Towards model predictive control of a prosthetic leg actuated by a momentum exchange device

Master of Science Thesis

For the degree of Master of Science in Systems and Control & BioMechanical Design at Delft University of Technology

J. Kreuk

August 2, 2021

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

CONTENTS

Towards model predictive control of a prosthetic leg actuated by a momentum exchange device

Jesper Kreuk

***Abstract*—Some above the knee amputees take a smaller step with their prosthetic leg. A momentum exchange device (MED) can increase the step length by exchanging angular momentum between the device and the leg. However, there are no controllers for MEDs during regular gait. The goal of this study is to build a model predictive controller (MPC) that controls an MED to help the amputees achieve a certain step length. The MPC uses a model of an unimpaired walking human to predict its movement. Two different models are evaluated, a compass-gait biped (CGB) model and a neuromusculoskeletal (NMS) model. The parameters of the CGB model are estimated with a grey-box estimation method. An NMS model of an unimpaired human is used to simulate the controllers as well. For control, both linear and nonlinear hybrid model predictive control methods are used. By evaluating the behaviour of the controllers, insights are gained in whether the chosen models and control methods are suitable to achieve the goal. It is concluded that the prediction of the CGB model is insufficient for the control algorithm used. The model is capable of approximating the swing of the leg, but it is unable to accurately predict the future states. The NMS model may be more accurate, but is at this moment in time not practical for control. The model is computationally expensive, observing all its states is difficult and the model can currently not be initiated in any desired state, which is required for control. The results indicate that the chosen combinations of models and control methods are not well suitable to achieve the goal. Control with an unimpaired human model unfolded many difficulties and control of an impaired human is even more difficult. A control method that does not rely on an accurate step length prediction might be more suitable. However, the identified CGB model and the control methods contribute to further research on the subject or other applications within the field.**

I. Introduction

Transfemoral amputees, patients who are amputated above the knee, can have an asymmetric gait due to their prosthetic leg. For example, the step length can be either larger [1], [2] or shorter [3] on the prosthetic side. A prosthetic leg actuated by a momentum exchange device (MED), like a control moment gyroscope (CMG) [4], [5] or a reaction wheel (RW) [6], can help patients take a larger step. These devices can exchange angular momentum such that a torque is experienced on one ligament instead of two. Conventional actuators provide a torque on two ligaments due to the reaction torque [7]. MEDs are of interest because torques on one ligament show superior performance in increasing the step length [7]. Appendix A gives more details and a visual representation of a CMG. The use of MEDs for walking applications is an active research area in both human (e.g. [5], [8]) and robotic (e.g. [9], [10]) applications. Recently controllers were built that assist human leg movement [11], [12]. However, both studies focus on fall prevention. Unfortunately, many people abandon their prosthesis due to a lack of need [13], [14]. To make the prosthesis more desirable in practice, the MED on the prosthesis should have a function while walking as well.

A model predictive controller (MPC) is used to control the MED. An MPC uses a model, called the internal model, to predict the future. First, a cost function is designed. The cost function penalises states and control actions over a time horizon. At each time step, the MPC searches for the control sequence that minimises the cost function. The resulting control sequence contains the optimal control input for each time step of the horizon. Only the first control input is executed, the control input corresponding to the current time step. The next time step the process is repeated. MPC is an attractive control technique for the intended application because it can deal with actuator constraints and is predictive [15]. Both CMGs and RWs have actuator states where they can provide no torque in the desired direction. Actuator constraints can help avoid these states. Additionally, the predictive capabilities of the MPC can predict the future trajectory of the human and adjust accordingly. MPC is not a popular controller for prosthetic devices, but it is widely used across industries in general, including biped robotics [16]–[19]. Hybrid and nonlinear aspects of MPCs are of special interest. A hybrid system is a system with both continuous and discrete dynamical components [19]. A walking human is a hybrid system because the continuous nonlinear dynamics change at discrete time events, when a foot hits or leaves the ground. Nonlinear model predictive control (NMPC) can deal with hybrid nonlinear systems, but this can result in non-convex optimisation problems that are hard to solve [20]. The algorithm may not find the optimal solution. Mixed logical dynamical (MLD) systems do have algorithms that find the optimal solution, but use linearised dynamics. However, they can switch between multiple linearisations that can approximate the nonlinear dynamics [19], [21]. Switching logic is implemented with the use of binary auxiliary variables.

This paper attempts to design an MPC to reach a certain step length in the swing phase. The controller controls the torque output of the MED on the right leg of a simulated unimpaired walking human. The paper is split into two parts, a modelling and a control part. In the modelling part, two different internal models are described and prepared for MPC, a compass-gait biped (CGB) model and a neuromusculoskeletal (NMS) model. In the control part, controllers are designed for both models. For the CGB model, an MLD-MPC and an NMPC are designed. The NMS model is controlled by an NMPC as well. Simulations are done to evaluate whether the controllers show desirable behaviour. Insights from this study are useful to eventually design a controller for a prosthetic leg actuated by an MED. In the process, MLD-MPC is introduced to the field of prosthetics. All the developed code and data files are available on Github [22].

II. Methods 1: Modelling

A. Objectives and assumptions

The objective of the modelling part is to describe and prepare the models for MPC. A compass-gait biped (CGB) model and a 2D neuromusculoskeletal (NMS) model are used for prediction. For simulation a 3D NMS model is used. The CGB model is chosen because of its simplicity [23]. A simple model reduces the computational power required by the controller such that it may be implementable in real-time. If the computation is too slow, the prosthetic device may not perform as desired and can consequently compromise the patient's safety. The NMS models are chosen because they can replicate human movement quite well and are implemented in the practical MATLAB-Simulink environment [24], [25]. This model is likely too slow for a real-time optimisation of the control input, but there are workarounds to circumvent this (see subsection VI-C).

The reader may not be familiar with some of the terminology associated with human walking used in this paper. The term "heel strike" is used for the event where the swinging foot hits the ground and "toe off" for the event where the foot lifts off the ground. When both feet are on the ground, the human is in the double stance phase. The sagittal plane is the plane that splits the body into a left and a right part. When walking, the plane is spanned by the walking direction and the vertical direction. This plane is where most of the leg movement occurs. Therefore, all positions are projected on and evaluated in this plane. The step length is measured from one point of the foot at initial contact to the same point at the other foot at the next moment of contact with the ground [26]. The measure is visualised in Figure 1.

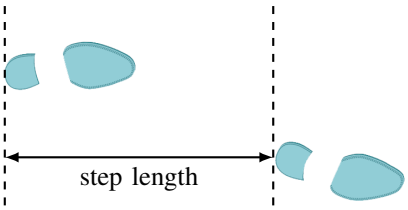

Fig. 1: The definition of the step length.

In this paper, only the swing phase of the actuated leg is controlled. However, control is required in the other phases as well. For example, a CMG has singularities where it cannot provide torque in the desired direction (see Appendix A). The controller in the other phases should ensure that the CMG at the beginning of the swing is far from the singularity. This will result in undesirable torques on the leg during the non-swing phases as well, but these are not considered.

It is assumed that the positions of all body positions are exactly known. Advanced optic setups (e.g. [27]) can measure these positions in a lab environment. However, the prosthesis should work in any environment. Encoders or potentiometers can measure some angles directly [28], and inertial measurement units (IMUs) can measure the acceleration of the limbs [29]. It is assumed that some combination of these sensors can compute the exact position. The controller uses these positions for state calculation.

The controller will eventually be used to help amputees above the knee, but this research simulates the effects of an MED on an unimpaired human. The purpose is to gain insights into challenges in controlling an MED with a human in the loop. The patient's adaptation to the MED is not modelled because this is not exactly known how the patient adapts and attempting to model the adaptation will add a lot of complexity. However, torques from the MED cause the patient to reach different states, and the patient's torques are state-dependent, so the model is responsive to torques of the MED. The added mass from the MED is ignored because the MED is eventually placed on a prosthetic leg. The total mass of the ligaments are thus adjustable.

B. Basics of the CGB model

A compass-gait biped (CGB) model is selected as a candidate internal model for the MPC because it is simple and can describe a human-like gait [23]. The CGB model has two legs of mass m and a hip mass m_H. The centre of mass (CoM) of the leg is at a distance a from the foot. The total length of the leg

$$L = a + b, \tag{1}$$

where b is the distance from the CoM of the leg to the hip. The leg angles are θ_1 and θ_2 for left and the right leg, in this case the stance and swing leg, respectively. The hip and the ankle are modelled as hinges. A visual representation of the CGB model is shown in Figure 2.

The model does not have a double stance phase but an instantaneous transition between swing phases. This transition happens at heel strike when $\theta_1 = -\theta_2$. Because both legs of the model are the same length, the foot of the swinging leg scuffs the ground mid-swing. Under the assumption that the human has enough ground clearance with its swinging foot, any forces due to the scuffing of the feet are ignored.

To predict of the future states of the model, the joint torques are estimated. The ankle torque u_{ank} applies to the stance leg and the hip torque u_{hip} applies to both the legs. The torques are estimated with the concept of virtual gravity [23], [30], based on passive dynamic walking. A CGB can walk down a slope in a stable human-like gait without any actuation besides gravity; this is called passive dynamic walking [31]. With virtual gravity, the model walks on a flat surface, but the gravity vector is angled by angle ϕ to actuate the system. In this paper, multiple virtual gravity angles are used for the different segments to improve the fit [23]. The virtual gravity forces on the CGB are replaced by the joint torques u_{ank} and u_{hip}. Both torques are a function of the state

$$\vec{x} = \begin{bmatrix} \vec{\theta} \\ \dot{\vec{\theta}} \end{bmatrix}, \tag{2}$$

where

$$\vec{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}. \tag{3}$$

The torque induced by the MED is modelled as a torque on the swinging leg u. The equations of motions are detailed in appendix B.

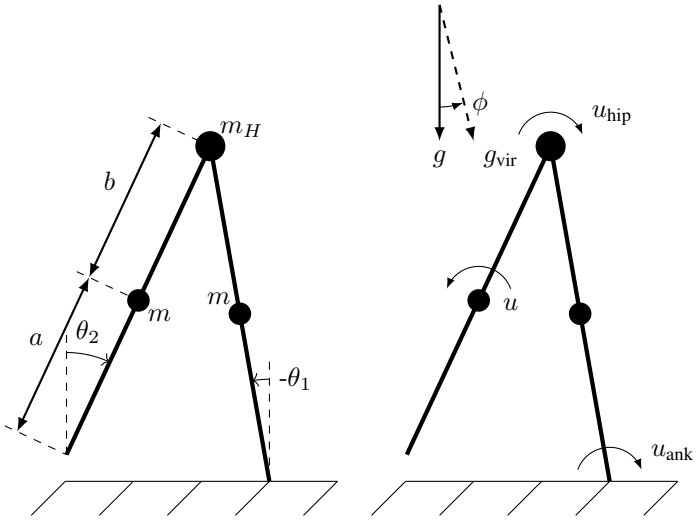

Fig. 2: The compass-gait biped model. The left side shows physical parameters. The right side the torques on the legs and the gravity and virtual gravity vector.

C. Position tracking of the CGB model

There are multiple ways to fit the low order CGB model on the human. Because of this, it is not clear what coordinates and formulas are to be used to calculate the state of the CGB model. The following reasoning is used: the stance leg should be on the ground, at the position of the stance foot. The CGB model then has two remaining degrees of freedom, the two angles $\vec{\theta}$. The horizontal coordinates in the walking direction p_x of the hip ($p_{x,\text{hip}}$) and the swinging foot ($p_{x,\text{foot}}$) are used to define these angles. The subscript L and R describe the left and right leg, respectively. These variables are visualised in Figure 3. The horizontal coordinates are chosen over the vertical coordinates as the objective is to achieve a certain step length in the horizontal direction.. The hip coordinate of the CGB model is offset by a constant c_{hip} such that heel strike is synchronised in time with the NMS model considering a standard gait of $1.2\,\text{m/s}$. In mathematical terms, the angle of the stance leg and swing leg are

$$\theta_1 = \arcsin\left(\frac{p_{x,\text{hip}} - p_{x,\text{foot,L}}}{L}\right) \tag{4}$$

and

$$\theta_2 = \arcsin\left(\frac{p_{x,\text{hip}} - p_{x,\text{foot,R}}}{L}\right), \tag{5}$$

respectively. Where

$$p_{x,\text{hip}} = \frac{p_{x,\text{hip,L}} + p_{x,\text{hip,R}}}{2} - c_{\text{hip}}. \tag{6}$$

Here $p_{x,\text{hip}}$ is the hip position of the CGB model, and $p_{x,\text{hip,L}}$ and $p_{x,\text{hip,R}}$ are the positions of the left and right hip of the NMS model, respectively.

