École Polytechnique Fédérale de Lausanne

Biorobotics Laboratory

Impedance controlled anthropomorphic exoskeleton model

Internship report

Author:
Chiel Lintzen
TU Delft

Supervisor:
Jesse van den Kieboom
EPFL Biorob

Professor:
Auke Jan Ijspeert
EPFL Biorob

June 17, 2010

Abstract

In this internship report the development of an impedance control method for an anthropomorphic lower extremity exoskeleton is described. This exoskeleton can be used to restore the gait of paralyzed people. Webots simulations are used to test the control method.

Therefore a realistic model of both the human as the exoskeleton has to be made. The lower extremities of the human model have 6 DOF all in the sagital plane. The human model has only passive elements simulating that the human is fully paralyzed. The actuation is done by the exoskeleton. The human model is compared to a Matlab model of the TU Delft. The joint angle trajectories of both models are approximately the same.

The exoskeleton model is anthropomorphic and is assumed to be rigidly attached to the user. The actuation is done directly at the joints, so that all supplied power is converted into joint rotation. The parameters of the impedance control, stiffness, damping and reference trajectory, are phase dependent. They are determined using particle swarm optimization. The optimizations are able to find a gait pattern of the human model wearing the exoskeleton. This gait pattern is compared to the gait of a healthy subject, tracked by the University of Twente. The joint angles of the simulated gait have large similarities with the tracked angles.

The developed model can be used to develop and test more complex control methods. By using a monotone fitness function and an adjusted fitness function the result can further improve.

Acknowledgement

I want to thank professor Ijspeert for hosting me at EPFL Biorob. It is a very good working environment in pleasant ambiance. I experienced the supervision of Jesse van den Kieboom as very good and well accessible. Also the discussions with Renaud Ronsse were very instructive.

Contents

1 Introduction

In this report the results are presented of my internship at the BioRob group of École Polytechnique Fédérale de Lausanne (EPFL). This internship is part of the MSc Biomedical Engineering education program at the TU Delft. During this internship my knowledge of modeling, control methods and optimizations of a human gait model is improved.

The goal of the project is to develop active impedance control for a lower extremity exoskeleton. This control method can eventually be used to restore the human gait. In the best case the human interact with the exoskeleton rather than react to it.

Initially, it is not practical to test the developed control method with a real exoskeleton. Therefore simulations are used to test the control algorithm. The simulation software used is Webots. Webots is a development environment used to model, program and simulate mobile robots. In this case the robot simulated is a human wearing an exoskeleton. First a good model of the human and the exoskeleton has to be made. Then the control algorithm can be tested.

The impedance control demands varying reference position, stiffness and damping of the joints depending on the phase of the human gait. These values cannot be calculated, but are determined by optimizations.

In chapter 2 the developed model is described and validated. In chapter 3 the implementation of the impedance control is presented. The conclusions are presented in chapter 4. Some recommendation for future research are made in chapter 5.

Figure 1: Webots segment model. Segments in green, joints in grey

2 Human model

In this section the development of the human model is described. At the Biorob group there was already a 12 DOFs Webots segment model present, shown in figure 1. The segments length and mass were obtained from a three-dimensional inverse model and log files provided by the University of Twente. The joints in the model are simulated with position controlled motors. The reference angle is set to zero to fix the angle between the segments. There are no dynamics implemented to describe tendon and ligament properties nor there is a ground contact model. For simplicity balance problems are not taken into account in this project, therefore movements are only possible in the sagital plane. Because this model is only used to simulate the human gait, the joints angles of the upper body are fixed and not further modified. Human limbs are not position controlled, but torque controlled. For this reason the position controlled motors of the lower extremities are substituted by torque controlled motors.

2.1 Passive damping and stiffness

Human joints cannot move without resistance and only in a certain range. This is caused by ligaments and the anatomy of the joints. These passive properties need to be present in the model for a more realistic result. The ligaments and tendons introduce passive damping and stiffness, which exert torque on the joints depending on the velocity and the angle.

2.1.1 Method

The damping and stiffness values are joint specific and are obtained from data provided by the University of Twente. The torque introduced by passive elements can be calculated as follows:

$$T_i(t) = T_k(\theta_{act}) + b \cdot \dot{\theta}_{act}(t) \tag{1}$$

$$\begin{aligned} T_k(\theta_{act}) \quad = \quad & -M_{min}(i) * e^{-P_{min}(i)*(\theta_{act}(i)-\theta_{min}(i))} + \\ & M_{max}(i) * e^{-P_{max}(i)*(\theta_{max}(i)-\theta_{act}(i))} \end{aligned} \tag{2}$$

Constant	Hip	Knee	Ankle
M_{min}	2.60	3.10	3.70
M_{max}	8.70	10.50	9.30
P_{min}	5.80	5.90	13.10
P_{max}	1.30	11.80	15.40
b	1.09	3.17	0.46
θ_{min}	-0.38	-1.21	-0.84
θ_{max}	0.74	0.10	0.39

Table 1: Passive damping and stiffness constants, obtained by the University of Twente

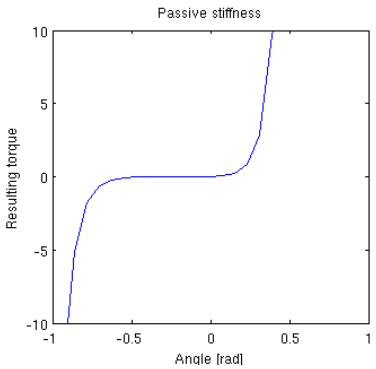

Figure 2: Passive stiffness of the ankle

The damping constant b relates the velocity $\dot{\theta}_{act}(t)$ linear to the applied torque T_i, with i the joint number. The torque added by the stiffness $T_k(\theta_{act})$ is calculated using equation 2, which is angle θ_{act} specific, see figure 2. The torque $T_k(\theta_{act})$ also defines the boundary angles of the joints. In table 2.1.1 the values of the constants can be found. The focus in this project is on the lower extremities, so damping and stiffness are only applied on the hip, knee and ankle.

