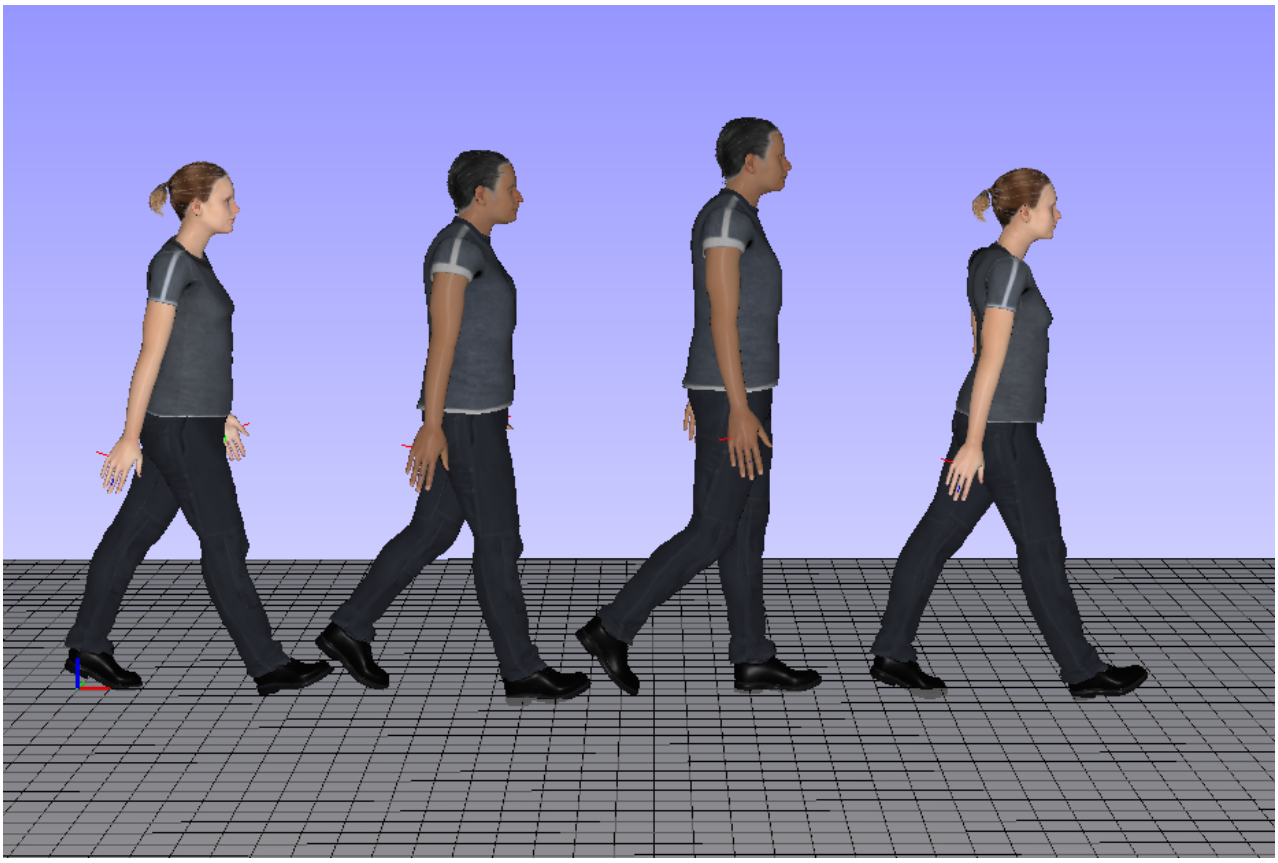
CHALMERS



Walking Generation for Virtual Manikin

Master's thesis in Complex Adaptive Systems

ELIN FÄGERLIND

Department of Mechanics and Maritime Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2018

MASTER'S THESIS IN COMPLEX ADAPTIVE SYSTEMS

Walking Generation for Virtual Manikin

ELIN FÄGERLIND

Department of Mechanics and Maritime Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2018

Walking Generation for Virtual Manikin
ELIN FÄGERLIND

© ELIN FÄGERLIND, 2018

Master's thesis 2018:65
Department of Mechanics and Maritime Sciences
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: +46 (0)31-772 1000

Cover:
Cover shows 4 walking manikins.

Chalmers Reproservice
Göteborg, Sweden 2018

Walking Generation for Virtual Manikin
Master's thesis in Complex Adaptive Systems
ELIN FÄGERLIND
Department of Mechanics and Maritime Sciences
Chalmers University of Technology

Abstract

Generating realistic walking motion for a virtual manikin is a problem that generally relies on motion capture data or advanced balancing models. I propose a simplified control method for generating walking cycles animations without using a physics engine, and without using almost any direct physics simulation, based on inverse kinematics for step guidance. I also show that this method can reproduce realistic walking patterns as well as a a diverse set of steps (side stepping, backwards stepping, etc.) through only two given manikin joint configurations. As the method is very cheap in computational cost as well as data dependency it can serve as an alternative when these are of primary concern.

Keywords: IPS, IMMA, virtual manikin, walking generation

Acknowledgements

I would like to thank Johan S. Carlson and the Fraunhofer-Chalmers Research Centre for Industrial Mathematics for the opportunity to do my Master's Thesis within the IPS project. I would also like to thank my supervisors at FCC, Niclas Delfs and Peter Mårdberg for the support in this project as well as my examinator at Chalmers, Krister Wolff. The data used in this project was obtained from mocap.cs.cmu.edu. The database was created with funding from NSF EIA-0196217.

Contents

Abstract i

Acknowledgements i

Contents iii

1 Introduction and Background 1

1.1 Purpose . . . 1
1.2 Limitations . . . 1
1.3 Background . . . 1
1.4 Related Work . . . 2
1.5 Approaches to Motion Control . . . 2
1.5.1 Joint Space Motion Control . . . 2
1.5.2 Stimulus–Response Network Control . . . 3
1.5.3 Constrained Dynamics Optimization Control . . . 3
1.5.4 Biologically Inspired Control . . . 3

2 Theory 4

2.1 Walking as a Complex System . . . 4
2.2 Framework . . . 4
2.3 Character Animation using Simulated Physics . . . 4
2.3.1 Using Joint Space Motion Control . . . 4

3 Method 5

3.1 Overview . . . 5
3.2 Platform . . . 5
3.3 Walking Summary . . . 5
3.4 Step Algorithm Summary . . . 6
3.5 Current Stance and Goal Step . . . 6
3.6 Inverse Kinematics . . . 6
3.6.1 Levenberg-Marquardt method . . . 7
3.7 PD-Controller . . . 8
3.8 Physics Simulation . . . 8
3.9 Enforced Limitations on Stepping Goals . . . 8
3.9.1 Heron's Formula against Tightrope Walking . . . 9
3.9.2 Forbidden Stepping Zone . . . 9
3.10 Step Types . . . 9
3.10.1 Inital Step . . . 10
3.10.2 Intermediate Step . . . 10
3.10.3 Final Step . . . 10
3.10.4 Feet Corrections . . . 10
3.11 Upper Body Movements . . . 11

3.12 Path Generation 11
3.13 Placement Correction 11
3.14 Parameter Settings 11

4 Results **13**
4.1 Comparison with Motion Capture Data 13
4.1.1 Results from Motion Capture Comparison 16
4.2 Comparison with Literature Data 16
4.3 Performance 18
4.4 Realism 18
4.5 Discussion 18
4.6 Conclusion 18
4.7 Future Work 18

References **19**

1 Introduction and Background

The human body is a highly adaptive complex system, and any effort to mimic it's characteristics has been a hardfought one. Though in the last decades several breakthroughs has been achieved when looking at subsystems of humans such as cognition and locomotion. In this Master's Thesis I have studied some methods for achieving adaptive generative locomotion for a virtual human manikin from both the field of computer simulation and humanoid robotics. The understanding of how you can generate realistic walking motion is of interest in such diverse fields as computer gaming, the movie industry, digital human modeling, robotics and biology.

1.1 Purpose

The purpose of this Master's Thesis is to develop a working generative method for realistic human walking animation. The framework for this development is the Intelligently Moving Manikin (IMMA): a virtual manikin with 162 degrees of freedom which is a part of the Industrial Path Solutions (IPS) program: a virtual room with a flat surface floor and the posibility to add waypoints for the IMMA and obstacles which the IMMA has to circumnavigate. The generated movements will be used with the already developed motion analysis software to achieve a more immersive sofware for human motion evaluation. The goal being that the manikin should be able to walk freely and naturally between workplace points, noting that there already exists tools for animating realistic manikin movements while the manikin stays in one and the same place.

1.2 Limitations

This Master's Thesis is limited around walking generation on flat ground. No running, jumping or climbing motions were therefore implemented.

1.3 Background

For half a century generating walking patterns has been a concern for robotics research, as well as, with the advancement in computer graphics, when simulating humans and animals in computer games, computer animations or computer simulations. In the human body, walking is controlled by a multitude of muscles contracting and relaxing, making the stiff skeleton assume the poses that together with the gravitational pull of the earth and the friction between soles of the feet and the ground comes together to form a walking motion.

In computer graphics, walking has generally been animated, often based off or inspired by motion capture recordings of real human motion. Some methods have been proposed to generate these animations through simulating a physical environment, in which a proposed control framework can realise walking for some virtual manikin living in that environment.

Since these proposed methods always rely on a very simplified model of the human body they could also be seen as efforts to simulate an advanced walking robot, why in this case the two fields of robotics and animation are so closely connected.

In these proposed methods the manikin is often controlled through setting the skeletal joint angles or torques in a similar way in which robotics servo joints are generally used. These has somewhat other characteristics than the muscle controlled joints in the human body, but seem to approximate them well enough.

In this Thesis, I use the IMMA manikin which has 162 joints corresponding to anatomical or approximated anatomical joints in the human body. (For example is the lateral plane ankle joint is placed in the middle of the tibia to better approximate the muscles at work) Out of these I use around 20 which in many humanoid robots have exact correspondants making the comparison with work done in that area highly viable.

Inspired by these simulated physics methods I will propose a simplified way for generating realistic walking animation for a virtual manikin that adapts to any proposed walking path on flat ground.

1.4 Related Work

Extensive study has been made on biped humanlike walking. In robotics the evocative nature of human-likness and the proved efficiency of human walking has fueled the research in this area. These approaches generally strive to generate a balanced walk for often clumsy robots. The effect of using real servos, relatively heavier legs and often fewer degrees of freedom are all aspects that need to be accounted for in the robotic case.

In computer graphics the want and need for generating walking patterns has emerged mainly in the last three decades with the success of computer generated films and videogames.

