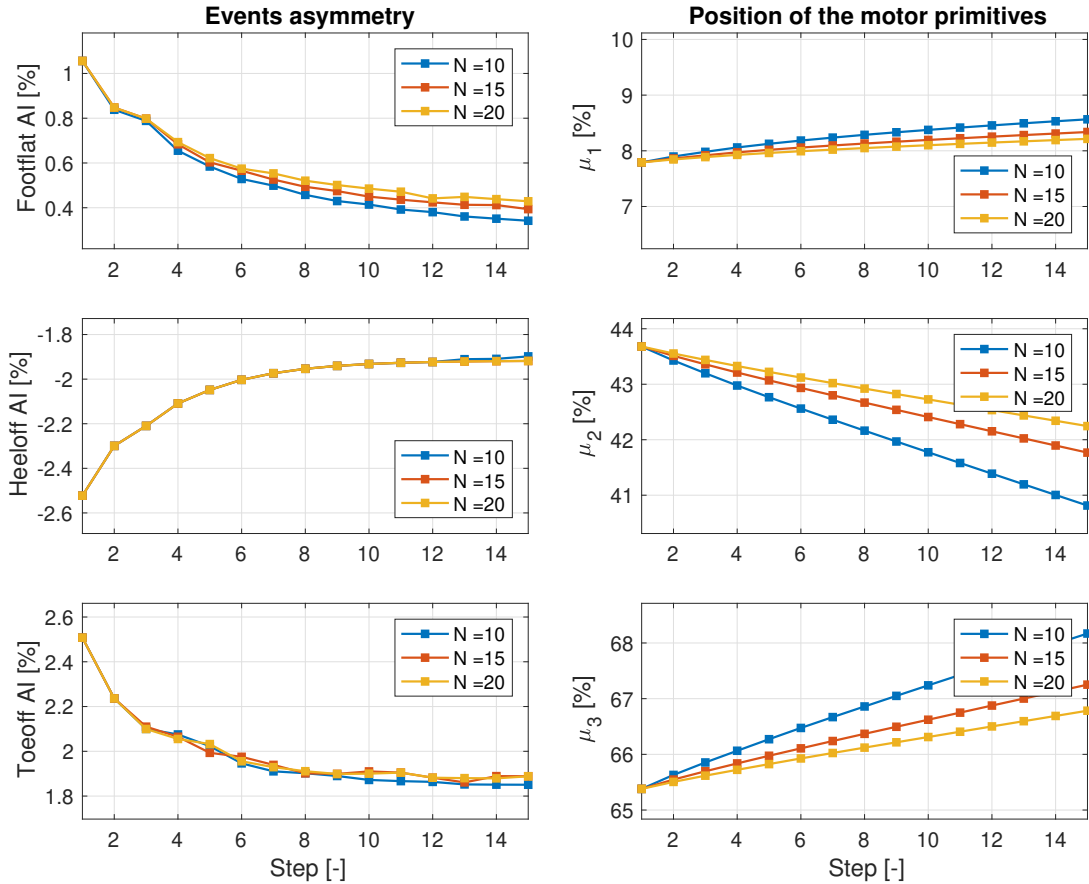


Figure 4.16: Evolution of events asymmetry and position of motor primitives when switching from normal ($T = 1.2$ [s]) to fast ($T = 1$ [s]) walking speed, using the simple adaptation mechanism.

Let us now move on to the sensitivity based adaptation strategy. First, the sensitivity analysis showing the impact of each μ on the time of occurrence of each event is presented in Figure 4.17. The linearization of the data around the nominal value is also displayed on the same figure. These

slopes are stored in the 3×3 matrix $\mathbf{S}$. Concerning diagonal terms, the curve of subplot (a) has a surprisingly steep negative slope, i.e. a negative correlation between μ_1 and the foot-flat event. The one of subplot (e) is slightly positive inducing a positive correlation between μ_2 and the heel-off event. Finally, the one of subplot (i) is surprisingly flat meaning no correlation between μ_3 and the toe-off event. Concerning non-diagonal terms, the curves of subplots (b), (c), (d), (g) and (h) are very flat as expected. However, the slope of the curve (f) is surprisingly steep and positive. The content of matrices $\mathbf{S}$ and $\mathbf{S}^{-1}$ are provided in eq. 4.9.

$$\mathbf{S} = \begin{bmatrix} -0.560 & -0.04 & 0.008 \\ 0.001 & 0.075 & 0.175 \\ 0.007 & -0.03 & -0.02 \end{bmatrix} \quad \mathbf{S}^{-1} = \begin{bmatrix} -1.63 & 0.397 & 3.446 \\ 0.547 & -4.31 & -45.64 \\ 0.246 & 7.547 & 19.478 \end{bmatrix} \quad (4.9)$$

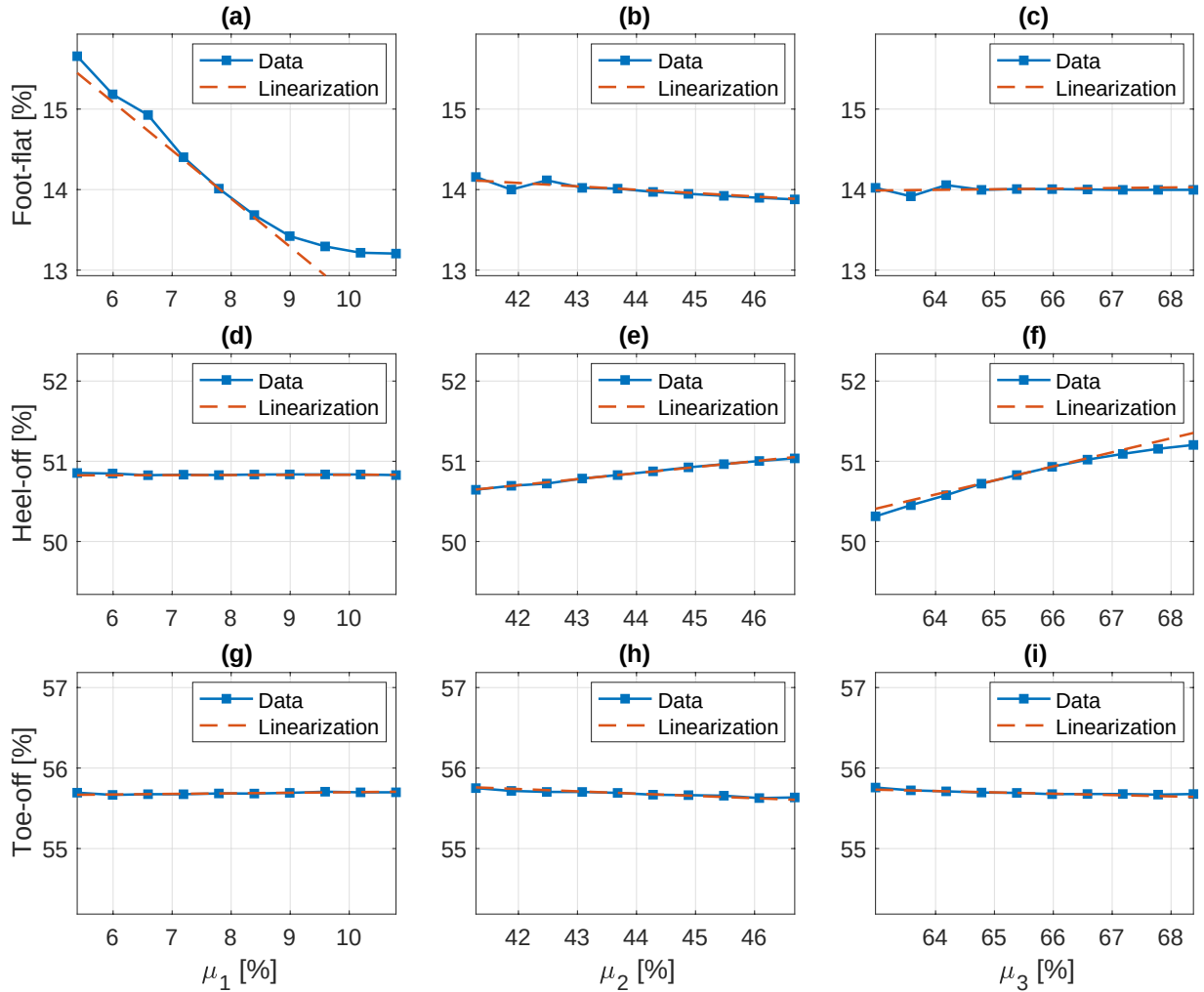


Figure 4.17: Sensitivity analysis of μ_i on the position of each event.

The results of the third part of the experiment using the inverse sensitivity matrix $\mathbf{S}^{-1}$ in the adaptation mechanism are presented in Figure 4.18. This time, one can observe that the μ evolve

much faster than for the first experiment and strike the hard limits preventing to fall below zero for μ_1 , or to flip positions between μ_2 and μ_3 . Events asymmetries also evolve faster and stabilise when the values of μ_i reach the boundaries. The same trends can be observed with AI tending to progressively decrease with the number of steps, except for the foot-flat AI which reaches 0 but ends up converging towards -1. Similarly to the first experiment, one can observe that a reduction of N results in a faster evolution of μ_i and AI.