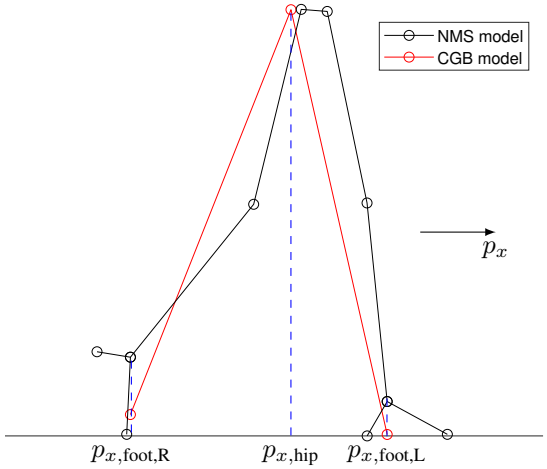

Fig. 3: An example of how the CGB model can fit the more complex NMS model.

The best position on the foot to calculate the p_x-coordinates is yet unclear. Some logical positions are the toe, ankle and heel because at these positions there is either a point of contact or a joint. The trajectories of angles $\vec{\theta}$ are obtained from the NMS model for all three positions and compared in the results. The position that results in the smoothest trajectories is chosen as a spike can cause the prediction to fluctuate due to the sudden change in the initial condition.

D. Grey-box estimation of the CGB model

The CGB model aims to predict the human trajectory as accurate as possible. Therefore, a parameter estimation technique is used to optimise the parameters of the CGB model. Instead of actual human data, data generated by the 3D NMS model (described in subsection II-E) is used. The CGB model is thus a reduced NMS model, but rather than using more typical model reduction techniques [32] the problem is treated as a grey-box estimation problem. An advantage of the grey-box estimation technique is that it can be used for actual human data as well, such that personal parameters are obtained. A benefit of the grey-box estimation method is that the parameters are physically meaningful and can therefore be valid over a wide range of states [33]. In this case, the model incorporates the knowledge that the CGB model can approximate the full swing.

The parameters are optimised to fit the trajectories of leg angles $\vec{\theta}$. The CGB model is optimised for the parameters m, m_H, a, ϕ_1 and ϕ_2. Where ϕ_1 and ϕ_2 are the angles between the gravity and the virtual gravity vector. The parameters m, m_H and a have an initial guess based on physiological data. They are allowed a maximum deviation of $30\,\%$ such that the model stays close to the actual physical values. The initial guesses and bounds are in Table I. The total leg length is fixed to $L = 1\,\text{m}$. This length is chosen because this approximates the length from the foot to the hip in stance (see Figure 3).

The optimisation is performed by the nonlinear grey-box estimation function 'nlgreyest' in MATLAB 2019a [34]. The

Parameter	m in kg	m_H in kg	a in m	ϕ_1 in rad	ϕ_2 in rad
Initial guess	13.25	53.50	0.55	0.17	0.17
Lower bound	9.28	37.45	0.38	0.052	0.052
Upper bound	17.23	69.55	0.71	0.44	0.44

TABLE I: The initial guess and bounds of parameters for the grey-box parameter estimation of the CGB model.

normalised root mean square error (NRMSE) between the data and the model is minimised, this is the default of the function.

$$\text{NRMSE}(\vec{y}, \vec{y}_{\text{ref}}) = 1 - \frac{||\vec{y}_{\text{ref}} - \vec{y}||}{||\vec{y}_{\text{ref}} - \text{mean}(\vec{y}_{\text{ref}})||}, \tag{7}$$

where $||$ indicates the 2-norm, $\vec{y}$ is the signal that is compared with a reference signal $\vec{y}_{\text{ref}}$. The NRMSE values range from $-\infty$ to 1, where 1 is a perfect fit and 0 as good as a line on the mean of $\vec{y}_{\text{ref}}$ [35]. This results in a continuous-time model. For MPC, the model is discretised with a time step $dt = 0.01\,\text{s}$, details are in Appendix B. A validation set is used to analyse how well the CGB model describes the data. This validation data set is from a later swing of the same simulation used for the identification data set. For analysis the NRMSE is used as well. The CGB model is simulated from multiple initial conditions to observe how this affects the NRMSE between the data and the predictions of the CGB model. These initial conditions are data points of the validation set.

The MLD-MPC method (see subsection IV-B) uses linear dynamic equations. Two linearisation points are compared for the CGB model. One linearisation point is the current state, similar to what is done for the MPC for fall prevention [12]. The other linearisation point is the middle of the current state and the state at heel strike. In general any convex combination between those two states is a valid linearisation point, such that

$$\vec{x}_{\text{lin}} = \vec{x}_{\text{now}}(1 - \lambda) + \vec{x}_{\text{hs}}\lambda, \tag{8}$$

where state x_{lin} is the linearisation point, x_{now} the current state, x_{hs} the state at heel strike and $\lambda \in [0, 1]$. Linearisation points at $\lambda = 0$ and $\lambda = 0.5$ are tested. In Appendix C more values for λ are evaluated. How well the linearised dynamics fit the data is also tested against the validation set and based on the NRMSE. The linearised models are initialised from multiple initial conditions corresponding to data points of the validation set as well.

E. Basics of the NMS model

Both a 2D [24] and a 3D NMS model [25] are considered in this paper. The latter is an extension of the 2D model and its use in this paper is to validate the controller based on the CGB model and generating the data for the aforementioned parameter estimation method. This model is suitable because it resembles a walking human and is able to react to torques provided by controller. The 2D model is used as an internal model for MPC. The 2D model is chosen over the 3D model because it is computationally more efficient due to its reduced degrees of freedom, but the 3D model is suitable as well when sufficient computational power is available.

The working principle of the 3D NMS model is visualised in the diagram of Figure 4. The skeleton is actuated by muscles. The forces of these muscles depend on stimulation signals from the reflex modules. The supraspinal control layer can adjust the foot's target position and can select which reflex modules to use. The 2D version works quite similar, but has no supraspinal control layer. Both models have seven segments, but the 3D model has more muscles and more mobility in the hip. Further details of the model are in the papers from Song, Geyer and Herr and references therein [24], [25].

Unfortunately, no method is found to initialise the model from any configuration. The main issue is that the model does not only depend on joint angles and velocities, but depends on internal signals like the muscle stimulation signals as well. If these signals are initialised with value zero, the model will collapse. It is difficult to initialise these signals in the complex model even if the desired values are known. Another issue with the model is that it uses the first generation Simscape library. Unfortunately that library is becoming obsolete, which gives some practical limitations. One limitation is that the newer versions of MATLAB no longer support this library, therefore an older release (2019a) is used. Another limitation is that it is not possible to save the state of the simulation and continue from there. The simulation must be initialised from the beginning. Besides the unsolved initiation issue, the model is ready for control [36]. More details about the initialisation issue are in Appendix D.

III. Results 1: Modelling

A. Position tracking of the CGB model

As described in section II, three positions are considered from which the leg angles $\vec{\theta}$ are computed: the heels, the ankles and the toes. Figure 5 shows the resulting angles for the three different positions, the angles are calculated by Equations 4 and 5. The ankle has the least spikes and is thus chosen as the best option. The remaining results use the position of the ankles to calculate leg angles $\vec{\theta}$.

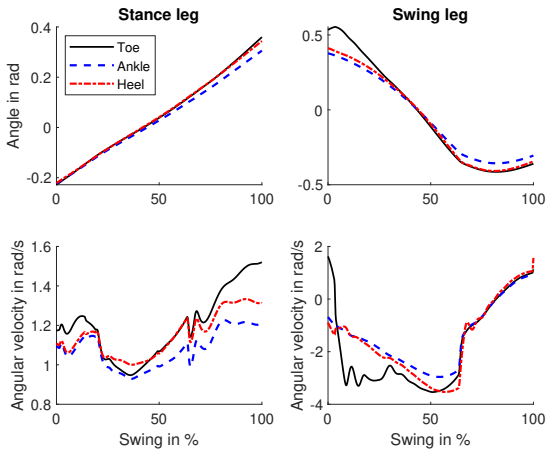

Fig. 5: The angles and angular velocity of the stance and swing leg calculated from different positions on the foot. The stance leg is described by θ_1 and the swing leg by θ_2. Time is normalised to a percentage of swing.

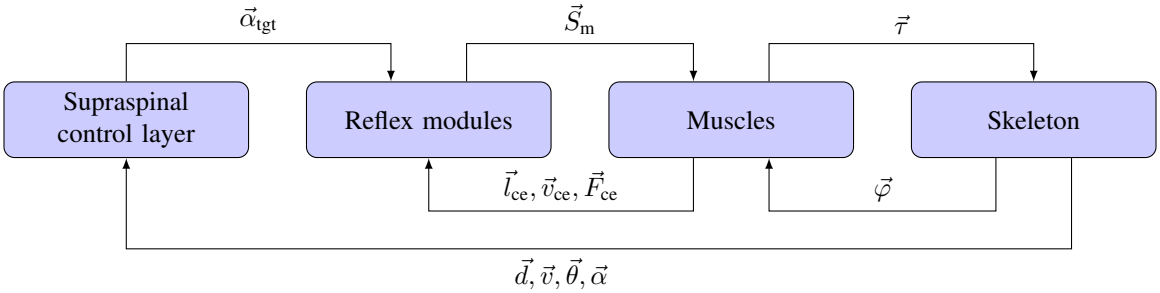

Fig. 4: A schematic representation of the NMS model. The signals described are the target leg angles $\vec{\alpha}_{\text{tgt}}$, the muscle stimulations $\vec{S}_{\text{m}}$, the muscle torques $\vec{\tau}$, the joint angles $\vec{\varphi}$, the lengths, velocities and forces of the contractile muscle elements $\vec{l}_{\text{ce}}, \vec{v}_{\text{ce}}$and $\vec{F}_{\text{ce}}$, the distances and velocities of the CoM with respect to the stance leg $\vec{d}$ and $\vec{v}$, the trunk angles $\vec{\theta}$ and the leg angles $\vec{\alpha}$. More details on the definitions and working principles are in the paper of Song [25].

B. Grey-box estimation of the CGB model

The optimised parameters of the CGB model are given in Table II. Figure 6 visualises how well the CGB models, the linear models and the nonlinear model, fit the data. The NRMSE between the predictions and the data are given in Table III. This table only shows the values from two initial conditions, at 0 and $50\,\%$ swing. These results are typical for all initial conditions. Both choices for linearisation points have similar NRMSE, but the linearisation point with $\lambda = 0.5$ tends to be higher at the start of the swing. Later in the swing both options have nearly the same performance. The NRMSE of the nonlinear model is typically higher than the linear model. However, there are (regions of) initial conditions where these typical results do not hold.

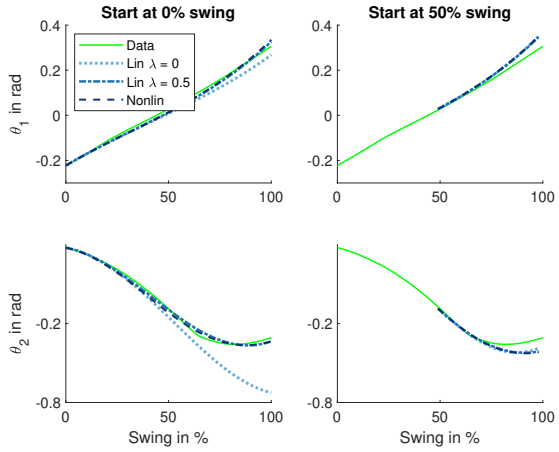

Fig. 6: A visualisation of how well the CGB model describes the data. The two linear and the nonlinear CGB models are compared. The percentage shows at which point of the swing the model was initiated. Time is normalised to a percentage of swing.

Parameters	m in kg	m_H in kg	a in m	ϕ_1 in rad	ϕ_2 in rad
Value	9.28	37.7	0.68	0.098	0.0524

TABLE II: The optimal parameters found by the grey-box method.

Model	θ_1 at 0%	θ_1 at 50%	θ_2 at 0%	θ_2 at 50%
lin $\lambda = 0$	0.86	0.74	0.44	0.75
lin $\lambda = 0.5$	0.91	0.75	0.86	0.74
nonlin	0.92	0.76	0.89	0.57

TABLE III: The NRMSE between the prediction of the models and the data the model attempts to predict.