Though it should be noted that the first computerbased walking pattern genenerator seems to have been implemented as early as in 1968 in the Soviet Union by mathematicians Konstantinov, Minahin and Ponomarenko. They modeled the walking pattern of a cat through a series of differential equations which they then solved using a BESM-4 computer, printing the result as film frames. [rrr74] The resulting film called Koshechka (Kitty) shows the cat walking a short 2D path and turning it's head. [rrr68] While using an approach basically equivalent to todays, this work got very little to no recognition (especially in the West) and seems not to not have influenced later advances in the field. [Wil13] Instead the breakthrough of computer generated animation came with success of the microchip computers of the 70s and 80s and the special effects work made by companies like ILM, Pixar and Disney. In 1982 the New York Instute of Technology screened a trailer for the never completed film "The Works" which tellingly opened with an animation of a walking humanoid robot.

The animation industry carrying a history from from the cartoon era when you had to create the movie frame by frame, seems to have focused on using classical animation [Las87] techniques while simulation were left to deal mainly with more and mored advanced rendering techniques. The video game and animation industry also relies very heavily on motion capture for making realistic movement as this is computationally cheap and leads to realistic motions. [Tha08] Motion capture has also been used since the very early days of animation [TJT95] so it was already a proven technique when CGI came about. Instead the insight that you can model the character motion as a simulated physics problem led researchers to borrow ideas from robotics research and control theory[RH91].

1.5 Approaches to Motion Control

In this section I use Geijtenbeek and Pronost's[GP12] three subpart classification for motion control approaches as it agrees with my impression of the main different areas of research for virtual humanoid motion control. Additionally I consider biologically inspired control which seems to be a growing field within engineering overall.

1.5.1 Joint Space Motion Control

This is the classic control theory approach towards motion control, with origins in servo motor control for controlling humanoid robots. A closed feedback loop is given reference points in joint space. Using classical control theory the joints moves toward this point in simulated real-time. To keep balance or generate walking the reference point in joint space is updated accordingly. This is the approach taken in the articles this thesis work is based upon. In 2007 van de Panne, Loken and Yin proposed a framwork for biped gait control called SIMBICON. [YLP07] This along with the further developments of a similar system; the Generalized Biped Walking Control by van de Panne, Coros, Beaudoin [CBP10] were taken as as a starting point for this thesis with the purpose of partly affirming them as well as exploring possible simplifications.

Basically their work takes continuosly updated speed and direction as an input to the motion controller and generates walking animation on a flat or nearly flat gorund. They the use a balance model that aims to keep the simulated physical manikin balanced, while a PD-controller tries to achieve certain key stance positions. For the simbicon these stances are modeled by the user, while the Generalized Biped Walking Control extends this to also use inverse kinematics to generate goal stances.

For this thesis, I will simplify this algorithm by removing the balance model, while keeping the other parts mentioned above.

1.5.2 Stimulus–Response Network Control

Inspired by biological systems, and with origins in artificial intelligence research this approach uses a generic control network for joint control based on some stimuli. The parameters of the network are set through some off-line optimization procedure to achieve desired behaviour. This has grown immensely in popularity in the last decade with motion controllers proven to achieve all kinds of different locomotion styles, including walking, running and jumping obstacles. In 2017 DeepMind showed emergence of complex movements from just training neural networks to control joints for (among other creatures) a 28-DoF manikin on trying to get as fas as it could on an uneven terrain course. [Nic+17] This approach seems though -while crucial to machine learning and artificial intelligence research- to not efficiently produce realistic motions. A more efficient approach is then to train on real recorded human motions, an approach that has been taken numerous times with impressive results, most recently in DeepMimic.[Pen+18] These endeavours of course demands available data as well as an existing control framework, but could very well serve as a natural continuation of this thesis work.

1.5.3 Constrained Dynamics Optimization Control

This approach seeks to optimize joint acctuator behaviour on-line for high-level onjectives subject to constraints from the equations of motion of the particular model. In 1988 Witkin and Kass[WK88] made the Luxo lamp, famously the star of one of the first succesful CGI films, recreate realistic motions conforming to such principles of traditional animation as anticipation, squash-and-stretch, follow-through, and timing developed in the first part of the twentieth century [TJT95] by trying to minimze muscle energy expenditure between two given still positions. This method has since been further developed working also with human figures. [PW99] The focus on optimization and the demand on computational power, as well as the need for clearly defined goals and constraints made me decide against using this approach.

1.5.4 Biologically Inspired Control

Increased understanding of the mechanisms that control real life has led to several methods that are relevant to this thesis work. Studies that suggest that human walking is atleast partially controlled by central pattern generator [DGP06], that is, neural networks in which neurons fire in an coupled oscillating fashion that can produce a stable walking cycle, has led to developing the same mechanisms for humanoid robots. [SJJ] Likewise the evolutionary process of mutation and gene mixing has inspired researchers to try to evolve model free walking controllers. [WW07] While these certainly produces walking control that works, they struggle with producing natural walking gaits.

2 Theory

2.1 Walking as a Complex System

The problem of getting from point a to point b in 3D-space via human motion is a complex system. We have multiple joints capable of independent movement, an infinite possibility of choosing step direction, orientation, step height, length etc. We can have multiple independent goals as body orientation, posture, path, and so on. This is a key aspect for understanding why generative methods are so elusive. Worth noting is that human babies learn to walk when about 12 months old at around the same time they start to show evidence of word-object relational language skills.[Jen+13][WH99] The recent years advancements in machine learning has also inspired researcher to make programs that learn to walk by themselves sometimes with astounishing resluts.[HKS17] Though unless combined with some kind of very well defined framework or real data such as motion capture this also seem to have a tendency to develop rather silly walks. [Nic+17]

2.2 Framework

To control a human character's walking we must first define what we are controlling. Now there have been examples where researchers have approximated muscle structures for controlling walking gaits, but the general approach is clearly to consider the skeletal joints as servo motors capable of achieving any angle. This is also the approach taken by me. Secondly humans and biped robots live in the real world and are therefore subject to the laws of nature. Gravity, friction, momentum, acceleration, speed, collisions are some of the aspects that should be taken into consideration when simulating realistic behaviour.

2.3 Character Animation using Simulated Physics

Geijtenbeek and Pronost[GP12] lists the following fundamental components to character animation using simulated physics:

- A component performing real-time physics simulation.
- A physics-based character to act in the simulation.
- A motion controller to control a physics-based character.

As the two first points are often dealt with by using third party tools like open dynamics engine or fully developed physics engines like bullet, unity and so on we will touch on them more lightly. They are also quite straight forward as using stiff bodyparts, rotating joints and classical physics approximates realistic human interaction well enough for most animation purposes. For more advanced purposes you would need to consider soft body physics, muscle action, and so on, but that is well beyond the scope of this master's thesis. The focus will be on motion control. In my case a physics-based character was supplied in the form of IMMA and the enviromental physics were developed to be crude but efficient.

2.3.1 Using Joint Space Motion Control

Since a simple motion control framework was desired I choose to work with the joint space motion control as developped by Van de Panne et. al [YLP07][CBP10] The foundation of this method is to consider the manikinin moving through space as a set of still joint configurations. These are connected through PD-control to achieve a smooth motion. It has been shown [YLP07] that as few as four still joint configurations are enough to produce a walk if you also simluate a physics environment with gravity, feet-ground collison and friction. These joint configurations typically include a stance position (foot is in contact with the ground) and one swing position (foot is lifted off the ground). I will use a simlified gravity model, feet-ground collison detection and friction modelled by locking the stance foot to the ground to achieve this. The stance position will be given by inverse kinematics to achieve different steps depending on walking direction and orientation.

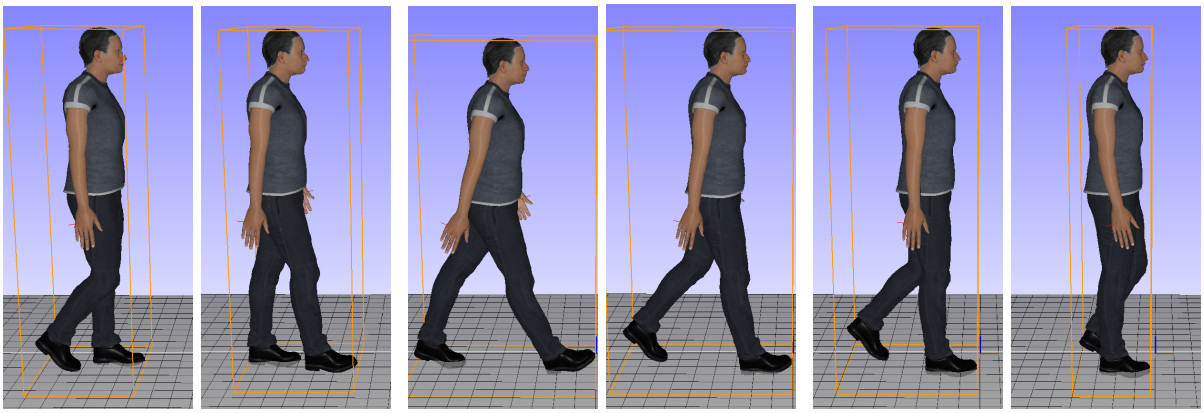


Figure 3.1: Manikin doing a single right foot step.

3 Method

3.1 Overview

The method choosen was based on the Simbicon[YLP07] and the Generalized Biped Walking Control [CBP10]. This was based on simplicity as well as ease on implementation. Since the purpose of the thesis was to develop a working walking generator and the time was limited, other methods was not tried out to any extent but estimated from literature studies. In general these were deemed too reliant on an advanced descriptions of the equations of motion or in the case of machine learning and neural network methods too reliant on a working base framework for control.

3.2 Platform

The project was developped as a add-on dll for the IMMA Manikin add-on package for the Industrial Path Solutions program. Developped in C++ I utilized the Eigen template library for linear algebra and the IMMA library for controlling the IMMA Manikin within IPS. Accessible for setting and getting actions were joint angles and four dimensional transform matrices for each joint, as well as global position and rotation of the Manikin.

3.3 Walking Summary

A walk is given as a series of waypoints on the floor of the simulation room. Each one of these points is given as 4D-transformation matrix describing translation and rotation of the unit vector.

These waypoints can be chosen to make a line segmented walking path around objects. Since each waypoint has an orientation, the manikin can be made to face in a certain direction at the end of the walk. The walk is partitioned in a series of single steps, successively controlled by single step algorithm. Care was taken to achieve each waypoint, such that if a waypoint is closer to the manikin than a standard step length, the manikin will make a shorter step. The orientation of the manikin is controlled by a P-controller making the manikin face the next waypoint.

A threshold distance was set such that if the overall distance of a walk was shorter than a steplength the manikin will start assuming correct end orientation at the beginning of the walk rather than first orienting itself to facing the next waypoint. This to achieve natural backward or sidestepping movements.

3.4 Step Algorithm Summary

The step algorithm I implemented can shortly be described by the following list:

1. → Current Stance
2. → Goal Step
3. Inverse Kinematics: Goal Step → Goal Stance
4. PD-Controller: Current Stance → Goal Stance
5. Gravity: Current Stance → Goal Stance
6. PD-Controller: Goal Stance → Current Stance

Where item 4 and 5 happen semi-simultaneously. In short: Take a current stance and a step goal. Determine swing foot and support foot. Using inverse kinematics, determine how to position the manikins joint angles to achieve the goal position for the swing foor. This is now the goal stance. Then use a discreet PD-controller to achieve this position. The simulated gravity will ensure that the swing foot will touch ground sooner or later. When it does, and the goal stance is approximately correct, use PD-controller to go to a new initial stance for the next step. Gravity is disregarded for this last movement. The algorithm is explained in full detail in this section.