The modified Webots model is validated by comparing it with the human Matlab model of the TU Delft. The implementation of the passive elements at the lower extremities is tested by performing a passive swing test in both models. Therefore, the model is attached at the back far above the ground so that ground clearance is guaranteed. The initial angle of the left hip is set to $-\pi/4$ and the initial angle of the right hip to $\pi/4$. The legs are released at $t = 0$ seconds.

2.1.2 Results & Discussion

The results of both simulation environments can be found in figure 3. The results confirm that the implementation of the passive damping and stiffness is correct, because the angles of the different models overlap well. Only at the right ankle some difference can be seen. This is caused by a small length difference between the left and right leg in the Matlab model. In the Webots model the legs have the same length.

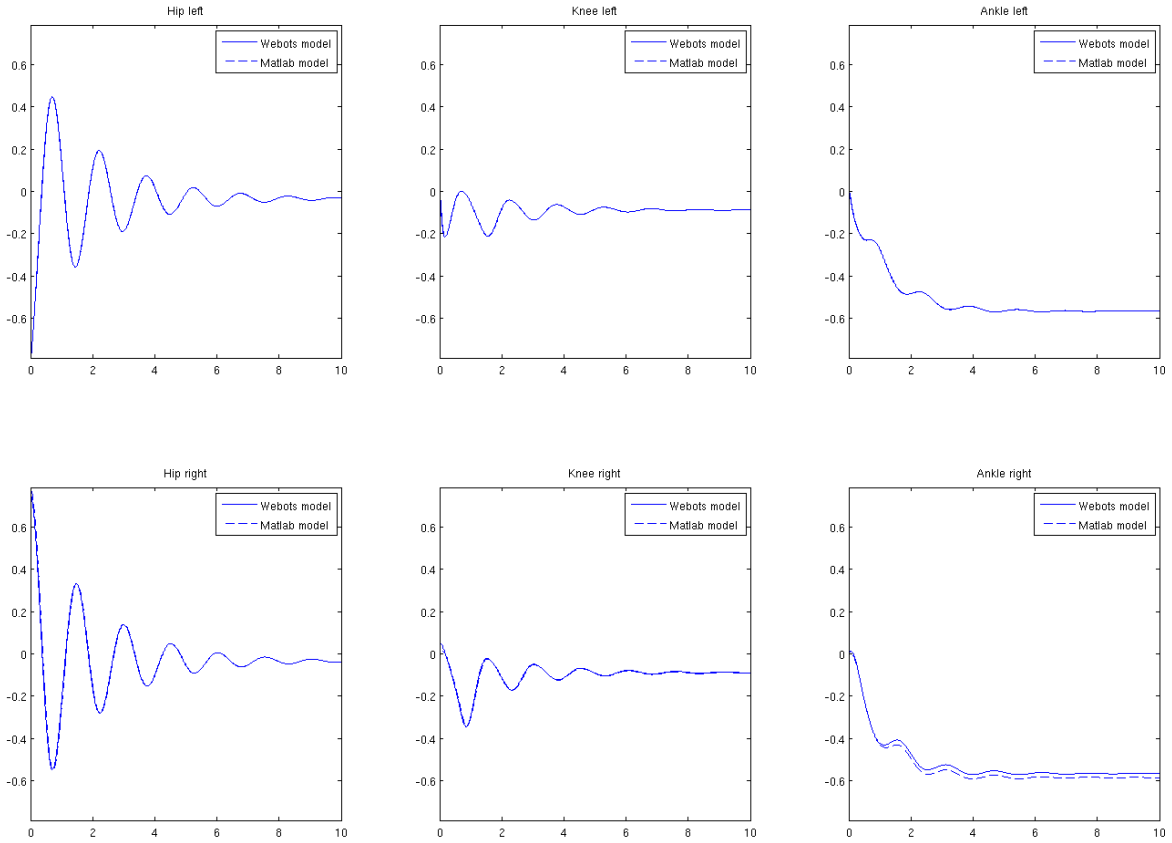

Figure 3: Passive stiffness and damping with initial positions of the hips $\pi/4$ and $-\pi/4$

2.2 Ground contact model

Every gait cycle the feet interact with the ground during toe off and heel strike. It is essential for finding a realistic walking pattern to have a realistic ground contact model. The used ground model is described and tested in this section.

2.2.1 Method

In Webots there is a ground contact model present. It consists of two parts: a normal and a tangential part. The normal part simulates the impact of the foot, the tangential part simulates the slip or friction of the foot.

The Matlab model simulates the ground as a large spring, the force becomes larger when the penetration in the ground is larger. In Webots the reaction of the ground is simulated by inverting the velocity $\dot{y}$ into the ground and by scaling its value:

$$\dot{y} = -\dot{y} \cdot bounce, \text{ with } bounce \text{ } [0..1] \tag{3}$$

In both the Matlab as the Webots model the tangential forces are calculated using Coulomb friction forces. Depending on the normal force F_n there is a tangential friction force F_f applied:

$$F_f = \mu \cdot F_n, \text{ with } \mu \text{ the coulomb friction component} \tag{4}$$

The implementation of the ground contact model is separately tested for the normal and tangential forces. To test the normal forces the model is fixated at the back. Both hip angles were initial set to $\pi/4$. The ground level is set at such height, that the legs impact into the ground in order to test the ground model. To test the tangential forces the ground level was set somewhat lower, so that the legs slide over the ground. The other initial conditions were the same. These two tests were performed with both the Webots as the Matlab model. The *bounce* parameter was set on 0.1 and $\mu = 2$.

2.2.2 Results & Discussion

In figure 4 the results can be found of the normal force simulation of the ground contact model. The trends of the Webots and Matlab models are the same. There is some strange behavior in the beginning of the Webots simulation, this is caused by setting the initial positions of the joints in the controller. When the initial conditions are applied in the Webots world the models calculate the same angles as in the Matlab model. Again some variation can be seen between the left and right leg, caused by the length difference. There are some small deviations in the final angles, which is not thought to be a problem for the simulation of the exoskeleton.