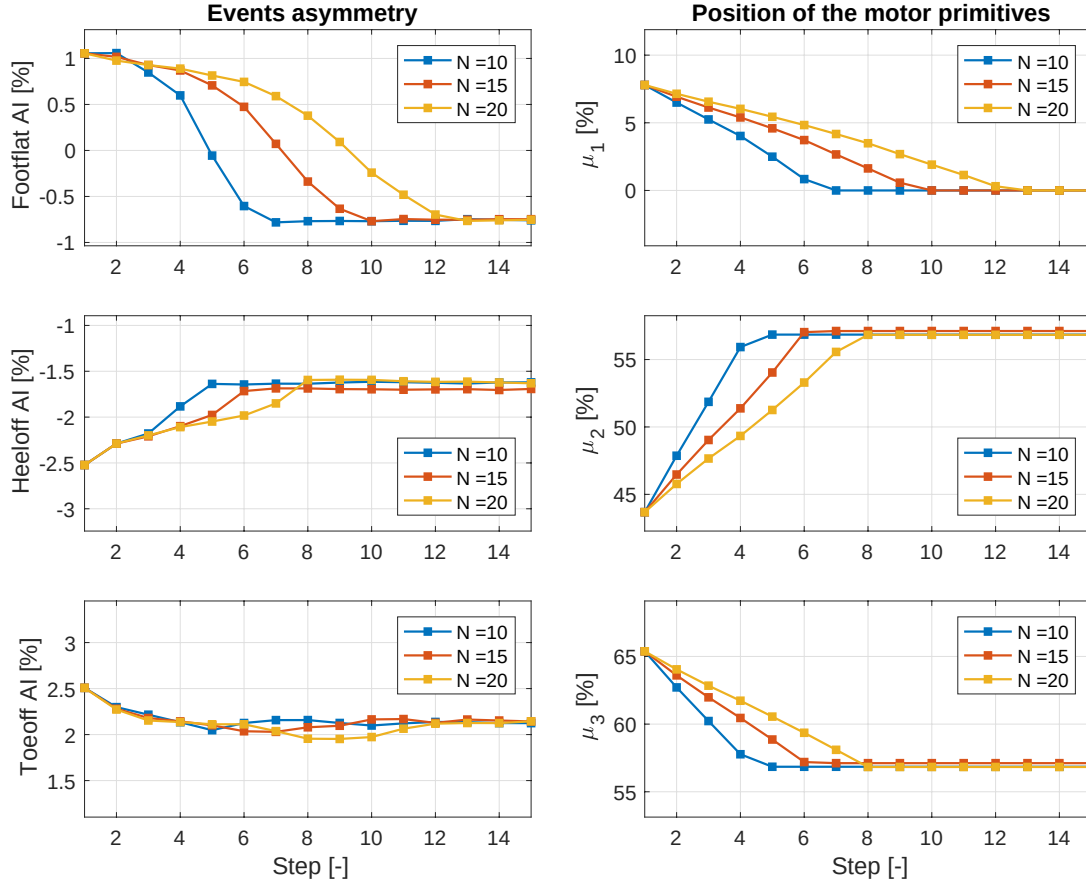


Figure 4.18: Evolution of events asymmetry and position of motor primitives when switching from normal ($T = 1.2$ [s]) to fast ($T = 1$ [s]) walking speed, using the sensitivity-based adaptation mechanism.

6.3. Discussion

Overall, the results obtained through these experiments are quite disappointing. The adaptation mechanism does not manage to cancel asymmetry, or at least converge towards a stable state which minimises asymmetries, either using the simple or the sensitivity-based strategy. Consequently, the μ_i keep evolving as the asymmetry is not cancelled until reaching a hard boundary. The sensitivity-based approach is more efficient but does not converge any better. Nevertheless, the experimental protocol is not ideal : biologically speaking, gait asymmetries are generated by the complex articulations of all lower-limb joints as explained in Section 1.1. of Chapter 1. In this case, the gait is mostly symmetric as the hip and knee joints are driven to identical trajectories and are therefore symmetric. Only the prosthetic foot does not behave properly. Yet, the algorithm does not aim at mimicking the exact motion of the sound foot but rather reducing asymmetries by the reaction of upper joints, otherwise we would have opted for the echo control strategy (Section 2.2. in Chapter 1). The simulation envi-

ronment also introduces inconsistent behaviours from a dynamic and kinematic standpoint, mainly due to the imperfect ground contact model and the unique vertical degree of freedom of the trunk preventing forward progression. Among other things, the toe-off event occurs much earlier (Figure 4.8) than in a standard gait cycle (Winter [30]), making the push-off movement of the foot completely useless as performed in the air. This might explain why the impact of μ_3 is so small on the occurrence of all events. In conclusion, even though the results are not very encouraging, it could be interesting to test this adaption mechanism on a real prosthesis and validate or not its interest.

1. Challenges and adjustments of the strategy

Now that we have shown the interest of this novel control strategy, the next step is to evaluate its performance on the ELSA prosthesis. The migration of the algorithm from the simulation environment to the prosthesis environment requires additional adjustments. Indeed from a mechanical point of view, the parallel contribution of the SEA and the LPS introduces new challenges.

The proposed strategy assumes that the ankle prosthesis is a pure torque source, and indeed, ELSA is theoretically able to produce such torque patterns. However, its internal structure described in Section 1. of Chapter 2 introduces some limitations. The joint torque is produced both by the LPS and the SEA. The SEA generates a torque τ_{SEA} which is the difference between the desired torque τ_{des} and the contribution already generated by the LPS τ_{LPS} .

$$\tau_{SEA} = \tau_{des} - \tau_{LPS}$$

The torque reference is reached thanks to the LPS being engaged during the stance phase and disengaged during the swing phase. Particular attention must be paid to disengage the spring at the end of the push-off, otherwise preventing dorsiflexion during early-swing, the two springs being stretched in opposite directions. This step requires the parallel spring to be fully unloaded, in other words the ankle must reach sufficient plantarflexion. To be more precise, the plantarflexion following push-off must at least be equal to the angular position reached at footflat where the parallel spring has been locked (Figure 5.1 (1)-(2)). However, the compliant behaviour introduces uncertainty about the ability to reach the reference position for varying external forces. With this in mind, we cannot enforce a desired torque pattern without ensuring the LPS to be disengaged after push-off. One solution might be to hold maximum plantarflexion reference position when reaching (3) in Figure 5.1 until the LPS has been disengaged.

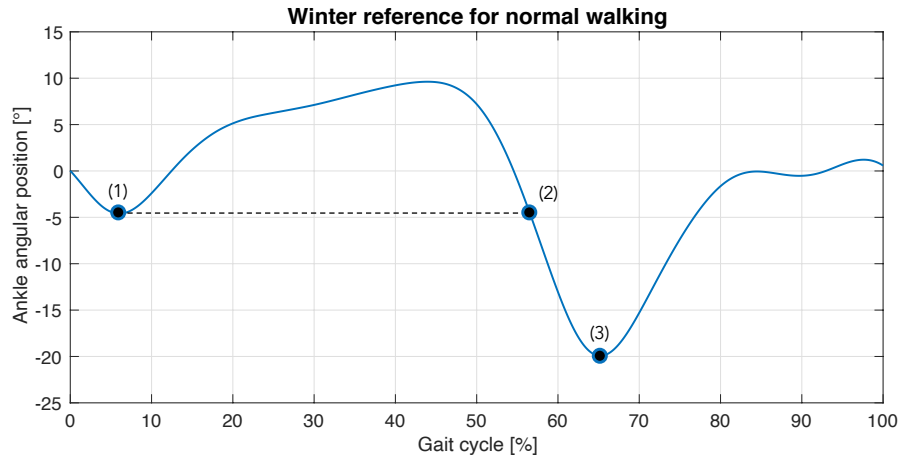


Figure 5.1: Sound ankle position trajectory (Winter), adapted from [30]. (1) The LPS is engaged and starts to stretch, (2) the LPS is unloaded, (3) maximum plantarflexion following push-off.

2. Implementation

This section focuses on a first-draft C++ implementation of the control strategy based on ROS to perform further experiments on the embedded system. The Robot Operation System (ROS) is a programming tool widely used in the field of robotics. The particularity of ROS lies in its modular architecture in the form of nodes, i.e. separate programs, each one being dedicated to a particular task, and which can easily communicate with each other. This architecture facilitates collaborative work. In our case, several nodes have thus been created to match the structure of Figure 3.1. The resulting network is depicted in Figure 5.2. Compared to Figure 3.1, some blocks have been assembled to reduce the amount of code. Note that ROS is designed for distributed computing, meaning that several components (robots, computers, mobile devices, etc.) connected to the same local network can be linked. In that case, all topics are shared and a node makes no assumption about where in the network it runs, except if it has to communicate with a piece of hardware physically connected to one of the components. Thanks to this feature, the new nodes (blue) can seamlessly be executed on a local computer or on the prosthesis, and all ROS topics can be remotely monitored.

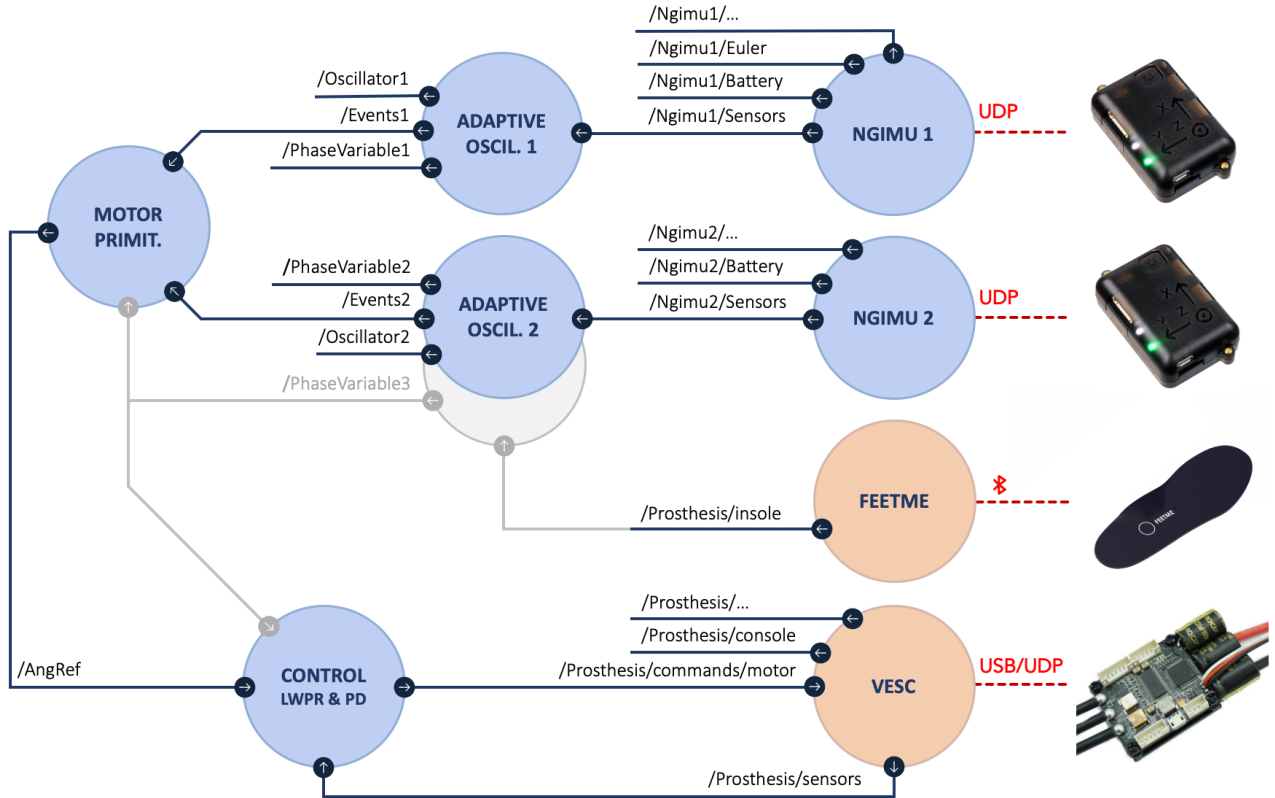


Figure 5.2: Synthetic diagram of the implemented ROS network. Nodes are depicted as circles (beige = already existing, blue = new, grey = to do), and topics as lines. Similar nodes designated with a digit use instances of the same code, with different parameters. (Source VESC : <https://www.banggood.com/>).