IV. Methods 2: Control

A. Objectives and assumptions

The main objective is to reach a certain step length by controlling an MED with an MPC. This controller can be subdivided into a high-level, mid-level and low-level controller [17]. The high-level controller detects the intention. The mid-level controller translates the intent to a reference for the low-level controller. The latter computes the optimal control to track or reach the reference. For example, the high-level controller detects that the human tries to walk at a speed of $1.2\,\text{m/s}$. The mid-level controller than computes that the step length should be $0.7\,\text{m}$. The low-level controller, for example a feedback controller like an MPC, than computes the optimal control to reach the step length.

The focus will be on the low-level controller. It is assumed that a combination of a high- and a mid-level controller provides a reference to be tracked by the low-level controller. Such a reference can for example be obtained by the capture point method [37], [38] or by neural networks [39]. The low-level controller will be an MPC. Both the CGB model and the 2D NMS model are evaluated as internal models for the MPC.

It is assumed that any torque within the lower and upper bound is executable, there is no delay and the state of the MED is not considered. As mentioned before, the state can have influence on the torques the MED can provide. Appendix A shows a way to implement the dynamics and the state dependency of a CMG. The torque of the MED is limited between 0 and $5\,\text{Nm}$. The lower bound is set to zero because an MED is used is to enable reaching a larger step length, for which a positive torque is required. A maximum torque of $5\,\text{Nm}$ seems realistic as well [11], [12].

The step length is not a suitable control reference. As mentioned before, the step length is measured from one point

of the foot at initial contact to the same point at the other foot at the next moment of contact with the ground. An unfortunate consequence is that the position of that point at initial contact should be remembered. Additionally, if a model is initiated in mid-swing it does not have a clearly defined step length. For control it is more practical to have a measure that only depends on the current configuration. Therefore, the distance from one ankle to the other z is controlled to the reference z_{ref}. The letter z is chosen to indicate that this value is not a sensor output (see subsection II-A), commonly denoted by y.

For the CGB controllers the diagram of Figure 7 holds. The controller uses the state $\vec{x}$ and the reference z_{ref} to calculate the torque u of the MED. This torque is applied on the human and a new state is obtained. This in the loop structure is not used for the NMS controller, because the current state of the human can not be used by the controller, see subsection II-E.

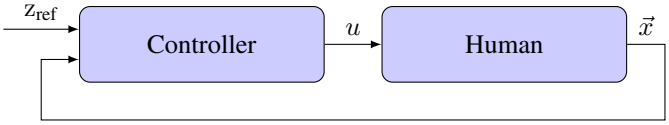

Fig. 7: A block diagram of how the controller implemented in the loop.

B. Hybrid MLD-MPC based on the CGB model

As mentioned in the introduction, MLD models are suitable for MPC [19], [21]. Other hybrid models are suitable as well, like piecewise affine functions or max-min-plus-scaling systems. These models are equivalent to MLD systems under mild assumptions [40]. MLDs are chosen because the binary variables are intuitive to use.

One difficulty is that the reference, the distance from one ankle to the other, is only well defined in space. The time it takes to reach the reference is not of importance. This issue is solved with hybrid dynamics, the state is fixed after heel strike occurs. If a sufficiently large horizon N is chosen, the feet are fixed in the configuration where heel strike occurred at the end of the horizon. The controller should control z such that it close to the reference z_{ref} at the end of the horizon. The dynamics are

$$\vec{x}(k+1) = \begin{cases} \boldsymbol{A}\vec{x}(k) + \boldsymbol{B}_u u(k) + \vec{B}_{\text{aff}} & \text{in prosthetic swing} \\ \vec{x}(k) & \text{after heel strike,} \end{cases} \tag{9}$$

where matrices $\boldsymbol{A}, \boldsymbol{B}_u$ and vector $\vec{B}_{\text{aff}}$ are obtained by discretisation and linearisation of the CGB model, see Appendix B. These dynamics are written as an MLD system in Appendix E.

The cost function J_1 penalises the distance between $\hat{\mathrm{z}}$ and its reference z_{ref} as well as the torque. The latter is to use less torque if possible, but the focus should be on reaching the reference. In mathematical terms

$$J_1\big(\vec{x}(k), \vec{u}, N\big) = Q_{\mathrm{z}}\big(\hat{\mathrm{z}}(N) - \mathrm{z}_{\text{ref}}\big)^2 + \vec{u}^{\mathrm{T}}\boldsymbol{Q}_u\vec{u}, \tag{10}$$

where Q_{z} and $\boldsymbol{Q}_u$ contain weights, $\hat{\mathrm{z}}$ is the estimated value of z and

$$\vec{u} = \begin{bmatrix} u(k) \\ \vdots \\ u(N-1) \end{bmatrix}. \tag{11}$$

The controller uses Algorithm 1 when simulated on the 3D NMS model. To minimise J_1 the optimisation problem is written as a mixed integer quadratic programming problem (MIQP), which is solved with Gurobi [41]. Details on how the MIQP formulation is obtained are in Appendix F. A sufficiently large N is chosen by trail and error. Another way to find N is to estimate when heel strike occurs and add some time steps to make sure heel strike is reached within the horizon. Such an estimation is possible with e.g. forward simulation or with a ballistic approximation [42]. As mentioned before, the controllers main objective is to reach the reference. Therefore, the parameter Q_{z} is chosen much larger than the elements of $\boldsymbol{Q}_u$. The weights on $\vec{u}$ are used to favor solutions that use little torque.

Algorithm 1:

Result: Optimal actuator torque u_{opt}
while *walking* **do**
  if *prosthetic swing* **then**
    Set a sufficiently large N
    $\min_{\vec{u}} J_1(\vec{x}(k), \vec{u}, N)$
    $u_{\text{opt}} = \vec{u}(1)$
  else
    $u_{\text{opt}} = 0\,\mathrm{Nm}$
  end
end

C. NMPC based on the CGB model

The NMPC is designed similar to the MLD-MPC. Both use Algorithm 1, cost function J_1 and the values of the parameters from the CGB model and cost function are the same. However, the NMPC uses the nonlinear dynamics of the CGB model are instead of the linearised dynamics. These equations of motion are integrated with the fourth-order Runge Kutta method. Similar to the linear case, once the foot hits the ground, the state is fixed as well. Fixing the state was achieved by an if-else statement in the dynamics.

For optimisation of J_1, the Matlab function 'fmincon' is used [43]. This function requires a starting point, in this case a control sequence $\vec{u}$. The optimisation algorithm then uses a gradient descent method to find a better control sequence. Unfortunately, the algorithm can converge to a sub-optimal local minima [44]. Different starting points can lead to different local minima and thus to different solutions that are sub-optimal. However, in this case the same optimum is found from all tested initial guesses. The tested initial states are the zero initial condition, several multisines and random distributions. This indicates that the model is not sensitive to local minima, although this is not proven. It is assumed that the optimisation algorithm is not sensitive to local minima and that the algorithm will converge to the global minimum.

D. NMPC based on the NMS model

The 2D NMS model is used as an model for MPC as well. The NMS model is much more complex than the CGB model (see section II). Consequently, a single simulation with a given actuator torque is much more computationally expensive. To reduce the complexity, a constant actuator torque u_{con} is chosen for the entire horizon. The controller thus optimises a single variable instead of a control sequence. However, an MPC calculates a new constant torque each time step, so the executed torque is not necessarily constant over the whole swing.

As mentioned before, the objective is to reach a certain distance between the stance and the swing foot. The time it takes is irrelevant. Therefore, the 2D NMS model is simulated until heel strike. The cost function is slightly different to J_1 as well. It is assumed that only one constant torque can reach the desired reference. Therefore there is no need to differentiate between multiple solutions, no penalty on the torque is included in the cost function. The cost function

$$J_2\big(\vec{x}(k), u_{\text{con}}\big) = \big(\hat{z}(x(k), u_{\text{con}}) - z_{\text{ref}}\big)^2. \tag{12}$$

Evaluations of J_2 with different u_{con} showed that J_2 is approximately a quadratic function, but not completely smooth, see Appendix G. Parabolic interpolation is chosen as a suitable algorithm, because this algorithm can use the quadratic shape to quickly converge to the optimum [44]. The optimisation algorithm is described in Algorithm 2. The optimisation stops when $J_2 < 10^{-8}$, such that the foot is placed within $1\,\text{mm}$ of the reference.

To test the controller a reference test and a disturbance test are performed. Unfortunately the controller cannot be tested on a more complex model, because it cannot be initiated from a desired state (see subsection II-E). Instead, the model is simulated for 3 swings, where tests are done in the third swing. In the reference test the controller tries to reach a range of references, from $0.75\,\text{m}$ to $0.95\,\text{m}$. In this test no disturbances are added and the MED is placed on either the shank and the thigh. The location that uses the least torque to reach a reference is chosen for the rest of the disturbance test as well. In the disturbance test a the disturbance is applied for a short duration ($0.01\,\text{s}$) at the beginning of the swing phase and controlled to the reference $z_{\text{ref}} = 0.8\,\text{m}$.

Algorithm 2:

Result: Optimal actuator torque
while *walking* **do**
  if *prosthetic swing* **then**
    Choose 3 inputs u_1, u_2 and u_3
    Evaluate $J_2(u_i, k)$ for $i = 1, 2, 3$
    while $J_2(u, k) > 10^{-8}$ **do**
      Fit a quadratic function through the last 3 evaluated points of J_2
      Minimise the quadratic function to find new input u_{new}
      Evaluate $J_2(u_{\text{new}}, k)$
    end
    u_{opt} is the input where $J_2(u, k) < 10^{-8}$.
    If u_{opt} is out of bounds, go to the nearest bound.
  else
    $u_{\text{opt}} = 0\,\text{Nm}$
  end
end

E. Controller comparison

The time it takes for the controller to calculate the optimal torque is compared between all controllers. The computations are performed on a HP ZBook Studio x360 G5 with an Intel Core i7-9750H processor. The computation time depends on the initial condition as the horizon is not the same for each initial condition. Additionally, the optimisation algorithm might need more iterations from a certain initial condition. The computation time should not exceed the step time, therefore the maximum required computation time is most interesting. The longest horizon is just after toe off, therefore the first state of the swing is used to compare the speed of the controllers.

Three controllers based on the CGB model are tested on the 3D NMS model. Two are MLD-MPC, where one is linearised in the current state ($\lambda = 0$) and the other is linearised in the middle of the remaining swing ($\lambda = 0.5$). The third controller is an NMPC. The controllers activate at the start of the sixth swing with the right foot of the 3D NMS model. Without any control $z_{\text{hs}} = 0.5882\,\text{m}$, where z_{hs} is the ankle to ankle distance at heel strike. When using the maximum $5\,\text{Nm}$ of torque for the full swing $z_{\text{hs}} = 0.7903\,\text{m}$. With this in mind, the parameters of Table IV are chosen for all three controllers. The torques are applied on either the shank or the thigh, both options are tried. Because the NMPC based on the 2D NMS model still has initialisation issues, it is not used as a controller in a simulation environment.

Parameter	Q_z	$\boldsymbol{Q}_u$	z_{ref}
Value	10^8	$\boldsymbol{I}$	$0.7\,\text{m}$

TABLE IV: The parameters of cost function J_1, where $\boldsymbol{I}$ is the identity matrix.

V. Results 2: control

A. Control using the CGB model

The performance of the three CGB-controllers are compared in Figure 8. All three controllers do not reach the reference of $0.7\,\text{m}$, instead z_{hs} is between 0.60 and $0.61\,\text{m}$. For all three simulations the torque is applied on the shank. When applying the torque on the thigh instead of the shank the optimal torques are close to the trajectories shown in Figure 8, but z_{hs} is slightly lower, approximately half a centimeter.

The computation time of the controllers are in Table V. The computation time of both MLD-MPCs are similar and therefore grouped.

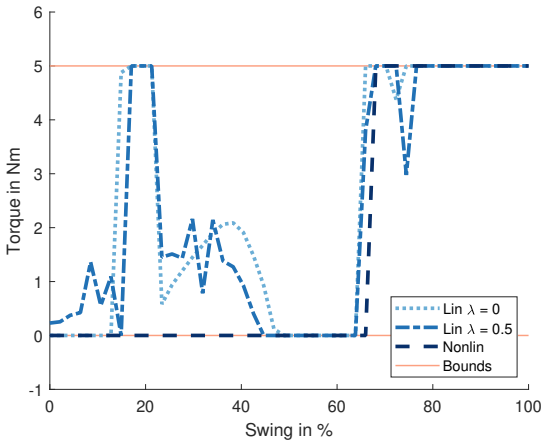

Fig. 8: The results of all control algorithms using the CGB model. Time is normalised to a percentage of swing.