In figure 5 the results can be found of the tangential simulation of the ground contact model. The left leg shows the best results for this test. Because of the leg length difference in the Matlab model the right leg collides at 0.7 seconds. This can best be seen in the angle of the right ankle. The Webots simulation again shows the problem with setting the initial positions. The trends however are the same.

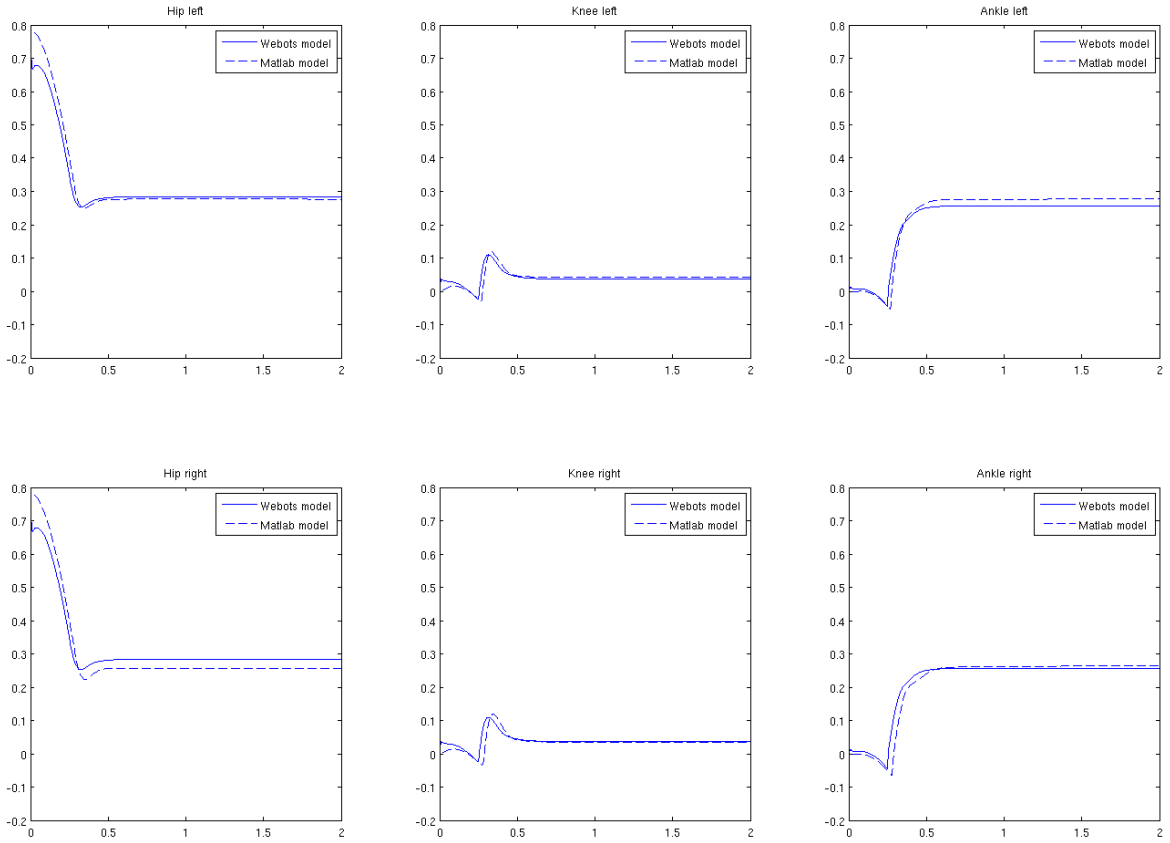

Figure 4: Feet collide with the ground, initial position of the hips $\pi/4$

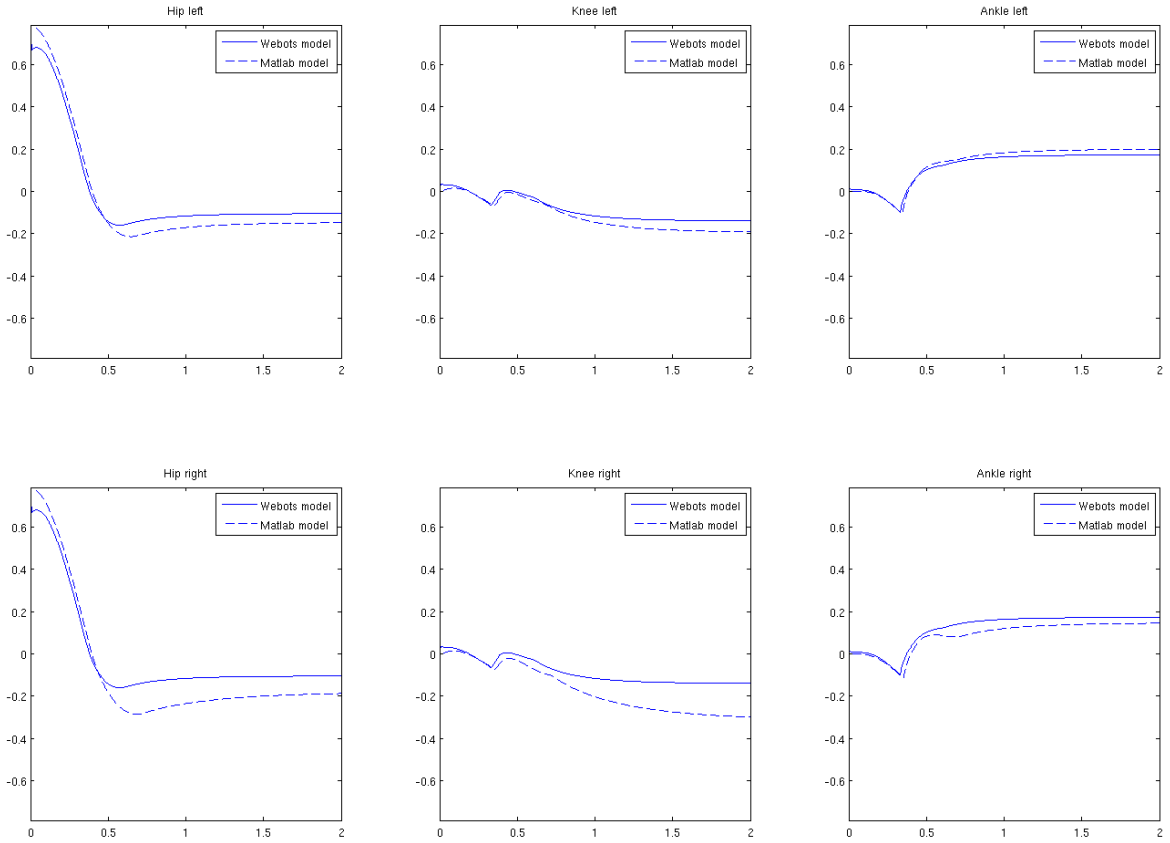

Figure 5: Feet sliding over the ground, initial position of the hips $\pi/4$

3 Exoskeleton model

In this section the exoskeleton model is described and tested. First, all individual components of the exoskeleton are discussed in section 3.1. In section 3.2 the developed model is compared to human gait data.

3.1 Method

The simulated exoskeleton is a massless and anthropomorphic structure, it fits exactly along the joints and the segments of the human body. It is assumed that the exoskeleton is rigidly attached to the user. The actuation is done directly at the joints, so that all supplied power is converted into joint rotation. The exoskeleton is impedance controlled. The values for reference position, stiffness and damping are phase dependent and they are determined by an optimization algorithm combined with polynomial fitting. In the following sections the components are explained one by one.

3.1.1 Impedance control

Robots were initially constructed to perform position tasks. These robots were made very rigid to obtain reasonable positional accuracy by utilizing simple control laws [3]. However, a problem with position-based control is that precise motion tracking requires high output impedance. This is needed in order to dictate a joint trajectory, but it results in a stiff exoskeleton. It forces the user to react to the exoskeleton rather than interact with it [8].

Joint torques however can also be generated by an impedance-based approach. There are several ways to do this. Sup et al. [8] characterize the active knee and ankle behavior with passive spring and damper constants that vary over some finite states. The power is added by switching between the gaits. Pratt et al. [4] use series elastic actuators to actuate the joints. The spring decouples the actuator and the mass of the leg, resulting in low impedance, making it more compliant.

In this project impedance control is used to actuate the joints. The control law used is:

$$T_i = k(t) \cdot (\theta_{act}(t) - \theta_{ref}(t)) + b(t) \cdot (\dot{\theta}_{act}(t) - \dot{\theta}_{ref}(t)) \tag{5}$$

With:
- T_i Torque applied on joint i
- $k(t)$ Stiffness value depending on the time t
- $b(t)$ Damping value depending on the time t
- $\theta_{act}(t)$ Current angle depending on the time t
- $\theta_{ref}(t)$ Reference angle depending on the time t

The phase dependent stiffness, damping and reference angle are determined by polynomial fitting and an optimization algorithm. The time dependent stiffness, damping and reference angle are calculated by the CPG network. The reference velocity is calculated by differentiating the reference position.

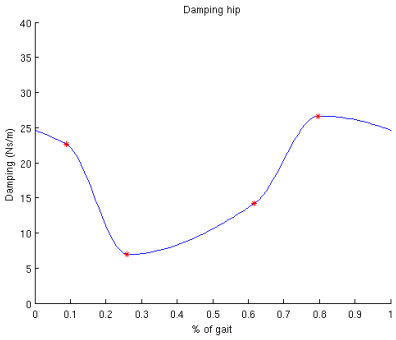

Figure 6: Monotone Hermite interpolating polynomial

3.1.2 Polynomial fitting