3.5 Current Stance and Goal Step

The current stance is given by the manikin at any time. The swingfoot is then determined by which foot is located higher in the lateral plane. If both feet are at the same height the step is determined to be an initial step and the algorithm is slightly modified. (see section 3.10) The goal step is given as the desired foot placement and orientation of the swing foot of the current step.

3.6 Inverse Kinematics

The inverse kinematic problem is in this case: given a point in space how should joint parameters be set for the manikin to reach that point with one foot while keeping the other foot fixed in position.

For the manikins inverse kinematics the joints affecting the swing foot location are toes(1 Degree of Freedom), ankle (3 DOF), knee(1 DOF), hip(3 DOF), opposing hip(3 DOF), opposing knee(1 DOF), opposing ankle(3 DOF) and opposing toes (1 DOF), all only rotational. To control step orientation I use both the heel and toes of the swingfoot as effectors. This gives a two × three dimensional vector t as a function of 16 rotational angles.

$$\mathbf{t} = \mathbf{f}(\theta) \tag{3.1}$$

We then get the end effector velocity using the Jacobian as:

$$\dot{\mathbf{t}} = \mathbf{J}(\theta)\dot{\theta} \tag{3.2}$$

Since we seek to solve (3.1) for a given t and given that we know our current position and angle vector we can use (3.2) to approximately do this. If

$$\mathbf{t} = \mathbf{t_0} + \boldsymbol{\Delta}\mathbf{t} \tag{3.3}$$

and

$$\boldsymbol{\Delta}\mathbf{t} \approx \mathbf{J}(\theta_\mathbf{0})\boldsymbol{\Delta}\theta \tag{3.4}$$

then we get an approximate solution if we solve:

$$\mathbf{y} = \mathbf{J}(\theta_\mathbf{0})\mathbf{x} \tag{3.5}$$

for the given $\mathbf{y} = \mathbf{t} = \mathbf{t_0} + \boldsymbol{\Delta}\mathbf{t}$.

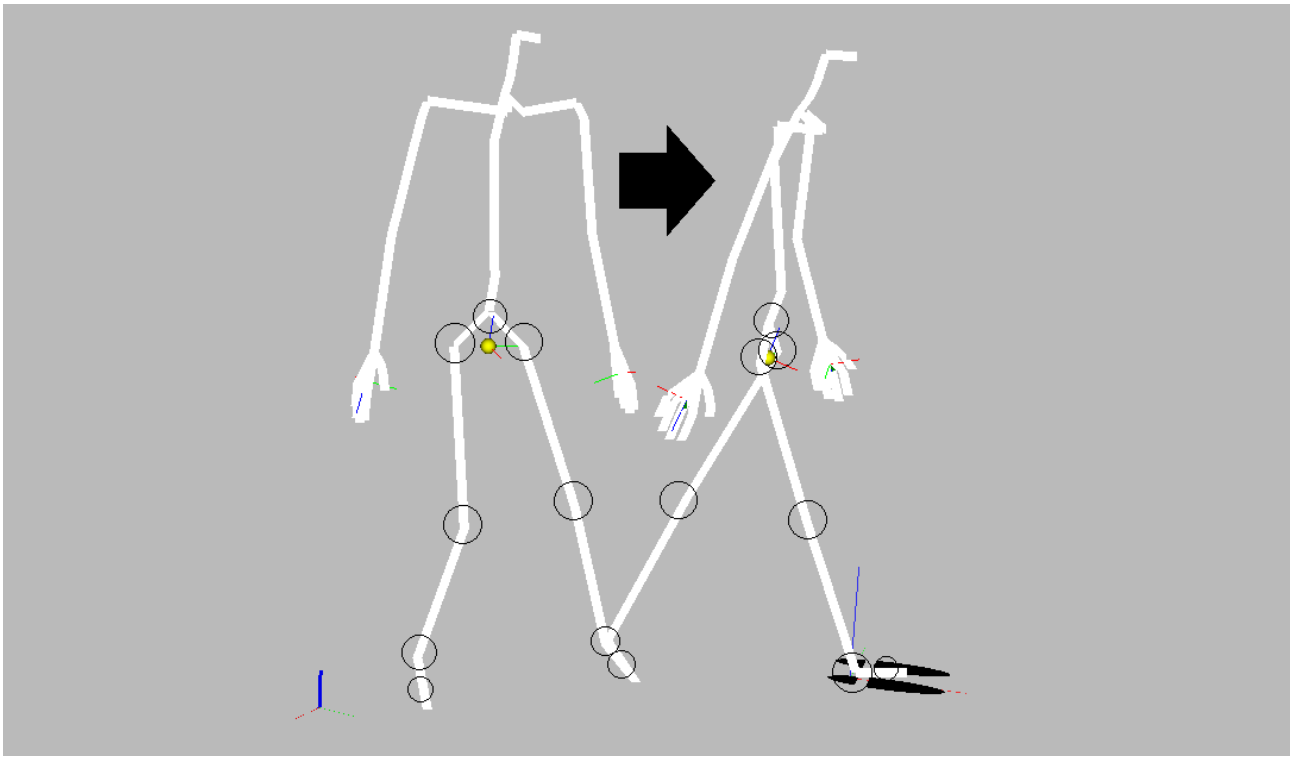


Figure 3.2: Inverse Kinematics problem: How to reach a stepping point through rotation of the joints marked by circles.

3.6.1 Levenberg-Marquardt method

I choose the Levenberg-Marquardt method for solving this equation. There are other methods, of which jacobian transpose was also tried briefly, but due to [Bus04] reported efficiency and stability the Levenberg-Marquardt method was chosen.

The method is to minimize the following quantity:

$$||\mathbf{J}\Delta\theta - \mathbf{y}||^2 + \lambda^2||\Delta\theta||^2 \tag{3.6}$$

where $\lambda \in R$ is a non-zero damping constant. This is equivalent to minimizing:

$$||\begin{pmatrix}\mathbf{J}\\ \lambda\mathbf{I}\end{pmatrix}\Delta\theta - \begin{pmatrix}\mathbf{y}\\ 0\end{pmatrix}|| \tag{3.7}$$

Now (3.7) is minimal if:

$$\begin{pmatrix}\mathbf{J}\\ \lambda\mathbf{I}\end{pmatrix}\Delta\theta = \begin{pmatrix}\mathbf{y}\\ 0\end{pmatrix} \iff \tag{3.8}$$

$$\iff \begin{pmatrix}\mathbf{J}\\ \lambda\mathbf{I}\end{pmatrix}^T\begin{pmatrix}\mathbf{J}\\ \lambda\mathbf{I}\end{pmatrix}\Delta\theta = \begin{pmatrix}\mathbf{J}\\ \lambda\mathbf{I}\end{pmatrix}^T\begin{pmatrix}\mathbf{y}\\ 0\end{pmatrix} \iff$$

$$\iff (\mathbf{J}^T\mathbf{J} + \lambda^2\mathbf{I})\Delta\theta = \mathbf{J}^T\mathbf{y} \iff$$

$$\iff \Delta\theta = (\mathbf{J}^T\mathbf{J} + \lambda^2\mathbf{I})^{-1}\mathbf{J}^T\mathbf{y}$$

Since $\mathbf{J}$ is a 6x16 matrix we'd rather invert:

$$(\mathbf{J}\mathbf{J}^T + \lambda^2\mathbf{I}) \tag{3.9}$$

So we get:

$$\Delta\theta = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T + \lambda^2\mathbf{I})^{-1}\mathbf{y} \tag{3.10}$$

as our solution.

3.7 PD-Controller

A PD-controller was implemented to approximate the muscle work to reach a certain stance. This is done on a joint per joint basis with the possibility to give different joints different controller parameters to achieve a smooth gait cycle. A joint individual acceleration cap was also added to avoid jittery step response.

The PD-controller is given by:

$$f = K_p(x_{ref} - x) + K_d(-\dot{x}) \tag{3.11}$$

where the first term is the proportional control term which if added to the x value will bring it closer to x_{ref} and the second term is the derivative term which will dampen the speed at which x will approach x_{ref}. K_p and K_d are the non-negative coefficients for controlling the behaviour of the controller.

3.8 Physics Simulation

Unlike the models developed in [YLP07] and [CBP10] my model uses a much, much more simplified physics simulation model implemented very much ad hoc. The manikin is basically used as an animation manikin for which the joints could be set in any angle. Gravity is used only for a rotating the manikin around the stance foot until the swing foot touches the ground. The center of gravity is approximated as being in between the hip-joints. When both feet are touching the ground at the same time the simulated gravity effects are temporarily shut off and the body is straightened up through PD control. The gravity is then turned on again at the beginning of the next step. No friction is used but instead the heel or toe point of the stance foot is locked in place while the manikin moves around it. If the rotation of the manikin exceeds a certain value around any horisontal axis, it's deemed that it's losing balance and a new stepping point is proposed by the inverse kinematics to save grace.

Gravitational effects were approximated by first calulating torque around the stance foot by

$$\boldsymbol{\tau} = r \times \mathbf{F} \tag{3.12}$$

where r is the distance between the stance foot ankle or toes and the approximated center of gravity, and $\mathbf{F}$ is gravitational accelaration times a unit mass. Varying the mass of the manikin was tried, but in the end scrapped as I did not see any improvements from doing it. This was then scaled linearly by trial and error to rotational speed disregarding any form of momenta calculations.

3.9 Enforced Limitations on Stepping Goals

As the step goal is free to be anywhere and in any direction around the manikin I had to enforce some limitations to avoid unnatural or too difficult manuveurs for the manikin.

Note that as stepping distance tended to small values it became increasingly difficult to achieve the fine motor skills involved when doing these kinds of steps in real life, while maintaining the simplistic model. When for example taking a step that involved putting one foot just ahead of the other, the gravity from hip displacement was to weak to achive enough rotation in the sagittal plane resulting in an extremely slow step. On top of this the rotation in the frontal plane was centered around the same axis as the swing foot location making the manikin was prone to falling over.

In a similar fashion scissored steps resulted in an smaller displacement of the center of gravity in the frontal plane than regular steps, making these steps slow and awkward too. Furthermore since the model has no self collision awareness they resulted in very awkward looking leg collisions.

Several methods where considered to deal with the problems, including additional body rotation based on stepping intention, as well as trajectory following to avoid leg collisons and ground reaction force manipulation on base ideas from [YLP07].

In the end the simple solution was to avoid these steps altogether. Partly I felt these steps were undesired when walking myslef (and this for the very same reasons of balance and stability) and partly to avoid unforseen consequences of a more complex model.