NGIMU This first node acquires and broadcasts on ROS topics the stream of data coming from the IMUs. Each NGIMU device (from x-io Technologies) is connected to the local WiFi network which can be either an external router or the hotspot generated by the prosthesis. These sensors embed a microprocessor and are able to provide a bunch of information from accelerations (angular, linear,

earth) to position (rotation matrix, Euler angles, quaternion), along with altitude, temperature, battery level, etc. It is configured to send the desired information as UDP package to the device of choice on the local network, at the desired rate. With this in mind, a custom code has been implemented in C++ to listen on a specific port for incoming UDP packages, decode these packages, and broadcast the data on dedicated topics. This way, the code can be reused for further projects. For the purpose of this thesis, only the gyroscope information is acquired.

Adaptive oscillators The raw angular velocity signals are received by the nodes **Adaptive oscillators** and learned by the adaptive oscillators. The event detection algorithm is directly applied on the learned signal. The node broadcasts three types of information : the learned signal, the current gait phase, and the phase value at each event detected during the previous step (published only at heel-strike). Note that the third phase variable accurately synchronised with the physical touchdown of the prosthetic foot and mentioned in Section 2.2. of Chapter 3 has not yet been implemented and should be in order to fully evaluate the proposed strategy on the prosthesis. This would either require to use the pressure insole or a simple mechanical switch.

Motor primitives This node reads the phase value published by the adaptive oscillators of the amputated leg and evaluates the three motor primitives to generate an angular reference position for the ankle prosthesis. In addition, it subscribes to both events topics and performs the heuristic primitive adaptation at every heel-strike.

Control Looking now at the actuation part, the **Control** node receives the reference and current position, as well as the phase variable to perform the feedback-error learning method and generate a torque reference pattern.

FeetMe & VESC These two nodes developed by F. Heremans handle the bidirectional communication with the FeetMe pressure insole and Vedder electronic speed controller (VESC) (STM32 based) which interfaces all the sensors / actuators and runs lower-level control. Note that the code running on the VESC motor controller should be properly adapted to allow full control of the joint torque from the ROS topics. For now, the VESC runs the middle-level controller developed by F. Heremans based on a finite-state machine.

ROS is very convenient for testing and monitoring. However, it is only compatible with Linux-based platforms. It introduces in this case some limitations due to the low computational capacity of the embedded Raspberry PI, and the limited bandwidth of the serial communication with the VESC. The main risk is that the maximum reachable control frequency does not allow to perform the desired compliant position control. An enhanced version of the middle level controller should be almost entirely implemented on the VESC to maximise performance.

3. Experimentation

Once the controller described in the previous section was implemented, we performed four simple real-life experiments to validate some parts of the control scheme or highlight technical issues. Real-time gait analysis has been performed as a first experiment in an attempt to demonstrate that we are able to acquire real data similar to the simulation, learn these raw data, and validate the detection algorithm on a sound subject.

3.1. Method

At an initial stage, a healthy subject equipped with one IMU on the side of each tibia (Figure 5.3) walked on a treadmill at his natural walking speed (4 [km/h]). IMUs are connected to the local network and communicate with a personal computer running the nodes **NGIMU** and **Adaptive oscillators** (Section 2. of Chapter 5). For this experiment, the analysed signals are the same as for the simulation described in Section 3. of Chapter 4. The first 30 seconds of the experiment are not recorded to reach a steady-state.

At the next stage, an experiment was carried out with the ELSA prosthesis mounted on an adapter and one IMU on each tibia (Figure 5.4). The purpose of this experiment is to demonstrate that the real-time gait analysis is also operational when walking with the prosthesis, using the well-tried current middle-level controller of F. Heremans. However, due to the exceptional circumstances experienced in 2020 resulting from the Covid-19 pandemic, the specific experimental equipment required (i.e. a treadmill) could not be used. For this reason, the healthy subject simply walked at his natural walking speed over a dozen steps on flat ground. The angular velocity of each leg was collected by the IMUs, this time connected to the local hotspot of the prosthesis. Only the nodes **NGIMU**, **Adaptive oscillators**, and **VESC** were used, allowing to monitor the angular position of ELSA.



Figure 5.3: Experiment #1 : IMUs on the lateral side of the left and right tibias.



Figure 5.4: Experiment #2 : ELSA active ankle prosthesis during an overground walking experiment performed by an able-bodied subject wearing an adapter.

3.2. Results

The results of the first experiment are very similar to those obtained with the simulation environment in Section 3. of Chapter 4 and are provided in Appendix C. Since this first step is conclusive, let us focus on the second experiment. The results obtained with the ELSA prosthesis are provided in Figure 5.5. One can observe that, here again, the adaptive oscillators are able to effectively learn the raw signal of the shank angular velocity gathered by the IMUs and to provide a phase variable that linearly increases from 0 to 100 [%] between two successive heel-strike events (Figure 5.5 (a)). In addition, the second subplot (Figure 5.5 (b)) demonstrates that the learned signal is smooth enough to systematically detect the gait events throughout the stride. Finally, the third subplot (Figure 5.5 (c)) shows the evolution of the ELSA angular position over time. The gait events are displayed on the angular position profile at their corresponding detection time.

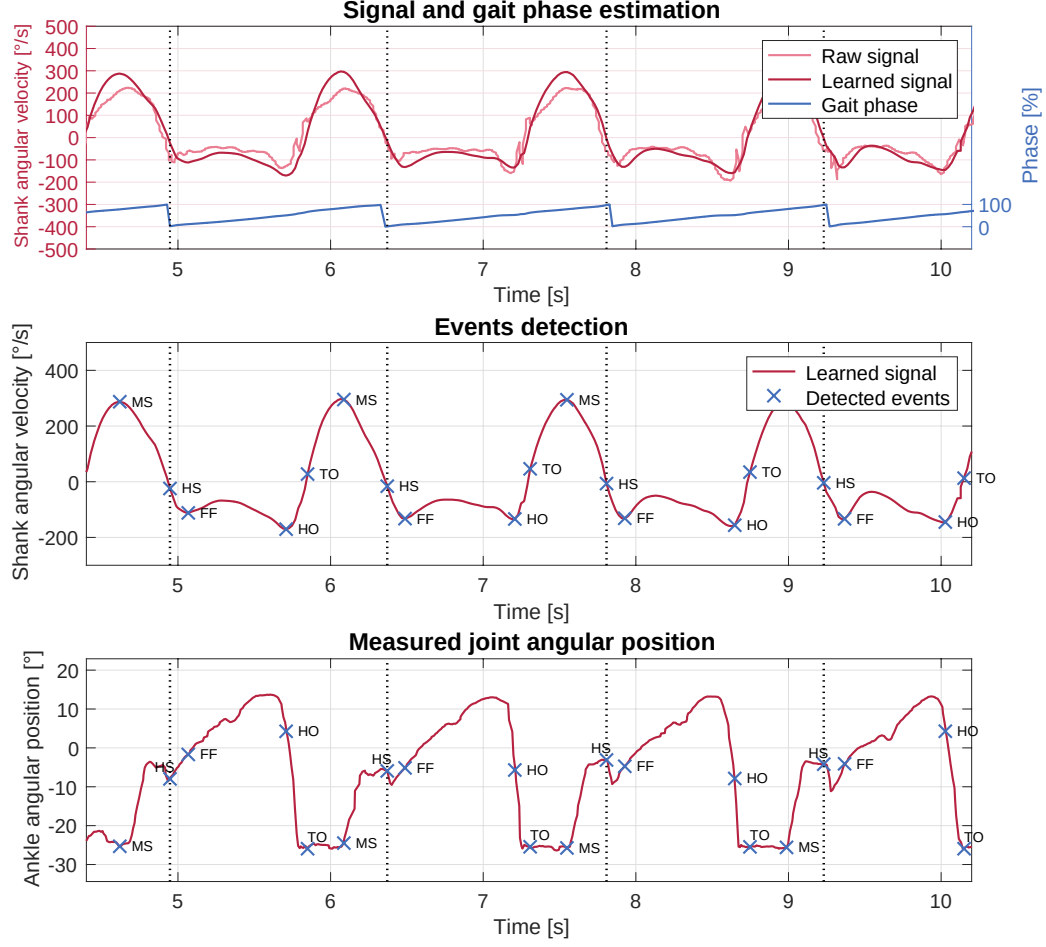


Figure 5.5: (a) Shank angular velocity [°/s] (raw and learned signals) and phase variable [%] over time [s] for the prosthetic side. (b) Learned signal of the shank angular velocity [°/s] over time for the prosthetic side with the detected events. (c) Measured angular position of the prosthetic joint [°] over time with the detected events. The vertical dotted lines show the time at which the heel-strike events are detected by the algorithm on the raw signal. The crosses correspond to an event detection : heel-strike (HS), foot-flat (FF), heel-off (HO), toe-off (TO) and mid-swing (MS).