The values for the reference trajectory, active stiffness and active damping are found with an optimization algorithm. However these three parameters all depend on the phase and calculating new values for every phase step cause a lot of calculation effort. An optimization algorithm however can also be used in combination with polynomial fitting. The optimization algorithm calculates in this case only 4 points of a walking phase and a curve is fit through those points.

The used polynomial fitting is monotone piecewise cubic Hermite interpolating polynomial (pchip) [2]. The Hermite curves are used to smoothly interpolate between key points, see figure 6. The advantages of this method are that there is no overshoot and less oscillations in comparison to other fitting methods.

The walking gait is a rhythmic movement so the found curves need to be rhythmic as well. Therefore the pchip method is extended so that the start and end point of the found polynomial are the same and have the same derivatives.

3.1.3 CPG network

The control of locomotion is regulated in a lot of animal species with a central pattern generator (CPG). CPGs are biological neural networks that can produce coordinated multidimensional rhythmic signals, under the control of simple input signals. Artificial neural networks are made to better understand the biological neural networks and to profit from the advantages. A model, approximating the human CPG, makes online trajectory generation possible [6, 7].

At Biorob [7] a system was developed of coupled adaptive nonlinear oscillators that can learn arbitrary rhythmic signals in a supervised learning framework. Parameters as intrinsic frequencies, amplitudes and coupling weights can automatically be adjusted to replicate a teaching signal. When the teaching signal is removed the repeating trajectory remains. The system does not need external optimization algorithms, because the learning is embedded into the dynamical system, nor any preprocessing of the teaching signal. The developed system can modulate the speed of locomotion and even allow reversal of the direction.

In this project there are only simple open loop CPG networks used. The idea of implementing the CPG network is that in a next step easily a more

sophisticated network can be integrated. Then more benefit is taken from the properties of CPG networks. In this case each joint has its own open loop CPG network in which the stiffness, damping and reference angle curves are determined. The time dependent curves are found using the following differential equations:

$$\dot{\theta}_{ref}(t) = c \cdot (\theta_{pchip}(\phi) - \theta_{ref}(t)) + \dot{\theta}_{pchip}(\phi) \cdot \dot{\phi} \tag{6}$$

$$\dot{\phi}(t) = \omega + sin(\theta_{ref,left}(t) - \theta_{ref,right}(t) - \phi_{bias}) \tag{7}$$

With:		
	$\theta_{ref}(t)$	Reference trajectory depending on time t
	$\dot{\theta}_{ref}(t)$	First time derivative of $\theta(t)$
	c	Convergence factor
	$\theta_{pchip}(\phi)$	Reference angle depending on phase ϕ
	$\dot{\theta}_{pchip}(\phi)$	First phase derivative of $\theta_{pchip}(\phi)$
	ω	Angular frequency
	ϕ_{bias}	The phase bias, between $\theta_{ref,left}$ and $\theta_{ref,right}$

The damping and stiffness curves are found by substituting θ in equation 6 with respectively b and k. The CPG network is in this case used for smooth switching between the initial straight stance position to the walking movement. The CPG network also maintains the phase difference between the left and right joints at π by the coupling in equation 7. The frequency of the oscillators is equal to ω. The frequency defines the duration of the gait cycle. In this project the frequency is set on 1 Hz.