3.9.1 Heron's Formula against Tightrope Walking

Heron's formula gives the area of the triangel with sides ABC as

$$\text{Area} = \sqrt{(s(s-A)(s-B)(s-C)}$$

where

$$s = \frac{1}{2}(A+B+C)$$

If C is the longest side, this formula gives twice the perpendicular distance between the point AB and the line C. If a goal step had both the heel and the toe of the swinging foot being both less than 5 cm from the line going through the toes and and heel of the supporting foot and extending about a foot's length in both directions, the risk of the manikin loosing balance was found to severly increase. The goal step was in these cases therefore moved 15 cm in the swing foot direction, i.e. to the left or to the right, to achieve a more balanced pose. This measure was done to avoid making the manikin step on it's own feet of losing balance when walking as if on a tightrope.

3.9.2 Forbidden Stepping Zone

To try to keep a natural step and avoid leg-leg interference (the legs are otherwise permitted to pass through each other) a no stepping zone was also added to the left of the left supporting foot and to the right of the right supporting foot as a cone extending from the supporting foot's toes with angle 2/3 π. If the goal step is inside this cone, it's moved to a balancing step next to the supporting foot. This works as a swing foot switch, such that the next step, with opposing swing foot, better reflects the desired target step.

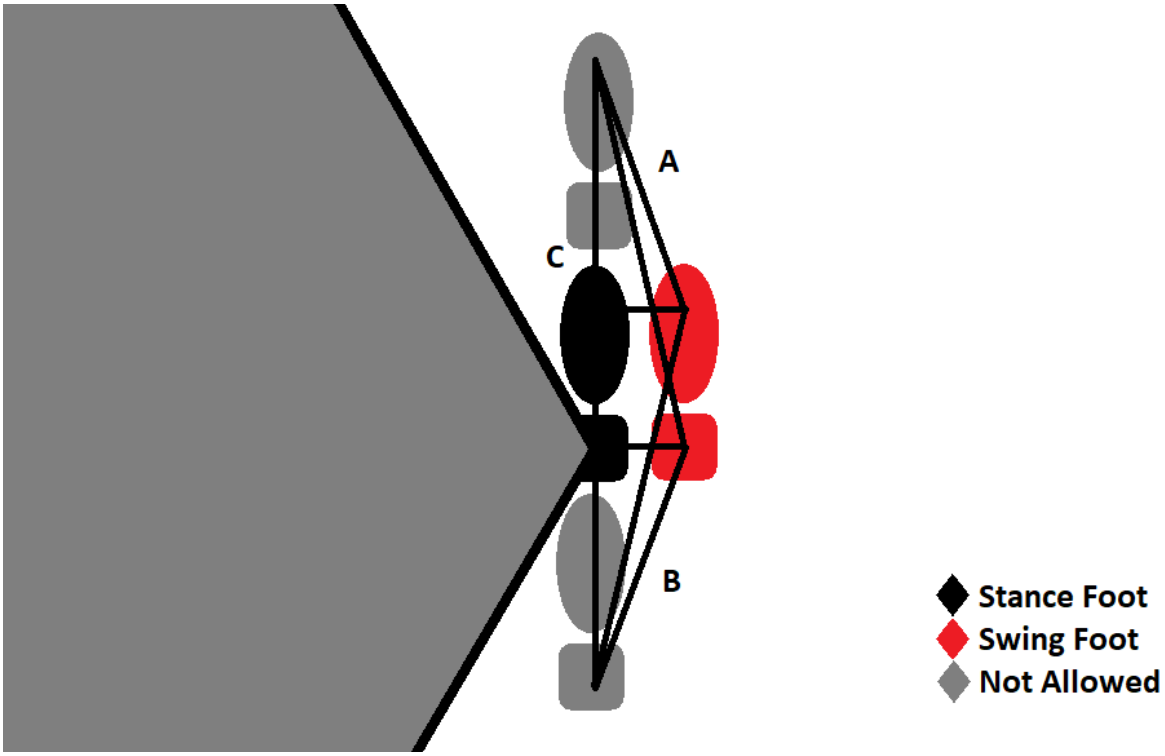


Figure 3.3: Forbidden Steps

3.10 Step Types

All steps in a walk is controlled by the same function createStep. The initial facing direction of the manikin, the distance to the final walk goal and the final facing direction will influence these steps in the following way: if a walk is short the manikin will begin assuming the final facing direction directly and if needed proceed to

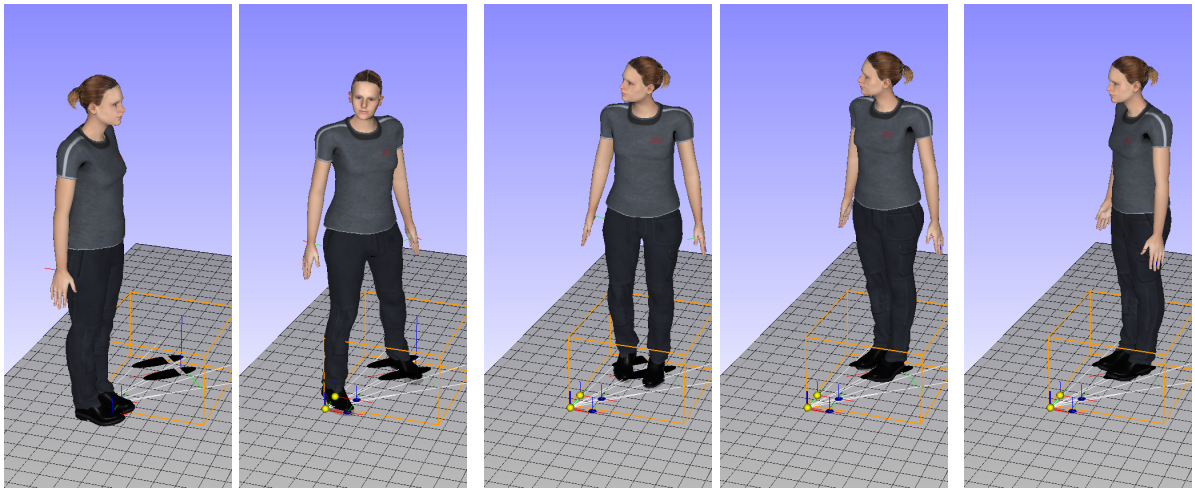


Figure 3.4: Manikin taking a side step and turn.

take a series of backwards steps or side steps to reach it's goal. If a walk is a bit longer, the manikin will first face the next waypoint and move forwards facing it. The goal steps will be determined by this information.

3.10.1 Inital Step

If both feet are close to the ground, createStep assumes the manikin is standing and about to start a walk. The swing foot is decided by in which direction the goal position for the next step is located compared to the initial facing direction. The step is also designated to be an initial step and gravity is temporarily shut of to give the manikin some time to lift the swing foot while not falling over and an intermediate goal stance is added to lift the swing foot a bit of the ground. When this stance is achieved the goal stance reverts to the one given by the inverse kinematics and gravity is reinstated.

3.10.2 Intermediate Step

An intermediate step is defined as going from a midswing stance where one foot is lifted and the manikin is in near balance to a midswing stance where the opposite foot is lifted. The higher foot in the lateral plane as the step begins is decided to be the swing foot. Thus create step needs no direct information about earlier steps in the walking cycle. A significant break is when the swingfoot strikes the ground. The movement is then constrained to keep the placement of the swingfoot while the constraint keeping the stance foot still is lifted. The manikin is then straightend up and assumes the opposite side midswing stance.

3.10.3 Final Step

For the final step, the stepping algorithm is used with a slight modification. Similar to the initial step, final step is decided by when the given goal position is standing with both feet on the ground. To achieve the exact correct end position the manikin is made to partly slide until it has reached that position (Correcting the displacement by about 5-10 cm). It also assumes a standard neutral position.

3.10.4 Feet Corrections

Even though the inverse kinematics works well with the broad movements of the legs, it doesn't give exact solutions. Therefore corrections was added for the rotation of the feet around the tibia axes. Both to keep the stance foot from sliding and to make the swing foot anticipate the step. This was done through a simple P-controller rotating the swing foot towards the final goal orientation and for the stance foot to rotate to keep a fixed orientation between frames.

3.11 Upper Body Movements

The upper body movements where set as linear functions of the leg movements and body torque in the following way: The back joints were set dependent on the body rotation to keep the spine close to a horizontal alignement. The arm joints were set based on the hip rotation as they were percieved to be highly correlated when studying video references of human gaits. This is also consistent with research on arm movement during normal walk. [MBD13] A weak correlation to body rotation was also added to smoothen the movements further. The body rotation and neck joint was set based on the current walking goal by a simple P-controller with the neck rotating at twice the speed of the body and thus anticipating the turn.

3.12 Path Generation

The path for the IMMA is given as a series of waypoint positions in the worldframe. These have direction and position information for the manikin. If the path is straight only the starting point and finishing point were given. If there is an obstruction, a path generator will create a piecewise linear path that takes the manikin around it and a series of waypoints defining this path. Cubic Hermite interpolation was first used to give a smooth walking path but was then discarded as I felt this problem would be better dealt with at the path generator side as my model has no sense of it's surroundings and taking wider turns would thus potentially lead to graphical collisions with the surrounding environment. For a walk the distance was first calculated as the sum of straight line distances between waypoints. The approximated number of steps for the manikin is then calculated as the distance divided by the normal step length (calculated by leg height) rounded upwards. This gives a new normal step length as the distance divided by the number of steps. To ensure exactness or obtain smoother walking, a distance variable decides at which time the manikin should use the next waypoint as walking goal. If this is set to zero the manikin will ensure that it reaches the waypoint by taking a smaller step, if set to the step length the manikin will continue walking straight towards the next waypoint without having to take a smaller intermediating step.

3.13 Placement Correction

Finally to ensure balanced and consistent steps, the stance foot placement in the lateral plane is adjusted throughout the walk by a P-controller to ensure that the stance foot is in contact with the ground. This relaxed constraint sometimes leads to the manikin taking some steps below ground level, but in the end ensures a smooth walking animation.

3.14 Parameter Settings

Since a step is defined from the moment a foot lifts off the ground until it touches down again the walking characteristics can be controlled by the rate of falling over and the controller for the joints. The foot placement can in this context be seen as just the setpoint given by the brain of the human or in our case the inverse kinematics. An effort was therefore made to set the PD-control parameters to achieve motions mimicking motion captured walking cycles. At the same time torque parameters were set to achieve smoothness. That is the torque should result in foot touchdown at about the same time as the PD-controller has reached it's current setpoint.