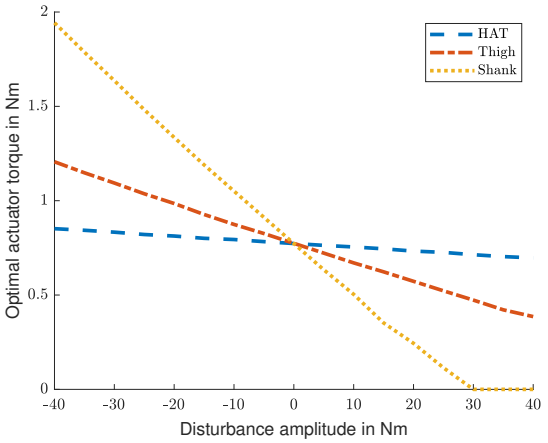

Fig. 10: The results of the disturbance tests with torque disturbances for the NMS controller.

B. Control using the NMS model

The results of the reference tests are visualised in Figure 9. Because applying the torque on the shank is more effective, this is done for the remaining results.

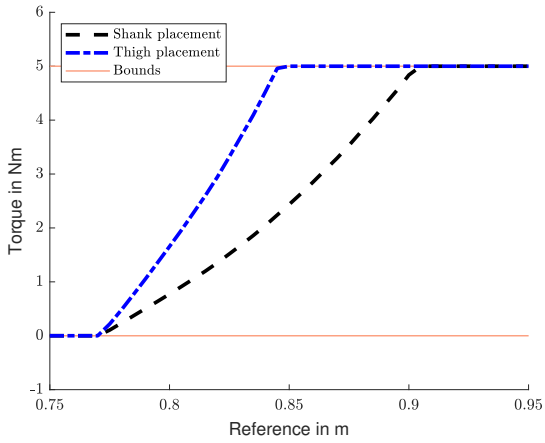

Fig. 9: The results of the reference tests for the NMS controller.

The results of the disturbance tests are shown in Figure 10 and Figure 11. The torque and push disturbances in the figure are applied to the shank and thigh of the swinging leg, as well as the HAT. The disturbances are tested on the other ligaments as well, but the closer the disturbance to the stance foot, the less it affected the optimal torque.

The parabolic interpolation algorithm uses four to ten evaluations of J_2 to find a suitable solution. Ten evaluations take approximately 35 s, which is the number shown in Table V.

Controller	MLD-MPC CGB	NMPC CGB	NMPC NMS
Time in s	0.2	0.06	35

TABLE V: The computation time of the different control algorithms from the start of the swing.

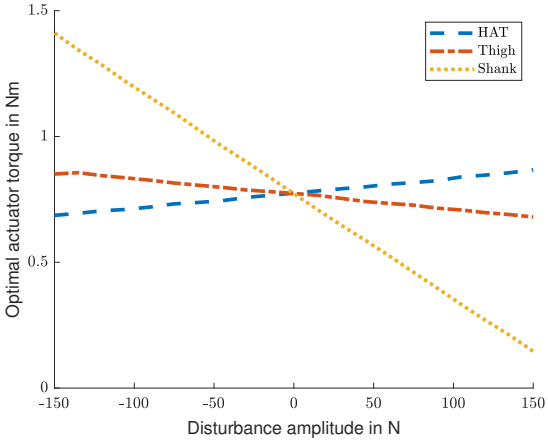

Fig. 11: The results of the disturbance tests with push disturbances for the NMS controller.

VI. Discussion

A. Modelling

The results of subsection III-B show how a nonlinear CGB model can be fitted on human like data. The fit, made from the beginning of the swing, describes the angles θ_1 and θ_2 of the CGB model quite well with a NRMSE of 0.92 and 0.89 on a validation set respectively. However, when initialised from another initial condition than the start of the swing phase the performance deteriorates. This makes sense because the fit was optimised for the initial conditions at the start of the swing phase.

Linearisation of the nonlinear CGB model should reduce the quality of the fit, because it is an approximation of the nonlinear model. This is also noticeable in the results. When comparing the NRMSE values from the start of the swing, the nonlinear model has the best fit (see Table III). The linearisation with $\lambda = 0.5$ is better than the linearisation with $\lambda = 0$ at the start of the swing. For example, the prediction of the linear model with $\lambda = 0$ that starts from 0 % swing has a NRMSE of only 0.44 when predicting θ_2. The other

linearisation option with $\lambda = 0.5$ is more accurate in that case with an NRMSE of 0.86. However, starting from $50\,\%$ of the swing the model with $\lambda = 0.5$ performs similar to if not slightly worse than the model with $\lambda = 0$. Generally, the NRMSE is slightly higher when $\lambda = 0.5$ if more initial points are evaluated, but there are exceptions. These exceptions could be coincidental.

Pre-processing data before system identification can help achieving a better fit [45]. The data in this paper was not pre-processed, because the data is exact. There is no bias or disturbance from sensors. However, filtering the data anyway may reduce the spikes of the controller. A better result may also be achieved by splitting the swing into multiple segments and fitting a (non)linear model through each segment [46]. Another limitation is that the model is only tested on data from a regular walking gait simulated by a model. A disturbed gait or actual human data could reduce the quality of the prediction due to unmodeled dynamics and noise respectively. Furthermore, no MED was used in generating the data for grey-box estimation. How the MED effects the states was based on mechanics. Finally, the leg length can be optimised as well, it was now set to the length of $1\,\mathrm{m}$ based on visual inspection.

Overall, the angle prediction of the CGB models seem reasonable if only an approximation is required. However, the model and the identification technique make many (aforementioned) rough assumptions. Therefore the models are not suitable for an accurate prediction. Whether the models are accurate enough or not, they are usable as an internal model for MPC.

B. Control using the CGB model

Unfortunately, none of the controllers based on the CGB model are able to reach the reference. The main reason the controllers fail is because the prediction of the state at heel strike is not accurate enough. If the model incorrectly predicts that the foot will reach the reference or further, the controller will provide no torque even though torque is required. This is the case for part of the swing phase (e.g. between $45\,\%$ and $65\,\%$ of the swing phase). Around $65\,\%$ of the swing the internal model realises that it can no longer reach the reference and consequently the controller provides maximum torque for the remainder of the swing.

The computation of the optimal torque is quite efficient, with computational times not larger than $0.2\,\mathrm{s}$ at the start of the swing for any of the CGB controllers. In the current implementation the NMPC is even faster. The reasonable computation times give confidence in that the controllers are implementable in real-time if the controllers are optimised for computation time. There are ways to increase the efficiency. For example, the step size is now set to $0.01\,\mathrm{s}$. If the step size is increased, the size of the optimisation problem is reduced because it reduces the number of variables. There are other options for reducing computation time as well, like different optimisation algorithms or only evaluating likely solutions [21], [44], [47].

Before a CGB controller is usable in a practical application, at least the following steps must be taken. First, the dynamics of the MED must be implemented. Appendix A gives some options on how the dynamics can be implemented. Secondly, the controller must reset the MED to the optimal configuration such that it can again be used in the prosthetic swing phase. While resetting the MED provides torques on the leg. Therefore, the full gait cycle should be analysed. Finally, a good observer must be made to estimate the states of the CGB, this can be especially challenging because the exact position of body parts are hard to obtain, as discussed in subsection II-A.

The analysis of the controller in simulation indicated that the predictive performance of the CGB mode as an internal model for MPC is, at least in the current form, insufficient. However, the approximation may be good enough for certain applications, especially if some of the aforementioned limitations of the grey-box estimation method are addressed. The proposed control method is strongly dependent on the predicted state at heel strike. As this is the final point of the prediction, this state is the most uncertain. The MLD-MPC and NMPC control method may be interesting for other control approaches as well, for example, where a reference trajectory is tracked instead of a reference point [48].

C. Control using the NMS model

Compared to the thigh, a torque on the shank seems more effective when increasing the step length, because the same reference is reached with less torque. Placement on the shank is also easier from a practical point of view, because the shank of transfemoral amputees is fully amputated while the stump of the thigh limits space. In a study where free torques, the torques provided by MEDs, were tested for their performance on both the shank and the thigh, the shank was more effective as well, but the difference was small [7]. The authors of that study considered more control flexibility than the constant torque for the full swing cycle used in this study. However their results and the results of this paper are from simulations and can thus differ in practice. There are other factors to keep in mind as well, like comfort. Even if the MED is more effective on the shank, placement on the thigh could be more comfortable for the user and still be effective enough. Another benefit of placement on the thigh is that free torques on the thigh are more effective for toe clearance [7]. Placing the CMG on the shank can cause the foot to hit the ground when used for fall prevention [11], so an increase in toe clearance is desired.

The disturbance tests show that the optimisation algorithm is robust in the sense that slight deviations to the nominal gait do not result in excessive control spikes. Moreover, the relation between the optimal torque and the disturbance are of an affine nature even though the model is highly nonlinear. These affine relations in combination with the affine relation between the optimal torque and the step length reference may somehow be exploitable by a controller.

The biggest limitation of the NMS controller is that the NMS model cannot be initiated in a desired state, as explained in subsection II-E. Consequently, the controller can not be tested in a simulation environment. However, it is likely that the prediction is more accurate and therefore results in better control performance because the 2D NMS model is much

more complex than the CGB model. On the downside, the NMS controller takes approximately $35\,\mathrm{s}$ to find the optimal control. The NMS model therefore seems to complex for a real-time application. Additionally, a more complex function than a constant torque for the full swing might be desired, which can increase the number of variables to be optimised. To reduce computation time the model can be upgraded such that it can run from any initial state. This will severely reduce the computational cost as the 2 strides do not have to be simulated every evaluation of the cost function J_2. Other options like precompiling the model, better initial guesses and the relaxation of the stopping condition can also reduce the computation time.

In order to use the 2D NMS model in a practical application, at least the same steps covered in subsection VI-B and the following two should be made. Firstly, the software must be upgraded such that the NMS model can be initialised from any position. Secondly, instead of doing the optimisation in real-time, the controller should uses a function that maps the current state to (an approximation of) the optimal torque. Suitable functions are e.g. a neural network [49] or a PWA function [50], [51]. This function can be computed offline, such that only the function has to be evaluated in real-time.

D. Future research

There is currently no sufficient MPC algorithm that can reach a set step length by using an MED on a leg, and this is only the first step before an MPC can be build for an actual prosthesis. For an actual prosthesis, the control approach of this paper might be too complex because in this approach the controller must accurately predict the desired reference and the pathological gait. If one of these predictions is off, the controller may show fluctuations, like the CGB controllers in Figure 8. Measuring intent and modelling pathological gait are both difficult to do at this point in time [11], [17]. The attempt to model and control a healthy human unfolded many practical challenges. Therefore, another control method for controlling MEDs is advised. The controller method may however be good enough for fall prevention, because in that case an exact prediction of the future is not required [12].

Instead of predicting the shortcoming of the human and using the MED to fill up the gap, a shared control approach could be interesting [52]. A major difference between this approach with respect to the control approach used in this paper is that the goal of the controller is not to reach a certain step length, but to reduce the required work of the human such that he or she is capable of placing his or her foot as desired. Shared control schemes may still use MPC and models to predict the future and provide torques accordingly [53].

Besides further development of control techniques, clinical trials with MEDs should be conducted as well. These trials could answer open questions on how humans respond to MEDs. For example, how does the human adapt to torques of the MED? Or, how obtrusive are torques in undesired directions caused by CMGs [5]? Finally, it should be researched whether an MED is a desirable solution to step asymmetry and whether gait symmetry should be the objective of the controller in the first place. Maybe another objective, like reducing the required work of the human, is better. It is advised to first answer some of these fundamental questions before further developing the MPC, because the control approach depends on the answers.

VII. Conclusion

As of now, there is no suitable controller that controls a momentum exchange device to achieve a certain step length during a normal walking gait. For model predictive control, the compass-gait biped model found with the nonlinear grey-box parameter estimation method is insufficient in predicting the step length. However, the model can roughly predict the swing phase of a walking human. The use of a neuromusculoskeletal model for prediction has other issues. The controller's performance based on this model is not verified because it could not be initiated. Additionally, observability and a high computation time remain an issue. Simulations of this model do indicate that a momentum exchange device to control the step length is more effective on the shank than on the thigh.

This research shows that finding an accurate and practical model for the model predictive control problem is challenging. Therefore it is suggested to find a control approach that does not depend on an accurate actual and desired step length prediction, unlike the controllers in this paper.

Acknowledgments

I would like to thank my supervisors prof. dr. ir. H. Vallery and dr. ir. A.J.J. van den Boom for their help and feedback. They were able to perfectly balance the disciplines of BioMechanical Design and Systems & Control. I would also like to thank Mahdi Nabipour and Nathan Timmers for sharing their knowledge and files on the subject of the paper. Finally, I would like to thank Bram Haanen for helping me as a study partner and keeping me motivated during this Covid period.