3.3. Discussion

As expected, the adaptive oscillators and the event detection algorithm prove to be reliable in real-life experiments with the ELSA prosthesis (5.5 (a), (b)). In addition to the simulation results, this real-life experiment demonstrates that the detected heel-strike event using the algorithm of H. Silva et al. [35] (Figure 5.5 (c)) matches the expected location shown in Figure 3.10. Besides heel-strike, we are not able to prove that the detected events using Silva's algorithm correspond exactly to the physical events. The use of the embedded insole or an external slow-motion camera would be required to validate this aspect.

On another issue, the joint angular position pattern followed by the ELSA prosthesis (Figure 5.5 (c)) is different from that obtained by F. Heremans in Figure 2.7. Indeed, in Figure 5.5, one can observe a delay of ~ 0.3 [s] before the prosthetic foot dorsiflexion following toe-off. In other words, the angular position of the prosthetic foot remains constant from the toe-off to the mid-swing events. In the same way, if we take a closer look at the angular position obtained with a transtibial amputee

in Figure 2.16, one can observe the same anomaly. This could explain the lack of confidence of the user who feared to stumble and fall during the experiment and his tendency to lift his knee higher to ensure enough foot clearance during the swing phase. In order to enquire about the cause of the observed anomaly, additional experiments have been carried out and are described below.

4. Technical issues encountered with ELSA

In our way to performing early experiments on the prosthesis, involving the actuation part of our control strategy, we bumped into a major issue which might be linked to the observed anomaly. The low-level controller seems to struggle in performing simple current and position tracking. This section describes our observations coming from two simple experiments, and our current understanding of the problem.

4.1. Method

For the purpose of these two experiments, the middle-level controller is simplified to its paroxysm : the LPS is disengaged and the current FSM is replaced by a single state where references are directly provided from ROS topics to the low-level controller of the SEA.

The first experiment consists in isolating and reproducing the dorsiflexion movement inherent to the early-swing phase. To do so, the pylon of the prosthesis is held with the hand such that the entire system is sufficiently far from the ground. The prosthesis is manually put in maximal plantarflexion, reproducing the conditions following toe-off (Figure 5.6). The low-level controller is set in nut position tracking, with maximal desired speed of 400 [mm/s], maximal desired torque of 40 [Nm], and maximal desired current of 10 [A]. These are the parameters values used by F. Heremans in the current controller at early-swing. As a reminder, the nut position range goes from 0 (max. dorsiflexion) to 50 [mm] (max. plantarflexion). From there, a step reference position is applied to reach quasi-maximal dorsiflexion, i.e. $x_{\text{nut}} = 2$ [mm]. The nut is expected to reach the reference position within ~ 0.1 [s].

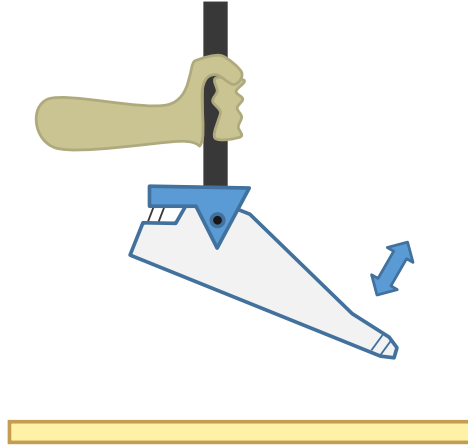


Figure 5.6: Schematic of the position tracking experiment, reproducing early-swing. The pylon is held with the hand, sufficiently far from the ground. The prosthesis is put in maximal plantarflexion.

The second experiment consists in performing a simple current tracking to verify the good behaviour of the low-level controller at its lowest level. Ideally, this experiment should be performed

The low-level controller is set in current tracking mode, meaning that no other command/limitation than current are considered by the controller. From there, a step reference is applied, from 0 to 0.1 [A], then 0 to 0.3 [A], then 0 to 0.5 [A], and finally 0 to 1 [A]. The measured current is expected to reach the reference within a few milliseconds.

4.2. Results

Results of the first experiment are provided in Figure 5.9. One can rapidly observe that the reference is absolutely not tracked properly. The nut position seems to move **away** from the reference when the step is applied. It repeatedly strikes the physical end-of-stroke and seems unstable.

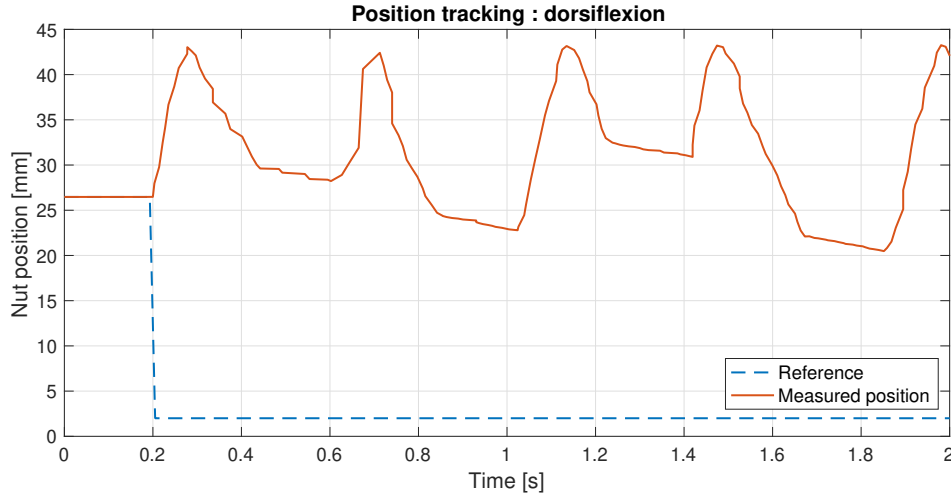


Figure 5.9: Position tracking with nut reference position going from quasi-maximal plantarflexion to quasi-maximal dorsiflexion. Blue dotted line : reference, red solid line : measured position.

The results of the second experiment are provided in Figure 5.10. Here again, the measured current does not converge to the desired value. For step amplitude of 0.1 [A] and 0.3 [A] (subplots (a) and (b)), the measured current oscillates within the range $\pm$ the step amplitude, and vibrations can be heard from the motor. Starting from 0.5 [A] (subplots (c) and (d)), the system becomes completely unstable with a current rising to 2 [A]. Violent oscillatory movements of the nut and pylon are observed, from one end-of-stroke to the other.

4.3. Discussion

We can obviously say that something is wrong with the low-level controller. Concerning the first experiment, we are surprised to observe such an unstable behaviour, much worse than what was observed when using the full controller of F. Heremans in Sections 3.2. and 3.3. of Chapter 2. It would have been interesting to monitor the state of the inner loops (speed reference, current reference) but these values were not published by the VESC node designed by F. Heremans and we have not managed to modify it appropriately. A major difference between our experiment and a real swing motion is the dynamic aspect. We have not considered the forward acceleration of the entire limb which makes dorsiflexion a bit more difficult. Even though coefficients might be slightly too high for a static experiment, we believe that there is a deeper problem to be resolved.

Results of the second experiment are more concerning since the inability of performing accurate current tracking theoretically prevents any type of higher level control. The current loop is the

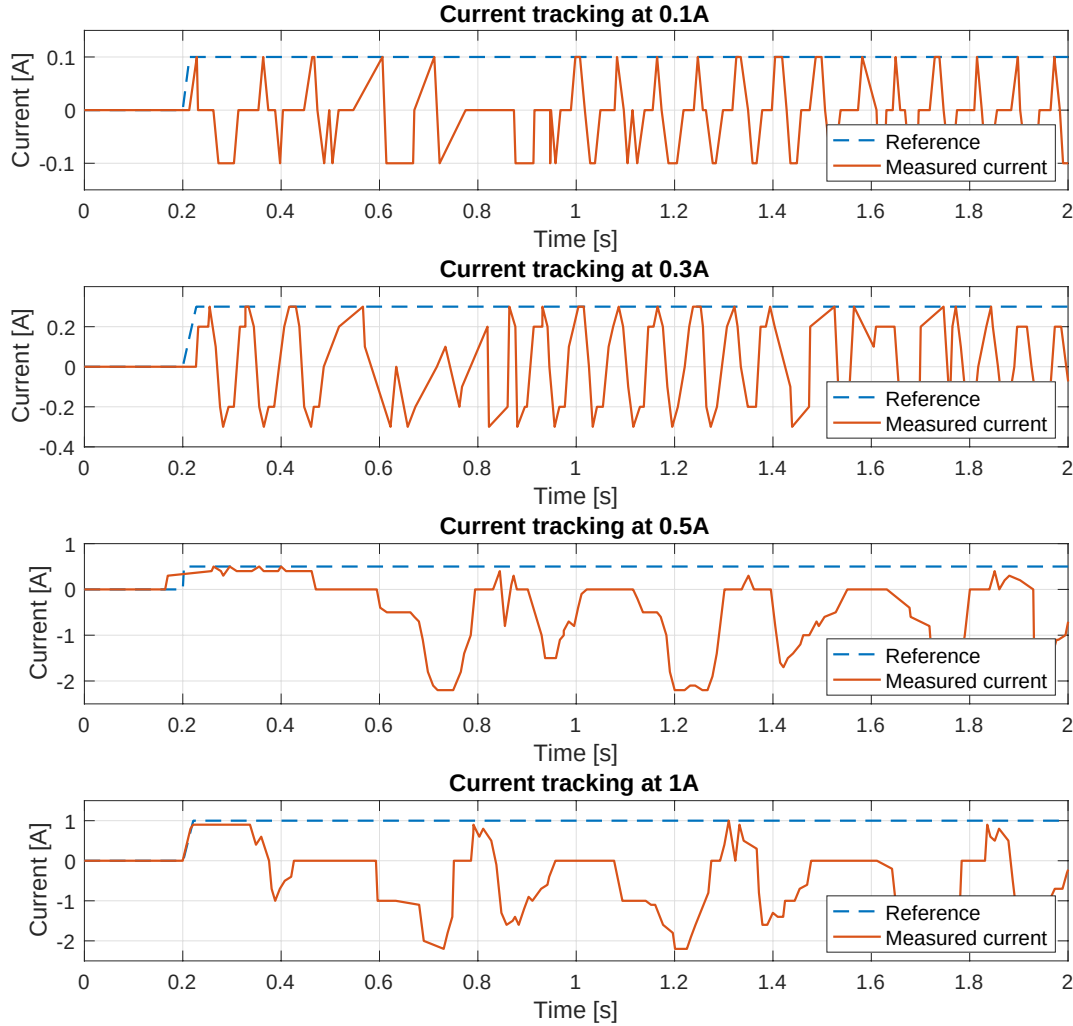


Figure 5.10: Current tracking for different step amplitudes. Blue dotted line : reference, red solid line : measured current.

inner most one. We have not managed to discover the source of the problem, meaning that further experiments and debug sessions will be required. However, as deep as we have dug, the current reference provided from ROS topics is properly received by the VESC and never overwritten by another value. These anomalies may be due to the presence of a mechanical defect within the ELSA prosthesis and suggests that the mechanical issue encountered at Ossür may have damaged an internal component of the prosthesis. This might be a deficient sensor, contact failure or actuator failure.