A timestep of 0.01 s was used for the simulation, and for the PD-Control, a proportional constant $K_p = 0.099$ and a damping constant that varied between joints from $K_d = 0.833\sqrt{K_p}$ to $K_d = 0$ as given below:

Joint	Toe	Ankle	Knee	Hip	Hip Sagittal
K_d	0.26	0.26	0.026	0.26	0
Max. $\ddot{\theta}$	0.05	0.05	0.25	0.05	0.05

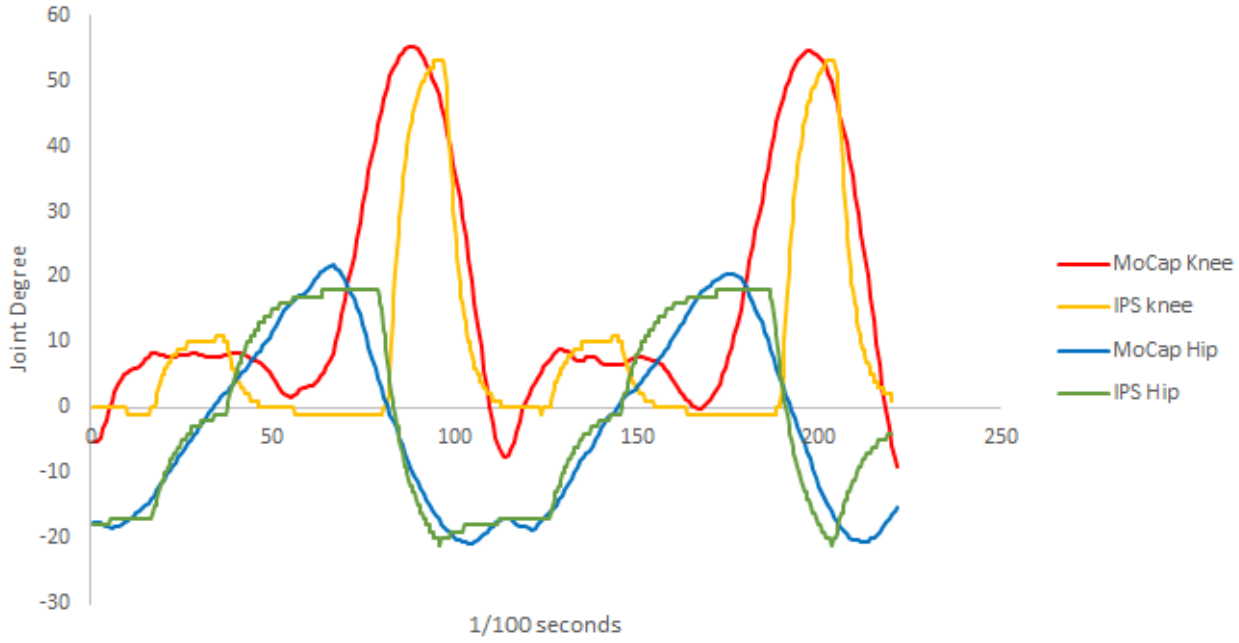


Figure 3.5: Comparison with MoCap 35:13

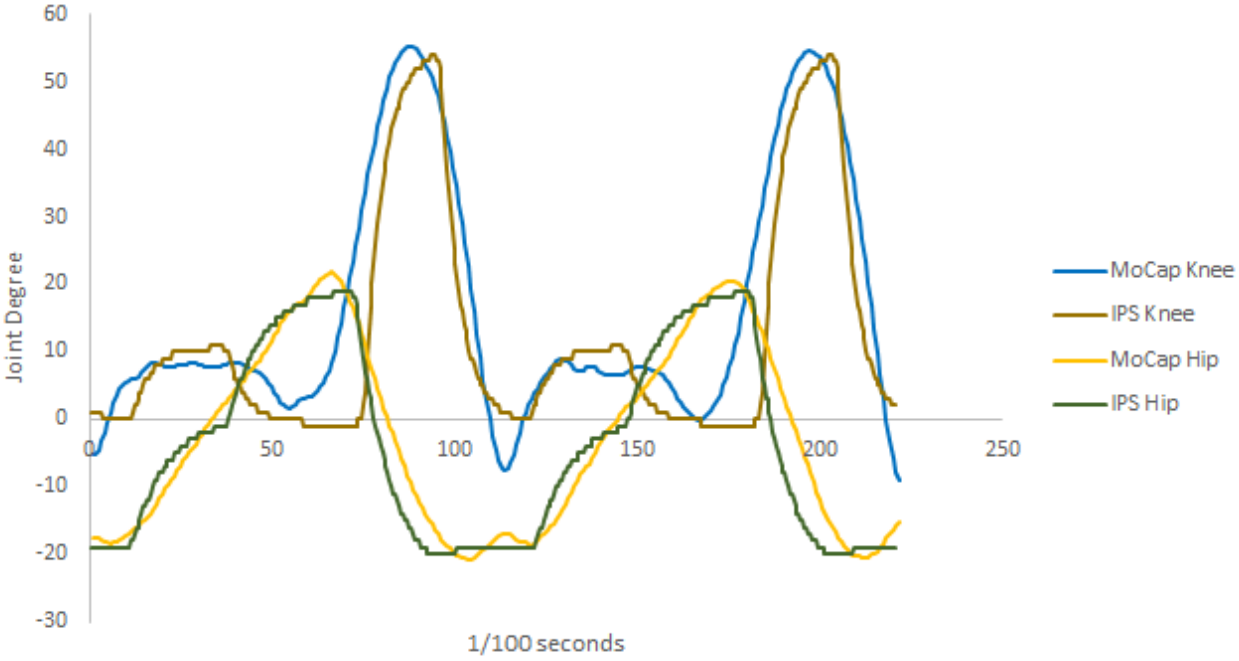


Figure 3.6: Comparison with MoCap 35:13 with torque correction

Figure 3.14 and 3.6 shows how how parameters were adjusted to achieve a realistic gait cycle. The figure shows the result from two early versions of the algorithm were the body torque has been adjusted to achieve an earlier swing foot touchdown in figure 3.6. Note also that the acceleration cap for the PD-controller has been adjusted to achieve smoother movements.

4 Results

4.1 Comparison with Motion Capture Data

The parameters for the PD-controllers were set using comparison with available motion capture data. This data was acquired through Carniege Mellon University's Motion Capture Database.

Motions were chosen from the Locomotion:walking subcategory with a preference towards higher subject numbers in accordance with the recommendations from the site. Motion capture videos of a straight line natural walk from 10 different subjects were then picked. 29,32,35,39,43,45,46,47,49 and 56. These subjects were of unknown sex but their approximated lengths from measured tibia bone lengths spanned from 175 to 193 cm, with a mean length of 187 cm. This calculation was based on given tibia lengths and height estimation as per [DP03]. As the tibia length in the motion capture skeleton is not exactly corresponding to the skeletal bone this height estimation is very rough estimate. The only conclusion drawn is that they were all of adult stature.

The data for joint degrees were taken from the amc-format which stores joint Euler angles in textformat. These were not consistent in zero angle placement, probably due to the motion capture relying on placing white markers on specific parts of the body, a task that should be very difficult to get consistent between indiviudals. As it where, the angular displacement was both interconsistent as well as consistent with other sources. I choose to base my comparison mainly on the right hip joint angle and right knee joint angle as these would be expected to be largest in amplitude and therefore not as vunerable to measurement errors during the motion capture process as well as being the main driving forces of the walk.

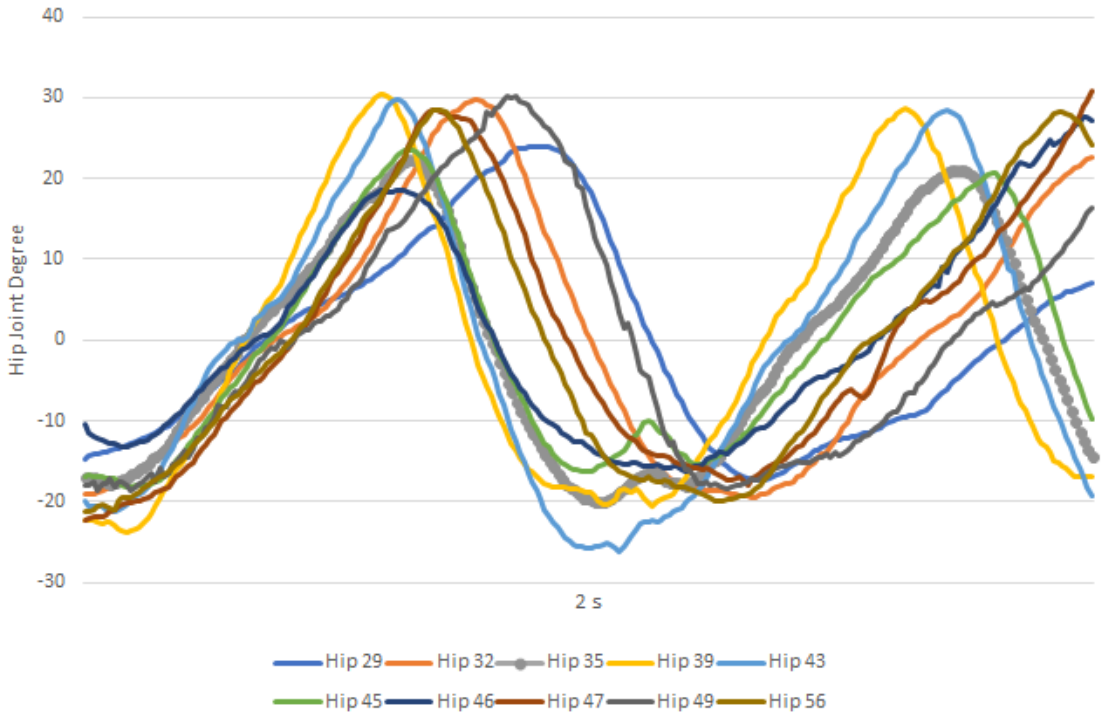


Figure 4.1: 2 seconds of regular walking influence on right hip joint.

As can be seen in figures 4.1-4.2 there is some variance in both frequency and amplitude as would be expected but the general shape and response speed of joint movements can be clearly seen. This data was therefore used as as a guideline for setting the PD-controller and body torque parameters.

As the arm movements from the motion capture data was much more varied but also changing over time within a walk sequence of just a couple of seconds it played a limited role in setting arm movement parameters. The effort was instead put on using video references and trying to mimick what was perceived as natural swinging motion. Still as can be seen in Figure 4.3 some likeness to motion capture was achieved here as well.

The comparison shows that using only two finite state joint poses a credible walking gait cycle was achieved.