References

[1] A. O. Petersen, J. Comins, and T. Alkjær, "Assessment of gait symmetry in transfemoral amputees using c-leg compared with 3r60 prosthetic knees," *JPO: Journal of Prosthetics and Orthotics*, vol. 22, no. 2, pp. 106–112, 2010.

[2] A. D. Segal, M. S. Orendurff, G. K. Klute, M. L. McDowell, J. A. Pecoraro, J. Shofer, and J. M. Czerniecki, "Kinematic and kinetic comparisons of transfemoral amputee gait using c-leg and mauch sns prosthetic knees." *Journal of Rehabilitation Research & Development*, vol. 43, no. 7, 2006.

[3] M. J. Highsmith, B. W. Schulz, S. Hart-Hughes, G. A. Latlief, and S. L. Phillips, "Differences in the spatiotemporal parameters of transtibial and transfemoral amputee gait," *JPO: Journal of Prosthetics and Orthotics*, vol. 22, no. 1, pp. 26–30, 2010.

[4] H. Schaub, S. R. Vadali, J. L. Junkins *et al.*, "Feedback control law for variable speed control moment gyros," *Journal of the Astronautical Sciences*, vol. 46, no. 3, pp. 307–328, 1998.

[5] A. Berry, D. Lemus, R. Babuška, and H. Vallery, "Directional singularity-robust torque control for gyroscopic actuators," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 6, pp. 2755–2763, 2016.

[6] D. J. Block, K. J. Åström, and M. W. Spong, "The reaction wheel pendulum," *Synthesis Lectures on Control and mechatronics*, vol. 1, no. 1, pp. 1–105, 2007.

[7] S. Jabeen, A. Berry, T. Geijtenbeek, J. Harlaar, and H. Vallery, "Assisting gait with free moments or joint moments on the swing leg," in *2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*. IEEE, 2019, pp. 1079–1084.

[8] D. Lemus, A. Berry, S. Jabeen, C. Jayaraman, K. Hohl, F. C. van der Helm, A. Jayaraman, and H. Vallery, "Controller synthesis and clinical exploration of wearable gyroscopic actuators to support human balance," *Scientific reports*, vol. 10, no. 1, pp. 1–15, 2020.

[9] T. L. Brown and J. P. Schmiedeler, "Reaction wheel actuation for improving planar biped walking efficiency," *IEEE transactions on robotics*, vol. 32, no. 5, pp. 1290–1297, 2016.

[10] J.-H. Park and B.-K. Cho, "Development of a self-balancing robot with a control moment gyroscope," *International Journal of Advanced Robotic Systems*, vol. 15, no. 2, p. 1729881418770865, 2018.

[11] N. Timmers, "Simulating gait with the 3r60 knee prosthesis and a control moment gyroscope," 2020.

[12] M. Nabipour, S. A. A. Moosavian, and H. Vallery, "Comparing the optimized performance of fully and under-actuated lower-limb wearable devices," 2021.

[13] G. Kejlaa, "Consumer concerns and the functional value of prostheses to upper limb amputees," *Prosthetics and orthotics international*, vol. 17, no. 3, pp. 157–163, 1993.

[14] E. A. Biddiss and T. T. Chau, "Upper limb prosthesis use and abandonment: a survey of the last 25 years," *Prosthetics and orthotics international*, vol. 31, no. 3, pp. 236–257, 2007.

[15] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer science & business media, 2013.

[16] K. Ekkachai and I. Nilkhamhang, "Swing phase control of semi-active prosthetic knee using neural network predictive control with particle swarm optimization," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 11, pp. 1169–1178, 2016.

[17] M. R. Tucker, J. Olivier, A. Pagel, H. Bleuler, M. Bouri, O. Lambercy, J. del R Millán, R. Riener, H. Vallery, and R. Gassert, "Control strategies for active lower extremity prosthetics and orthotics: a review," *Journal of neuroengineering and rehabilitation*, vol. 12, no. 1, p. 1, 2015.

[18] P.-B. Wieber, R. Tedrake, and S. Kuindersma, "Modeling and control of legged robots," in *Springer handbook of robotics*. Springer, 2016, pp. 1203–1234.

[19] B. d. Schutter and M. Heemels, "Modeling and control of hybrid systems," 2015. [Online]. Available: https://www.webedu.nl/bestellen/tudelft/

[20] D. Mayne, "Nonlinear model predictive control: Challenges and opportunities," *Nonlinear model predictive control*, pp. 23–44, 2000.

[21] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[22] J. Kreuk, "Med_mpc," https://github.com/JesperKreuk/MPC_MED, 2021.

[23] F. Asano, M. Yamakita, and K. Furuta, "Virtual passive dynamic walking and energy-based control laws," in *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)(Cat. No. 00CH37113)*, vol. 2. IEEE, 2000, pp. 1149–1154.

[24] H. Geyer and H. Herr, "A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities," *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 18, no. 3, pp. 263–273, 2010.

[25] S. Song and H. Geyer, "A neural circuitry that emphasizes spinal feedback generates diverse behaviours of human locomotion," *The Journal of physiology*, vol. 593, no. 16, pp. 3493–3511, 2015.

[26] F. Huxham, J. Gong, R. Baker, M. Morris, and R. Iansek, "Defining spatial parameters for non-linear walking," *Gait & posture*, vol. 23, no. 2, pp. 159–163, 2006.

[27] E. Demircan, L. Sentis, V. De Sapio, and O. Khatib, "Human motion reconstruction by direct control of marker trajectories," in *Advances in Robot Kinematics: Analysis and Design*. Springer, 2008, pp. 263–272.

[28] C. M. Lara-Barrios, A. Blanco-Ortega, C. H. Guzman-Valdivia, and K. D. Bustamante Valles, "Literature review and current trends on transfemoral powered prosthetics," *Advanced Robotics*, vol. 32, no. 2, pp. 51–62, 2018.

[29] M. Kok, J. D. Hol, and T. B. Schön, "Using inertial sensors for position and orientation estimation," *arXiv preprint arXiv:1704.06053*, 2017.

[30] F. Asano, Z.-W. Luo, and M. Yamakita, "Biped gait generation and control based on a unified property of passive dynamic walking," *IEEE Transactions on Robotics*, vol. 21, no. 4, pp. 754–762, 2005.

[31] M. Garcia, A. Chatterjee, A. Ruina, and M. Coleman, "The simplest walking model: stability, complexity, and scaling," 1998.

[32] B. Besselink, U. Tabak, A. Lutowska, N. van de Wouw, H. Nijmeijer, D. J. Rixen, M. Hochstenbach, and W. Schilders, "A comparison of model reduction techniques from structural dynamics, numerical mathematics and systems and control," *Journal of Sound and Vibration*, vol. 332, no. 19, pp. 4403–4422, 2013.

[33] N. R. Kristensen, H. Madsen, and S. B. Jørgensen, "Parameter estimation in stochastic grey-box models," *Automatica*, vol. 40, no. 2, pp. 225–237, 2004.

[34] "nlgreyest - estimate nonlinear grey-box model parameters," https://www.mathworks.com/help/ident/ref/iddata.nlgreyest.html, accessed: 02-07-2021.

[35] "goodnessoffit - goodness of fit between test and reference data," https://www.mathworks.com/help/releases/R2019a/ident/ref/goodnessoffit.html?searchHighlight=goodnessoffit&s_tid=doc_srchtitle, accessed: 14-07-2021.

[36] "Save and restore simulation operating point," https://nl.mathworks.com/help/simulink/ug/saving-and-restoring-simulation-operating-point.html, accessed: 02-07-2021.

[37] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *2006 6th IEEE-RAS international conference on humanoid robots*. IEEE, 2006, pp. 200–207.

[38] A. L. Hof, R. M. van Bockel, T. Schoppen, and K. Postema, "Control of lateral balance in walking: experimental findings in normal subjects and above-knee amputees," *Gait & posture*, vol. 25, no. 2, pp. 250–258, 2007.

[39] C. Jayaraman, C. K. Mummidisetty, M. V. Albert, R. Lipschutz, S. Hoppe-Ludwig, G. Mathur, and A. Jayaraman, "Using a microprocessor knee (c-leg) with appropriate foot transitioned individuals with dysvascular transfemoral amputations to higher performance levels: a longitudinal randomized clinical trial," *Journal of neuroengineering and rehabilitation*, vol. 18, no. 1, pp. 1–13, 2021.

[40] W. P. Heemels, B. De Schutter, and A. Bemporad, "Equivalence of hybrid dynamical models," *Automatica*, vol. 37, no. 7, pp. 1085–1091, 2001.

[41] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2021. [Online]. Available: https://www.gurobi.com

[42] H. Van Der Kooij, R. Jacobs, B. Koopman, and F. Van Der Helm, "An alternative approach to synthesizing bipedal walking," *Biological cybernetics*, vol. 88, no. 1, pp. 46–59, 2003.

[43] "fmincon - find minimum of constrained nonlinear multivariable function," https://www.mathworks.com/help/optim/ug/fmincon.html, accessed: 02-07-2021.

[44] T. van den Boom and B. de Schutter, "Optimiztion in systems and control," 2018. [Online]. Available: https://www.webedu.nl/bestellen/tudelft/

[45] M. Verhaegen and V. Verdult, *Filtering and system identification: a least squares approach*. Cambridge university press, 2007.

[46] D. Corona, M. Lazar, B. De Schutter, and M. Heemels, "A hybrid mpc approach to the design of a smart adaptive cruise controller," in *2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control*. IEEE, 2006, pp. 231–236.

[47] T. Marcucci, R. Deits, M. Gabiccini, A. Bicchi, and R. Tedrake, "Approximate hybrid model predictive control for multi-contact push recovery in complex environments," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 31–38.

[48] M. A. Holgate, A. W. Bohler, and T. G. Suga, "Control algorithms for ankle robots: A reflection on the state-of-the-art and presentation of two novel algorithms," in *2008 2nd IEEE RAS & EMBS international conference on biomedical robotics and biomechatronics*. IEEE, 2008, pp. 97–102.

[49] M. A. Botto, T. J. van den Boom, A. Krijgsman, and J. S. da Costa, "Constrained nonlinear predictive control based on input-output linearization using a neural network," *IFAC Proceedings Volumes*, vol. 29, no. 1, pp. 2574–2579, 1996.

[50] L. H. Csekő, M. Kvasnica, and B. Lantos, "Explicit mpc-based rbf neural network controller design with discrete-time actual kalman filter for semiactive suspension," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 5, pp. 1736–1753, 2015.

[51] R. Oberdieck and E. N. Pistikopoulos, "Explicit hybrid model-predictive control: The exact solution," *Automatica*, vol. 58, pp. 152–159, 2015.

[52] D. P. Losey, C. G. McDonald, E. Battaglia, and M. K. O'Malley, "A review of intent detection, arbitration, and communication aspects of shared control for physical human–robot interaction," *Applied Mechanics Reviews*, vol. 70, no. 1, 2018.

[53] L. Song, H. Guo, F. Wang, J. Liu, and H. Chen, "Model predictive control oriented shared steering control for intelligent vehicles," in *2017 29th Chinese Control And Decision Conference (CCDC)*. IEEE, 2017, pp. 7568–7573.

[54] D. Corona and B. De Schutter, "Adaptive cruise control for a smart car: A comparison benchmark for mpc-pwa control methods," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 2, pp. 365–372, 2008.
[55] "Using operating point data for model initialization," https://www.mathworks.com/help/physmod/simscape/ug/using-operating-point-data-for-model-initialization.html, accessed: 16-07-2021.
[56] "Simulink.blockdiagram.getinitialstate - return initial state data of block diagram," https://www.mathworks.com/help/simulink/slref/simulink.blockdiagram.getinitialstate.html, accessed: 16-07-2021.
[57] "Hysdel - hybrid system description language," http://people.ee.ethz.ch/~cohysys/hysdel/, accessed: 02-07-2021.
[58] "Hybrid toolbox for matlab," http://cse.lab.imtlucca.it/~bemporad/hybrid/toolbox/, accessed: 02-07-2021.