This thesis focused on the development of a new control approach for active ankle prostheses and especially intended for the ELSA prosthesis. Its goal was two-fold : first, to develop a compliant control strategy adapting to the user's dynamics and second, to reduce gait asymmetry throughout the walking gait cycle. Indeed, active lower-limb prostheses induce complex control schemes that usually struggle to provide adaptive assistance to the user, preventing them from recovering a complete locomotion ability. The interest of this thesis was therefore to develop a novel control approach that can potentially enhance the user's comfort and improve his quality of life.

The first part of this work consisted in identifying and clarifying the main control strategies already used in the literature for active lower-limb orthotics and prosthetics in order to gather interesting ideas and have a clear view of the challenges to overcome. These challenges are mainly linked to the need of a safe prosthesis-amputee interaction, the need of adaptation to different locomotive tasks and walking speeds, and the complexity of the human walking gait cycle itself.

Another important section was devoted to the ELSA prosthesis developed by F. Heremans, including its mechanical features and current controller. Two new experiments were carried out with an amputee at Ossür facilities and at the CUSL to evaluate the performance and the limitations of the current control strategy running on the latest prototype.

All these information allowed us to outline a novel control strategy. The idea is to perform an adaptive compliant position control which tries to minimise gait asymmetry using adaptive oscillators for gait analysis and a linear combination of adaptable motor primitives as a reference signal. Adaptive oscillators perform continuous learning of the raw ankle angular velocity signal providing a smooth approximation as well as a gait phase linearly increasing from 0 to 100 [%]. The analysis of the learned signal by means of a robust gait events detection algorithm allows to detect key events throughout the gait cycle. An asymmetry index is then generated by comparing the phase variable values at each event between both legs. Based on the sign and magnitude of the asymmetry index, simple heuristic laws are used to adapt motor primitives. Concerning actuation, a novel bio-inspired method based on the functionality of the cerebellum and which purpose is to provide a real-time inverse dynamics model estimation, was experimented. This method is designed to learn from the user's behaviour and from the environment and is based on the locally weighted projection regression (LWPR) non-linear function approximator, providing a feedforward contribution, combined with a feedback contribution for error compensation.

From there, experiments were carried out in a simulation environment as a preliminary validation stage to assess the performance of the proposed strategy in terms of real-time gait analysis, position tracking, compliance and asymmetry correction. Regarding gait analysis, the adaptive oscillators proved to be able to provide a reliable gait phase variable estimate and to efficiently learn the angular velocity signal of the shank. The smoothness and synchronisation of the learned signal with raw data lay the foundation of gait events detection algorithm. The latter demonstrated its robustness in detecting key events throughout the gait cycle. These results were observed both in the simulation environment and in real-life with the ELSA prosthesis. It suggests that such a technique could be used in a final product to evaluate asymmetry and progression in the gait cycle. Regarding position

tracking, we were able to demonstrate that the achievable performance provided by the feedback-error-learning mechanism, combining feedback and feedforward contributions, are at least as good as a good feedback-based regulator. We proved that the inverse dynamics model governing feedforward contribution can be learned from scratch in a few steps and adapt to the user in real-time. Then, the compliance analysis showed that the combination of the LWPR and the feedback regulator allows to significantly increase the compliance of the system, i.e. by a factor 2 or 3, compared with a pure feedback regulator. Finally, the asymmetry correction mechanism showed disappointing results on the simulation environment. The asymmetry tends to be slightly reduced but not cancelled, and the system does not achieve to converge to a stable state. These bad results could arise from the restrictions of the simulation environment where all uncontrolled joints are driven to predetermined trajectories. For this reason, further investigations should be performed with real subjects.

The last section focused on the application of the controller on ELSA, with associated challenges and adjustments mainly linked to the mechanical specificities of the prosthesis. Indeed, a major subject of concern comes from the LPS being engaged during the stance phase and disengaged during the swing phase which prevents to consider the actuator as a pure source of torque. A first-draft implementation of the controller based on ROS has been provided and partially validated. The entire control scheme could unfortunately not be tested on the prosthesis due to unexpected technical issues and difficult circumstances encountered in 2020 due to the COVID-19 pandemic.

Consequently, future works will aim to precisely identify and solve the technical issues. Only then can the strategy be finely adapted to the particular features of ELSA based on previous recommendations. The asymmetry correction algorithm might also be improved to ensure convergence or efficiency, eventually harnessing the concept of sensibility analysis. Finally, many experiments will have to be performed on the prosthesis to tune each parameter (LWPR, PD regulator) and assess the overall performance of this controller.

To conclude, throughout this thesis, we showed that the combination of IMUs with adaptive oscillators was a suitable tool to evaluate asymmetry between both legs and estimate the progression of each leg in the gait cycle. Furthermore we were able to demonstrate the interest of the LWPR as a powerful tool to efficiently predict the user's inverse dynamics model. This technique could be used for many active prosthetic applications.

A stride is characterised by a chain of events that starts when one limb comes into contact with the ground and ends when the same limb is back in contact with the ground. This chain of events constitutes the gait cycle. One stride is composed of two different phases : the stance phase and the swing phase. The stance phase refers to the time during which the foot is in contact with the ground and corresponds to around 60% of the normal gait cycle while the swing phase refers to the time during which the foot is off the ground and corresponds to the remaining 40% of the normal gait cycle. The stance phase has a dual function : weight acceptance and single-limb support. The main function of the swing phase is limb advancement. The terms *single support* or *double support* are used when either one foot or both feet are in contact with the ground respectively [7].

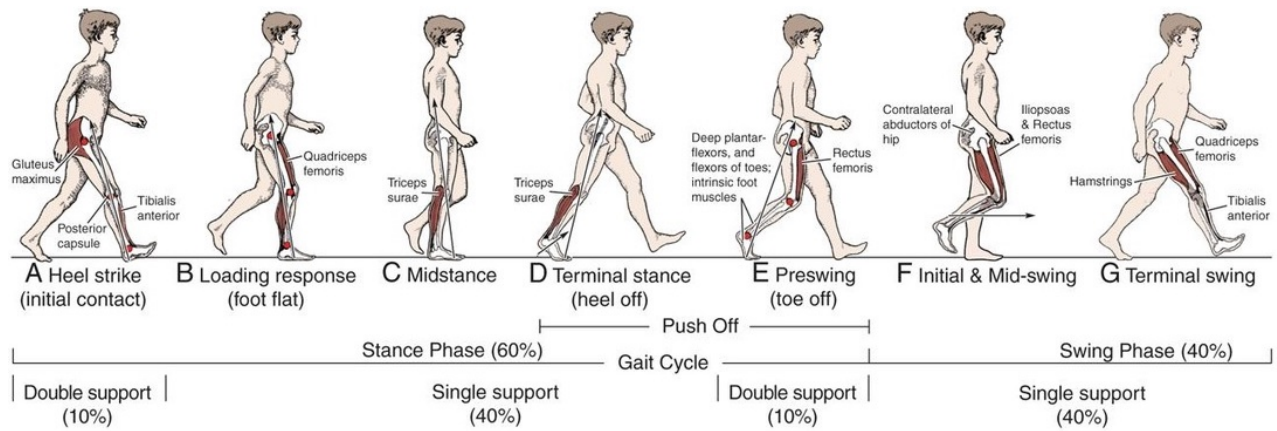


Figure A.1: Normal gait cycle [7].

A complete cycle can be divided into seven different periods (Figure A.1) [7] :

- A. **Initial contact** : Initial contact occurs at *heel-strike* and refers to weight acceptance. This event allows limb stabilisation before forward body weight shifting.
- B. **Loading response** : Shortly after *heel-strike*, the foot flattens on the ground through plantar flexion. This period ends when the opposite toe is off the ground. The loading response plays a role in shock absorption thanks to the slight knee flexion and stabilises the limb even more.
- C. **Mid stance** : The hip and the knee are extended such that the body weight is aligned with the supporting foot while the opposite foot is in the swing phase.
- D. **Terminal stance** : The terminal stance starts when the heel of the supporting foot leaves the ground, that is the *heel-off* event. This event makes the body weight move forward.
- E. **Pre-swing** : The pre-swing starts at the *toe-off* event resulting from a forward push that moves the body weight to the opposite limb.

F. **Initial swing and Mid-swing** : The foot lifts off the ground while the other foot is in mid stance. The foot then swings forward and reaches maximum speed at *mid-swing*.

G. **Terminal swing** : This period ends when the foot comes back into contact with the ground, i.e. at *heel-strike*.

The gait cycle described above is the result of a complex muscle coordination inside the body. There are two types of contraction mechanisms : concentric and eccentric mechanisms. Concentric muscle action is characterised by the muscle shortening in order to generate mechanical energy while eccentric muscle action allows to absorb energy as it lengthens [11, 52]. We can identify two main groups of muscles responsible for the **ankle movement** during walking : the *tibialis anterior* muscle and the *triceps surae* muscle composed of the *gastrocnemius* and *soleus* muscles (Figure A.1). The *tibialis anterior* controls the foot dorsiflexion. The *triceps surae* provides the required propulsive push-off for *heel-off* and *toe-off* [53].