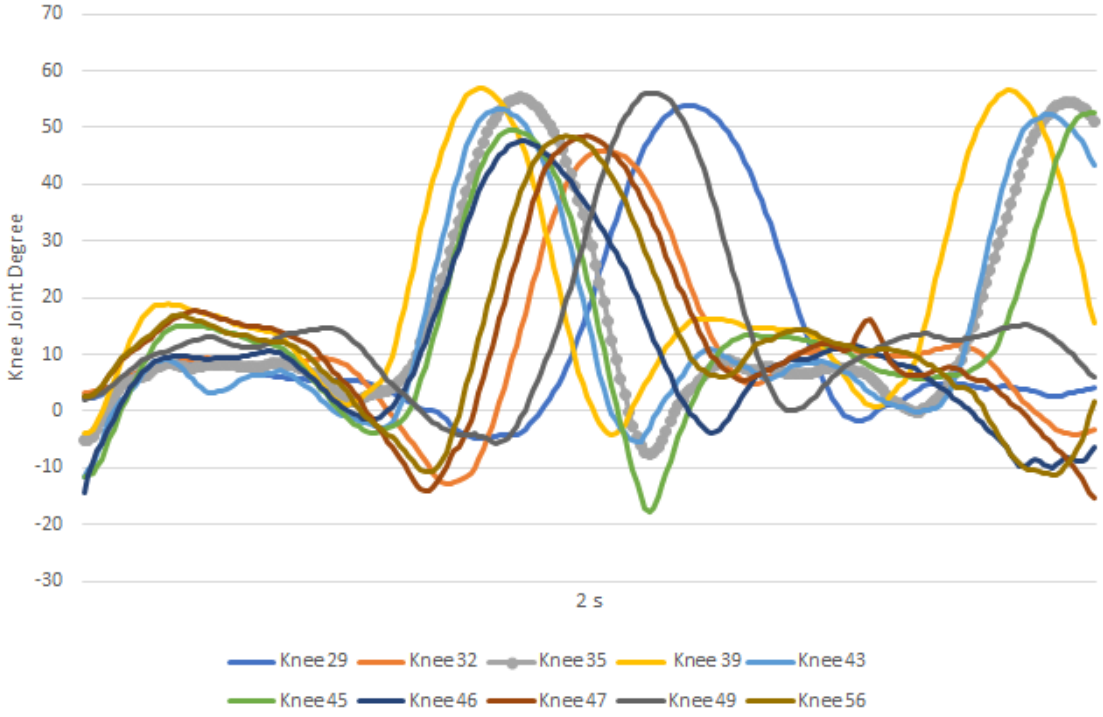


Figure 4.2: 2 seconds of regular walking influence on right knee joint.

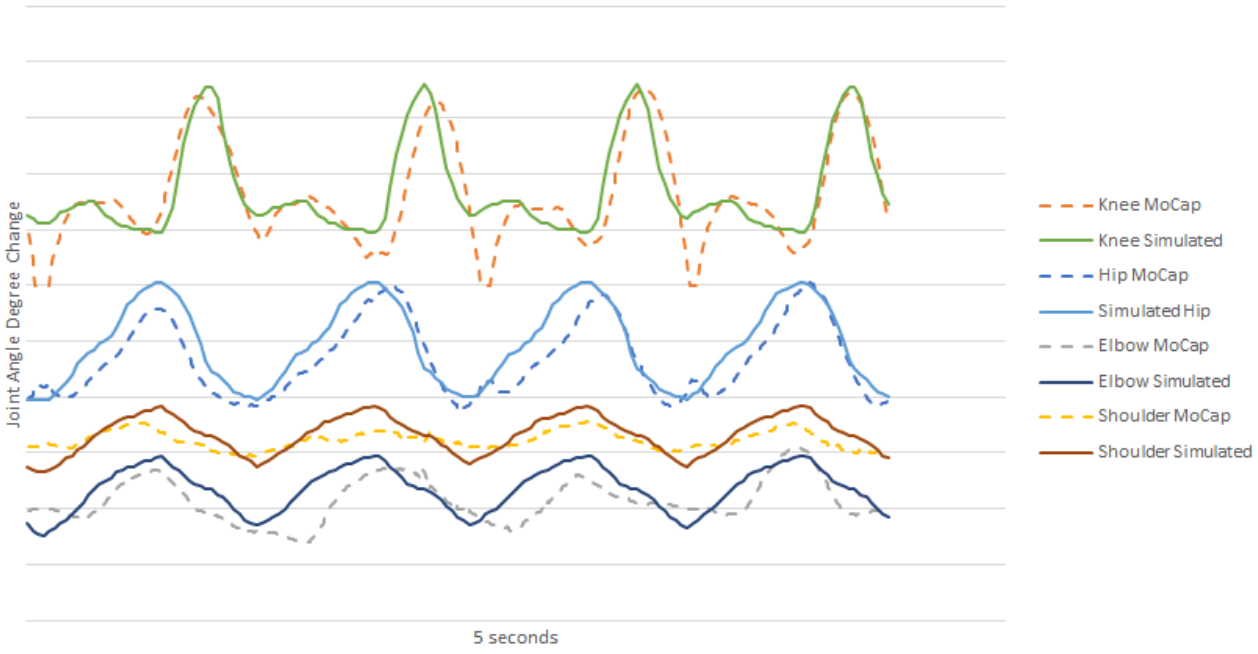


Figure 4.3: Comparison with MoCap 46:01 over 5 seconds of regular staright line walking. Each horizontal bar representing a 20 degree shift.

Movements that required small steps seems to be significantly harder to achieve. Without a collision aware physics simulation or any weight distribution possibilities for the manikin it's hard to pinpoint the finer mechanics involved in taking short adjusting steps. In figure 4.4 we see the result when walking backwards. Instead of achieving a balanced backwards gait, the algortim takes a fast series of small backward steps.

Likewise figure 4.5 shows that the algorithm fails to recreate the nuances in a pivot turn even though the

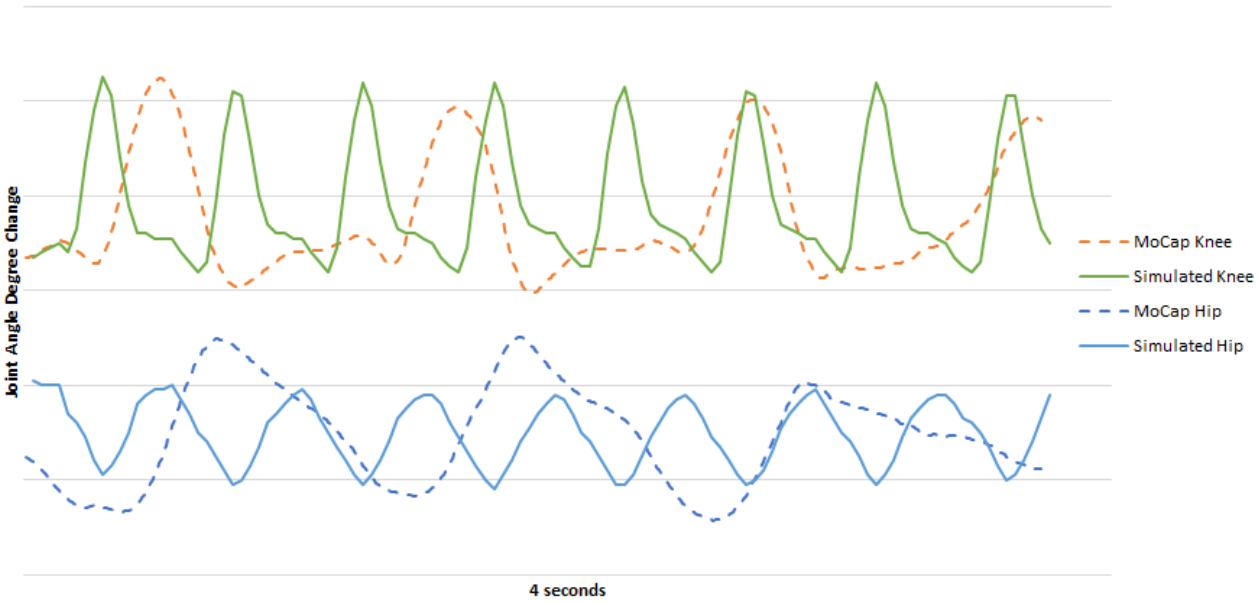


Figure 4.4: Comparison with MoCap 69:41 over 4 seconds of walking straight backwards. Each horizontal bar representing a 20 degree shift.

timing (around 1.5 seconds) is about right. Figure 4.6 also shows some of the problems that arise with the ground contact detection we implemented. When not allowing for this freedom in foot placement severe issues with the stability of the manikin arose instead. That is to ensure stable results from the algorithm independent of the dimensions of the manikin a lot of constraints had to give way. This sometimes lead to a bit akward results like uncomfortably looking fast steps or once in a while an uncommonly slow step.

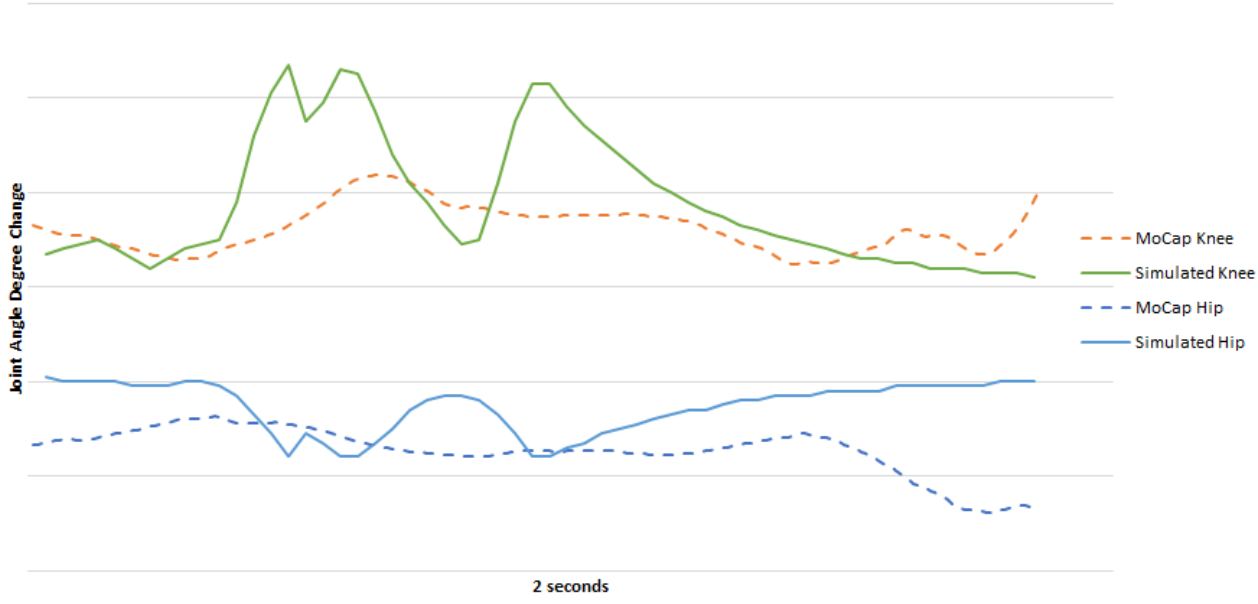


Figure 4.5: Comparison with MoCap 69:41 over 2 seconds of a pivot turn 180 degrees to the left. Each horizontal bar representing a 20 degree shift.