APPENDIX

Appendix A: Actuator dynamics

As mentioned in the introduction, MEDs can have saturation and singularity issues. Additionally, the maximum torque of the MED can also be state dependent. Tracking the actuator state and modelling the dynamics can be helpful when dealing with these issues. Moreover, the ability of an MPC to deal with actuator constraints is one of the main reasons the control approach is chosen for this case. This section discusses how the dynamics of the actuator can be included. Attention is focused on CMGs, but similar approaches can be used for RWs. An advantage of a CMG is that the torque experienced on the leg can be much larger than the torque used to turn the CMG. A CMG is visualised in Figure 12. The wheel spins fast around the $\vec{g}_s$ axis with a constant angular velocity of Ω. The CMG can rotate around the gimbal axis $\vec{g}_g$, this induces a torque around the $\vec{g}_t$ axis. The amplitude of the torque

$$T = I_{W_s}\dot{\gamma}\Omega, \tag{13}$$

where I_{W_s} is the moment of inertia of the wheel and $\dot{\gamma}$ the angular velocity around the gimbal axis. For this equation to be true components from other rotations like the rotation of the CMG structure itself are neglected because the spinning velocity should be much larger than the other velocities.

The direction of the torque changes as the CMG rotates around its gimbal axis. The torque is desired in the sagittal plane, therefore a scissor-pair of CMGs can be used, see Figure 13. The torques from the two CMGs cancel each other out such that the torque is only in the $\vec{b}_t$ direction with amplitude

$$T_{\text{sci}} = 2I_{W_s}\Omega\cos(\gamma)\dot{\gamma}. \tag{14}$$

However, both a single CMG and a scissor-pair of CMGs can induce no torque in the $\vec{b}_t$ direction if $\gamma = \pi/2\,\text{rad}$. This is the singularity of the CMG, this configuration should be avoided by the controller.

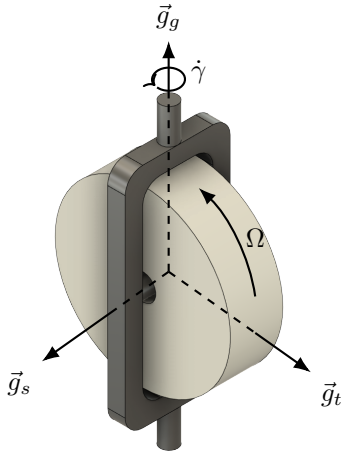

Fig. 12: A control moment gyroscope.

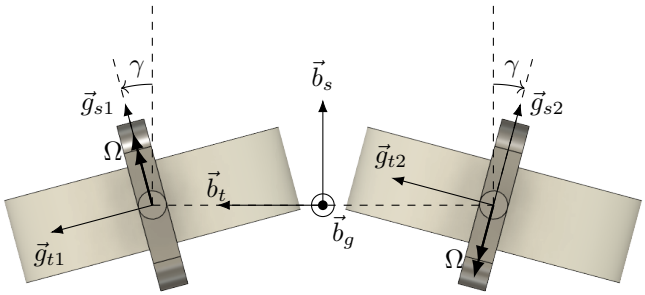

Fig. 13: A scissor-pair of control moment gyroscopes. The numerical subscripts denote the corresponding control moment gyroscope.

The state of the CMG can be tracked by augmenting the state $\vec{x}$ with the angle γ and assuming $\dot{\gamma}$ can be directly controlled [12]:

$$\dot{\gamma} = u_{\text{cmg}} \tag{15}$$

If a scissor pair of CMGs is used, Equation 26 can then be changed to

$$\boldsymbol{S}\vec{u}_{\text{cgb}} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} u_{\text{ank}} \\ u_{\text{hip}} \\ 2I_{W_s}\Omega\cos(\gamma)u_{\text{cmg}} \end{bmatrix} \tag{16}$$

The state γ can is then obtained by integrating the input u_{cmg}.

Unfortunately the amplitude of the torque has a nonlinear relation with angle γ. For MPCs that use linear dynamics, the dynamics can be linearised in the current state [12]. This does have the disadvantage that the controller may think it can provide more torque than is actually possible. This should not be an issue for the nonlinear controllers because they can implement the nonlinear dynamics.

One way to check whether the control sequence is realistic, is with angular momentum. The maximum amount of angular momentum that can be exchanged from $\gamma = 0\,\mathrm{rad}$ to $\gamma = \pi/2\,\mathrm{rad}$ for a scissor-pair of CMGs

$$H_{\text{max}} = 2I_{W_s}\Omega. \tag{17}$$

The remaining amount of angular momentum that can be exchanged is

$$H_{\text{rem}} = 2I_{W_s}\Omega(1 - \sin(\gamma)). \tag{18}$$

The optimal control sequence calculated by the MPC will correspond to an optimal torque sequence $T_{\text{opt}}(k)$, whether the CMG can provide this torque if

$$\sum_{k=0}^{N} T_{\text{opt}}(k)\,dt \leq H_{\text{rem}}, \tag{19}$$

where N is the prediction horizon, k the time step and dt the step size.

Instead of a single linearisation point, Equation 14 can be approximated by multiple linear functions, each active in a certain region. The MLD-MPC control approach can handle multiple linearisation points [54].

Appendix B: Dynamics of the CGB model

The dynamics of the CGB model are expressed in the literature [30]. Definitions of the variables are in subsection II-B, note that clockwise rotation is positive for angles $\vec{\theta}$. The concept of virtual gravity is used to predict the torques in the legs. The gravity vector does not point straight down, but acts under an angle ϕ_i with $i = 1, 2$. The virtual gravitational forces are replaced by a torque in the ankle u_{ank} and the torques in the hip u_{hip} with formulas

$$u_{\text{ank}} = (m_H l + ml + ma)g\cos(\theta_1)\tan(\phi_1) - mbg\cos(\theta_2)\tan(\phi_2) \tag{20}$$

and

$$u_{\text{hip}} = mbg\cos(\theta_2)\tan(\phi_2), \tag{21}$$

respectively [23]. All torques applied on the system are gathered in $\vec{u}_{\text{cgb}}$. The third entry of $\vec{u}_{\text{cgb}}$ is added to represent the torque of the momentum exchange device u, this torque is only applied on the swinging leg, described by θ_2 and defined such that a positive torque results in an increase in step length. The resulting dynamics are

$$\boldsymbol{M}(\vec{\theta})\ddot{\vec{\theta}} + \boldsymbol{C}(\vec{\theta}, \dot{\vec{\theta}})\dot{\vec{\theta}} + \vec{g}(\vec{\theta}) = \boldsymbol{S}\vec{u}_{\text{cgb}}, \tag{22}$$

where

$$\boldsymbol{M}(\vec{\theta}) = \begin{bmatrix} m_H l^2 + ma^2 + ml^2 & -mbl\cos(\theta_1 - \theta_2) \\ -mbl\cos(\theta_1 - \theta_2) & mb^2 \end{bmatrix}, \tag{23}$$

$$\boldsymbol{C}(\vec{\theta}, \dot{\vec{\theta}}) = \begin{bmatrix} 0 & -mbl\sin(\theta_1 - \theta_2)\dot{\theta}_2 \\ mbl\sin(\theta_1 - \theta_2)\dot{\theta}_1 & 0 \end{bmatrix}, \tag{24}$$

$$\vec{g}(\vec{\theta}) = \begin{bmatrix} -(m_H l + ma + ml)\sin\theta_1 \\ mb\sin\theta_2 \end{bmatrix} g \tag{25}$$

and

$$\boldsymbol{S}\vec{u}_{\text{cgb}} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} u_{\text{ank}} \\ u_{\text{hip}} \\ u \end{bmatrix}. \tag{26}$$

For dynamics are linearised and discretised such that they fit the MLD-MPC framework. First the system is written in state space form where state $\vec{x}$ and angles $\vec{\theta}$ are as defined in Equations 2 and 3 respectively. In state space form the derivative of the state

$$\dot{\vec{x}} = \begin{bmatrix} \dot{\vec{\theta}} \\ (\boldsymbol{M}(\vec{\theta}))^{-1}(\boldsymbol{S}\vec{u}_{\text{cgb}} - \boldsymbol{C}(\vec{\theta}, \dot{\vec{\theta}})\dot{\vec{\theta}} - \vec{g}(\vec{\theta})) \end{bmatrix} = \vec{f}(\vec{x}, u). \tag{27}$$

The function can be linearised around the a linearisation point $(x_{\text{lin}}, u_{\text{lin}})$ by the first order Taylor approximation where

$$\dot{\vec{x}} \approx f(\vec{x}_{\text{lin}}, u_{\text{lin}}) + \left.\frac{\partial f(\vec{x}, u)}{\partial \vec{x}}\right|_{\substack{x=\vec{x}_{\text{lin}} \\ u=u_{\text{lin}}}} (\vec{x} - \vec{x}_{\text{lin}}) + \left.\frac{\partial f(\vec{x}, u)}{\partial u}\right|_{\substack{x=\vec{x}_{\text{lin}} \\ u=u_{\text{lin}}}} (u - u_{\text{lin}}) \tag{28}$$

$$= \boldsymbol{A}_c\vec{x} + \boldsymbol{B}_{u,c}u + \boldsymbol{B}_{\text{aff},c} \tag{29}$$

The linearised equations are discretised with a zero order hold.

$$\vec{x}(k+1) \approx \vec{x}(k) + dt\ \dot{\vec{x}}(k) \tag{30}$$

$$\approx \vec{x}(k) + dt\boldsymbol{A}_c\vec{x} + dt\boldsymbol{B}_{u,c}u + dt\boldsymbol{B}_{\text{aff},c} \tag{31}$$

$$= \boldsymbol{A}\vec{x}(k) + \boldsymbol{B}_u\vec{u}(k) + \boldsymbol{B}_{\text{aff}}. \tag{32}$$

With time step k, step size dt and constant matrices $\boldsymbol{A}$, $\boldsymbol{B}_u$ and $\boldsymbol{B}_{\text{aff}}$.

Appendix C: Optimal linearisation point

The nonlinear equations of the CGB walker are linearised for the MLD-MPC (see Equation 8). In the paper two linearisation points are compared, one where $\lambda = 0$ and one where $\lambda = 0.5$. The linearisation where $\lambda = 0$ is used because this does not depend on a prediction of the future and is thus easier to use. The linearisation where $\lambda = 0.5$ is used because this is in the middle of the current state and the expected end state and does thus not favor one point over the other. This appendix attempts to find the optimal choice for λ.

The trajectories predicted by the linearised dynamics are compared with two different references. One set of references are the trajectories of the nonlinear dynamics, the other set of references are from data of the simulation of the 3D NMS model, called the data set. The latter trajectories are the same as the green trajectories in Figure 6. The models are initialised as follows: first a point from the data set is chosen, this is the initial condition. From this initial condition the nonlinear CGB model is simulated. Then, from the same initial condition, the linear CGB model is simulated for a range of λ. The trajectories $\vec{\theta}_1$ and $\vec{\theta}_2$ from the linear models, called $\vec{\theta}_{1,\text{lin}}$ and $\vec{\theta}_{2,\text{lin}}$ respectively, are compared to corresponding references $\vec{\theta}_{1,\text{ref}}$ and $\vec{\theta}_{2,\text{ref}}$. Two tests are done with one with the reference trajectories from the the nonlinear model and one from the data set. The comparison is based on the NRMSE (see Equation 7). The NRMSE of θ_1 and θ_2 are added and the sum is maximised. This is the same as minimizing the cost function

$$J_3 = -\left(\text{NRMSE}(\vec{\theta}_{1,\text{lin}}, \vec{\theta}_{1,\text{ref}}) + \text{NRMSE}(\vec{\theta}_{2,\text{lin}}, \vec{\theta}_{2,\text{ref}})\right) \tag{33}$$

The λ that minimises J_3 is the optimal value λ_{opt}.

This test gives a rough idea of what λ is best to use. The optimisation algorithm is a grid search because only a rough estimation is needed and cost function J_3 is sometimes concave. The range of λ is from 0 to 1 with steps of 0.01. For each λ two values of J_3 are found, corresponding to the two references. The values of λ_{opt} are plotted in Figure 14.

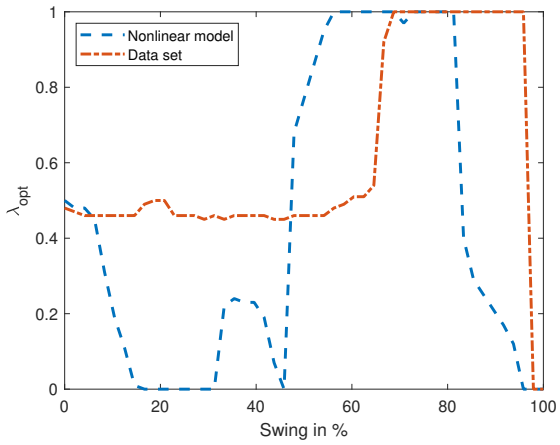

Fig. 14: Optimal value of λ based on cost function J_3. Two different sets of references are used and compared. One is from simulation of the nonlinear CGB model, the other is from the data set obtained by simulating the 3D NMS model.