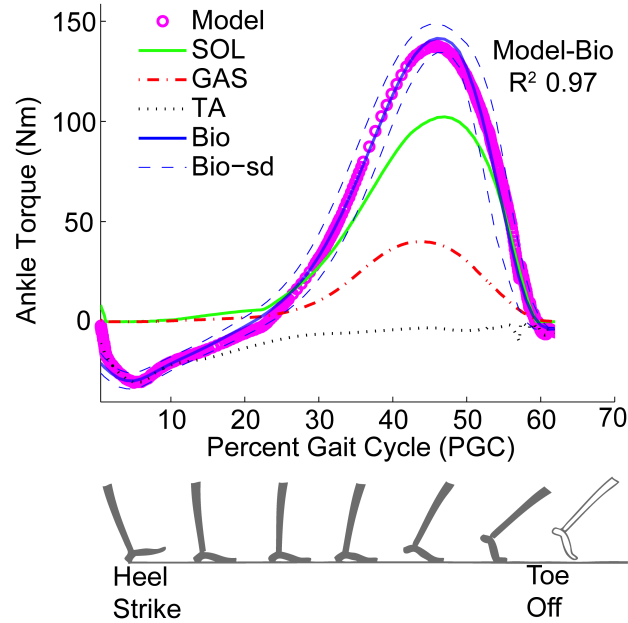


Figure A.2: Amount of ankle torque generated by different muscles involved in the walking gait cycle [54]. SOL = soleus, GAS = gastrocnemius, TA = tibialis anterior, Bio = total ankle torque, Bio-sd = standard deviation.

Figure A.2 shows the ankle torque generated by previously mentioned muscles. The gait cycle starts with an eccentric contraction of the tibialis anterior to prevent the foot from flapping during loading response and early stance, i.e. from *heel-strike* to *foot-flat*. Then the triceps surae contracts concentrically with torque generated by the soleus being about twice bigger than the gastrocnemius. Muscle activity is assumed to cease during the swing phase because of the leg inertia of motion that swings the leg forward although the *tibialis anterior* keeps controlling foot dorsiflexion to allow the forefoot to clear the ground during initial swing and to hold the ankle in position for heel-strike during terminal swing [53].

APPENDIX B _____

_____ PROTOCOL

Evaluation of the ELSA prosthesis with an amputee

Experimental protocol

Heremans Francois

June 8, 2020

1 Introduction

The ELSA prosthesis (Efficient Lockable Spring Ankle, Fig 1) aims at restoring the biological function of the ankle-foot complex for amputees. More precisely, this system reproduces with high fidelity the biomechanics of the missing limb. Indeed, by restoring the gait symmetry, it is expected that this prosthesis will prevent the onset of comorbidities and thus improve the quality of life of the patients. This document describes an experimental study, which objective is to validate the performance of the device with an amputee. In particular, the mechanical and control aspects will be evaluated. The following section introduce the experimental protocol required to perform these tests. The tested hypotheses will be laid down. Then, the experimental setup will be presented. Finally, the test protocol will be detailed.

2 Hypotheses

In the context of these tests, the evaluated hypotheses are the following:

1. The active ELSA prosthesis provides more energy during walking than the passive daily-life prosthesis.
2. The active ELSA prosthesis restores the symmetry of the gait by matching the biomechanics of the contra-lateral sound leg: the kinematic and kinetic profiles are similar.

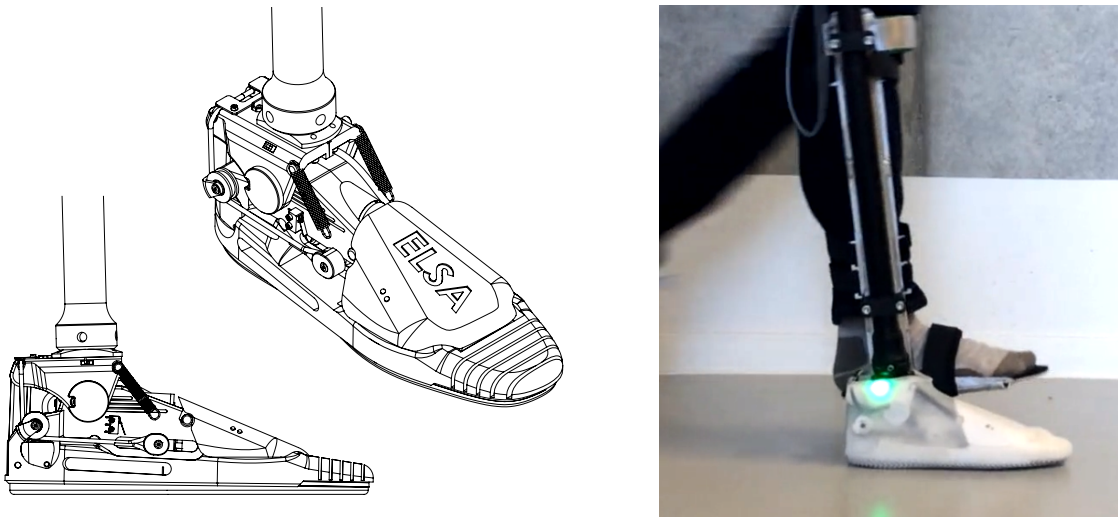


Figure 1 – The ELSA prosthesis to be tested with an amputee.

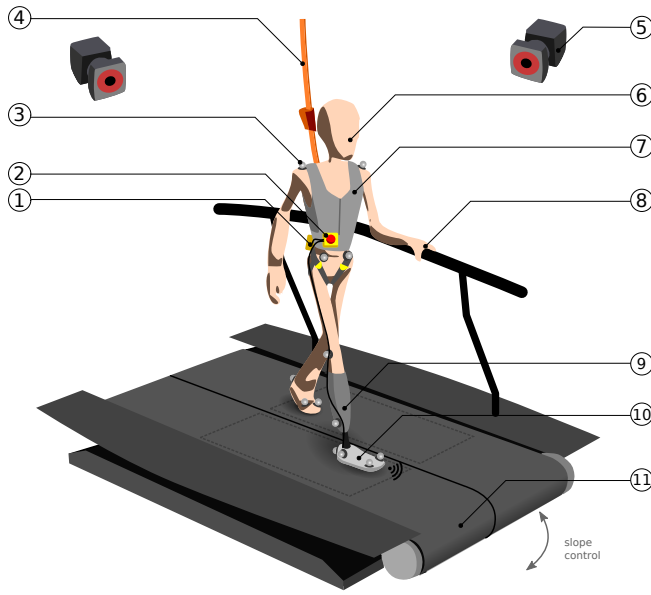


Figure 2 – Experimental setup: The amputee (6) walks with the device (10) connected to his socket (9) on an instrumented treadmill (11). Reflective markers (3) part of a motion capture environment (5) allow the motion kinematics to be acquired. The prosthesis is powered by an external battery (1) through an emergency stop button (2). For safety, the user wears a harness (7) connected to the ceiling by means of a rope (4). A lateral handrail (8) is also available.

3 Experimental setup

The trials will be carried out at the gait lab.

3.1 Equipment

The experimental area is composed of a motion capture environment (MoCap) and an instrumented treadmill fitted with force plates. This equipment measures both the body kinematics and interaction forces with the ground. The OpenSim software then reconstructs the joint torque by inverse dynamics using an anatomical model of the subject.

3.2 User

The test pilot is recruited on a voluntary basis and must be, apart from his amputation, both physically and mentally healthy. A trans-tibial amputation is favored in order to reduce the balance problems inherent to a more proximal amputation. No monetary compensation is provided.

The collected data is recorded electronically in anonymous files without any traceability of the subject.

The pilot will have access to the complete information regarding this experiment and his written consent will be requested.

3.3 Setup

The experimental setup is depicted in Fig. 2. The amputee (6) walks with the device (10) connected to his socket (9) on an instrumented treadmill (11). Reflective markers (3) part of a motion capture environment (5) allow the motion kinematics to be acquired. The prosthesis is powered by an external battery (1) through an emergency stop button (2). For safety, the user wears a harness (7) connected to the ceiling by means of a rope (4). A lateral handrail (8) is also available.

3.4 Safety

The required safety measures are set up to minimize any risk of accident. On one hand, in case of instability, the pilot can at all times grab the lateral handrail with his hand. On the other hand,

the pilot is equipped with a safety harness connected to the ceiling by a rope. This harness stops any fall and limits the risk of injury. Moreover, the experimenter has access to an emergency stop button that, upon activation, immediately brings the treadmill to a stop.

The device itself is controlled by the experimenter via a monitoring dashboard. On top of that, the pilot wears an emergency stop button that can halt the power to the device.

4 Experimental protocol

The complete protocol is depicted on Fig. 3. The first part consists in setting up the equipment and attaching markers to the subject. Then, the pilot will go through the three trials experiment with his daily-life prosthesis. His natural walking speed with that prosthesis will be identified.

During this first recording, the user is requested to walk at different speeds on the treadmill (natural speed $\pm 15\%$). The trials order is randomized. The slope of the treadmill is maintained level during the whole experiment. Each trial is divided in two sequences. The first sequence lasts 30 seconds and is not recorded. It allows the gait to reach a steady state. Then, a 30 seconds recorded sequence takes place.

Following this first recording, the pilot will be fitted with the robotic prosthesis and will go through a learning phase in order to familiarize with the device. The treadmill speed will be progressively increased until it reaches the natural walking speed identified with the passive device.

Then, the pilot repeats the same three trials with the robotic prosthesis. The internal data from the device is also recorded and synchronized offline with the motion capture data. If the equipment allows a direct processing of the recorded data, the device can be adjusted and the experiment can be repeated until a satisfactory outcome is obtained.

5 Conclusion

The outcome of this study will be used to improve the performance of the device. Moreover, the results could be the object of a scientific publication.