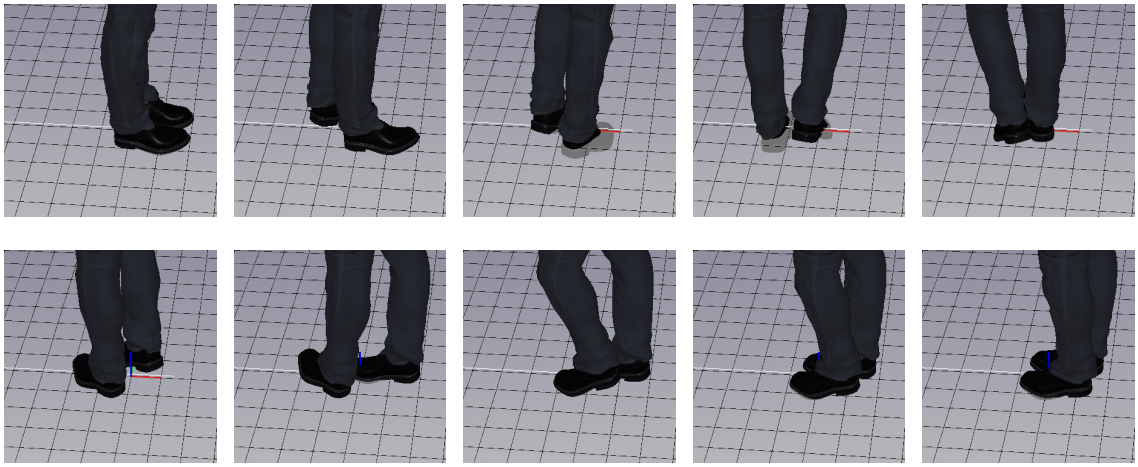


Figure 4.6: Manikin doing a pivot turn.

4.1.1 Results from Motion Capture Comparison

To compare with motion capture data mean hip angle amplitude was calculated from the 10 straight walking sequences to be 49 degrees with a standard deviation of 5.2 degrees for an unknown adult. This is compared with the results from 22 simulations of men and women walking 5 meters straight from the 1, 10, 20, ... , 80, 90, 99:th percentiles of the ANSUR US dataset. The results are shown in the following tabular:

Percentile	1	10	20	30	40	50	60	70	80	90	99	Mean	SD
Men	47	51	49	49	48	48	47	47	46	45	43	47.3	2.1
Women	51	49	49	47	47	47	47	45	45	51	49	48	2.1

Likewise the mean wavelength for the angular motion of the hip of the motion capture data was calculated to be 1.27 seconds with a standard deviation of 0.14 seconds. This compares to simulated values of 1.1 seconds for men regardless of height and weight and and 1.03 for women regardless of height and weight. This is interesting since the algorithm doesn't distinguish between men and women, but could be an effect of exact skeleton joint locations. I note that a comparison between a male and a female of the same height and weight gave the same discrepant results.

4.2 Comparison with Literature Data

To achieve even more realism a comparison with available statistical walking data was made. The mean speed and step length as reported in [CVS97] for adult females and adult males were taken as a guides to further adjust parameters until a set was achieved that simultaneously could recreate approximate free step length and walking velocity for a man and a woman of mean proportions and weight.

The mean length of men participating in [CVS97] was given to be 1.72 m (Mean age: 45.24, weight: 73.7 kg) This corresponds to the 30:th percentile of the ANSUR US men dataset available in the IPS. The mean free steplength and velocity for the men in the study was calculated to be 0.72 m and 1.344 m/s. This compares favourable to our model which achieved a steplength of 0.70 m and a velocity of 1.31 m/s for a single male individual from the 30th percentile of the ANSUR dataset.

The mean length of women participation in [CVS97] was given to be 1.61 m (Mean age: 45.23, weight: 59.6 kg) This corresponds to the 38:th percentile of the ANSUR US women dataset. The mean free steplength and velocity for the women in the study was calculated to be 0.66 m and 1.26 m/s. Using the same parameter values and running instance of the walking model as for the men, the algorithm achieved a steplength of 0.66 m and a velocity of 1.31 m/s for a single female individual from the 38th percentile of the ANSUR dataset.

The joint angles over 4 seconds of these walks are compared with one of the motion capture walks from the the Carnegie Mellon University dataset can be seen in Figure 4.9 showing close similarity. Worth noting is that the cadence is just slightly above 1 step/second with the female having a slightly faster walking pace than the taller man. This corresponds well to the 55-60 steps/minute reported in [CVS97].

It should be noted that another study puts the comfortable speed a bit higher at 1.39 m/s for men and 1.36 m/s for women [Boh97], but as this study didn't report steplength we considered it more comfortable to go with the first. Both also reported a standard deviation of about 0.2 m/s while the second reported that the mean walking speed only decreased with about 0.1 m/s over a 60 year age span.[Boh97] Taken together these studies seem to imply that preferred walking speed is an higly individual property rather than a function of age and height and that women compensate for their shorter stature by a higher cadence.

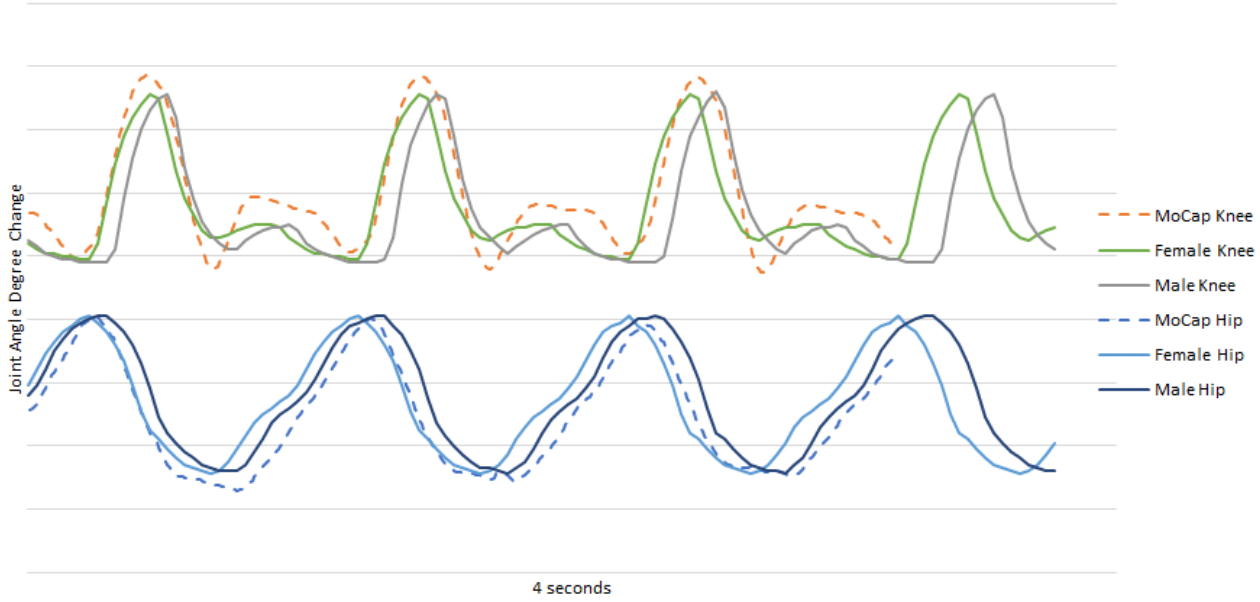


Figure 4.7: Comparison with MoCap 39:14 over 4 seconds of regular straight line walking. Each horizontal bar representing a 20 degree shift.

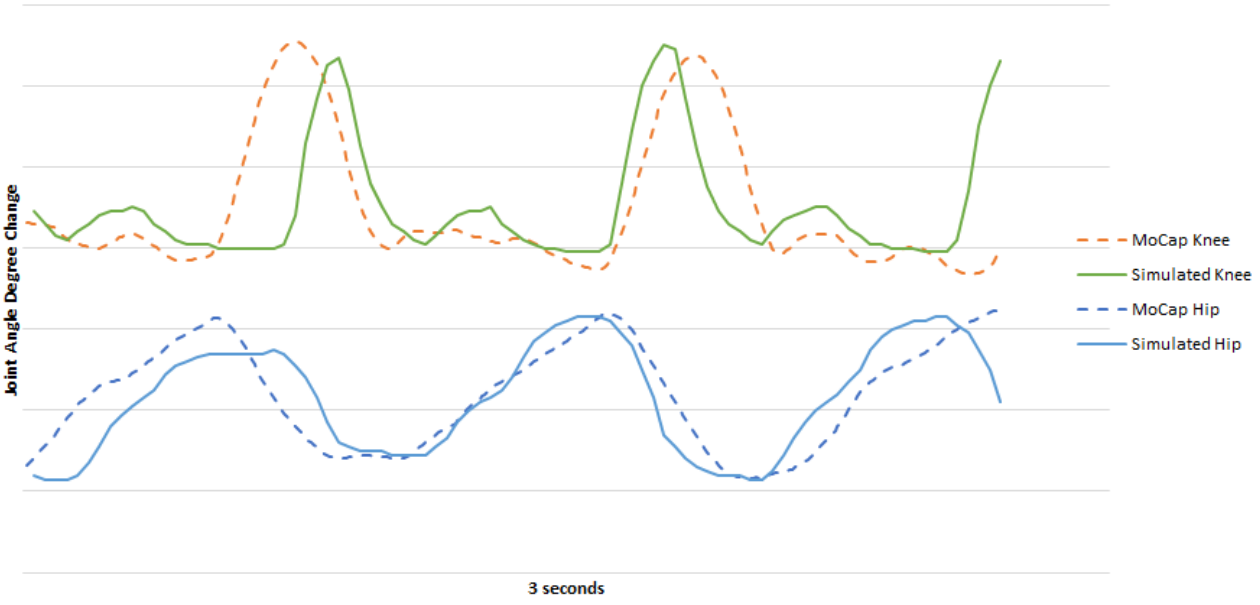


Figure 4.8: Comparison with MoCap 69:32 over 3 seconds of regular walking in a 90 degree left turn. Each horizontal bar representing a 20 degree shift.

4.3 Performance

IPS is a simulation tool in which developers can simulate many manikin persons at one and the same time. This demands a certain efficiency from the parts involved. When tried on a computer with a i-7 4.2 GHz processor it took 11 seconds to generate 10 individual 10 second walks within the IPS program. This gives the non-optimized algorithm almost a 10 $\times$ realtime running speed for a single manikin.

4.4 Realism

The notion of realism in computer simulated imagery should be discussed a bit before I conclude this work. Computer generated animation has in a few decades gone from joy over being able to draw something that looks like a teapot[Leh12] to making real time animated video games that are indistinguishable from watching a filmed sequence. See for example games like Last of Us or this discussion on styles and realism in video games: [Jär02]. In coming years virtual reality promises to make this line between the real and simulated even more blurred. Visual perception and true physical nature are on the other hand not equivalent and should be considered partly separate. This implies that even if I have shown that the simulated manikin reproduces some motions that agrees with data this doesn't make it inherently realistic. I argue that I have shown that the alorithm reproduces an approximation of human walking, but due to the subjectivity of perception and the indivudality of human walking an objective qualitative measure can't really be produced.