The results of this test show that $\lambda = 0.5$ is indeed a good choice if the goal is to approximate the nonlinear CGB model because λ_{opt} is approximately 0.5 for most of the swing. However, if the goal is to approximate the data set the optimal value for λ is not clear. The large fluctuations in λ_{opt} are probably due to modelling errors. As mentioned in the paper, the CGB model is unable to accurately model the swing data. Consequently the prediction of the CGB model tends to diverge from the data. Depending on how the linear model diverges, another λ_{opt} is found that best compensates the deviation.

Appendix D: Initialisation and the model of Timmers

In the paper the 2D NMS model from Geyer and Herr and the 3D NMS model from Song are used [24], [25]. As mentioned there, these model use the first generation Simscape library, which is becoming obsolete. Newer versions of MATLAB (2019b+) no longer support this library. Timmers updated the model to the second generation of Simscape with minimal modifications [11], both are able to achieve a steady walking gait with approximately the same parameters. This section shows an interesting difference between the model of Timmers and Song and details attempts to initialise the model of Timmers. The 2D NMS model showed difficulties with initialisation, as discussed in subsection II-E. This model could however be replaced with any of the 3D models. This was not done in the paper because the 3D models are slower than the 2D models and no initialisation solution is found for either of the 3D models. Because the model from Timmers is the most recent and updated to the second generation of Simscape, the attempts to initialise that model are detailed.

Firstly, a difference between the model from Timmers and Song is covered. Interestingly, the two models do show different quite trajectories for $\vec{\theta}$ and $\dot{\vec{\theta}}$, as shown in Figure 15. The trajectories of the figure are obtained from measurements on the ankle using Equations 4 and 5. Especially the angular velocities $\dot{\vec{\theta}}$ are different. Because the models are nearly the same, the difference in trajectories may be quite different from person to person as well, or worse, within the same person but for different swings. The latter is at least not the case in simulation, all swings are approximately the same.

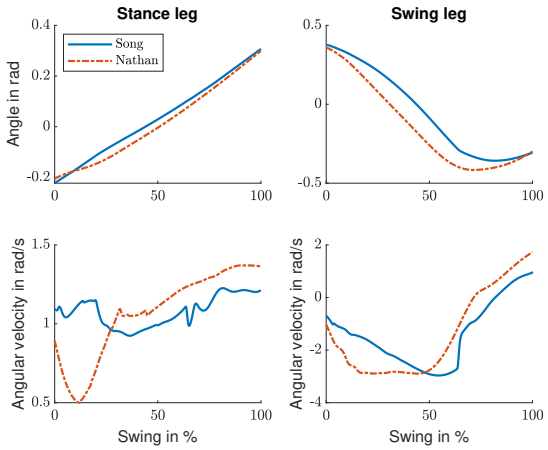

Fig. 15: Comparison of the state trajectories between the model of Timmers and Song.

Secondly, attempts to initialise the model of Timmers are covered as a reference for future research. The model needs to be initialised at a certain state, but many internal signals, like the muscle stimulation signals, are hard to determine and initialise. The idea is to simulate the human walker for some time and save the signals of the simulation. To initialise from a certain state a simulation state close to the desired initial state is chosen and the joint angles and velocities are changed. All other signals, like the muscle stimulation signals, remain the same. Three different features were used in the attempt. The first feature can to save the operating point [36]. This is not possible in the first generation of Simscape, but it is possible in the second generation, like the generation used in the model of Timmers. With this feature the simulation is paused and saved at a certain point, the operating point. The model is able to continue from the saved operating point, but unfortunately it appears to be impossible to change the joint angles and velocities. The second and third feature have another issue, which is the same for both. The second feature uses the Simscape log [55] and the third the 'getInitialState' function [56]. Both are able to save and adjust the joint angles and velocities, but are unable to save all signals. When initiated without all signals the simulated human falls to its knees.

Appendix E: Writing the CGB model as an MLD system

The goal of this section rewrite the dynamics as presented in Equation 9 as an MLD system. HYSDEL is a language that can build MLD systems from coded logic rules [57]. This language works well in combination with the hybrid toolbox for MPC [58]. However, the latter is not suitable for the control law in question, because the software requires a reference trajectory for each time step and can not have a reference for only the end state. The HYSDEL language was not used because defining the MLD by hand gave more insight in debugging the control law.

The state $\vec{x}$ is augmented with the binary state variable

$$x_p = \begin{cases} 1 & \text{during prosthetic swing} \\ 0 & \text{else} \end{cases}. \tag{34}$$

The augmented state

$$\vec{\bar{x}}(k) = \begin{bmatrix} \vec{x}(k) \\ x_p(k) \end{bmatrix}. \tag{35}$$

The system should switch between dynamics when heel strike occurs. At heel strike $\theta_1 + \theta_2 = 0$. However, some steps are taken to ensure that a heel strike occurred. This is done with the by introducing the binary variables $\delta_{\text{imp}}, \delta_{\text{ls}}$ and δ_{sw}. These variables represent whether the swinging foot is on or in the ground (imp from impact), whether the CGB is in late swing (ls) and whether the dynamics of the systems should switch (sw) respectively. First of all, because the system is sampled at certain time steps, it is unlikely that the exact moment of impact where $\theta_1 + \theta_2 = 0$ is sampled. Therefore, it is checked whether the threshold is crossed. The binary variable

$$[\delta_{\text{imp}} = 1] \Leftrightarrow [\theta_1 + \theta_2 \geq 0]. \tag{36}$$

Additionally, the CGB model should be in late swing before the switch occurs, meaning it should have crossed the half way point of the swing. At the half way point both legs of the CGB model are close to the ground because the model has no knees, therefore there is a high risk of a false impact trigger. Therefore, a constant is added to ensure the system does not see the scuffing of the feet as a step at the point where the legs overlap. This is done with the binary variable

$$[\delta_{\text{ls}} = 1] \Leftrightarrow [\theta_1 \geq c], \tag{37}$$

where c is set to 0.001. The binary variable is thus equal to one once the CGB has crossed the half way point of the swing. Finally, the dynamics should switch (sw) if the CGB model is in late swing, an impact has occurred and the system is still in the state prosthetic swing ($x_p = 1$), so

$$[\delta_{\text{sw}} = 1] \Leftrightarrow [\delta_{\text{imp}} + \delta_{\text{ls}} + x_{\text{p}} \geq 2.5]. \tag{38}$$

The value 2.5 is chosen instead of 3 to make sure the condition is triggered if there are small numerical issues.

The dynamics from Equation 9 are written in a single dynamic equation:

$$\vec{\bar{x}}(k+1) = \begin{bmatrix} \vec{x}(k+1) \\ x_p(k+1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{4\times4} & \boldsymbol{B}_{\text{aff}} \\ \vec{0}_{1\times4} & 1 \end{bmatrix} \begin{bmatrix} \vec{x}(k) \\ x_p(k) \end{bmatrix} + \begin{bmatrix} \vec{0}_{4\times1} \\ -1 \end{bmatrix} \delta_{\text{sw}}(k) + \begin{bmatrix} \boldsymbol{A} - \boldsymbol{I}_{4\times4} & \boldsymbol{B}_u \\ \vec{0}_{1\times4} & 0 \end{bmatrix} \begin{bmatrix} \vec{z}_1(k) \\ z_2(k) \end{bmatrix}. \tag{39}$$

Where

$$\vec{z}_1(k) = \vec{x}(k) x_p(k) \tag{40}$$

and

$$z_2(k) = u(k) x_p(k) \tag{41}$$

are real valued auxiliary variables. The auxiliary variables are gathered in the vector

$$\vec{w}(k) = \begin{bmatrix} \vec{z}_1^{\text{T}}(k) & z_2(k) & \delta_{ls}(k) & \delta_{\text{imp}}(k) & \delta_{\text{sw}}(k) \end{bmatrix}^{\text{T}}. \tag{42}$$

Using this vector, the dynamics of the MLD system are rewritten:

$$\vec{\bar{x}}(k+1) = \begin{bmatrix} \boldsymbol{I}_{4\times4} & \boldsymbol{B}_{\text{aff}} \\ \vec{0}_{1\times4} & 1 \end{bmatrix} \begin{bmatrix} \vec{x}(k) \\ x_p(k) \end{bmatrix} + \begin{bmatrix} \boldsymbol{A} - \boldsymbol{I}_{4\times4} & \boldsymbol{B}_u & \vec{0}_{4\times1} & \vec{0}_{4\times1} & \vec{0}_{4\times1} \\ \vec{0}_{1\times4} & 0 & 0 & 0 & -1 \end{bmatrix} \vec{w}(k) \tag{43}$$

$$= \bar{\boldsymbol{A}} \vec{\bar{x}}(k) + \bar{\boldsymbol{B}}_w \vec{w}(k). \tag{44}$$

The controller should reach a certain distance between the feet z. As mentioned in subsection II-A, it is assumed that the positions of the body parts are perfectly known from some observer and $\vec{\theta}$ is thus perfectly computed. In that case the distance between the feet

$$\text{z}(k) = L\sin(\theta_1) + L\sin(-\theta_2) \approx L\theta_1 - L\theta_2 = \begin{bmatrix} L & -L & 0 & 0 & 0 \end{bmatrix} \vec{\bar{x}}(k) = \boldsymbol{C} \vec{\bar{x}}(k). \tag{45}$$

So the dynamics are described by Equation 44 and the controlled variable is estimated by Equation 45. However, the system is also subjected to constraints. First of all binary variables from Equations 36, 37 and 38 introduce constraints to ensure the logic is valid [21]. Constraints appear as follows:

$$[f(x) \leq 0] \Leftrightarrow [\delta = 1] \text{ iff } \begin{cases} f(x) \leq M(1-\delta) \\ f(x) \leq -\varepsilon - (m-\varepsilon)\delta \end{cases} \tag{46}$$

where $M = \max(f(x))$, $m = \min(f(x))$ and $f(x)$ is a function that should be less or equal than zero for the binary variable δ to be equal to one. For example, Equation 36 has the corresponding function $f(x) = -\theta_1 - \theta_2$. The machine precision ε is introduced to make all inequalities non strict. For example, if $f(x) < a$, where a is a constant, then $f(x) + \varepsilon \leq a$.

The minimum and maximum of θ_1 and θ_2 are set to $-\frac{\pi}{2}$ and $\frac{\pi}{2}$ for both variables. Equations 36, 37 and 38 give the following constraints

$$-\theta_1 + \frac{\pi}{2}\delta_{\text{ls}} \leq \frac{\pi}{2} - c, \tag{47a}$$
$$\theta_1 + (-\frac{\pi}{2} - \varepsilon)\delta_{\text{ls}} \leq -\varepsilon + c, \tag{47b}$$
$$-\theta_1 - \theta_2 + \pi\delta_{\text{imp}} \leq \pi, \tag{47c}$$
$$\theta_1 + \theta_2 + (-\pi - \varepsilon)\delta_{\text{imp}} \leq -\varepsilon, \tag{47d}$$
$$-x_p - \delta_{\text{imp}} - \delta ls + 4\delta_{\text{sw}} \leq 1.5, \tag{47e}$$
$$x_p + \delta_{\text{imp}} + \delta_{\text{ls}} - (1+\varepsilon)\delta_{\text{sw}} \leq 2.5 - \varepsilon. \tag{47f}$$

The real valued auxiliary variables from Equations 40 and 41 also introduce constraints. The real valued auxiliary variable z can be enforced equal to the product between a binary variable δ and a real valued variable ξ with the following conditions:

$$z(k) = \delta(k)\xi(k) \text{ iff } \begin{cases} z(k) \leq M_\xi \delta(k) \\ -z(k) \leq -m_\xi \delta(k) \\ z(k) \leq \xi(k) - m_\xi(1-\delta(k)) \\ -z(k) \leq -\xi(k) + M_\xi(1-\delta(t)) \end{cases}, \tag{48}$$

where $m_\xi = \min(\xi)$ and $M_\xi = \max(\xi)$. This leads to the constraints

$$-\vec{M}_x x_p(k) + \vec{z}_1(k) \leq 0, \tag{49a}$$
$$\vec{m}_x x_p(k) - \vec{z}_1(k) \leq 0, \tag{49b}$$
$$-\vec{x}(k) - \vec{m}_x x_p(k) + \vec{z}_1(k) \leq -\vec{m}_x, \tag{49c}$$
$$\vec{x}(k) + \vec{M}_x x_p(k) - \vec{z}_1(k) \leq \vec{M}_x, \tag{49d}$$
$$-M_u x_p(k) + z_2(k) \leq 0, \tag{49e}$$
$$m_u x_p(k) - z_2(k) \leq 0, \tag{49f}$$
$$-u(k) - m_u x_p(k) + z_2(k) \leq -m_u, \tag{49g}$$
$$u(k) + M_u x_p(k) - z_2(k) \leq M_u \tag{49h}$$

All constraints from Equations 47 and 49 can be written as

$$\boldsymbol{E}_x \vec{\vec{x}}(k) + \boldsymbol{E}_u \vec{u}(k) + \boldsymbol{E}_w \vec{w}(k) \leq \vec{E}_{\text{aff}}. \tag{50}$$

Where $\boldsymbol{E}_x, \boldsymbol{E}_u, \boldsymbol{E}_w$ and $\vec{E}_{\text{aff}}$ are constant matrices or vectors. Other constraints can be added as well, for example bounds on the states $\vec{\vec{x}}$ and inputs $\vec{u}$ can be added within the same structure. This is used to constrain the torques of the MED between $0\,\text{N}$ and $5\,\text{N}$. Additionally, the angles $\vec{\theta}$ were constrained between $\pm\frac{\pi}{2}$. This makes sure there are no solutions where a leg does a full rotation around a hinge.