3.1.4 Optimization

In this section the optimization is discussed. First the optimization algorithm is explained, second the fitness function is presented, third the physical stops are discussed, then the sagital plane fixation is introduced and at last the explode functions are enumerated.

Optimization algorithm The values for the reference position, stiffness and damping, that are used by the polynomial fitting, are found by particle swarm optimization (PSO) [1]. PSO spreads a predefined number of solutions (particles named) in the parameter space and calculates their fitness. The movements of the particles over iterations are guided by the current best particles in the parameter space. The best particle of all particles and iterations is called the global best. The best solution of a particle over all iterations is called the local best. Both global as local best particles are used to calculate the position of the particle in the next iteration. This results in the following equations:

$$x_i(k) = x_i(k-1) + \dot{x}_i(k) \tag{8}$$

$$\dot{x}_i(k) = \dot{x}_i(k-1) + c_1 \cdot R \cdot (x_{i,local} - x_i(k-1)) + c_2 \cdot R \cdot (x_{global} - x_i(k-1)) \tag{9}$$

With:		
	$x_i(k)$	The position of particle i at iteration k
	$x_i(k-1)$	The position of particle i at iteration $k-1$
	$\dot{x}_i(k)$	The velocity of particle i at iteration k

$\dot{x}_i(k-1)$	The velocity of particle i at iteration $k-1$
R	Random number
c_1, c_2	Scaling factor for convergence to local or global best
$x_{i,local}$	Local best of particle i
x_{global}	Global best

The dimension of the parameter space depends on the amount of free parameters. This also determines the dimension of the position vector of each particle. In this case the position, stiffness and damping are interpolated between 4 points, which need both a x and a y coordinate. This results in 24 parameters per joint. Because the parameters are the same for the left and the right leg with a phase difference of π, there are in total 72 parameters.

For this project the number of iterations was set on 300, with a population size of 60 particles. The boundaries for the optimization are parameter specific. The position boundaries are set on θ_{min} and θ_{max}, see table 2.1.1. The stiffness boundaries are set between 0 and 400 N/m and the damping boundaries between 0 and 40 Ns/m.

Fitness function Next the fitness function has to be defined. There are a lot of possibilities, so after testing the following fitness function is used:

$$Fitness = \frac{Time}{Torque_{Active}} \cdot Distance \tag{10}$$

$Time$ is the elapsed simulation time, it gives a boost to the optimization, so that more solutions are found that do not collapse. The maximal simulation time is set on 10 seconds, which makes the $Time$ range between 0 and 10. $Torque_{Active}$ is the active torque integrated over the time, it minimize the total torque needed to make a movement. The human gait is very energy efficient so it is expected that minimizing the total active torque will result in a human like gait. The $Distance$ is added to achieve a certain speed.

Physical stops In section 2.1.1 passive stiffness and damping is implemented at the joints for a more realistic swing behavior. This also defines the joint angle range, because when the joint angle becomes larger the passive force will repulse the joint, see figure 2. However, the exponential curve is only realistic up to certain angles, outside this range the passive torque becomes too high. This makes the walking pattern instable and let the model collapse. It is hard for the optimization algorithm to find a solution if a lot of solutions collapse.

Physical stops are implemented to reduce the chance of overstretching. The stops are placed in the Webots world at: $\theta_{min} - 0.1 < \theta_{act} < \theta_{max} + 0.1$. θ_{act} is the angle joint, θ_{min} and θ_{max} are defined in table 2.1.1.

Sagital plane fixation The upper body of the model is also fixated in the sagital plane to make the optimization easier. With this constrain the body cannot tilt or rotate in the coronal plane. The implementation in the Webots world is done by attaching two fixed sliders at the back. One slider makes it possible to move in the direction of the gait, the other makes it possible to move up and down.

With the introduction of this constrain and the physical stops the model cannot fall on the ground anymore, but it will hang in its physical stops. Therefore, there is chosen to make the back joint a little flexible, by reducing the maximum force of the position controlled motor.

Explode functions The parameter space contains regions that result in physically unrealistic solutions. Calculation effort can be saved by stopping the calculation, when the solutions are out of the physical possible range. This is done with the following explode functions:

- If very high torques are applied, it is possible that the physical stops do not work, resulting in too large angles. Therefore the simulation is terminated, when the joint angle θ_{act} becomes too small or too large in comparison to the minimum angle θ_{min} respectively to the maximum angle θ_{max}, see table 2.1.1: $\theta_{min} - 0.2 < \theta_{act} < \theta_{max} + 0.2$
- Falling or jumping of the model is prevented by terminating the simulation, when the height of the center of mass y_{CoM} of the model becomes too low or too high: $0.4 < y_{CoM} < 1.5$

After the simulation is stopped, the fitness value is still calculated and used in the optimization. The fitness gets however a penalty, because the fitness function takes the simulation time into account, see equation 10.

3.2 Results & Discussion

In this section the simulation results of the human model with exoskeleton are presented and discussed. There is a lot of variation between the results of different optimization runs. The initial positions of the particles have a large influence on the final solution. Therefore the optimization is run several times. The solution with the best fitness value is used to present the results in this section.

3.2.1 Fitness curves

In figure 7 the maximum and mean fitness values are showed over 300 iterations. This is done for the individual parts of the fitness function as well as the total fitness function, equation 10.

The *time* fitness shows a nice result, the maximum *time* fitness is after approximately 30 iterations always at its maximum value and the mean *time* fitness keeps rising.