4.5 Discussion

The goal of this thesis was to find and evaluate an algorithm for generating realistic walking animation for the IMMA manikin. This goal was partially reached. Realism was reached foremost during straight walking as shown by the data and motion capture analysis. For small step movements the solution sometimes looks awkward, robotic and inefficient but the algorithm seems to always find an accaptable looking set of motions to reach it's goals. One glaring problem is that the algorithm needs to slide the manikin to the last part of the way to reach a correct goal positions. Even if I partly blame this fault on the inherent underactuation of the system I think a way of dealing with this would be to add a more developed physics simulation as in [YLP07] and [CBP10] which seems to be better at handling small steps. I also believe this would make the animation more realistic and believable while also making it usable for more applications. This said I think it is interesting that so much can be accomplished with such a simple model as this one.

4.6 Conclusion

The results I argue shows that an acceptable walking generator can be achieved using extremely simplified physics, simple classical control theory and as little as two predefined finite states. This very efficient approach towards generating complex behaviour I argue could be very useful wherever computational efficiency is of importance and motion capture data scarce. Finally we note that as a generative method it should be able to produce a credible walk regardless of the given input pathway parameters; something it did without a hint of the contrary.

4.7 Future Work

While the algorith works for it's purpose there are several ways to improve it and make it more versatile and robust. I would foremost propose to develop a more advanced physics model, that includes some kind of balance sense for the manikin and a ground reaction force model between the feet and the ground that can be utilized for the more difficult small steps. To better comply with the IMMA usage, which includes optimizing movement efficiency under some ergonomic rule based constraints, it would also be of interest to develop this method further using a constrained optimization model. The method used is also so simple that I think it could be easily adapted for neural network control, seemingly recent years most promising method for generating versatile walking motions. [HKS17] [Pen+18]

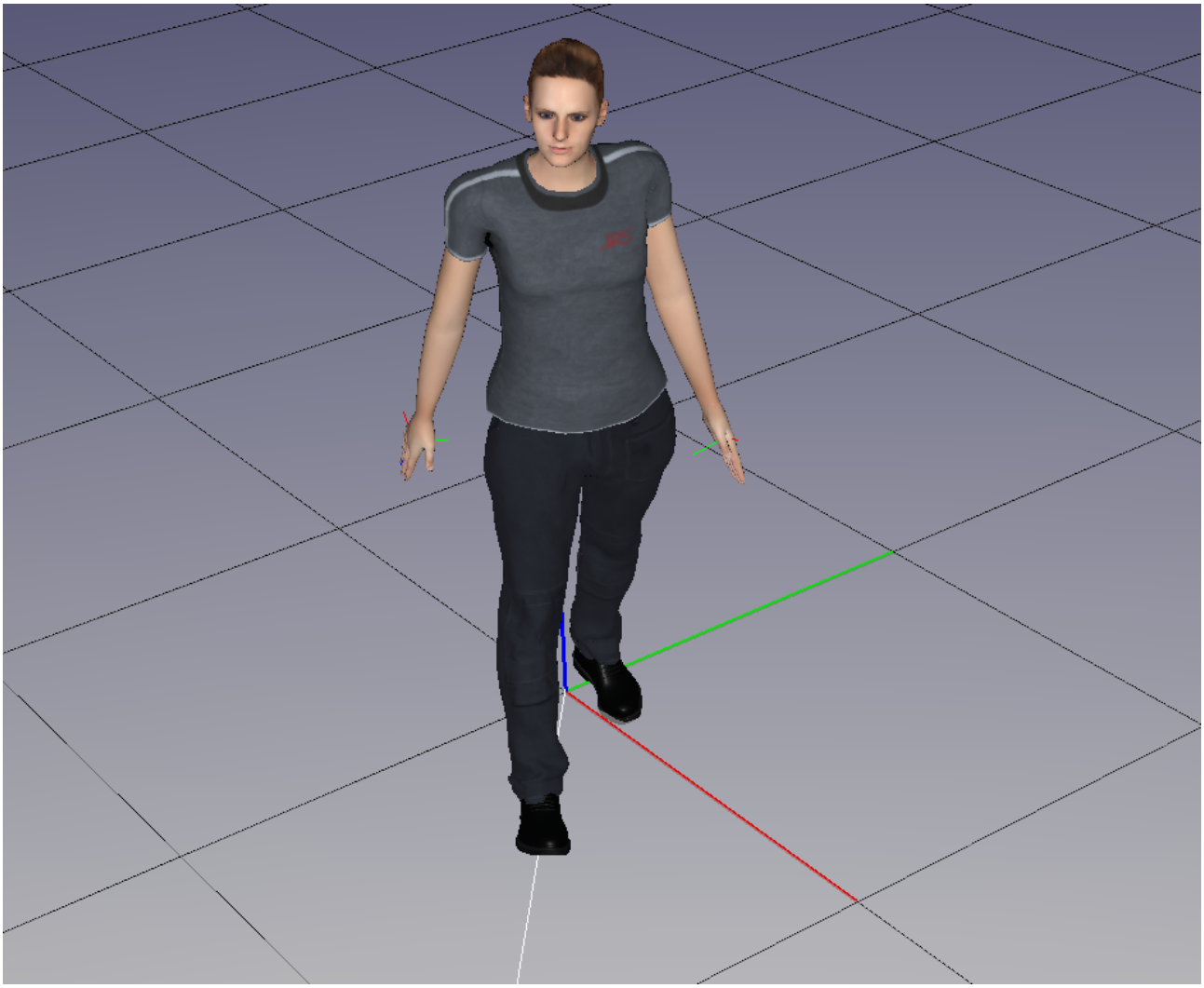


Figure 4.9: Swing foot leads the way, stance foot stays put.

References

[Boh97] R. W. Bohannon. Comfortable and maximum walking speed of adults aged 20—79 years: reference values and determinants. *Age and Ageing* **26**.1 (1997), 15–19. DOI: 10.1093/ageing/26.1.15.

[Bus04] S. R. Buss. *Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods*. Tech. rep. IEEE Journal of Robotics and Automation, 2004.

[CBP10] S. Coros, P. Beaudoin, and M. V. D. Panne. Generalized biped walking control. *ACM SIGGRAPH 2010 papers on - SIGGRAPH 10* (2010). DOI: 10.1145/1833349.1781156.

[CVS97] J. Crosbie, R. Vachalathiti, and R. Smith. Age, gender and speed effects on spinal kinematics during walking. *Gait Posture* **5**.1 (1997), 13–20. DOI: 10.1016/s0966-6362(96)01068-5.

[DGP06] R. DIMITRIJEVIC, Y. Gerasimenko, and M. M. PINTER. *Evidence for a Spinal Central Pattern Generator in Humans*. Feb. 2006.

[DP03] I. Duyar and C. Pelin. Body height estimation based on tibia length in different stature groups. *American Journal of Physical Anthropology* **122**.1 (2003), 23–27. DOI: 10.1002/ajpa.10257.

[GP12] T. Geijtenbeek and N. Pronost. Interactive Character Animation Using Simulated Physics: A State-of-the-Art Review. *Computer Graphics Forum* **31**.8 (Dec. 2012), 2492–2515. DOI: 10.1111/j.1467-8659.2012.03189.x.

[HKS17] D. Holden, T. Komura, and J. Saito. Phase-functioned Neural Networks for Character Control. *ACM Trans. Graph.* **36**.4 (July 2017), 42:1–42:13. ISSN: 0730-0301. DOI: 10.1145/3072959.3073663. URL: http://doi.acm.org/10.1145/3072959.3073663.

[Jär02] A. Järvinen. "Gran Stylissimo: The Audiovisual Elements and Styles in Computer and Video Games." *CGDC Conf.* 2002.

[Jen+13] O. G. Jenni et al. Infant motor milestones: poor predictive value for outcome of healthy children. *Acta Paediatrica* **102**.4 (Apr. 2013). DOI: 10.1111/apa.12129.

[Las87] J. Lasseter. Principles of traditional animation applied to 3D computer animation. *Proceedings of the 14th annual conference on Computer graphics and interactive techniques - SIGGRAPH 87* (1987). DOI: 10.1145/37401.37407.

[Leh12] A.-S. Lehmann. Taking the Lid off the Utah Teapot Towards a Material Analysis of Computer Graphics. **2012** (Jan. 2012).

[MBD13] P. Meyns, S. M. Bruijn, and J. Duysens. The how and why of arm swing during human walking. *Gait Posture* **38**.4 (2013), 555–562. DOI: 10.1016/j.gaitpost.2013.02.006.

[Nic+17] Nicolas et al. *Emergence of Locomotion Behaviours in Rich Environments*. July 2017. URL: https://arxiv.org/abs/1707.02286.

[Pen+18] Peng et al. *DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills*. Apr. 2018. URL: https://arxiv.org/abs/1804.02717.

[PW99] Z. Popovic and A. P. Witkin. "Physically Based Motion Transformation". *SIGGRAPH*. 1999.

[RH91] M. H. Raibert and J. K. Hodgins. "Animation of Dynamic Legged Locomotion". *Computer Graphics / Proceedings of SIGGRAPH*. Vol. 25. 4. July 1991, pp. 349–358.

[rrr68] Константинов, Н. Н., Минахин, В. В., and Пономаренко, В. Ю. *Кошечка*. 1968.

[rrr74] Константинов, Н. Н., Минахин, В. В., and Пономаренко, В. Ю. Программа, моделирующая механизм и рисующая мультфильм о нём. *Проблемы кибернетики* 28 (1974), 193–209.

[SJJ] J. Shan, C. Junshi, and C. Jiapin. Design of central pattern generator for humanoid robot walking based on multi-objective GA. *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)* (). DOI: 10.1109/iros.2000.895253.

[Tha08] D. Thalmann. "Motion Modeling: Can We Get Rid of Motion Capture?" *Motion in Games*. Ed. by A. Egges, A. Kamphuis, and M. Overmars. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 121–131. ISBN: 978-3-540-89220-5.

[TJT95] F. Thomas, O. Johnston, and F. Thomas. *The illusion of life: Disney animation*. Disney Editions, 1995.

[WH99] A. Woodward and K. Hoyne. Infants Learning about Words and Sounds in Relation to Objects. *Child Development* **70**.1 (1999), 65–77. DOI: 10.1111/1467-8624.00006.

[Wil13] B. Wilson. Computer Art Across the Iron Curtain: Early Digital Character Design in Kitty. *Animation Journal* **22**.4 (Jan. 2013), 4–25.

[WK88] A. Witkin and M. Kass. Spacetime constraints. *ACM SIGGRAPH Computer Graphics* **22**.4 (Jan. 1988), 159–168. DOI: 10.1145/378456.378507.

[WW07] K. Wolff and M. Wahde. Evolution of Biped Locomotion Using Linear Genetic Programming. *Climbing and Walking Robots: towards New Applications* (Jan. 2007). DOI: 10.5772/5088.

[YLP07] K. Yin, K. Loken, and M. V. D. Panne. Simbicon. *ACM Transactions on Graphics* **26**.99 (2007), 105. DOI: 10.1145/1239451.1239556.