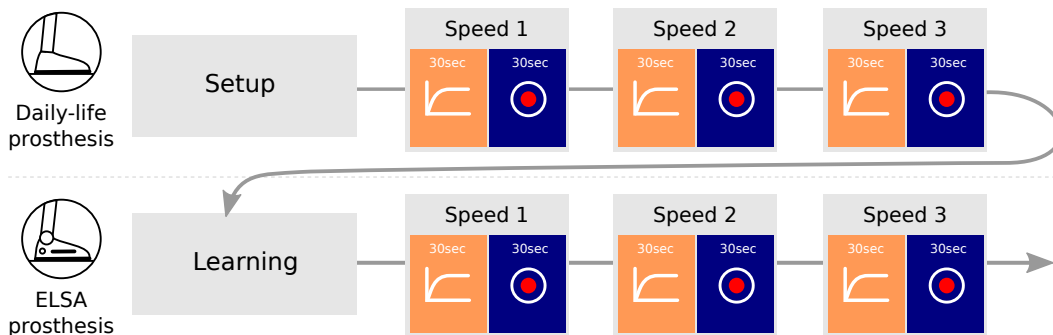


Figure 3 – Experimental protocol to test the ELSA prosthesis with an amputee.

For this experiment, a healthy subject equipped with one IMU on the lateral side of each tibia walked on a treadmill at a constant speed of 4 [km/h] for 2 minutes. Figure C.1 shows the raw and learned signals of the shank angular velocity and the estimated gait phase for both the left and right legs. One can observe that the adaptive oscillators are able to learn the overall shape of the raw signal. In addition, the learned signal is well synchronised with the raw signal. Regarding the gait phase variable, it varies from 0 to 100 [%] between two successive heel-strike events.

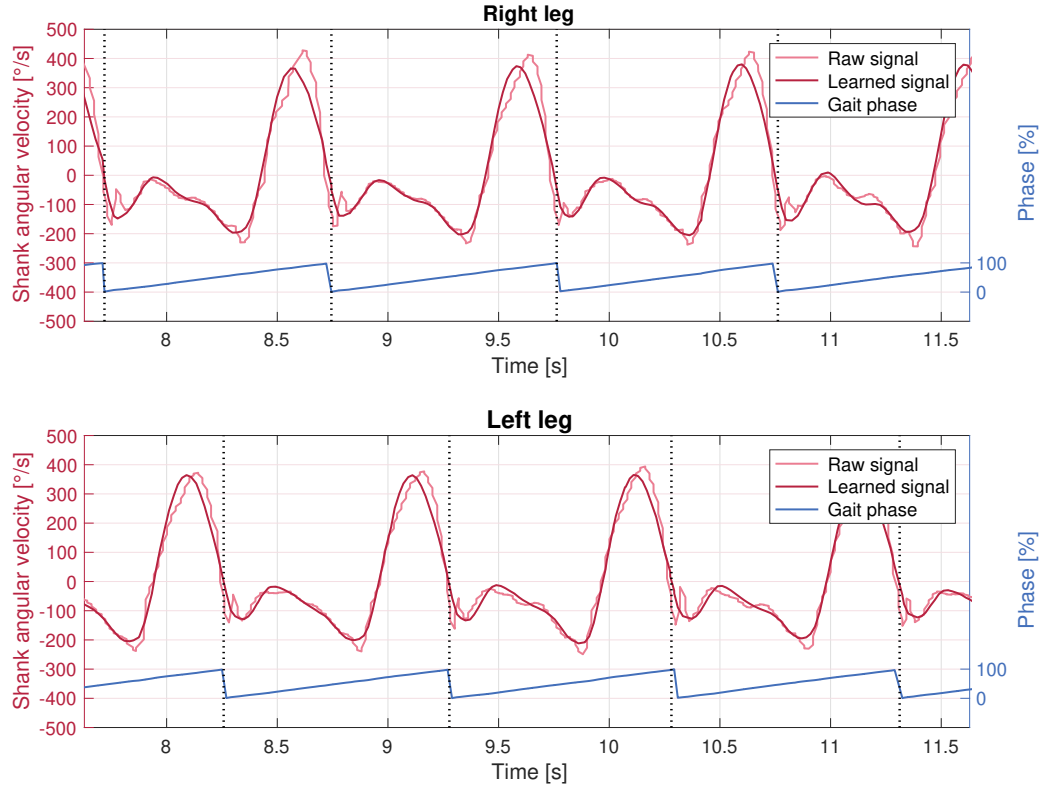


Figure C.1: Shank angular velocity [°/s] (raw and learned signals) and phase variable [%] as a function of time [s] for the right leg (a), and the left leg (b). The vertical dotted lines show the time at which the heel-strike events are detected by the algorithm on the raw signal.

The results obtained for the gait events detection are provided in Figure C.2. One can observe that the detection algorithm is able to systematically detect the five events.

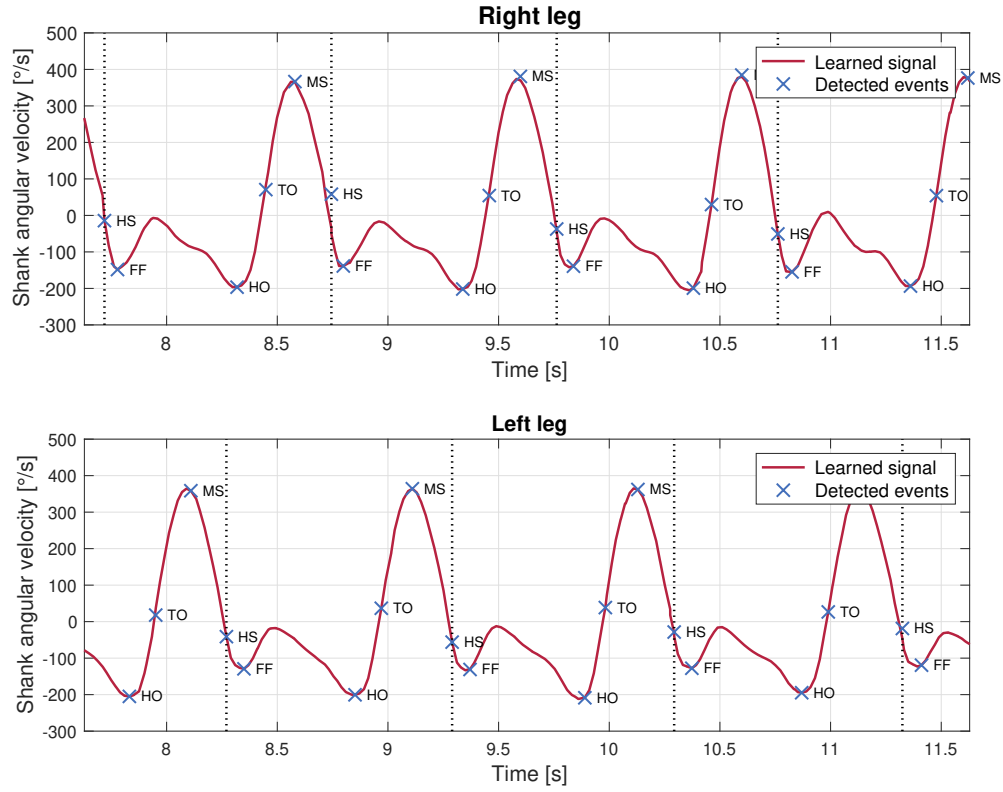


Figure C.2: Learned signal of the shank angular velocity [°/s] over time for both the right (a), and left (b) legs. The vertical dotted lines show the time at which the heel-strike events are detected by the algorithm on the raw signal. The crosses correspond to an event detection : heel-strike (HS), foot-flat (FF), heel-off (HO), toe-off (TO) and mid-swing (MS).

A .zip file is provided and contains the Matlab and C++ scripts described below.

1. Matlab

The folder called `simpleBiped` contains the entire `Robotran` project used for the preliminary validation described in Chapter 4. The hierarchical structure is described below.

- `../animationR` : Holds the animation file `.anim` for further visualisation in `Robotran`.
- `../dataR` : Holds the data file `.mbs` which is a complete description of the dynamical model in `Robotran`.
- `../resultsR` : Holds the standard numerical results `.res` of the last simulation (position, velocity, acceleration, ..., of each joint)
- `../symbolicR` : Holds the symbolic equations automatically generated by `Robotran` and based on the dynamical model to perform the simulation.
- `../userfctR` : Holds all the functions taking part in the control strategy and required by `Robotran`. For the sake of clarity, these functions are not described here, but every function is thoroughly commented.
- `../workR` : All scripts or functions that can be manually executed.
 - `main.m` : Main script used to configure and start a walking simulation. Many parameters and options are proposed to set the number of steps, the gait period, the LWPR and the feedback regulator parameters, the weights of the motor primitives, etc, to enable or disable the adaptation mechanism, to perform simple sensitivity analysis of μ_i on AI_i , and finally to save and monitor the results.
 - `impedanceAnalysis.m` : This script performs an impedance analysis over a range of frequencies. A Bode diagram is generated for the stance and swing phases separately and an impedance model is fitted.
 - `singleRun.m` : Simulates a single step with parametric perturbation and computes the average impedance magnitude. Intended to be called from `impedanceAnalysis.m`.
 - `tuneLWPR.m` : This script initialises a LWPR model and updates it offline based on previously saved simulation data.
 - `impModel.m` : Evaluates the magnitude of the impedance model $Z(s) = Ms + B + K/s$.
 - `impedanceFit.m` : This script performs an offline impedance model fitting based on the impedance magnitudes measured in `impedanceAnalysis.m` for a range of frequencies.
 - `eventsSensitivity.m` : This script loads the data from three single μ sensitivity analysis and computes the 3×3 sensitivity matrix which provides the impact of each μ on the position of each event.

- `asymPlots.m` : This script generates a figure to monitor the asymmetry indexes and position of motor primitives while performing simple or sensitivity-based asymmetry correction.
- `../Data` : Holds all data required to perform the simulation, i.e. reference trajectories from the literature, initial conditions of the adaptive oscillators, of the events detection algorithm, etc.

Another folder called `miscellaneous` is provided and holds two files of lesser significance.

- `amplExtr.m` : This script illustrates two methods for oscillation amplitude extraction. The first method uses the Hilbert transform to extract the envelope of the signal. The second method uses frequency analysis to extract the component at the desired frequency.
- `effectSampling.m` : This script illustrates the effect of sampling frequency on the shape of a sine wave. It allows to observe that at least four sample points are required to maintain an amplitude close to one. This conclusion was helpful to perform the impedance analysis in Section 5.1. of Chapter 4.

All these Matlab files are stored on a Git repository accessible at the following address : <https://git.immc.ucl.ac.be/decoeyer/simpleBiped>.