The maximum *distance* fitness goes up after 40 iterations, so when the *time* fitness is optimized. There is chosen to take the absolute value of the fitness before calculating the mean *distance*. This is done because there is only a slight difference in parameters values for walking forward and backward, resulting in positive or negative fitness values. Otherwise the negative and positive values are cancelled out in the mean fitness. As can be seen in the figure the mean *distance* fitness increases as well.

The *torque* fitness shows some strange behavior. The optimization algorithm decreases the *torque* fitness, apparently it gives more profit to increase the

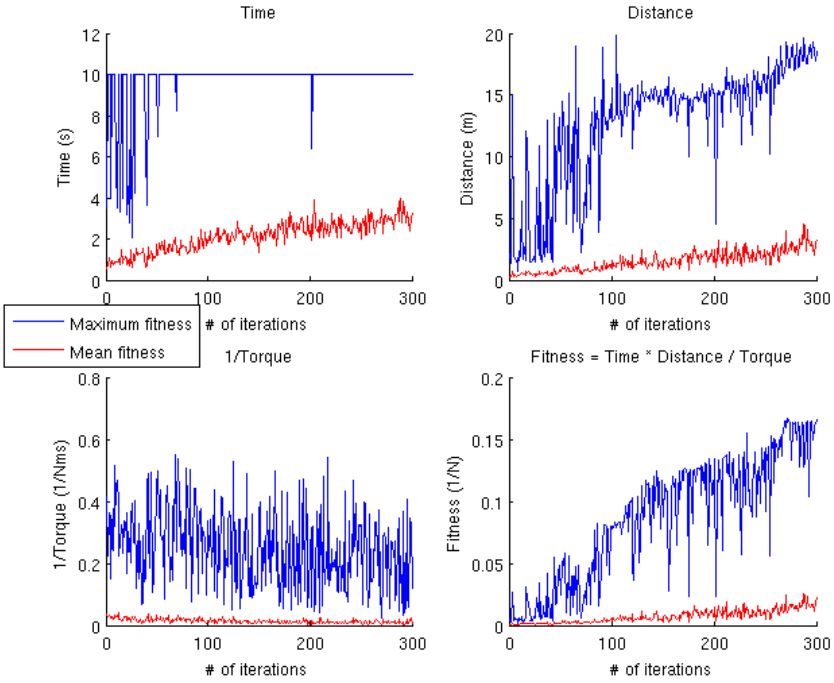

Figure 7: Fitness function over iterations

distance than to decrease the *torque*. This can be changed by increasing the order of the *torque* in equation 10.

The $fitness$ function increases over all iterations, although the curve is very erratic. This is an indication of the difficulty of this optimization problem. A small variation in the parameter values can result in a completely different gait or collapsing body.

3.2.2 Gait cycle

In this section the gait cycle of the maximum fitness is analyzed. The maximum fitness is: 0.166 N^{-1}, with the $Torque$ at 1041 Nms, the *distance* at 17.32 m over a *time* of 10 s. The speed, after switching from stance to walking, is approximately 1.9 m/s. In figure 8 snapshots of the simulated gait of the maximum fitness can be seen. The pictures are taken with an interval of 0.1 seconds.

The gait cycle looks natural, the ankle is used for toe off and the foot is in the right position for the heel strike. The foot clearance is guaranteed, without using the rotation of the hip.

However, the simulated gait is very fast compared to the optimal human walking speed of 1.33 m/s [5]. This results in a model which makes large steps and has a short double stance phase. This could be avoided by putting more emphasis on minimizing the total torque.

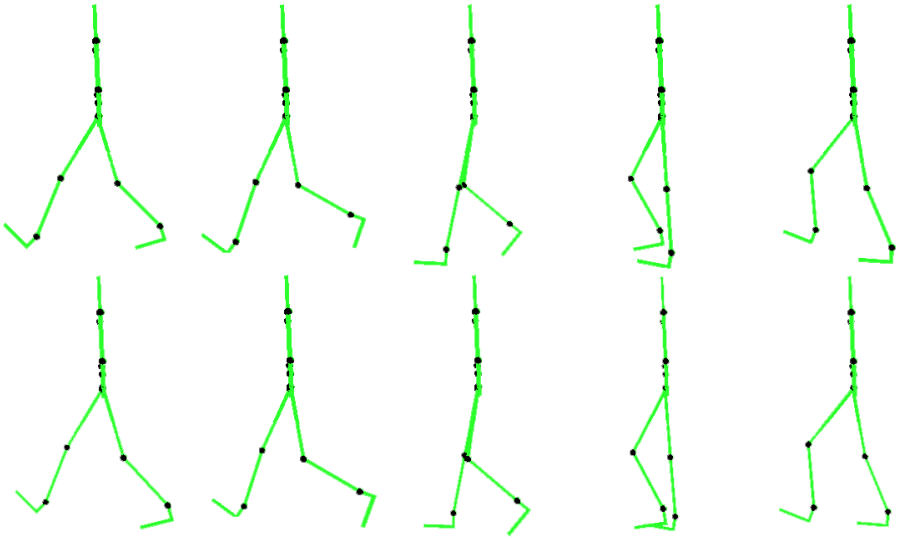

Figure 8: Simulated gait, sample frequency 0.1 s

3.2.3 Simulation curves

In figure 9 simulation data of the left leg can be found from 10 till 13 seconds. In this period 3 gait cycles were completed, the grey area marks the swing phase. In the figure the interpolated reference angle, stiffness and damping of all lower extremity joints are shown in blue. The red crosses are the optimized interpolation points. The actual angle and real recorded human angles from the University of Twente are plotted in the same figure as the reference angle. The Twente data is scaled to fit the speed of the simulated data.

First there has to be noticed that there is an error in the implementation of the monotone piecewise cubic Hermite interpolating polynomial, section 3.1.2. The interpolated curves are not monotone as clearly can be seen in for example the stiffness of the ankle and the damping of the knee. This even results in negative stiffness values for both the hip and ankle. This introduces some strange behavior in the position of the ankle discussed below.