Appendix F: Formulating the MIQP problem

The quadratic cost function from Equation 10 is used for control and should be minimised. This cost function can be written as a mixed integer quadratic programming problem, the equation can be solved by Gurobi. The desired from is

$$\begin{array}{ll} \min_{\vec{V}} & \vec{V}^{\mathrm{T}}\boldsymbol{H}\vec{V} + \vec{f}^{\mathrm{T}}\vec{V} \\ \text{s.t.} & \boldsymbol{A}_{\text{con}}\vec{V} \leq \vec{b}_{\text{con}} \end{array}, \tag{51}$$

where

$$\vec{V} = \begin{bmatrix} u(k) & u(k+1) & \dots & u(N) & \vec{w}^{\mathrm{T}}(k) & \vec{w}^{\mathrm{T}}(k+1) & \dots & \vec{w}^{\mathrm{T}}(N) \end{bmatrix}^{\mathrm{T}} \tag{52}$$

is the vector of variables to be optimised. The goal of this section is to find $\boldsymbol{H}, \boldsymbol{A}_{\text{con}}, \vec{b}_{\text{con}}$ and $\vec{f}$, which are all constant matrices or vectors.

Equation 10 is repeated here for clarity:

$$J_1(\vec{x}(k), \tilde{\vec{u}}, N) = Q_{\mathrm{z}}(\hat{\mathrm{z}}(N) - \mathrm{z}_{\text{ref}})^2 + \tilde{\vec{u}}^{\mathrm{T}}\boldsymbol{Q}_u\tilde{\vec{u}}, \tag{10}$$

The tilde represents a sequence, like control input sequence from Equation 11. The sequence of auxiliary variables $\tilde{\vec{w}}$ and the sequence of states $\tilde{\vec{x}}$ are similarly defined:

$$\tilde{\vec{w}} = \begin{bmatrix} \vec{w}(k) \\ \vdots \\ \vec{w}(N-1) \end{bmatrix}, \tag{53}$$

$$\tilde{\vec{x}} = \begin{bmatrix} \vec{\bar{x}}(k) \\ \vdots \\ \vec{\bar{x}}(N-1) \end{bmatrix}. \tag{54}$$

Note that $\tilde{\vec{x}}$ contains the sequence of augmented states $\vec{\bar{x}}$, the bar is removed for simplicity of the notation. Additionally, the current state $\vec{\bar{x}}(k)$ is assumed to be known, denoted by $\vec{\bar{x}}_k$.

The function J_1 is written as a function of $\vec{V}$. The estimation of the distance between the two feet

$$\hat{\mathrm{z}}(N) = \boldsymbol{C}\hat{\vec{\bar{x}}}(N) \tag{55}$$

is substituted. The hats indicate that the variable is estimated. In this appendix the dynamics from Equation 44 are used. However, the code is made for a more general case, where the dynamics also include an affine term $\vec{B}_{\text{aff}}$ and an input term $\bar{\boldsymbol{B}}_u u$ [22]. However, these terms are not required in this case, therefore the final state

$$\begin{aligned} \hat{\tilde{\vec{\bar{x}}}}(N) &= \bar{\boldsymbol{A}}^N\vec{\bar{x}}_k + \begin{bmatrix} \boldsymbol{A}^{N-1}\bar{\boldsymbol{B}}_w & \dots & \bar{\boldsymbol{A}}\bar{\boldsymbol{B}}_w & \bar{\boldsymbol{B}}_w \end{bmatrix}\tilde{\vec{w}} && (56) \\ &= \boldsymbol{R}_x\vec{\bar{x}}_k + \boldsymbol{R}_w\tilde{\vec{w}}. && (57) \end{aligned}$$

The final state is substituted in the function of J_1, such that

$$\begin{aligned} J_1 &= \boldsymbol{Q}_{\mathrm{z}}(\boldsymbol{C}\boldsymbol{R}_x\vec{\bar{x}}_k + \boldsymbol{C}\boldsymbol{R}_w\tilde{\vec{w}} - \mathrm{z}_{\text{ref}})^2 + \tilde{\vec{u}}^{\mathrm{T}}\boldsymbol{Q}_u\tilde{\vec{u}} && (58) \\ &= \begin{bmatrix} \tilde{\vec{u}}^{\mathrm{T}} & \tilde{\vec{w}}^{\mathrm{T}} \end{bmatrix}\begin{bmatrix} \boldsymbol{Q}_u & 0 \\ 0 & \boldsymbol{R}_w^{\mathrm{T}}\boldsymbol{C}^{\mathrm{T}}Q_{\mathrm{z}}\boldsymbol{C}\boldsymbol{R}_w \end{bmatrix}\begin{bmatrix} \tilde{\vec{u}} \\ \tilde{\vec{w}} \end{bmatrix} + 2\begin{bmatrix} 0 & (\vec{\bar{x}}_k^{\mathrm{T}}\boldsymbol{R}_x^{\mathrm{T}}\boldsymbol{C}^{\mathrm{T}} - \mathrm{z}_{\text{ref}}^{\mathrm{T}})Q_{\mathrm{z}}\boldsymbol{C}\boldsymbol{R}_w \end{bmatrix}\begin{bmatrix} \tilde{\vec{u}} \\ \tilde{\vec{w}} \end{bmatrix} && (59) \\ &\quad + \vec{\bar{x}}_k^{\mathrm{T}}\boldsymbol{R}_x^{\mathrm{T}}\boldsymbol{C}^{\mathrm{T}}Q_{\mathrm{z}}\boldsymbol{C}\boldsymbol{R}_x\vec{\bar{x}}_k + \mathrm{z}_{\text{ref}}^{\mathrm{T}}Q_{\mathrm{z}}\mathrm{z}_{\text{ref}} - 2\vec{\bar{x}}_k^{\mathrm{T}}\boldsymbol{R}_x^{\mathrm{T}}\boldsymbol{C}^{\mathrm{T}}Q_{\mathrm{z}}\mathrm{z}_{\text{ref}} && (60) \\ &= \vec{V}^{\mathrm{T}}\boldsymbol{H}\vec{V} + \vec{f}^{\mathrm{T}}\vec{V} + C && (61) \end{aligned}$$

The constant C is not required for the optimisation problem, because this is no function of $\vec{V}$. The optimal value of $\vec{V}$ is does not depend on C.

The constraints from Equation 50 should hold each time step, such that

$$\begin{bmatrix} \boldsymbol{E}_x & \boldsymbol{0} & \dots & 0 \\ \boldsymbol{0} & \boldsymbol{E}_x & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{E}_x \end{bmatrix}\tilde{\vec{x}} + \begin{bmatrix} \boldsymbol{E}_u & \boldsymbol{0} & \dots & 0 \\ \boldsymbol{0} & \boldsymbol{E}_u & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{E}_u \end{bmatrix}\tilde{\vec{u}} + \begin{bmatrix} \boldsymbol{E}_w & \boldsymbol{0} & \dots & 0 \\ \boldsymbol{0} & \boldsymbol{E}_w & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{E}_w \end{bmatrix}\tilde{\vec{w}} \leq \begin{bmatrix} \vec{E}_{\text{aff}} \\ \vec{E}_{\text{aff}} \\ \vdots \\ \vec{E}_{\text{aff}} \end{bmatrix} \tag{62}$$

$$\boldsymbol{\epsilon}_x\tilde{\vec{x}} + \boldsymbol{\epsilon}_u\tilde{\vec{u}} + \boldsymbol{\epsilon}_w\tilde{\vec{w}} \leq \vec{\epsilon}_{\text{aff}} \tag{63}$$

The sequence of states can be estimated with the dynamics, such that

$$\vec{\bar{x}} = \begin{bmatrix} I \\ \bar{\boldsymbol{A}} \\ \bar{\boldsymbol{A}}^2 \\ \vdots \\ \bar{\boldsymbol{A}}^{N-1} \end{bmatrix} \vec{\bar{x}}_k + \begin{bmatrix} \vec{0} & \vec{0} & \vec{0} & \dots & \vec{0} \\ \bar{\boldsymbol{B}}_w & \vec{0} & \vec{0} & \dots & \vec{0} \\ \bar{\boldsymbol{A}}\bar{\boldsymbol{B}}_w & \bar{\boldsymbol{B}}_w & \vec{0} & \dots & \vec{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{\boldsymbol{A}}^{N-2}\bar{\boldsymbol{B}}_w & \bar{\boldsymbol{A}}^{N-3}\bar{\boldsymbol{B}}_w & \bar{\boldsymbol{A}}^{N-4}\bar{\boldsymbol{B}}_w & \dots & \vec{0} \end{bmatrix} \vec{\bar{w}} \tag{64}$$

$$\vec{\bar{x}} = \boldsymbol{T}_x \vec{\bar{x}}_k + \boldsymbol{T}_w \vec{\bar{w}} \tag{65}$$

By substituting this in Equation 63 the following inequality constraints are obtained:

$$\boldsymbol{\epsilon}_x \boldsymbol{T}_x \vec{\bar{x}}_k + \boldsymbol{\epsilon}_x \boldsymbol{T}_w \vec{\bar{w}} + \boldsymbol{\epsilon}_u \vec{\bar{u}} + \boldsymbol{\epsilon}_w \vec{\bar{w}} \leq \vec{\epsilon}_{\text{aff}}. \tag{66}$$

By rewriting the desired form is found.

$$\begin{bmatrix} \boldsymbol{\epsilon}_u & \boldsymbol{\epsilon}_x \boldsymbol{T}_w + \boldsymbol{\epsilon}_w \end{bmatrix} \begin{bmatrix} \vec{\bar{u}} \\ \vec{\bar{w}} \end{bmatrix} \leq \boldsymbol{\epsilon}_{\text{aff}} - \boldsymbol{\epsilon}_x \boldsymbol{T}_x \vec{\bar{x}}_k \tag{67}$$

$$= \boldsymbol{A}_{\text{con}} \vec{V} \leq \vec{b}_{\text{con}} \tag{68}$$

Appendix G: Quadratic relation J_2 and u_{con}

As described in subsection IV-D, the cost function J_2 (see Equation 12) was evaluated for multiple values of u_{con} to evaluate the behaviour of the model and the sensitivity to local minima. The model was initiated from its default initial state and simulated for three swings. From the start of the third swing a constant external torque u_{con} is applied to the shank of the swinging leg. Upon heel strike the distance from the stance ankle and the swinging ankle z is extracted and used to evaluate J_2. Figure 16 shows the results of this experiment.

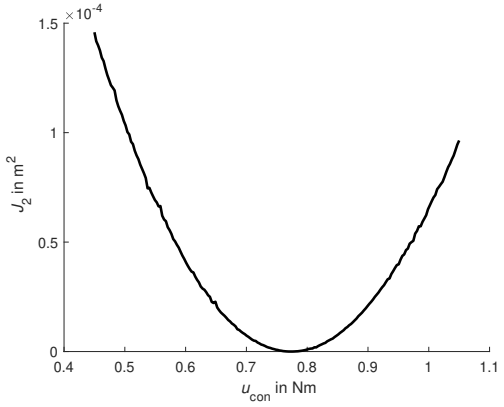

Fig. 16: The relation between cost function J_2 and the constant torque u_{con} when this torque is applied to the shank from toe off to heel strike.

The relation between J_2 and u_{con} is approximately parabolic. However, the curve is not completely smooth, so there are local minima. Because the shape is close to a parabolic, it is concluded that the parabolic interpolation optimisation method is a suitable optimisation method to find the optimal u_{con} that minimises J_2 [44].