2. C++

A folder called `ROS` contains the C++ implementation of the four ROS nodes `NGIMU`, `Adaptive oscillators`, `Motor primitives`, and `Control` described in Section 2. of Chapter 5. Also, a python script called `bag_to_csv.py` is provided and allows to convert topics information saved in a `.bag` file to a `.csv` file for further analysis in Matlab.

Finally, a folder called `ngimuAcquisition` contains a C++ code allowing to acquire any type of data possibly streamed by x-io NGIMUs across the WiFi network. This code could easily be used in further projects involving these IMUs. Until now, it is only compatible with `macOS` and `Linux`, but a `Windows` version will soon be available. A `README` file describes the simple procedure required to start acquiring data.

- [1] emedicine. Gait analysis after amputation. <https://emedicine.medscape.com/article/1237638-overview#a1>. Accessed: 20.12.19.
- [2] AccessProsthetics. 15 limb loss statistics that may surprise you. <https://accessprosthetics.com/15-limb-loss-statistics-may-surprise/>. Accessed: 01.02.20.
- [3] M. Windrich et al. Active lower limb prosthetics: a systematic review of design issues and solutions. *BioMedical Engineering OnLine*, 15:140, 2016.
- [4] M. McLaughlin. Lower limb amputation and gait. *Braddom's Rehabilitation Care: A Clinical Handbook*, 10:57–65, 2018.
- [5] G. Rümenapf and S. Morbach. Amputation statistics — how to interpret them? *Deutsches Ärzteblatt International*, 114:128–129, 2017.
- [6] MossRehab. Lower extremity amputation types & levels. <https://www.mossrehab.com/amputation-and-prosthetics/lower-extremity-amputation>. Accessed: 20.12.19.
- [7] Physiopedia. Gait in prosthetic rehabilitation. https://www.physio-pedia.com/Gait_in_prosthetic_rehabilitation. Accessed: 03.12.19.
- [8] Protokinetics. Measuring spatiotemporal parameters: Spatial parameters, speed and walk ratio. <https://www.protokinetics.com/2018/12/21/measuring-spatiotemporal-parameters-spatial-parameters-speed-and-walk-ratio/>. Accessed: 19.11.19.
- [9] Protokinetics. Measuring spatiotemporal parameters: Time & stance phase. <https://www.protokinetics.com/2018/12/21/measuring-spatiotemporal-parameters-time-stance-phase/>. Accessed: 19.11.19.
- [10] Protokinetics. Using gait asymmetry to define gait quality. <https://www.protokinetics.com/2019/04/08/using-gait-asymmetry-to-define-gait-quality/>. Accessed: 19.11.19.
- [11] E. Prinsen et al. Adaptation strategies of the lower extremities of patients with a transtibial or transfemoral amputation during level walking: A systematic review. *Arch Phys Med Rehabil*, 92:1311–1325, 2011.
- [12] L. Nolan et al. Adjustments in gait symmetry with walking speed in trans-femoral and trans-tibial amputees. *Gait and Posture*, 17:142–151, 2003.
- [13] P. Adamczyk et al. Mechanisms of gait asymmetry due to push-off deficiency in unilateral amputees. *IEEE Trans Neural Syst Rehabil Eng.*, 23(5):776–785, 2015.
- [14] A. Rutkowska-Kucharska et al. Relationship between asymmetry of gait and muscle torque in patients after unilateral transfemoral amputation. *Hindawi Applied Bionics and Biomechanics*, 2018, 2017.

-
- [15] H. Herr et al. Bionic ankle-foot prosthesis normalizes walking gait for persons with leg amputation. *The Royal Society*, 279(1728):457–464, 2011.
 - [16] H. Bateni et al. Kinematic and kinetic variations of below-knee amputee gait. *Prosthetic and orthotic science*, 14:1, 2002.
 - [17] K. Patterson et al. Evaluation of gait symmetry after stroke: A comparison of current methods and recommendations for standardization. *Gait & Posture*, 31:241–246, 2010.
 - [18] R. Gailey et al. Review of secondary physical conditions associated with lower-limb amputation and long-term prosthesis use. *Journal of Rehabilitation Research & Development*, 45:15–30, 2008.
 - [19] M. Grimmer. Powered lower limb prostheses. *Technische Universität Darmstadt*, February 2015.
 - [20] M. Edwards. Lower limb prosthetics. *Mosby Clinician’s Guide to Assistive Technology*, 18:297–310, 2002.
 - [21] R. Jimenez-Fabian et al. Review of control algorithms for robotic ankle systems in lower-limb orthoses, prostheses, and exoskeletons. *Medical Engineering & Physics*, 34:397–408, 2012.
 - [22] T. Lenzi et al. Speed adaptation mechanism - robotic prostheses can actively regulate joint torque. *IEEE Robotics & Automation magazine*, 21:4:94–107, 2014.
 - [23] U. Tahir et al. Case study : A bio-inspired control algorithm for a robotic foot-ankle prosthesis provides adaptive control of level walking and stair ascent. *Frontiers in Robotics and AI*, 5:36, 2018.
 - [24] M. Tucker et al. Control strategies for active lower extremity prosthetics and orthotics: a review. *Journal of Neuroengineering and Rehabilitation*, 12:1, 2015.
 - [25] A. Hansen et al. Effective rocker shapes used by able-bodied persons for walking and fore-aft swaying: Implications for design of ankle-foot prostheses. *Gait & Posture*, 32:181–184, 2010.
 - [26] P. Song et al. Impedance control of robots: An overview. *IEEE, 2nd International Conference on Cybernetics, Robotics and Control*, pages 51–55, 2017.
 - [27] F. Heremans et al. The low-cost and lightweight elsa prosthesis faithfully reproduces the walking dynamics of a sound ankle. *Not published*, 2019.
 - [28] F. Heremans et al. Design of an energy efficient trans-femoral prosthesis using lockable parallel springs and electrical energy transfer. *International Conference on Rehabilitation Robotics (ICORR)*, pages 1305–1312, 2017.
 - [29] F. Heremans et al. Bio-inspired design and validation of the efficient lockable spring ankle (elsa) prosthesis. *IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*, 2019.
 - [30] D. Winter. Biomechanics and motor control of human movement. *John Wiley & Sons*, October 2009.
 - [31] A. Calanca et al. A review of algorithms for compliant control of stiff and fixed-compliance robots. *IEEE*, 21:613–624, 2015.

-
- [32] P. Catalfamo et al. Gait event detection on level ground and incline walking using a rate gyroscope. *Sensors (Basel)*, 10:6:5683–5702, 2010.
 - [33] A. Laudanski et al. A concurrent comparison of inertia sensor-based walking speed estimation methods. *33rd Annual International Conference of the IEEE EMBS*, September 2011.
 - [34] H. F. Maqbool et al. Real-time gait event detection for lower limb amputees using a single wearable sensor. *38th Annual International Conference of the IEEE EMBC*, 2016.
 - [35] H. Silva et al. Walk distance estimation using an ankle-mounted inertial measurement unit. *World Congress on Engineering 2017*, 1, 2017.
 - [36] W. Sheng et al. The effectiveness of gait event detection based on absolute shank angular velocity in turning. *2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 899–904, 2019.
 - [37] T. Yan et al. An oscillator-based smooth real-time estimate of gait phase for wearable robotics. *Autonomous Robots*, 41:759–774, 2016.
 - [38] R. Ronsse et al. Real-time estimate of velocity and acceleration of quasi-periodic signals using adaptive oscillators. *IEEE TRANSACTIONS ON ROBOTICS*, 29:NO. 3, 2013.
 - [39] Q. Li et al. Walking speed estimation using a shank-mounted inertial measurement unit. *Journal of Biomechanics*, January 2010.
 - [40] R. Garate et al. Experimental validation of motor primitive-based control for leg exoskeletons during continuous multi-locomotion tasks. *Frontiers in Neurobotics*, 11:15, 2017.
 - [41] S. Aoi et al. Neuromusculoskeletal model that walks and runs across a speed range with a few motor control parameter changes based on the muscle synergy hypothesis. *Scientific Reports*, 9:369, 2019.
 - [42] M. Capolei et al. A biomimetic control method increases the adaptability of a humanoid robot acting in a dynamic environment. *Frontiers in Neurorobotics*, 13, 2019.
 - [43] E. Massi et al. Combining evolutionary and adaptive control strategies for quadruped robotic locomotion. *Frontiers in Neurorobotics*, 13:71, 2019.
 - [44] S. Tolu et al. Bio-inspired adaptive feedback error learning architecture for motor control. *Biological Cybernetics*, 106:507–522, August 2012.
 - [45] S. Tolu et al. A cerebellum-inspired learning approach for adaptive and anticipatory control. *International Journal of Neural Systems*, 30:1, 2020.
 - [46] S. Tolu et al. Adaptive and predictive control of a simulated robot arm. *International Journal of Neural Systems*, 23:3, 2013.
 - [47] S. Vijayakumar. Incremental online learning in high dimensions. *Neural Computation*, 17:2602–2634, 2005.
 - [48] J. Porrill et al. Recurrent cerebellar architecture solves the motor-error problem. *The Royal Society*, 271:789–796, 2004.

-
- [49] Van der Noot et al. Bio-inspired controller achieving forward speed modulation with a 3d bipedal walker. *The International Journal of Robotics Research*, 37(1):168–196, 2018.
- [50] Chwała et al. Three-dimensional analysis of the pelvic and hip mobility during gait on a treadmill and on the ground. *Acta of bioengineering and biomechanics / Wrocław University of Technology*, 14(2):83–89, 2012.
- [51] H. Vallery et al. Compliant actuation of rehabilitation robots. *IEEE Robotics & Automation Magazine*, 15 : 3:60–69, 2008.
- [52] G. Kerckhofs. Lgbio2040 biomechanics. *UCLouvain*, 2018-2019.
- [53] Physiotherapist Dr. Sunit. Gait cycle muscle activity. <https://www.youtube.com/watch?v=jJvxa4xsX58>. Accessed: 04.12.19.
- [54] P. Krishnaswamy et al. Human leg model predicts ankle muscle-tendon morphology, state, roles and energetics in walking. *PloS Computational Biology*, 7, 2011.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/epl