Position The figures of the position of the hip and knee show that the actual joint angle is well attracted by the interpolated reference angle. However, the ankle reference position is only followed well half the gait cycle. This is probably caused by the negative stiffness values of the ankle. The reference trajectory in equation 6 has to become opposite, when the stiffness is negative. In the figure can be seen that when the optimized interpolation point is flipped in the x-axis it lies near the actual angle.

In green the recorded data of Twente is plotted. For the actual angle of the hip and knee the trends are comparable to the Twente data. For the ankle it is again somewhat less clear. The actual angle is extreme if the Twente data is extreme, but in between the amplitude of the actual angle is much larger than the amplitude of the Twente data. This can again be caused by the negative stiffness value or the absence of hip abduction, making the foot clearance more

difficult.

Stiffness It feels logical that during the swing phase the joint stiffness is low to let the leg swing freely. This can clearly be seen at the stiffness of the knee. At the hip and ankle stiffness this is somewhat less clear, however, the stiffness of the hip is at the lowest level during the swing phase and the stiffness of the ankle is decreased rapidly.

At the end of the swing phase the stiffness is expected to be high to prepare for the ground contact. This is not well reflected in the simulation results. The hip and ankle stiffness are instead really low at heel strike. Anyway, based on this figure it cannot be said that the hypothesis is wrong, because of the wrong implementation of pchip and because the active torque is not minimized. The knee stiffness is increasing at the end of the swing phase.

The high stiffness of the joints just before the swing phase is possibly caused by the need of high control gains for the toe off.

Damping The damping of the hip and knee is high just before the swing phase. It is useful to slow the limbs down, because in the swing phase the direction is inverted for both hip and knee. During swing phase the damping is decreased fast to make a movement with low friction. Apparently the hip need damping at the heel strike, where there is no damping needed at the knee.

The damping of the ankle remains more or less the same. Only just after heel strike the damping is increased to prevent oscillations with the ground.

3.2.4 Phase transition

In figure 10 the first 4 seconds of the simulation are shown. In this first seconds the model switch from stance position to walking. After approximately 1 gait cycle the gait becomes steady.

In this figure the soft implementation of the gait pattern can be seen by the CPG network. It results in a smooth movement. It would be interesting to see how other phase transitions, using a more advanced CPG network, will look like.

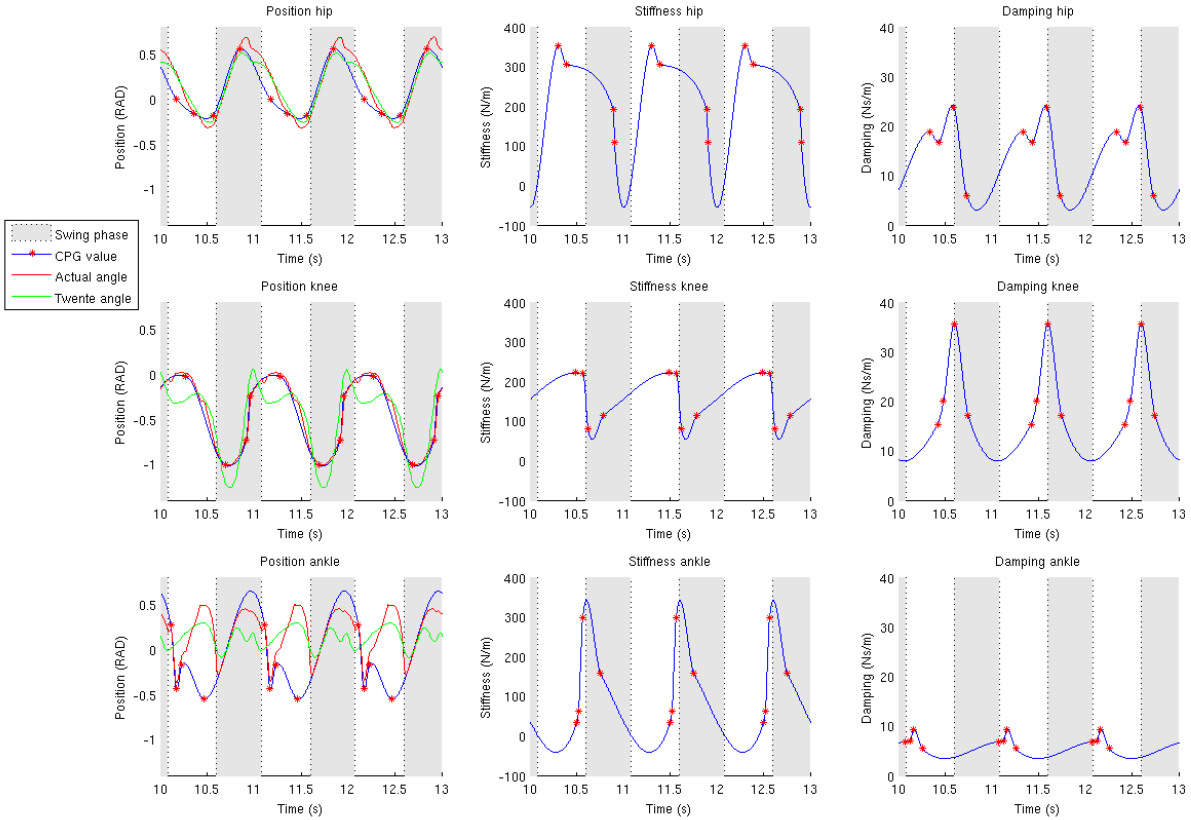

Figure 9: Joint angle, stiffness and damping

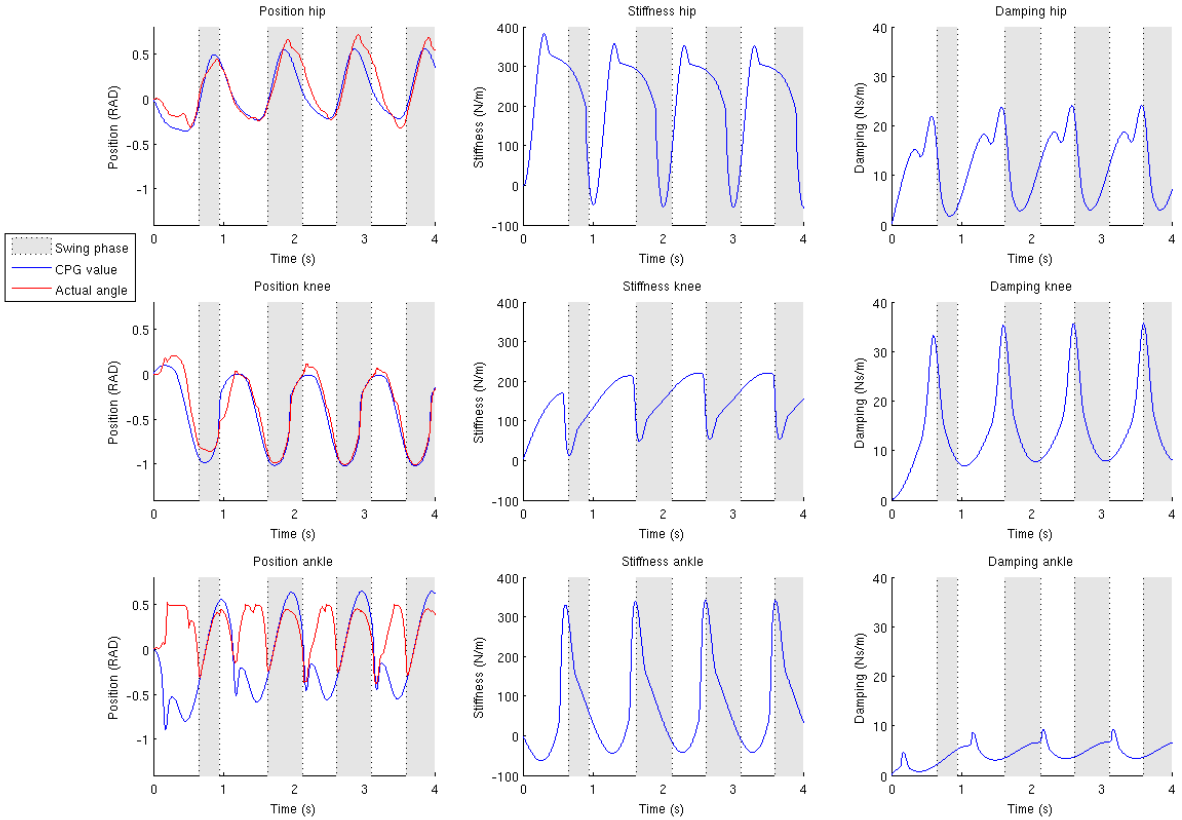

Figure 10: Starting of the simulation: Joint angle, stiffness and damping

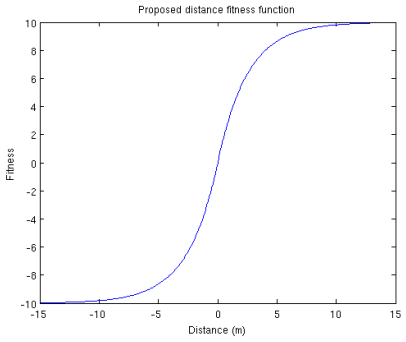

Figure 11: Proposed distance fitness function

4 Conclusion

This project shows that it is possible to actuate an exoskeleton with impedance control. The simulated joint angles show great similarities with the tracked human gait angles. The time dependent stiffness and damping values do not always show the expected values, but this may improve when the polynomial fitting is made monotone. It can be concluded that an optimization problem of 72 parameters is solvable.

The model made in this project can be used to test more sophisticating CPG networks. In this case only an open loop CPG network is used, but when this is extended to a closed loop network, it is for example also possible to adjust the gait pattern to the environment and deal with stability problems.

5 Future Directions

In this section some recommendations are done for future research.

First, the polynomial fitting has to be made monotone. The incorrect fitting of a curve through the 4 optimized points introduces a random effect, which makes the optimization more difficult.

Second, the total active torque is hardly decreased over the optimization runs. Apparently it is more efficient to increase the distance than to decrease the torque. This can be done in two ways: increasing the order of the torque in the fitness function or reshaping the distance fitness function. The proposed distance fitness curve can be found in figure 11. The preferred speed is the optimum speed of 1.33 m/s [5]. This results in an optimum distance of approximately 12 m, with a simulation time of 10 s and taken into account the phase transition between standing and walking. The fitness curve has both positive and negative values, because the variance in the parameters, between walking forward and backward, is really small. Therefore the fitness can also be negative, to make it possible to calculate the derivative of the fitness.

References

[1] R.C. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Proceedings of the sixth international symposium on micro machine and human science*, volume 43. New York, NY, USA: IEEE, 1995.

[2] F.N. Fritsch and R.E. Carlson. Monotone piecewise cubic interpolation. *SIAM Journal on Numerical Analysis*, 17(2):238–246, 1980.

[3] F.L. Lewis, DM Dawson, and C.T. Abdallah. *Robot manipulator control: theory and practice*. CRC, 2004.

[4] J.E. Pratt and B.T. Krupp. Series elastic actuators for legged robots. In *Proceedings of SPIE–The International Society for Optical Engineering*, volume 5422, pages 135–144. Citeseer, 2004.

[5] H.J. Ralston. Energetics of human walking. *Neural control of locomotion*, pages 77–98, 1976.

[6] L. Righetti, J. Buchli, and A.J. Ijspeert. From dynamic hebbian learning for oscillators to adaptive central pattern generators. In *Proceedings of the Third International Symposium on Adaptive Motion in Animals and Machines*, 2005.

[7] L. Righetti and A.J. Ijspeert. Programmable central pattern generators: an application to biped locomotion control. In *Proceedings of the 2006 IEEE international conference on robotics and automation*, pages 1585–1590. Citeseer, 2006.

[8] F. Sup, A. Bohara, and M. Goldfarb. Design and control of a powered transfemoral prosthesis. *The International journal of robotics research*, 27(2):263, 2008.